Data Mining and Classification for Traffic Systems using Genetic Network Programming

ZHOU, Huiyu

February 2011

Waseda University Doctoral Dissertation

Data Mining and Classification for Traffic Systems using Genetic Network Programming

ZHOU, Huiyu

Graduate School of Information, Production and Systems
Waseda University

February 2011

Abstract

Since the increase of the road traffic in modern metropolis, the need for traffic prediction systems becomes significant, while the traffic prediction aims at an accurate estimate of the traffic flow as an important item in recent traffic control systems. Concretely, the traffic prediction system analyzes data, especially real-time traffic data, predicts traffic situations, and its major role is to forecast the congestion levels in advance of hours and even days. Therefore, the traffic prediction system is becoming the key issue in the advanced traffic management and information systems, which reduces traffic congestions and improve traffic mobility.

Vast amount of traffic data are currently available using various components of the intelligent transportation system(ITS). Satellite-based automatic vehicle location technologies such as Global Positioning System (GPS) and cellular phones can determine the vehicle positions at frequent time intervals. These equipments collect the information on the vehicle positions and speeds, which are archived in a large amount of databases, enabling further analysis of the data about the traffic situations such as traffic density patterns.

The evolutionary computation method named Genetic Network Programming (GNP) has been proposed as an extension of typical evolutionary computation methods, such as Genetic Algorithm (GA) and Genetic Programming (GP). GNP-based data mining has been already proposed to deal with high density databases with large amount of attributes. In order to further extend the proposed data mining method using GNP to the real- time traffic system, time related association rule mining methods have been proposed and studied in this thesis. The extracted time related rules are stored though generations in a rule pool and analyzed to build a classifier, based on which the future traffic density information can be provided to the optimal route search algorithm of the navigation systems. Simulation studies on the prediction accuracy of extracted rules and the average traveling time of the optimal route using the future traffic information are carried out to verify the efficiency and effectiveness of the proposed mech-

anisms. Some analyses of the proposed methods are studied based on these simulation results comparing to the conventional methods.

Unlike the other traffic density prediction methods, the main task of GNP-based time related data mining is to allow the GNP individuals to self-evolve and extract association rules as many as possible. What's more, GNP uses evolved individuals (directed graphs of GNP) just as a tool to extract candidate association rules. Thus, the structure of GNP individuals does not necessarily represent the association relations of the database. Instead, the extracted association rules are stored together in the rule pool separated from the individuals. As a result, the structures of GNP individuals are less restricted than the structures of GA and GP, thus GNP-based data mining becomes capable of producing a large number of association rules.

In chapter 2, a method of association rule mining using Genetic Network Programming (GNP) with time series processing mechanism and attributes accumulation mechanism was proposed in order to find time related sequence rules efficiently in association rule extraction systems. In this chapter, GNP is applied to generate candidate association rules using the database consisting of a large number of time related attributes. In order to deal with a large number of attributes, GNP individual accumulates fitter attributes gradually during rounds, and the rules of each round are stored in a Small Rule Pool using a hash method, then the rules are finally stored in a Big Rule Pool after the check of the overlap at the end of each round. The aim of this chapter is to propose a method to better handle association rule extraction of the databases in a variety of time-related applications, especially in the traffic prediction problems. The algorithm which can find the important time related association rules is described and several experimental results are presented considering a traffic prediction problem.

In chapter 3, an algorithm capable of finding important time related association rules is proposed, where Genetic Network Programming (GNP) with not only Attribute Accumulation Mechanism (AAM) but also Extraction Mechanism at Stages (EMS) is used. Then, the classification system imitating the public voting process based on extracted time related association rules in the rule pool is proposed to estimate to which class the current traffic data belong. Using this kind of classification mechanism, the traffic prediction is available since the extracted rules are based on time sequences. Furthermore, the experimental results on the traffic prediction problem using the proposed mechanism are presented by the simple traffic

simulator.

In chapter 4, further improvements have been proposed for the time related association rule mining using generalized GNP with Multi-Branches and Full-Paths (MBFP) algorithm. For fully utilizing the potential ability of GNP structure, the mechanism of Generalized GNP with MBFP is studied. The aim of this algorithm is to better handle association rule extraction from the databases with high efficiency in variety of time-related applications, especially in the traffic density prediction problems. The generalized algorithm which can find the important time related association rules is described and experimental results are presented considering the traffic prediction problem.

Chapter 5 is devoted to a further advanced method for extracting important time related association rules using evolutionary algorithm named Genetic Network Programming (GNP), where Accuracy Validation algorithm is applied to further improve the prediction accuracy. The proposed method provides more useful mean to investigate the future traffic density of traffic networks and hence further help to develop traffic navigation systems. The aim of this algorithm is to better handle association rule extraction using prediction accuracy as one of the criteria and guide the whole evolution process more efficiently, then the adaptability of the proposed mechanism is studied considering the real-time traffic situations using a large scale simulator SOUND/4U. The experiments deal with a traffic density prediction problem using the database provided by the large scale simulator.

Chapter 6 describes a methodology for extracting important time related association rules using an evolutionary algorithm named fixed step GNP-based association rule mining. And based on the rule pool of the fixed prediction step, it is also proposed that the prediction of the future traffic is combined with a classical routing algorithm. The routing algorithm and prediction results are combined using a large scale simulator SOUND/4U. Simulation results showed that by providing future traffic information, the average traveling time for the testing vehicles can be improved, which proves that the proposed method can deal with the traffic prediction combined with the optimal route search problem fairly well.

In chapter 7, after studying each research topic in this thesis, AAM and EMS mechanisms have been proposed to improve the effectiveness of rule extraction, MBFP mechanism has also been proposed to further improve the efficiency of rule extraction and Accuracy Validation mechanism aiming at generating more general rules has been verified, finally the predicted

future information has been combined with the routing algorithm for navigation systems. In conclusion, the efficiency and effectiveness of the proposed methods have been proved based on the simulation results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Since the increase of traffic in modern metropolis, the need for traffic prediction system becomes significant, while the traffic prediction aims at an accurate estimate of the traffic flow as an important item in recent traffic control systems. Concretely, the traffic prediction system analyzes data, especially real-time traffic data, predicts traffic situations.

The traffic prediction system aims at an accurate estimate of the traffic flow as an important component of advanced traffic management and information systems, which reduces traffic congestions and improve traffic mobility.

Vast amount of traffic data are currently available using various components of the intelligent transportation system(ITS) [1]. Satellite-based automatic vehicle location technologies such as Global Positioning System (GPS) and cellular phones can determine vehicle positions at small time intervals. These equipments collect information about the vehicle positions and speeds, which are archived in large amount of databases, enabling further analysis of the data about the traffic situations such as traffic volume patterns.

<Related Works>

In order to prevent the congestions or provide the useful information for traffic management systems, when and where the traffic congestion or heavy traffic will occur should be predicted, then the navigation system can choose the route avoiding the sections with potential congestion and give higher priority to the sections with potential

low traffic. Many traffic density prediction methods belong to the traffic congestion prediction using regression analysis. Regression models have the advantage of accurate prediction, especially when model assumptions are thoroughly examined.

Common approaches to the problem of forecasting traffic situations are based on time series models. For example, autoregression models have been proposed and widely used for traffic flow prediction, especially when model assumptions are thoroughly examined. It is also mentioned that some of the time series models include the averaging or smoothing of input data over long time intervals, which may result in obscuring the association relationships in the database and thus affect the predictive performance[2].

Generally, the observation data collected either from the field or laboratory are noisy in nature. For example, in dynamic environments of traffic prediction, occasional abnormal changes caused by the failure of equipments hamper the analysis of traffic patterns. These abnormal changes are called noises in the database. Lingras and Osborne[3] found that both the regression and neural network models are robust in the presence of a small amount of noises. However, they also showed that, if the amount of noises increases, which means the increase of abnormal changes in the database, neural networks are more robust than statistical models.

Robustness to noises is necessary in traffic prediction since occasional changes frequently appear on the traffic network. Bad weather or equipments' failure may easily cause the abnormal changes of the records, thus traffic density prediction needs to find accurate traffic density patterns under the influence of the abnormal changes, in another words, noisy environments.

Soft Computing based methods such as Neural Networks can also be used to traffic congestion prediction, where the self-evolved parameters are used and also robustness under the noisy environment is obtained[3]. What's more, they have a good adaptive ability since the parameters can be adjusted automatically as the OD changes in the environment. The proposed mechanism uses an evolutionary based method to extract interesting association rules, while these rules are stored and used to construct an classifier model which predicts the traffic of the road networks, thus the proposed method can deal with various real-time traffic situations and show good mining performances under the noisy environments.

On the other hand, Dynamic Traffic Assignment(DTA)[4] has also attracted many attention, since it is capable of processing time-varying properties of traffic flows. However, the biggest issue is the requirement of time-dependent OD(original and destination) data and complex massive mathematical equations in the prediction process.

1.2 Evolutionary Algorithm

Optimization is a classical problem to deal with in all aspects of real world applications related to Physics, Mathematics, Economy or Biologies. Generally speaking, purely analytical methods proved efficiency in optimization topics, however, still suffer from one weakness, that is adaptivity.

The phenomena in nature rarely obey to simple rules and very hard to be accurately defined by mathematic differentiable functions, especially when the environment is continuously changing, which makes the global optimization more difficult to achieve.

Natural Process of Darwinian evolution intrigue the idea of Genetic based Optimization algorithm, which imitate the natural adaptation processes, and this kind of algorithm is called Evolutionary Algorithm. Holland first proposed Genetic Algorithm(GA) [5],[6] at the beginning of the 60s.

The basic idea is represent possible solutions as gene individuals with binary structure in a given population pool. Only the adaptive individual can survive the natural selection and undergo the process of mutation and crossover to generate next generation.

GA is intrinsically a robust search and optimization mechanism. Evolution process of GA is not a purposeful or directed process. Only the better individuals adapted to the environment survive. This population-based optimization results in stochastic optimization techniques that can often outperform classical analytical optimization methods when applied to difficult real-world problems.

Genetic Programming(GP)[7], [8] is the extension of the genetic model of learning into the space of programs. That is, the objects that constitute the population are not fixed-length binary strings that encode possible solutions to the problem at hand, and they are programs with the phenotype of tree structures.

GP is more expressive than binary string structures of GA. GP was firstly mainly used to solve relatively simple problems because it is computationally intensive. However, due to the development of modern computational technology and improvement of GP, it has recently produced many novel and outstanding results in areas such as electronic design, sorting, and searching optimizations.

Genetic Network Programming(GNP)[9],[10] has been proposed as an extended method of Genetic Algorithm (GA) and Genetic Programming (GP) by employing a directed graph as its genes. The applicability and efficiency of GNP has been studied by both virtual and real world applications. The directed graph of GNP contributes to creating quite compact programs and generality the partial observable processes in the network flows. Also, GNP can find solutions of problems without the bloating problem compared with GP, because of the fixed number of nodes in GNP.

1.3 Data Mining

The proposed method applies GNP to the data mining method to search for the potential associations among events and hence predict the future traffic density.

Searching for the pattern from databases is a process for obtaining associations where the occurrence of one event is related to other events. The association rule mining method, i.e., one of the most popular data mining methods with a wide range of applications is used in the proposed mechanism aiming at discovering association relations or correlations among attributes encoded within a database [11].

Data mining, also called knowledge discovery, is the analytical process of digging through and exploring the enormous sets of data in the search for consistent patterns and/or systematic relationships between attributes. It aims at extracting implicit, previously unknown information from data sets, which could be useful for many applications.

Nowadays, data mining has become an important field since huge amounts of data have been collected in various applications. Mining these data sets efficiently and effectively are too difficult and intricate when using conventional methods, especially for the sequential time related database in dynamic systems.

As one of the most popular data mining methods with a wide range of applicability, there is an association rule mining, where its major task is to detect relationships or associations between attributes in large databases. That is, the association rule mining aims at discovering association relations or correlations among attributes encoded within a database [11]. The relationship between data sets can be represented as association rules. An association rule has the form of $(X \Rightarrow Y)$, where X represents antecedent and Y represents consequent. The association rule "$X \Rightarrow Y$" can be interpreted as: the set of attributes satisfying X is likely to satisfy Y.

In many applications, such as information systems, web access traces, system utilization logs, transportation systems, etc., the data has naturally the form of time sequences. For example, in traffic systems, the sequential information such as "The 5th road has high traffic density, then, the 4th road will also have high traffic density on the same day" has been of great interest for analyzing the time-related data to find its inherent characteristics.

Concretely, conventional association rules are not enough to predict the future traffic situations in real time systems, which means that the association rules should be time related like the following: "If section X on the traffic map has high traffic density at time $t = 0$(current time), then section Y will also has high traffic density at $t = 10$(10 time steps later)."

Since mining real time data efficiently and effectively is too intricate when using conventional methods, especially for the time related sequential database in dynamic systems, e.g., traffic systems, the proposed mechanism uses an evolutionary based

method to extract interesting association rules, while these rules are stored and can be used to predicts the traffic of the road networks, thus the proposed method can deal with various real-time traffic situations and show good mining performances under the changing environments.

To meet the different needs of various applications, and also to analyze and understand the nature of various sequences, several models of sequential pattern mining have been proposed. The proposed association rule mining mechanism not only briefly studies definitions and application domains of these models, but also more introduces an evolutionary algorithm on how to find these patterns effectively and efficiently.

<Association Rule Mining Methods>

The most popular model in the association rule mining is Apriori algorithm, in which Agrawal et. al. proposed the support-confidence framework [12]. This algorithm measures the importance of association rules with two factors: support and confidence. Chi-squared value has also been applied to association rule mining. Brin et.al. suggested to measure the significance of associations via the chi-squared test for correlation used in classical statistics [13].
However, Apriori algorithm may suffer from large computational complexity for rule extraction when extracting from dense databases. Many approaches have been proposed to extract association information in various directions, including efficient apriori-like mining methods [14], [15]. The variations of Apriori approach such as the hash-based algorithm have also been studied for efficiency [16], [17].

Another kind of data mining uses Neural Networks [18], [19], which is a collection of neuron-like processing units with weighted connections between the units. However, they often produce incomprehensible models and require training data and long training times.

Genetic Algorithm (GA), proposed by J.H. Holland, has also been applied to data mining research in order to deal with dense databases. Holland proposed two kinds of approaches, Pittsburgh approach and Michigan approach. Pittsburgh approach represents an entire rule set as an individual, and evolve the population of candidate rule sets, while in Michigan approach, members of the population are individual rules and a rule set is represented by the entire population. Both of the approaches evolve the rules during generations and the individuals themselves represent the association relationships capable of acquiring inference rules. However, because a rule is represented as an individual or part of an individual in GA, it is hard for them to give us a complete picture of the underlying relationships in problem domains, thus not easy to extract

enough number of association rules.

Genetic Programming improved the expression ability of GA by evolving individuals as tree structures. Although, GP enables a more explicit representation of reference rules, it may suffer from the problem of loose structures and bloating, especially in dynamic problems.

Recently, many data mining methods have been proposed to satisfy the increasing demand for efficient information mining in time-related dynamic systems. A. K. H. Tung proposed the intertransaction association rule mining [20], which not only extracts relationship within the transaction, but also uses an extended Apriori method to obtain inter-transaction associations by defining and mining frequent intertransaction itemsets(FITI). This method uses the sequential association between transactions to represent temporal relations between transaction items. Since the method is basically an Apriori-based mining method, it shares the same disadvantage not able to deal with the databases with a very large number of attributes and high density.

TPrefixSpan algorithm [21] is a kind of nonambiguous temporal pattern mining and it can deal with temporal databases using temporal relationships between two intervals which was proposed by Kam and Fu in 2000 [22]. This method deals with interval-based event data, however, it overcomes the ambiguous problem with interval-based temporal mining by correctly describing the temporal relationship between every pair of events. As a result, the method is capable of building an unique temporal sequence for every time sequence of interval events. However, since there are 14 kinds of relationships between every pair of attributes, it would suffer from efficiency problems when processing the database with a large number of attributes.

Recurrent neural networks and associative memory can also deal with the time related database. Recurrent neural networks are used for extracting rules [23] as a form of Deterministic Finite-state Automata. The recurrent connection and time delay between neurons allows the network to generate time-varying patterns. Associative memory is mainly a content-addressable structure that maps specific input representations to specific output representations, and the time delay between neurons also enables to mimic the time related relationship patterns in real-time systems.

1.4 Contents of this Research

1.4.1 Motivations

GNP-based data mining method was first proposed by Dr.Shimada[24]. The databases used in the conventional GNP data mining method are transaction-related database, which means that every tuple in the database represents one of the transactions. A database transaction is a unit of events in the database that is treated in a coherent way.

Traditional transaction based rules without time series can tell what interest events happen together in one transaction, however, it can not represent at what time the event will happen or how long the event will persist.

Therefore, a method of time-related association rule mining will by proposed in this thesis[25] using Genetic Network Programming (GNP) to improve the efficiency and effectiveness of rule extraction in time related databases. In this thesis, the original time-related association rule mining will be explained and extended, and its mechanisms will be proposed in order to find time related rules more efficiently.

Unlike the other methods mentioned above, the main task of GNP-based time related data mining is to allow the GNP individuals to self-evolve and extract association rules as many as possible. What's more, it uses evolved individuals(directed graphs of GNP) just as a tool to extract candidate association rules. Thus, the structure of GNP individuals does not necessarily represent the association relations of the database. Instead, the extracted association rules are stored together in the rule pool separated from the individuals. As a result, the structures of GNP individuals are less restricted than the structures of GA and GP, and GNP-based data mining becomes capable of producing a large number of association rules.

In order to verify the efficiency and effectiveness of the proposed mechanism, this research is carried out by gradually studying the performance of the proposed method from simple ones to large scale real time ones in simulators. And the combination of the proposed prediction mechanism with the routing algorithm has also been studied in 2 kinds of simulators, and some analyses of the proposed methods are made based on these test results comparing to conventional methods.

1.4.2 Research Topics

In this thesis, there are five topics discussed based on the former mentioned motivations.

In chapter 2, a method of association rule mining using Genetic Network Programming (GNP) with time series processing mechanism and Attributes Accumulation Mechanism(AAM) was proposed in order to find time related sequence rules efficiently in association rule extraction systems. In this chapter, GNP is applied to generate candidate association rules using the database consisting of a large number of time related attributes. In order to deal with a large number of attributes, GNP individual accumulates fitter attributes gradually during rounds, and the rules of each round are stored in a Small Rule Pool using a hash method, then the rules are finally stored in a Big Rule Pool after the check of the overlap at the end of each round. The aim of this chapter is to better handle association rule extraction of the databases in a variety of time-related applications, especially in the traffic prediction problems. The algorithm which can

find the important time related association rules is described and several experimental results are presented considering a traffic prediction problem.

In chapter 3, an algorithm capable of finding important time related association rules is proposed where Genetic Network Programming(GNP) with Attribute Accumulation Mechanism (AAM) and Extraction Mechanism at Stages (EMS) is used. Then, the classification system imitating public voting process based on extracted time related association rules in the rule pool is proposed to estimate to which class the current traffic data belong. Using this kind of classification mechanism, the traffic prediction is available since the rules extracted are based on time sequences. And, the experimental results on the traffic prediction problem using the proposed mechanism is presented by a simple traffic simulator.

In chapter 4, since Genetic Network Programming(GNP)-based time related association rules mining method provides an useful mean to investigate future traffic density of road networks, it helps us to develop traffic navigation system. Further improvements have been proposed in this chapter about the time related association rule mining using generalized GNP with Multi-Branches and Full-Paths(MBFP) algorithm. For fully utilizing the potential ability of the GNP structure, the mechanism of Generalized GNP with MBFP is studied. The aim of this algorithm is to better handle association rule extraction from the databases with high efficiency in a variety of time-related applications, especially in the traffic density prediction problems. The generalized algorithm which can find the important time related association rules is described and experimental results are presented considering the traffic prediction problem.

Chapter 5 is devoted to extracting important time related association rules using evolutionary algorithm named Genetic Network Programming(GNP), where Accuracy Validation algorithm is applied to further improve the prediction accuracy. The proposed method provides an useful mean to investigate the future traffic density of traffic networks and hence help to develop traffic navigation systems. The aim of this algorithm is to better handle association rule extraction using prediction accuracy as one of the criteria and guide the whole evolution process efficiently, then investigate the adaptability of the proposed mechanism to the real-time traffic situations using a large scale simulator SOUND/4U. The experiments deal with a traffic density prediction problem using the database provided by the large scale simulator.

Chapter 6 describes a methodology and results of traffic prediction by extracting important time related association rules using an evolutionary algorithm named fixed step GNP-based association rule mining. The extracted rules provides an useful mean to investigate the future traffic density of traffic networks and hence to develop traffic navigation systems. The proposed methodology is implemented and experimentally evaluated using a large scale simulator SOUND/4U. The routing algorithm combined

with the traffic prediction results is studied using the environment of SOUND/4U.

In chapter 7, after giving the objectives and analyses of each research topic in this thesis, some conclusions about the proposed mechanisms are drawn based on the simulation results.

Chapter 2

Basic Time Related Association Rule Mining

2.1 Introduction

The major points of this chapter are included as follows:

- The database is time-related considering the real-time factor into consideration. Then, the same attribute now has different meanings at different time units.
- The extracted association rule has the following form: "$X \Rightarrow Y$", where each attribute of X and Y should have its time tag, and the rules obtained are actually show the relationships between data sets with time sequence.
- Time delay tags are used for the connections of the judgment nodes in GNP-based data mining, for example "$A(t = p)$" means the judgment of the attribute A at time p. Therefore, the searching for the database for calculating the confidence, support and chi-squared value becomes the two dimensional search.
- The concept of an Attribute Accumulation Mechanism (AAM) is introduced in order to systematically and efficiently explore the search space. Each round has its own attribute sub set and accumulate fitter attributes in the sub set gradually.

In section 2.2, the basic structure of time related database are introduced. In section 2.3, the algorithm of GNP-based time related sequence mining method will be described. Section 2.4 shows examples of applying the proposed method to traffic prediction using a simple simulator. And section 2.5 is devoted to conclusions.

Table 2.1: Conventional Transactional Database

T	A_1	A_2	A_3
1	8	3	6
2	5	4	6
3	8	1	10
4	2	9	9

Table 2.2: Time Related Database after Discretization

Time	A_1			A_2			A_3		
	Low	Middle	High	Low	Middle	High	Low	Middle	High
0001	0	0	1	1	0	0	0	1	0
0002	0	1	0	0	1	0	0	1	0
0003	0	0	1	1	0	0	0	0	1
0004	1	0	0	0	0	1	0	0	1

Middle threshold=4; High threshold=7;

2.2 Database Environment

Time related database is the database that consists of the tuples with data indexed by time, in other words, it is the database composed of sequential values or events which occur along the time. Time related database is the typical representation of the data gathered in many application fields, for example, stock markets, production systems, scientific experiments, medical applications, etc. Therefore, the time related data base can be considered as a kind of sequence database.

While the databases which consist of transactions are called transactional databases. Most modern relational database management systems fall into this category as shown in Table 2.1, where each tuple in the database represents one transaction. However, when processing real-time information systems, the databases are to be designed and developed to emphasize the need to satisfy time related requirements, like the necessity to predict the occurrence of the critical events in the time domain. For example, traffic prediction is one of these problems. As the traffic flow in each section of the roads is changing continuously as time goes on, the database should represent at what time the events in the road section occur, in order to better predict the traffic flow of the roads in the future,.

Traditional transaction-based rules without time series can tell what interesting events happen together, however, it can not represent at what time the event will happen or how long the event will last.

Thus, the database we handle here is no longer transaction-based, but time related-based, i.e., the tuples in the time series database represent the time unit instead of the transaction. The "Time" in Table 2.2 represents the time unit, and it can be very small ones as one second, one minute, or long ones like one year or one episode of a process and so on, thus its concrete meaning is related to the concrete problem to solve.

In the traffic systems we deal with, the database have continuous attributes like the traffic density of each section, so, we divided the continuous values to three categories, i.e., Low, Middle and High. Supposing that the Middle threshold is 4 and High threshold is 7, the attribute $A = 9$ is ranked as $A - High$, i.e., ($A - Low$, $A - Middle$, $A - High$)=$(0, 0, 1)$. Thus, Table 2.2 shows the binary values of 0s and 1s after discretizing Table 2.1 using the above thresholds.

In the traffic prediction problem, we assume there exist many cars on each section of the roads, thus the rows of the database are consisted of time units and the number of cars on every section becomes the attribute columns of the database. "The section named A has the traffic density of 10 cars on the time unit 2." is a typical case of the events in the traffic systems. This event information is recorded in row 2 and the column which represents A in the database. Since the value of the event is 10, we can classify it to $A - High$ using the thresholds mentioned above, and the event can be represented as ($A - Low$, $A - Middle$, $A - High$)=$(0, 0, 1)$ in the discretized database, where, the $A-Low$, $A-Middle$ and $A-High$ with binary values are called attribute here. Now the problem is to find time related interesting sequential relationships among attributes in the discretized database.

2.2.1 GNP for Association Rule Mining with Time Series

Association rule mining with time series is an extension of the GNP-based data mining [24] in terms of treating time related rules. In the proposed method, GNP individual examines the attribute values of database tuples using judgment nodes and calculates the measurements of association rules using processing nodes. Attributes of the database and their values correspond to judgment nodes and their judgment objects in GNP, respectively. Therefore, the connections of nodes are represented as candidates of association rules. The measurements include support and Chi-squared values. Judgment node determines the next node by a judgment result of (Yes/No). Fig.2.1 shows a basic structure of GNP for time-related association rule mining. P_1 here is a processing node and is a starting point of association rule mining. Each Processing node has an inherent numeric order ($P_1, P_2, ...$) and is connected to a judgment node. Yes-side of the judgment node is connected to another judgment node. No-side of the judgment node is connected to the next numbered processing node. The total number of tuples in the database moving to the Yes-side at each judgment node is calculated for every processing node. These numbers are the fundamental values for calculating

criteria(support and chi-squared values) of the association rules.

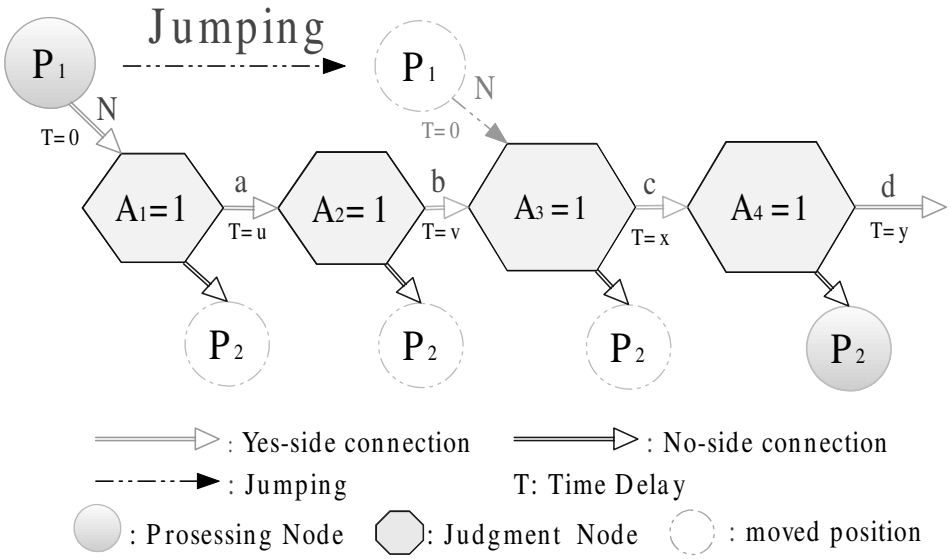
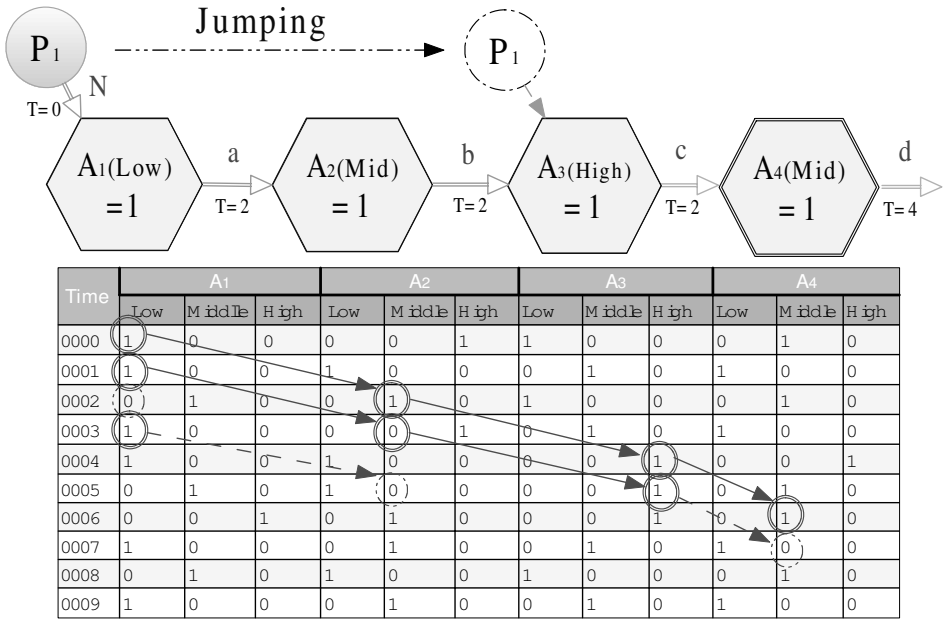


Figure 2.1: The basic structure of individuals

In the proposed method, the examination should consider both the attribute dimension and the time dimension concurrently, thus the method is basicly two dimensional. That is, not only the attributes but also the corresponding time delays should be considered: For example, as described in Fig.2.2: the judgment is not merely executed row by row, but the procedure is like the following: firstly judge the tuple at time 0000, and according to GNP individual structure of Fig.2.2, first $A_1(Low)$ is judged and if the value of $A_1(Low)$ at time 0000 is $'1'$, then move to the next judgement node named $A_2(Mid)$. Then, due to the time delay T =2 from $A_1(Low)$ to $A_2(Mid)$, we check the value of $A_2(Mid)$ at time 0000+2=0002. If the value of $A_2(Mid)$ at time 0002 is '1', continue the judgment likewise, if not, execute another turn of the judgment which begins from time 0001, 0002, 0003, . . . , until the end of the tuple.

2.2.2 Extraction of Association Rule

The total number of moving to Yes-side from the processing node at each judgment node is calculated for every processing node, which is a starting point for calculating association rules. In Fig.2.1 and Fig.2.2, N is the number of the total search, and a, b, c and d are the number of the searches moving to the Yes-side for each judgment node. The measurements are calculated by these numbers.



Time	A1			A2			A3			A4		
	Low	Middle	High	Low	Middle	High	Low	Middle	High	Low	Middle	High
0000	1	0	0	0	0	1	1	0	0	0	1	0
0001	1	0	0	1	0	0	0	1	0	1	0	0
0002	0	1	0	0	1	0	1	0	0	0	1	0
0003	1	0	0	0	0	1	0	1	0	1	0	0
0004	1	0	0	1	0	0	0	0	1	0	0	1
0005	0	1	0	1	0	0	0	0	1	0	1	0
0006	0	0	1	0	1	0	0	0	1	0	1	0
0007	1	0	0	0	1	0	0	1	0	1	0	0
0008	0	1	0	1	0	0	1	0	0	0	1	0
0009	1	0	0	0	1	0	0	1	0	1	0	0

Figure 2.2: Search method of time related data mining

Table 2.3 shows the measurements of the support and confidence of association rules. After the search described in Fig.2.2, we calculate the support and confidence of the rules like Table 2.3, but in order to calculate the chi-squared value of the rules, we need the support of the consequent part of the rules. Therefore, in the next step, we employ a jump method, which make the processing node randomly jump to another judgement node following the connection between nodes, where the judgment nodes after the jumped processing node correspond to the consequent part of the rules. Then, by counting the number of the movings to the Yes-side for each judgment node from the jumped processing node, we can calculate the support of the consequent part of the rules for obtaining the chi-squared values. This kind of jump mechanism is taken several times. For example, as represented in Fig.2.2, the processing node P_1 jumps from $A_1(Low)$ to $A_3(High)$, then the antecedent part of the candidate rules becomes "$A_1(Low) \wedge A_2(Mid)$" and the judgement nodes after the jumped P_1 will be the consequent part of the candidate rules: "$A_3(High) \wedge A_4(Mid) \ldots$ ". If P_1 node jumps from "$A_1(Low) = 1$" to "$A_2(Mid) = 1$", we are able to calculate the support of $A_2(Mid)$, $A_2(Mid) \wedge A_3(High)$ and so on, as a result, $chi - squared$ value can be calculated considering both the antecedent and consequent part of the candidate rules. We can repeat this like a chain operation in each generation. Thus, we can obtain the values for calculating the importance of the rules. Now, we define the important association rules as

Table 2.3: Measure of association rules

Association rules	support	confidence
$A_1(Low) \Rightarrow A_2(Mid)$	b/N	b/a
$A_1(Low) \Rightarrow A_2(Mid) \wedge A_3(High)$	c/N	c/a
$A_1(Low) \Rightarrow A_2(Mid) \wedge A_3(High) \wedge A_4(Mid)$	d/N	d/a
$A_1(Low) \wedge A_2(Mid) \Rightarrow A_3(High)$	c/N	c/b
$A_1(Low) \wedge A_2(Mid) \Rightarrow A_3(High) \wedge A_4(Mid)$	d/N	d/b
$A_1(Low) \wedge A_2(Mid) \wedge A_3(High) \Rightarrow A_4(Mid)$	d/N	d/c

the ones which satisfy the following:

$$\chi^2 > \chi^2_{min}, \tag{2.1}$$

$$support \geq sup_{min}, \tag{2.2}$$

where, χ^2_{min} and sup_{min} are the threshold of the minimum chi-squared and support value given by supervisors. In this definition, if the rule "$X \Rightarrow Y$' is important, then, $X \Rightarrow \neg Y, \neg X \Rightarrow Y, \neg X \Rightarrow \neg Y, Y \Rightarrow X, Y \Rightarrow \neg X, \neg Y \Rightarrow X$ and $\neg Y \Rightarrow \neg X$" are also important rules. If necessary, we can also use the confidence value in the definition. The extracted important association rules are stored in a pool all together through generations in order to find new important rules.

2.2.3 Attribute Accumulation Mechanism (AAM)

In order to deal with a large number of attributes, Attribute Accumulation Mechanism (AAM) has been proposed, where GNP individual accumulates better attributes in it gradually round by round of a sequence of generations.

The attribute accumulation mechanism proposed here first randomly selects a small attribute set of size s from the whole attribute set of size $S (S \geq s)$, then applies GNP based data mining algorithm using GNP individuals generated exclusively from the chosen attribute set and finally stores the extracted association rules in the corresponding Small Rule Pool(SRP) using hash functions. This whole procedure is called Round 0.

After the processing of Round 0, we get the corresponding SRP(0). For each of the rules stored in SRP(0), we check its overlap and sum up the count of the appearance of the attributes in the chosen attribute set, and finally sort the attributes from the most frequently used one to the least one. Then, the top $v\%$ $(0 \leq v \leq 100)$ of the attributes will be remained in the chosen attribute set. The attribute set of the next Round 1 is

then composed of the top $v\%$ of the attributes and attribute set randomly chosen from the original whole attribute set. Using the newly generated set, Round 1 searches the important association rules and stores the newly generated rules in the corresponding SRP, likewise.

The similar procedure is repeated Round by Round until the final condition is satisfied. The procedure is shown in Fig.2.3.

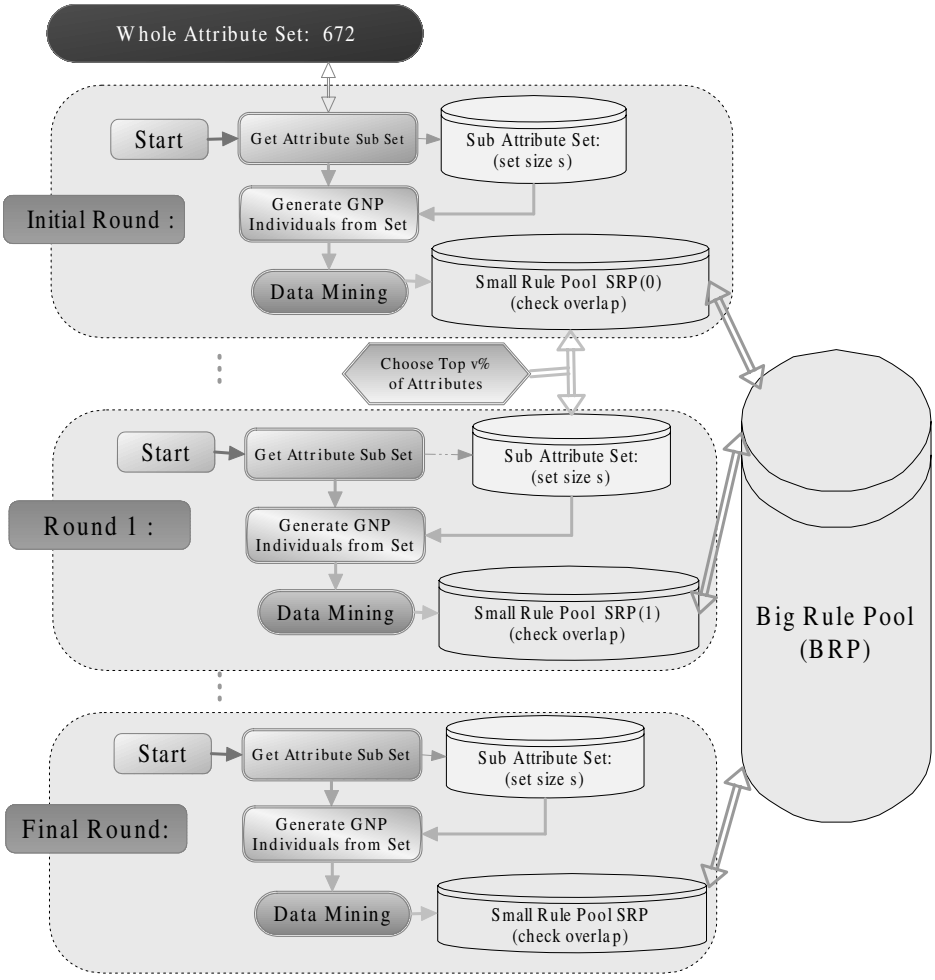


Figure 2.3: Attribute Accumulation Mechanism (AAM)

2.2.4 Rule Pool and Hash function

As mentioned before, there are two kinds of Rule Pools: Small Rule Pool(SRP), and Big Rule Pool(BRP). They both use hash functions to facilitate the speed of the

search and judge the overlap. The hash function we used here is defined by:

$$HV_r = (asum_r + tsum_r \ mod \ Tshift) \ mod \ BN, \tag{2.3}$$

$$asum_r = \sum_{i \in I(r)} \{In_i\}, \tag{2.4}$$

$$tsum_r = \sum_{i \in I(r)} \{Delay_i\}. \tag{2.5}$$

The symbols are as follows:
HV_r : the hash value of rule r.
$I(r)$: the set of attributes which is contained in rule r.
In_i : the index of attribute i.
$Delay_i$: the delay time of attribute i.
$asum_r$: the sum of the attribute indexes of rule r.
$tsum_r$: the sum of the attribute delay times of rule r.
$Tshift$: the number of shifts to differentiate the rules with the same sum of the attribute indexes but different sum of the attribute delay times.
BN : the total number of hash buckets.

In the small rule pool(SRP), we use the Eq.2.3, Eq.2.4 and Eq.2.5 to map the rules to the corresponding buckets of SRP, and each rule obtained in each generation of the round is checked for its overlap in SRP. SRP is actually a mapping mechanism from the rule to its relevant bucket in the form of hash tables.

Big Rule Pool(BRP) also employs the same hash function as SRP. Therefore, at the end of each round, we are able to check the rules in the buckets of SRP and BRP with the same hash value. Using this algorithm, the new rules are finally stored in the result pool–Big Rule Pool(BRP). The procedure of checking the overlap is shown in Fig.2.4.

In other words, when an important rule is extracted by GNP, it is checked whether an important rule is new, i.e., whether it is already in SRP or not in each generation of the rounds. If the rule is new, it is stored in the SRP. At the end of each Round, every rule in the SRP should be checked on whether it overlaps with the rules in BRP, and only the rule never exist in the BRP before can be considered as a real new rule. The appearance frequency of the attributes are also calculated using the rules in the BRP.

2.2.5 Fitness and Genetic Operators

When a new rule is generated, the rule like "$A \wedge A \Rightarrow A$" is considered useless in the original GNP-based data mining method. However, taking account of the time

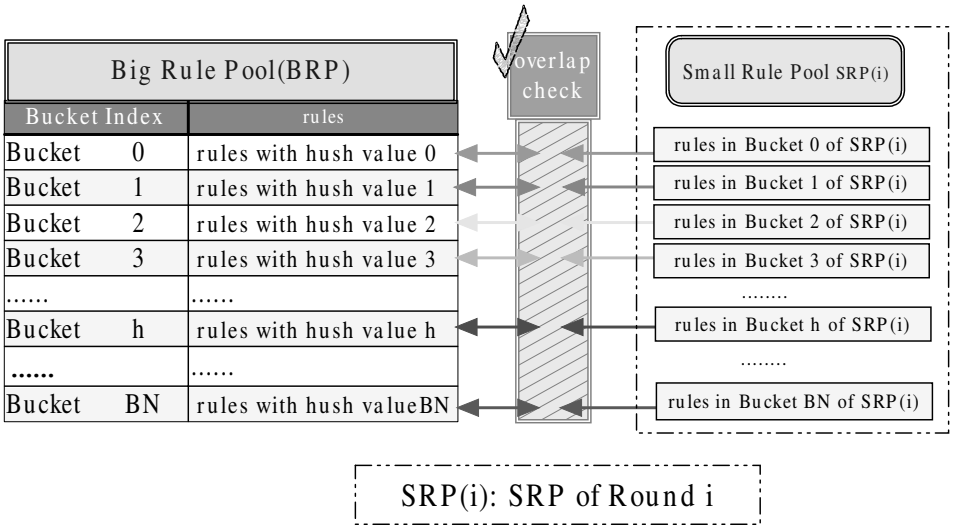


Figure 2.4: Checking the overlap of BRP and SRP(i)

series algorithm, even the same attribute has different meanings at different time units, so the rule like :

$$A_i(*)(t = p) \wedge \cdots \wedge A_i(*)(t = q) \Rightarrow$$
$$A_i(*)(t = r) \wedge \cdots \wedge A_i(*)(t = s),$$

where $p \leq q \leq r \leq s$, has its meaning. However, the number of this kind of rules should be controlled, since too many rules of this kind would harm the evolving process and produce a large number of GNP individuals with the same attribute. Now, we define the concept of multiplicity. Multiple rules here mean the rules which contain many different kinds of attributes. Therefore, the fitness function of GNP is defined as:

$$F = \sum_{r \in R} \{\chi^2(r) + 10(n_{ante}(r) - 1) + 10(n_{con}(r) - 1) + \alpha_{new}(r) + \alpha_{mult}(r)\}.$$

The symbols are as follows:

R : set of suffixes of important association rules which satisfy Eq.2.1 and Eq.2.2 in GNP individuals.

$\chi^2(r)$: chi-squared value of rule r.

$n_{ante}(r)$: the number of attributes in the antecedent of rule r.

$n_{con}(r)$: the number of attributes in the consequent of rule r.

$\alpha_{new}(r)$: constant defined as

$$\alpha_{new}(r) = \begin{cases} \alpha_{new}, & \text{if rule } r \text{ is new} \\ 0, & \text{otherwise} \end{cases}$$

$\alpha_{mult}(r)$: constant defined as

$$\alpha_{mult}(r) = \begin{cases} \alpha_{mult}, & \text{if rule } r \text{ has many kinds of} \\ & \text{different attributes} \\ 0, & \text{otherwise} \end{cases}$$

$\chi^2(r)$, $n_{ante}(r)$, $n_{con}(r)$, $\alpha_{new}(r)$ and $\alpha_{mult}(r)$ are concerned with the importance, complexity, novelty and diversity of rule r, respectively. At each generation, GNP individuals are replaced with the new ones by the selection policy and other genetic operations. We use four kinds of genetic operators:

- Crossover: The uniform crossover is used. Judgment nodes are selected as the crossover nodes with the crossover rate. Two parents exchange the gene of the corresponding nodes.
- Mutation-1: The connection of the judgment nodes is changed randomly by mutation rate-1.
- Mutation-2: The function of the judgment nodes is changed randomly by mutation rate-2.
- Mutation-3: The time delay between the judgment nodes is changed by mutation rate-3. The mutation range depends on the concrete problems to solve.

The individuals are ranked by their fitness and upper 1/4 individuals are selected. After that, they are reproduced four times, then four kinds of genetic operators are executed to them. These operators are executed for the gene of judgment nodes of GNP individuals. All the connections of the processing nodes are changed randomly in order to extract rules efficiently.

2.2.6 Summary of the proposed mechanism

The whole procedure of the proposed algorithm is shown in Fig.2.5. Two kinds of iterations are included here, the outer iteration represents the attribute accumulation mechanism, which means better attributes are gradually accumulated in the attribute sub set of each round and GNP individuals are generated based on the attributes in the attribute sub set during each round. The inner iteration represents the basic procedure of the time related data mining method, which uses GNP individuals to generate the candidate rules, calculate the support and chi-squared values of the candidate rules based on the time related databases.

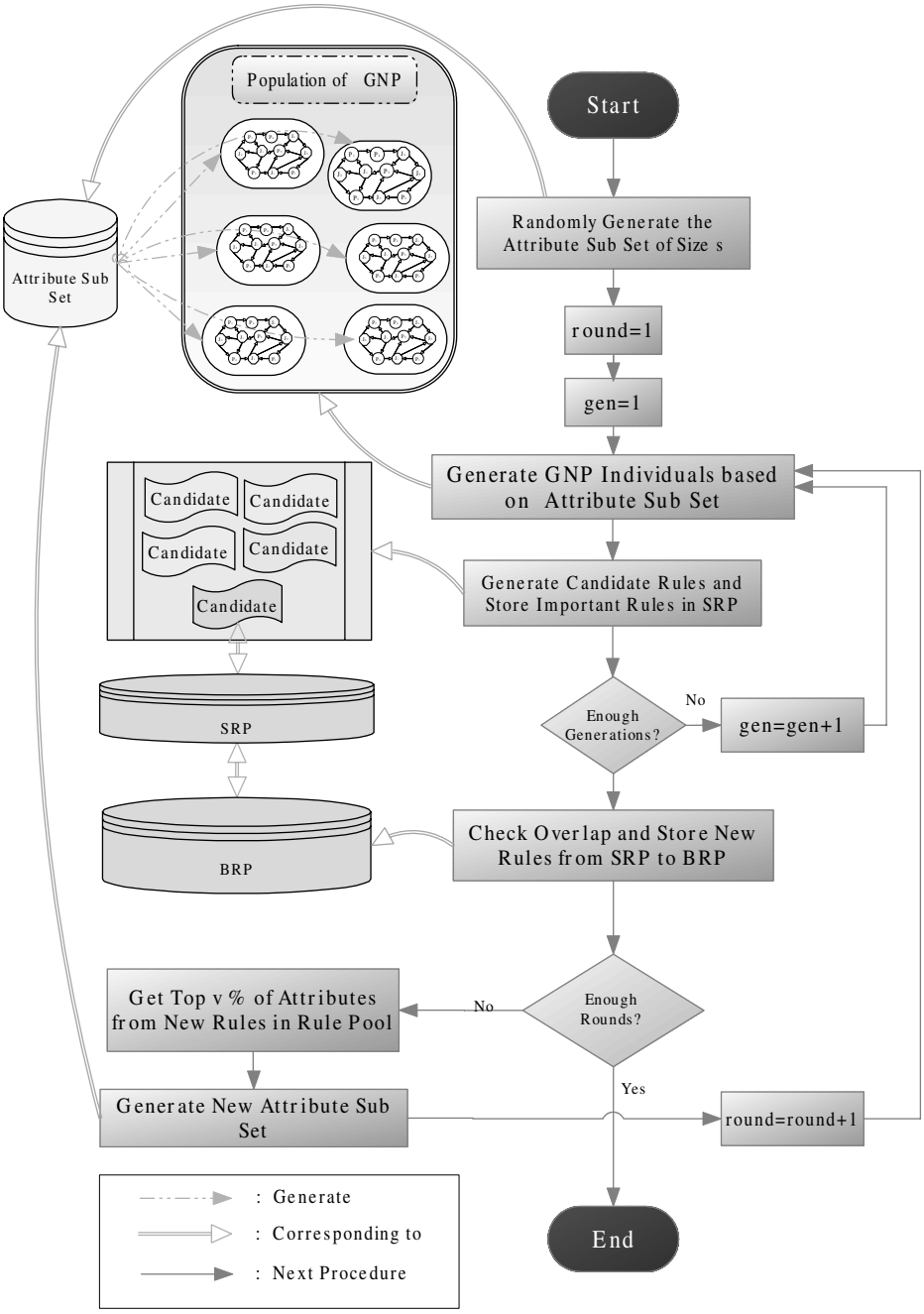


Figure 2.5: Flowchart of time related GNP data mining

2.3 Simulation

In this section, the effectiveness and efficiency of the proposed method are studied by simple traffic simulations.

2.3.1 Traffic Simulator

The main task of our simulator is to generate the databases to which we apply the proposed method.

Each section between two intersections in the road has two directions, and we assume each direction of the section represents different literals, i.e., items or attributes. The traffic simulator used in our simulations consists of the road model with 7×7 roads like Fig.2.6, i.e., each section has the same length, and the shape of the total road is like a grid network. Time shift in road setting of Fig.2.6 represents the time delay of the traffic lights between neighboring intersections.

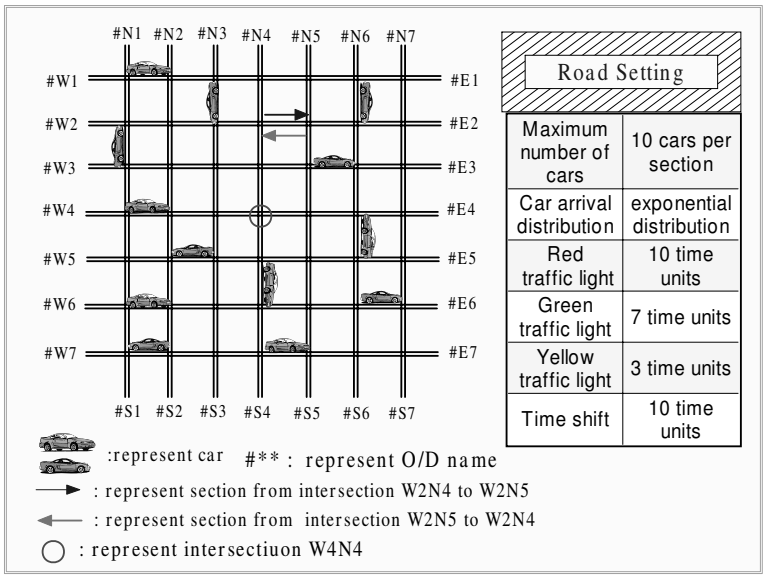


Figure 2.6: Road model used in simulations

Although the cars on the map share the same timer, they do not exactly have the same speed. Every time unit, all the cars can move forward by length 1 if and only if there exist spaces before them, thus the actual speed of all cars are influenced by the traffic lights or traffic jams. For example, if a car encounters the red light or traffic jam, it has to wait until the red light period passes or all the hindrances before it are moved. Therefore, the cars have different speeds depending on the concrete traffic situations.

Table 2.4: Database generated by simulator

Time	$W1N1, W1N2$			$W1N1, W1N3$		
	Low	Middle	High	Low	Middle	High
0001	0	0	1	1	0	0
0002	0	1	0	0	1	0
0003	0	0	1	1	0	0
0004	1	0	0	0	0	1

Middle threshold=4; High threshold=7;

Table 2.5: Example of OD (Origin/Destination)

O \ D	$\#N1$	$\#N2$	$\#N3$	$\#N4$
$\#N1$	...	7	1	8
$\#N2$	12	...	7	5
$\#N3$	8	0	...	6
$\#N4$	2	1	9	...

The database generated by our simulator is shown in Table 2.4. In this table, "$W1N1, W1N2$" and "$W1N1, W1N3$" represent sections in our simulator. The average traffic density of each section is discretized to Low/Middle/High groups, e.g., the average traffic density of section "$W1N1, W1N2$" has the value of 8 at time unit 0001, which means it belongs to the group High at time unit 0001 considering the thresholds.

The generation of cars is based on O/D (Origin / Destination) shown in Table 2.5. For example, in Table 2.5, the " #N1" is the name of a starting/end point, and the numerical value of 12 in the table means that the car traveling from the point named "$\#N2$" to the point named"$\#N1$" has the traffic flow of 12 vehicles per time unit. The

Table 2.6: Parameter setting for evolution

Items	Values
Number of judgment nodes	100
Number of processing nodes	10
Number of attributes	672
Number of time units	800
Number of generations per Round	50
Sub attribute set size	100

car traveling from the starting point to itself is forbidden here.

The parameter setting of the proposed data mining is presented in Table 2.6. Attributes here are the judgment node functions, for example, an attribute named "$W4N6$, $W4N7(Low)$" can be interpreted as section "$W4N6$, $W4N7$" has low traffic. We have $7 \times 8 \times 2$=112 sections here in our simulator and each section has two directions, so, there exist $112 \times 2 = 224$ sections in total. What's more, each section has 3 categories (Low/Mid/High), thus we have $224 \times 3 = 672$ attributes. Time units in Table 2.5 represents the number of total time units in our database. Simulator runs for 5000 time units, but, removes the first and last 500 time units for stabilization from the proposed data mining calculation. The samples are taken every 5 time units, i.e., the number of time units is (5000-500-500)/5= 800, thus 800 time units are used in the following simulations.

2.3.2 Simulation results

The aim of our data mining is to extract rules on the time-related association relations among all of the sections on the map, and our method can be applied to the database with a large number of attributes, e.g., 672 sections in our simulations.

<Test Case 1>

In test case 1, the number of rules stored in the Big Rule Pool(BRP) is compared between the proposed method and the conventional method without the attribute accumulation mechanism. Each round has the same number of generations of 50 and the selected set size is 100.

Fig.2.7 shows the number of rules obtained in the BRP versus round number. In the conventional method in Fig.2.7, GNP individuals are randomly initialized from the whole attribute set at the beginning of each "round". We can see from Fig.2.7 that the proposed method can extract important association rules efficiently, when compared with the conventional one.

The rule extracted is like the following:

$W4N5, W4N6, Low(t = 0) \Rightarrow$

$W4N5, W3N5, High(t = 9) \wedge W3N5, W3N6, High(t = 11)$

The above rule means that the section on the road named "$W4N5, W4N6$" has low traffic at time 0, then the section named "$W4N5, W3N5$" will probably have high traffic density at time 9, and the section named "$W3N5, W3N6$" will also possibly have high traffic density at time 11.

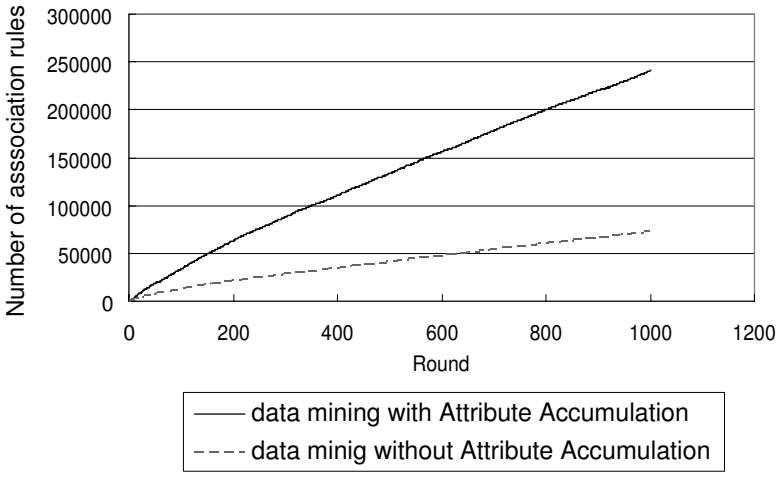


Figure 2.7: Comparison of the number of rules between the proposed method and conventional method

As Fig.2.8 describes, the above rule represents the trend that more cars will choose the section "$W4N5, W3N5$" and "$W3N5, W3N6$", which is triggered by the past low traffic density of the section "$W4N5, W4N6$". According to this information, we can update the routing algorithm more accurately.

<Test Case 2>

The aim of the second simulation is to study the relations between the selected attribute set size and the performances of the algorithm. With other parameter setting being unchanged, the attribute set size of 25, 50, 100, 125, 150, 200, 400 and 672 has been studied. The result is shown in Fig.2.9.

From Fig.2.9 we can see that the performance of the attribute set size 672 is almost the same as the result of the conventional method in Fig.2.7, since the number 672 is the total attributes size.

We can also see from Fig.2.9 that both too large or too small attribute set size will harm the performance of the algorithm and the best solution should be selected according to the concrete problem to solve. In our cases, the best solution is obtained when we use the attribute set size of 50.

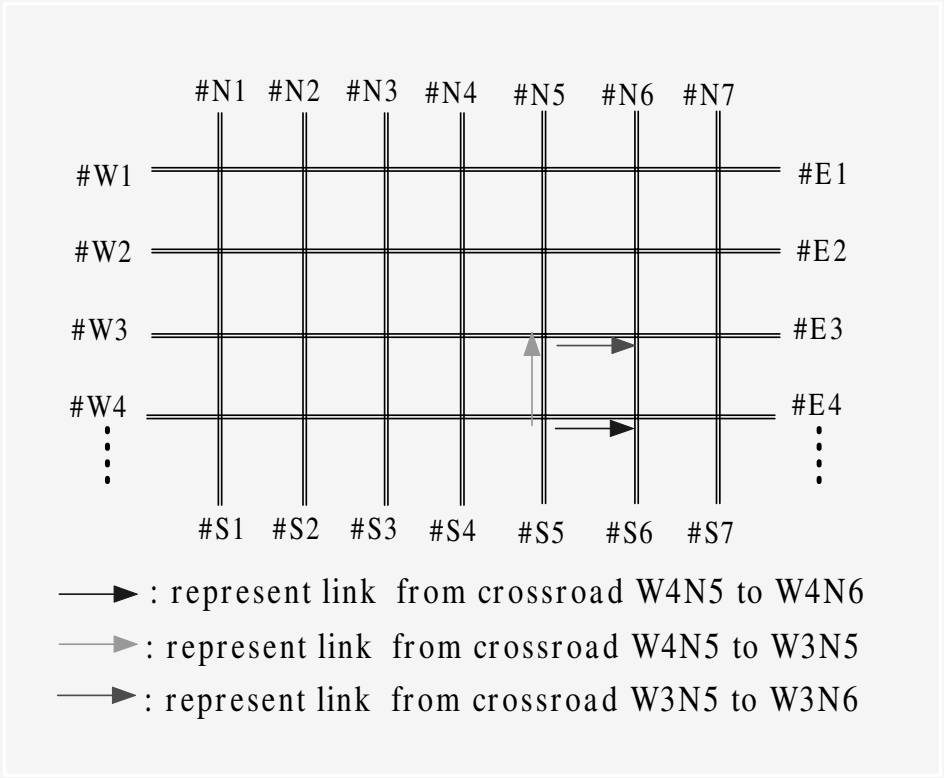


Figure 2.8: Map and rule

<Test Case 3>

The third simulation is to explore the relations between the generation size per round and the performances of the algorithm. With other parameter setting being unchanged, the performances of the algorithm with generation size of 25, 50, 100 and 150 are compared. The result is shown in Fig.2.10.

The average number of finding new rules per generation is shown in Fig.2.11 when changing the generation size per round. Since the time consumed in one generation is almost the same in four cases, the average number of extracted rules per generation in Fig.2.11 actually represents the time complexity of four cases.

The larger generation size certainly contributes to finding more rules, however, the time complexity increases. Considering this trade-off and that the different round has

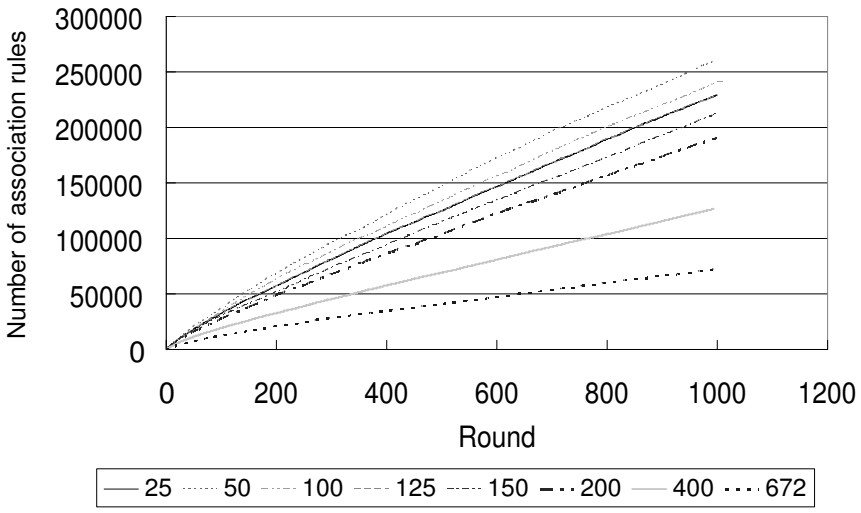


Figure 2.9: Number of rules in case of changing the attribute size

different convergence rate, the fixed size of generations per round is not a good method, which should be studied more in the future.

<Test Case 4>

The main purpose of the last simulation in this chapter is to study the effects of changing the accumulation percentage. With other parameter setting being unchanged, the top 10%, 20%, 40%, 50% and 70% attributes of the new rules in BRP are used as the attribute sub set of the next round. Here, the top attributes mean the attributes with the most frequent occurrence in BRP. The results are shown in Fig.2.12.

The figure in the upper side of Fig.2.12 represents the number of rules obtained from the first to 1000^{th} round. It is found from the figure that there is no huge difference among each other, since all of them are capable of obtaining a large number of association rules. However, we can examine the difference more clearly from the figure shown in the lower side of Fig.2.12, where the number of rules generated in the final 100 rounds is shown. We can see from the figure that both too large or too small accumulation percentage will harm the performance of the algorithm. And the best solution is obtained around 40% in our simulation.

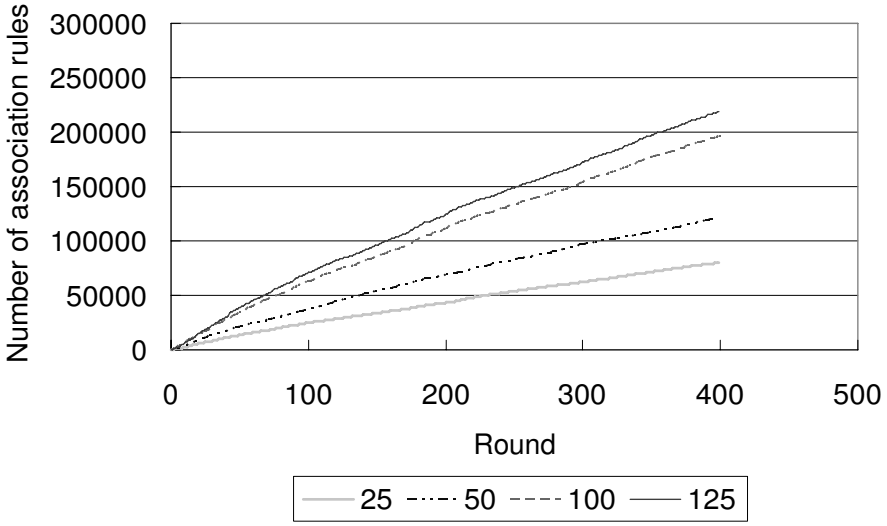


Figure 2.10: Number of rules in case of changing the generation size

2.4 Conclusions

In this chapter, a method of association rule mining using Genetic Network Programming with time series processing mechanism and Attribute Accumulation Mechanism (AAM) has been proposed. The proposed method can extract important time-related association rules efficiently. Extracted association rules are stored temporarily in Small Rule Pool(SRP) and finally all together in Big Rule Pool(BRP) through rounds of generations. These rules are representing useful and important time related association rules to be used in the real world. We have built a simple road simulator and examined the effectiveness and usefulness of the proposed algorithm. The results showed that the proposed method extracts the important time-related association rules in the database efficiently and the attribute accumulation mechanism improves the performance considerably. These rules are useful in time-related problems, for example, traffic prediction.

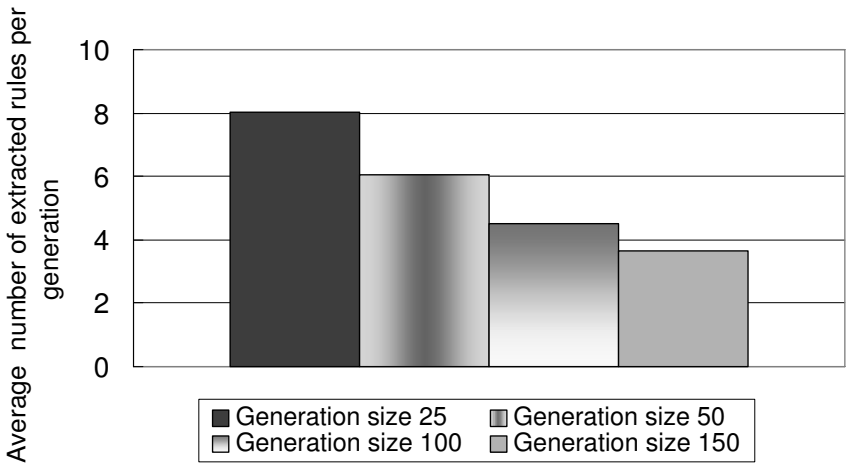


Figure 2.11: Average number of rules extracted per generation

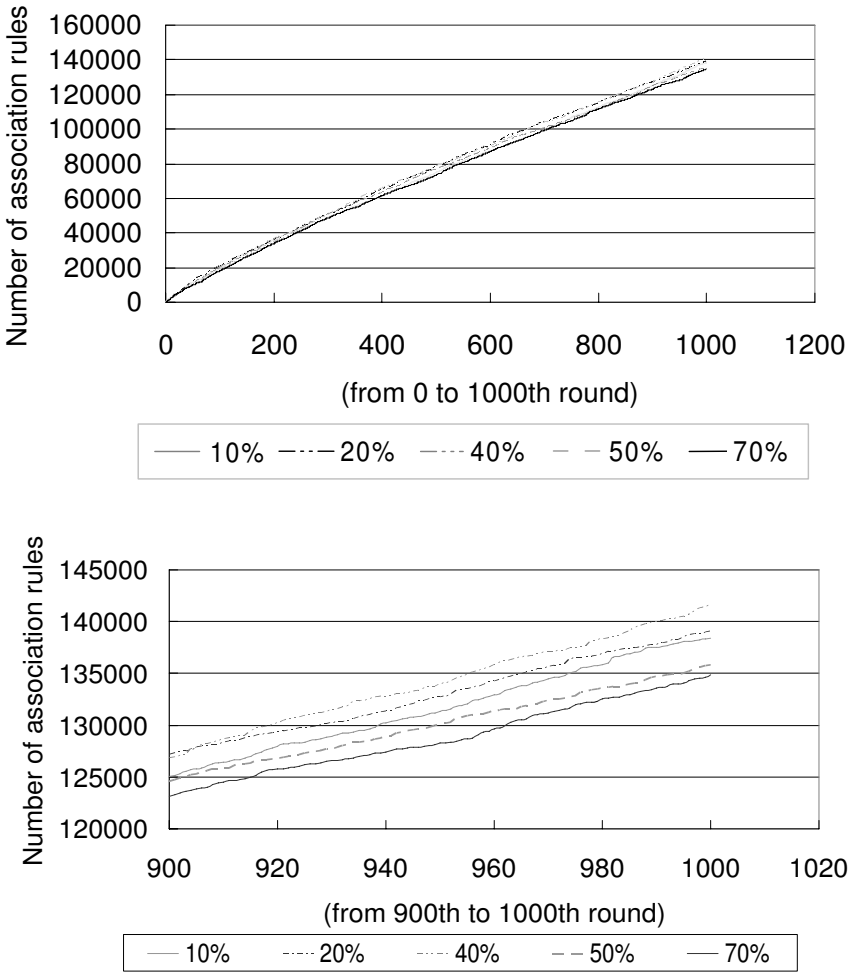


Figure 2.12: Number of rules in case of changing the accumulation percentage

Chapter 3

Class Association Rule Mining and Classification in Traffic Density Prediction

3.1 Introduction

In this chapter, a time related class association rule mining method is applied to estimate the real time temporal/sequential traffic situations. Although our lab. have already reported Genetic Network Programming(GNP) based data mining [24], which is a transaction related method, it means that every tuple in the database is not related to time series, where GNP is a newly developed evolutionary method whose chromosome is made of directed graphs [9][10][26].

The proposed method uses the time related database which has been introduced in Chapter 2.2, and it has the following features for the time related association rule mining and time related classification in traffic density prediction.

- One feature of the proposed method is that the proposed method can predict not only the traffic jam of a specific section on the road networks, but also can predict the low, middle and high traffic density of all of the sections on the road network, which might be useful in traffic applications.
- Extraction Mechanism at Stages(EMS) has been proposed to extract a fixed sufficient number of rules N_f for each class of each consequent attribute, i.e., each traffic density level of each section of road networks.
- In order to obtain a sufficient number of class association rules for each class, a kind of self-adaptive mechanism for decreasing the importance criteria and time addition mechanism has been proposed.

- Two unique timing methods have been proposed for matching the current traffic data with the time related association rules to improve the prediction accuracy.

The proposed rule based class association rule mining contains two stages: training and testing. In the training stage, the important rules are generated for classification by using the training data with class information. Then, the obtained rules are applied to the classification of the testing data.

The chapter is organized as follows: In section 3.2, the algorithm of the time related class association rule mining and classification using GNP is described. Section 3.3 shows the simulation environments, conditions and several experimental results. Section 3.4 is devoted to conclusions.

3.2 Time Related Class Association Rule Mining using GNP with EMS and AAM

3.2.1 Outline of the Proposed Method

The whole procedure of the proposed algorithm is like Fig.3.1. The first step is the rule extraction step. In rule extraction procedure, two kinds of iterations are included, i.e., the outer iteration represents the class association rule mining, which means to extract the time related class association rules of all the consequent attributes with Attribute Accumulation Mechanism(AAM) and Extraction Mechanism at Stages(EMS), while the inner iteration represents the basic procedure of the time related data mining, which uses GNP individuals to generate the candidate rules, then calculate the support, confidence and chi-squared values of the candidate rules based on the time related databases[25][27][28][29][30].

In order to testify whether the obtained time related rules is accurate or not, the second step is to use the extracted rules to predict the future events using the degree of matching between data and antecedent part of the rules in each class, and the class with the highest degree of matching will be assigned to data[24]. If the classification result is the same as the actual situation, then the prediction is correct, otherwise incorrect, and we can obtain the accuracy of the prediction in this way.

3.2.2 Association Rules

The following is a formal statement of the problem of mining association rules. Let $A = \{A_1, A_2, \ldots, A_k\}$ be a set of events, called items or attributes, e.g., it can represent the traffic density level(Low/Middle/High)of each section on the traffic network in traffic density prediction problem. Let G be total number of time points in the time

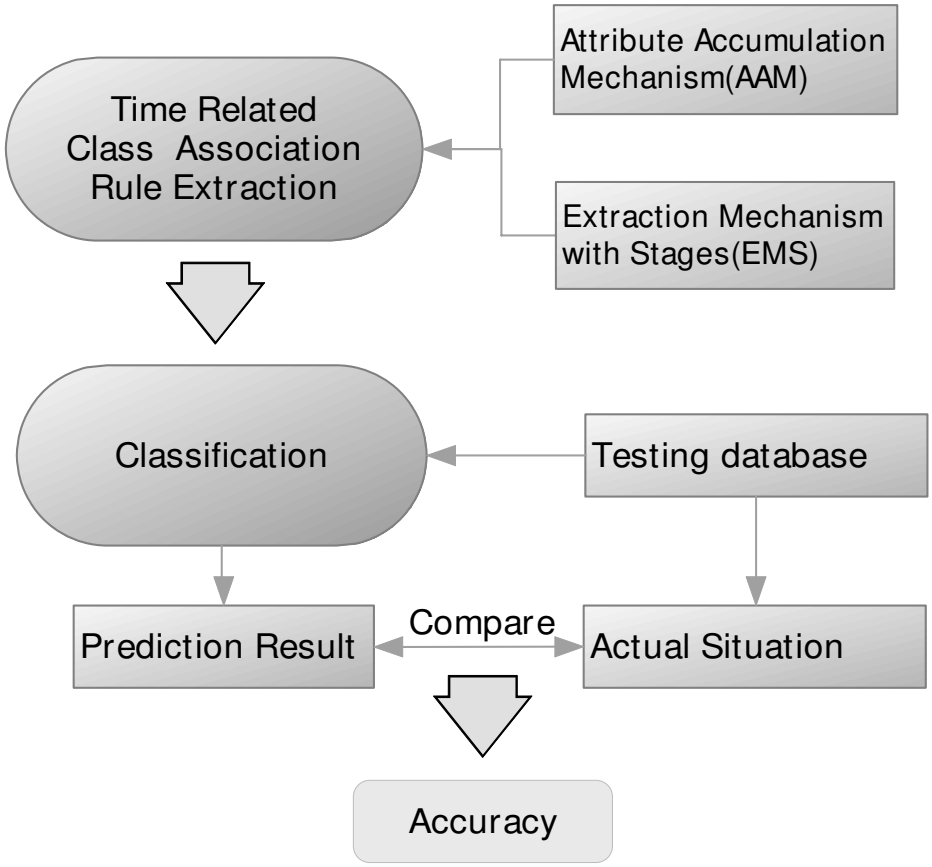


Figure 3.1: Basic steps of proposed algorithm

related traffic density database. Each time point is associated with a unique identifier whose set is called $TimeID$. Let D=$\{D_1, D_2, \ldots, D_k\}$ be a set of time points according to the time when the corresponding event(attribute)occurs, e.g., if attribute A_i occurs at time unit 0003, then D_i will be recorded as $D_i = 0003$. Now, the set of time related attribute A_i is defined as $A_t = \{A_{1(D_1)}, A_{2(D_2)}, \ldots, A_{k(D_k)}\}$.

Based on the definition, a time related association rule is an implication of the "$X \Rightarrow Y$" where $X \subseteq A_t$, $Y \subseteq A_t$. Then, X is called antecedent and Y is called consequent of the time related association rule.

In general, a set of items in A_t is called time transition. Fig.3.2 shows an example of the time transition. Each transition has its own associated measure of statistical significance called support. If the number of time transitions containing X in G equals t, and the total number of time transitions in G is N, then $support(X) = t/N$. The rule $X \Rightarrow Y$ has a measure of its strength called confidence defined as the ratio of

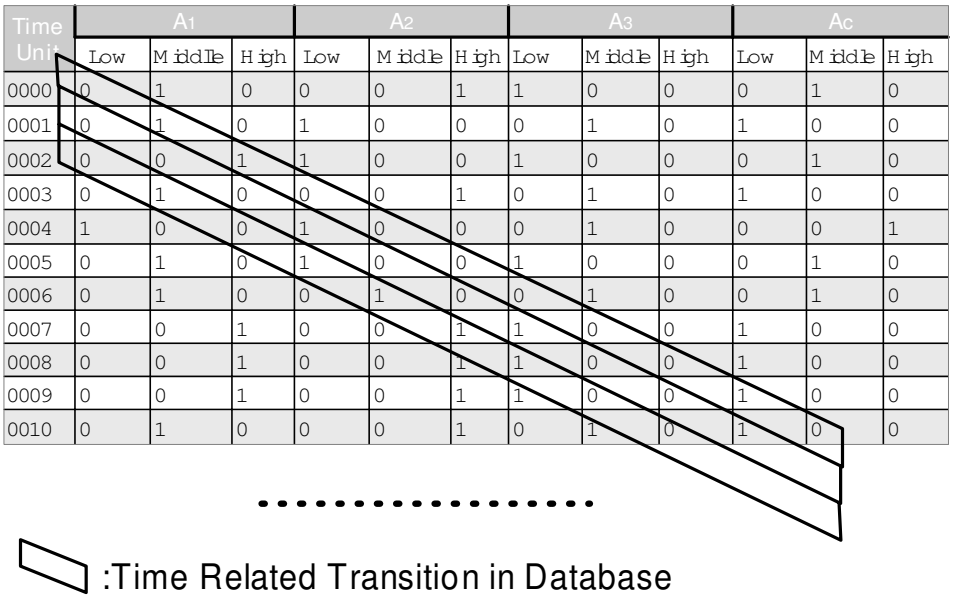


Figure 3.2: Time Transition

$support(X \cup Y)/support(X)$. Calculation of the chi-squared value of the rule $X \Rightarrow Y$ is described as follows. Let $support(X) = x$, $support(Y) = y$, $support(X \cup Y) = z$. If the events X and Y are independent, we can get $support(X \cup Y) = xy$. Table 3.1 is the contingency of X and Y ; the upper parts are the expectation values under the assumption of independence, and the lower parts are observational values.

Now, let E denote the value of the expectation under the assumption of independence. O is the value of the observation. Then, the chi-squared value is defined as follows:

$$\chi^2 = \sum_{AllCell} \frac{(O - E)^2}{E}. \tag{3.1}$$

The *AllCell* in the Eq.3.1 represents the four sections like XY, $X\neg Y$, $\neg XY$ and $\neg X\neg Y$ in the contingency Table 3.1.

We can calculate the chi-squared value using x, y, z and N of Table 3.1 as follows:

$$\chi^2 = \frac{N(z - xy)^2}{xy(1 - x)(1 - y)} \tag{3.2}$$

This has 1 degree of freedom. If it is higher than a threshold value (3.84 at the 95% significance level, or 6.63 at the 99% significance level), we should reject the indepen-

Table 3.1: The contingency of X and Y

	Y	$\neg Y$	$\sum_{row}$
X	Nxy Nz	$N(x-xy)$ $N(x-z)$	 Nx
$\neg X$	$N(y-xy)$ $N(y-z)$	$N(1-x-y+xy)$ $N(1-x-y+z)$	 $N(1-x)$
$\sum_{col}$	Ny	$N(1-y)$	N

(N: the number of time points (= $|TimeID|$))

dence assumption.

3.2.3 Structure of Rules

Let $A_i(*)(t=p)$ be an attribute in a database at time p and its value is 1 or 0. $A_i(*)$ represents $A_i(Low)/A_i(Mid)/A_i(High)$. The time related class association rule mining extracts the following association rules:

$$(A_j(*)(t=p)=1) \wedge \cdots \wedge (A_k(*)(t=q)=1) \Rightarrow (A_c(*)(t=r)=1),$$

where $(A_c(*)(t=r)=1)$ indicates the class of the consequent attribute.

Here, $p \leq q \leq r$, and the first t always equal 0 and other time points are the relative time shifts from the first attribute.

3.2.4 Time Related Class Association Rule Mining using GNP

Liu et al have proposed a method to integrate the classification rule mining and association rule mining [31]. The algorithm is called CBA (Classification Based on Associations). Two techniques are integrated by focusing on mining association rules whose consequent is restricted to a certain classification class attribute. The rules are called class association rules (CARs).

Building an accurate and efficient classifier model becomes essential subjects in the proposed Time Related Class Association Rule Mining method. The implicit interesting temporal/sequential patterns in the training data are fundamental to build a classifier model, which could predict the unknown classification attributes.

The proposed method extract important time related CARs using GNP supposing that the connections of judgement nodes are represented as association rules. The basic structure of it is also shown in Fig.3.3. The jump mechanism is also used in order to determine only the antecedent part, but we don't have to decide the consequent part of the candidates rules any longer. All the items of the consequent part are placed in the

Consequent Table(CT) shown in Fig.3.3. In addition, a sufficient number of rules for each item in the CT are obtained in the proposed method.

Fig.3.3 represents the transition starting from processing node P_1. The connection of the judgement nodes starting from P_1 represents possible candidate time related association rules. The transitions starting from processing $P_2, P_3, \ldots$ have been omitted in Fig.3.3 for simplicity. The network structure of the GNP makes it possible that different processing nodes can reuse the same judgement node, thus GNP has a compact structure.

Each processing node works as a starting point for calculating criteria of the candidate time related association rules, and if the no-side of the judgement node is taken, the next processing node automatically starts as shown in Fig.3.3.

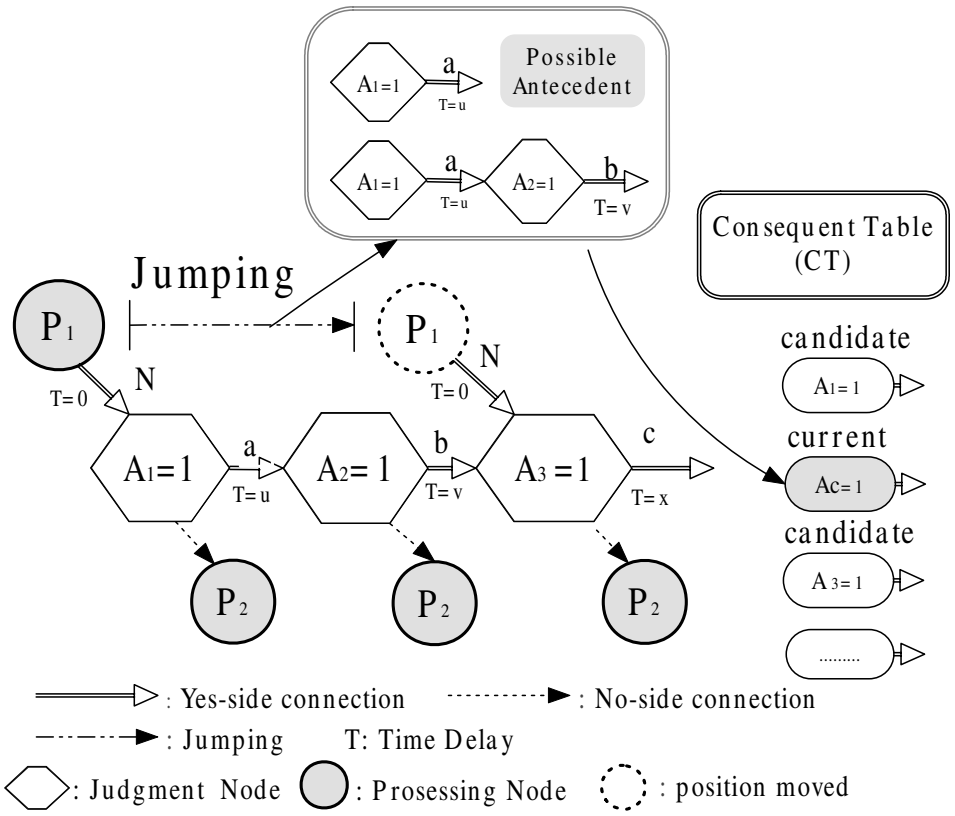


Figure 3.3: The basic structure of time related general association rule mining and time related class association rule mining

In the proposed method, the examination should consider both the attribute dimension and time dimension simultaneously, thus the method is fundamentally two dimensional. That is, not only the attributes but also the time delays should be consid-

ered: We will explain an example using the attributes of $A_1(Mid)$, $A_2(High)$, $A_3(High)$ and $A_4(High)$ as described in Fig.3.4. The judgment is not merely executed row by row, but the procedure is like the following: firstly judge the tuple at time 0000, and according to GNP individual structure of Fig.3.4, first $A_1(Mid)$ is judged and if the value of $A_1(Mid)$ at time 0000 is $'1'$, then move to the next judgement node named $A_2(High)$. Then, due to the time delay T =2 from $A_1(Mid)$ to $A_2(High)$, we check the value of $A_2(High)$ at time 0000+2=0002. The current classification consequent $A_c(Mid)$ at time 0000+2=0002 is also checked for the candidate rule of
$(A_1(Mid)(t = 0) = 1) \Rightarrow (A_c(Mid)(t = 2) = 1)$
using the time delay addition mechanism, that is, using the time delay after $A_1(Mid)$. If the value of $A_2(High)$ at time 0002 is '1', continue the judgment likewise, and continue to add the time delay after $A_2(High)$, i.e., $T = 2$ to check $A_c(Mid)$ at time 2+2=4 for the candidate rule of
$(A_1(Mid)(t = 0) = 1) \wedge (A_2(High)(t = 2) = 1)$
$\Rightarrow (A_c(Mid)(t = 4) = 1)$

The time delay addition method means that the consequent part is consistently checked at every possible node transition, thus during one database searching, there will be N_a possible candidate rules to be checked simultaneously, where, N_a represents the length of the antecedent part.

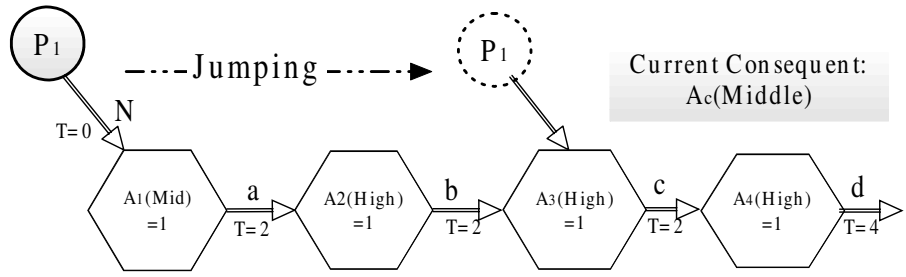
As show in Fig.3.4 while carrying out the two dimensional searching for rule 3, the counts moving to the Yes-side of rule 1 and rule 2 for the current consequent can be obtained simultaneously.

Conventionally, one database searching just generates one possible association rule, while N_a association rules can be generated using the time delay addition method, thus the time delay addition method increases the efficiency of the rule extraction.

The procedures something like these continue to execute another turn of the judgments which begins from time 0001, 0002, 0003, ... until the end of the tuple. By this kind of time delay addition method, we can check many candidate rules simultaneously within one jump, thus increase the efficiency of the process. The example is shown in Fig.3.4, where the current consequent is $A_c(Mid)$. When checking the count of moving to the Yes-side about rule 3, both rule 1 and rule 2 can be checked simultaneously by the time delay addition mechanism.

The searching process shown in Fig.3.4 exclusively concentrates on the searching process of processing node P_1. If the searching in the current time transition of P_1 failed, the searching automatically moves to the search starting from P_2 for the next time transition. In the other words, for each time transition, all of the possible candidate time related association rules are checked successively.

The total number of moving to Yes-side from the processing node at each judgment



Time Unit	A1			A2			A3			Ac		
	Low	Middle	High	Low	Middle	High	Low	Middle	High	Low	Middle	High
0000	0	1	0	0	0	1	1	0	0	0	1	0
0001	0	1	0	1	0	0	0	1	0	1	0	0
0002	0	0	1	0	0	1	1	0	0	0	1	0
0003	0	1	0	0	0	1	0	1	0	1	0	0
0004	1	0	0	1	0	0	0	0	1	0	0	1
0005	0	1	0	1	0	0	0	0	1	0	1	0
0006	0	0	1	0	1	0	0	0	1	0	1	0
0007	1	0	0	0	1	0	0	1	0	1	0	0
0008	0	1	0	1	0	0	1	0	0	0	1	0
0009	1	0	0	0	1	0	0	1	0	1	0	0

:Two dimensional checking point for antecedent

:Checking point for consequent

Condidate Rules	Time Delay				
Rule 1. A1(Mid) => Ac(Mid)	0	2	-	-	-
Rule 2. A1(Mid) ^ A2(High)=> Ac(Mid)	0	2	4	-	-
Rule 3. A1(Mid) ^ A2(High) ^ A3(High) => Ac(Mid)	0	2	4	6	-

Figure 3.4: Two dimensional searching method

node is calculated for every processing node, which is a starting point for calculating association rules. In Fig.3.4, N is the number of the total searches, and a, b, c and d are the numbers of the search moving to the Yes-side at each judgment node. The measurements are calculated by these numbers.

For example, as represented in Fig.3.4, the processing node P_1 jumps from $A_1(Mid)$ to $A_3(High)$, then the antecedent parts of the candidate rules becomes "$A_1(Mid)$", "$A_1(Mid) \wedge A_2(High)$" and "$A_1(Mid) \wedge A_2(High) \wedge A_3(High)$" and the current item in the Consequent Table(CT) is, for example, "$A_c(Mid)$". As a result, $chi-squared$ value can be calculated considering both the antecedent and consequent part of the candidate rules. We can repeat this in each generation by jumping the processing nodes in the time related class association rule mining using GNP. Thus, we can obtain the values for calculating the importance of the rules. Now, we define the important association

rules as the ones which satisfy the following:

$$\chi^2 \geq \chi^2_{min}, \tag{3.3}$$

$$support \geq sup_{min}, \tag{3.4}$$

$$confidence \geq conf_{min}, \tag{3.5}$$

where, χ^2_{min}, $\sup_{min}$ and $conf_{min}$ are the thresholds of the minimum chi-squared, support and confidence values given by supervisors. The extracted important association rules are stored in a pool all together through generations in order to find new important rules.

The threshold values are determined depending on concrete applications. Based on experimental parameter tuning results, chi-squared threshed value is usually set at 6.63 to ensure 99% significance level and the confidence value more than 0.8 is appropriate to maintain high prediction results in the model. The threshold for support values should be flexible depends on the databases.

3.2.5 Attribute Accumulation Mechanism(AAM) and Extraction Mechanism at Stages(EMS)

The conventional method extracts association rules using the whole attribute set without sub attribute sets and rounds. Since the whole attribute set we deal with is very large, which attributes are used for initializing individuals will largely influence the efficiency of the whole evolution process. When we use the Attribute Accumulation Mechanism(AAM) in order to deal with databases which have a large number of attributes, GNP individuals accumulate better attributes in it gradually round by round, where each round is a sequence of generations[32] as described in chapter 2.

The attribute accumulation mechanism reviewed here first randomly selects a small attribute set of size si from the whole attribute set of size $Si(Si \geq si)$, then applies GNP-based data mining algorithm using GNP individuals which are generated exclusively from the chosen attribute set and finally stores the extracted association rules in the corresponding rule pool using hash functions[32]. This procedure is so called Round 0.

After the processing of Round 0, we get the rules in the corresponding rule pool named SRP(0). For each of the rules stored in SRP(0), we check its overlap with others and sum up the counts of the appearance of each attribute in the chosen attribute set, and using the sum of the counts, the attributes are sorted from the most frequently used one to the least one. Only the top 20%(could be flexible according to the concrete

problem) attributes can be remained in the chosen attribute set. The attribute set of the next Round 1 is then composed of the top 20% of the attributes and attribute set randomly chosen from the original whole attribute set. Using the newly generated set, Round 1 searches the important association rules and stores the newly generated rules in the rule pool as described in chapter 2.

In further extension of AAM, the Extraction Mechanism at Stages(EMS) is proposed to extract time related class association rules, while each stage of the EMS is in charge of extracting the rules exclusively for each class of each section on the road networks using AAM, which implies that one stage consists of at least one round to obtain enough number of time related association rules.

Not only class association rules for one consequent attribute, i.e., one section on the traffic network, is extracted, but also, all the class association rules for all of the consequent attributes in the database, i.e., all of the sections on the traffic network are obtained by EMS. Using EMS method, the overall prediction model can be obtained for all necessary prediction requirements, and the navigation system could refer to the prediction model for the calculation of the real-time optimal route. Once the navigation system finishes the calculation of the optimal route, we can use the current traffic densities to predict again whether the sections of the optimal route will have a traffic jam.

Suppose that there exists three different classes for each section: Low/ Middle/High traffic density levels. When extracting rules, it is very important to extract enough number of time related rules for every possible classes, otherwise the information could not be sufficient for accurate classification, i.e., if the rules corresponding to Low class are missing, the remaining rules would lead to very low prediction accuracy due to incompleteness of information.

Therefore, the proposed method aims at getting enough number (N_f) of class association rules for each class of all attributes in order to acquire the high prediction accuracy of the classification. If the number of the extracted rules reaches a certain fixed number N_f for the current class of the consequent attribute, then the stage for the current class of the consequent attribute ends and another stage for the next item in the CT will begin, likewise.

Although the number of the rules, N_f, should be a sufficiently large value for all classes of the consequent attributes, sometimes it might occur that there do not exist enough number of rules for some classes of the consequent attributes at the current importance level. Thus, we proposed a method for decreasing the thresholds of the importance, where the criteria for important rules become self-adaptive and they are adjusted gradually according to the number of new rules extracted during the recent generations. If the number of new rules generated during the recent generations equal to 0, which means no more important rules can be found for the current class of the

consequent attribute in the Consequent Table(CT), the criteria for important rules are changed using the following equation until a certain fixed number N_f of rules are obtained .

$$\chi^2_{min} \leftarrow \chi^2_{min} \times s_1, \tag{3.6}$$

$$sup_{min} \leftarrow sup_{min} \times s_2, \tag{3.7}$$

$$conf_{min} \leftarrow conf_{min} \times s_3, \tag{3.8}$$

where, s_1, s_2 and s_3 belonging to $(0, 1)$ represent the step size of the decrease for the chi-squared, support and confidence values, respectively. χ^2_{min}, sup_{min} and $conf_{min}$ are initialized for each class of the consequent attributes of the Consequent Table(CT).

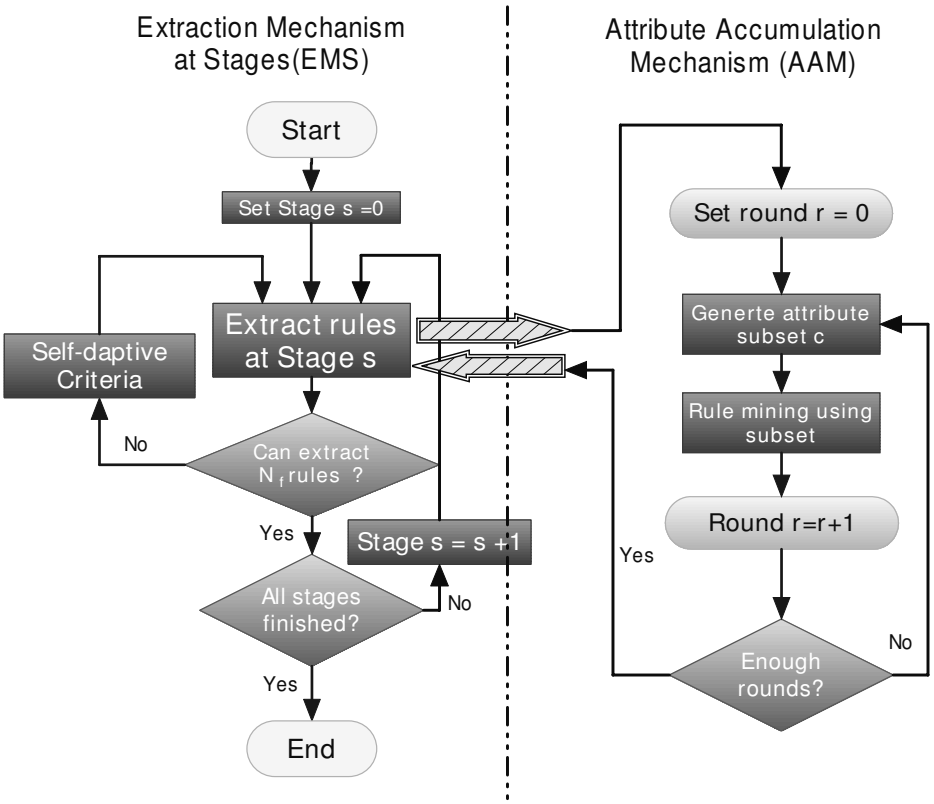


Figure 3.5: Flow chart of *AAM* and *EMS*

Extraction Mechanism at Stages(EMS) means a method for extracting a fixed num-

ber N_f of rules for each class of the consequent attributes, sequentially. EMS with Attribute Accumulation Mechanism(AAM) is used here to deal with consequent attributes sequentially.

As shown in Fig.3.5, the main flow chart consists of two kinds of iterations, the flow chart on the left side represents the main process of Extraction Mechanism at Stages(EMS) and each stage of EMS is in charge of exclusively extracting time related association rules for the current class of the consequent attribute. If enough number(N_f) of rules can not be obtained at the current stage, EMS uses the self-adaptive criteria.

The iteration procedure on the right side of Fig.3.5 describes AAM process, which is embedded in the rule extracting phase of EMS, thus each stage of EMS includes some rounds, or at least one round in AAM. Each round of AAM accumulates better attributes in the subset of the whole attribute set, which has the potential ability of generating more association rules, and those better attributes will have higher probability to be used in individuals of the later rounds. Thus, AAM is a process to strengthen the exploitation of the evolution.

The proposed GNP-based mining method to extract the association rules for each class of the consequent attributes in the Consequent Table(CT) is shown in Fig.3.6, where GNP based data mining consists of a sequence of generations.

For example, the first stage is in charge of extracting class association rules using $A_1(Low)$ as the consequent part. Attribute Accumulation Mechanism is the same as the one in the previous work[25], where GNP individual accumulates better attributes in it gradually round by round. At each stage, each round consist of a certain number of generations and has its accumulated chosen attribute subset, and the number of extracted rules will be checked at the end of each round. If the rules are not extracted, then a new round will begin with self adjusted criteria as shown before. Each stage ends if and only if a fixed N_f number of rules are extracted, thus a stage contains several rounds, at least, one round.

3.2.6 Classification

In order to test the prediction accuracy of the time related class association rules we extracted, these rules are applied to predict the traffic density in the future using a very simple classification mechanism.

Firstly we classify the extracted association rules in the pool to the classes of the consequent attributes. Every section attribute has three consequent attribute classes, i.e., *Low*, *Mid* and *High*.

Then, the association rules in each class of the consequent attributes are used to test whether the testing data satisfies the antecedent items of the rules based on two different kinds of timing mechanisms.

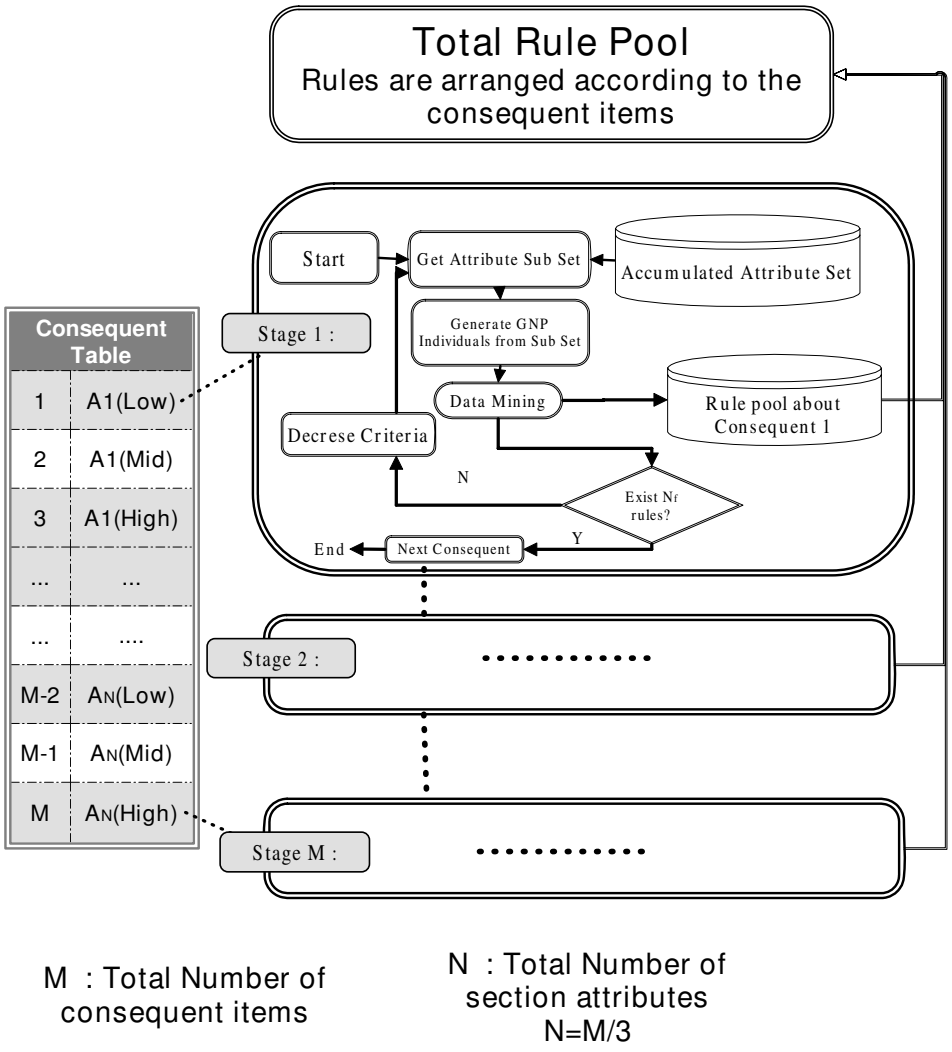


Figure 3.6: Procedure for extracting class association rules

The structure of the time related association rules represents the if-then type time sequential association relations and if the antecedent part of the rules satisfy the testing database, then it is probable that the consequent part will also occur at some time units later, therefore, the proposed method regards the antecedent part as the events already happened and the consequent part as the future events.

After defining the basic time span of the association rules, how to pinpoint the current time unit is essential to the time related classification mechanism. Two different mechanisms to choose the proper current time unit for each rule are proposed in this

model.

One of the mechanisms is just to set the time unit of the last attribute in the antecedent part as the current time unit, while another timing mechanism assigns the current time unit between the last attribute of the antecedent part and consequent attribute of association rules. These two mechanisms are described in Fig.3.7. In the N-Step prediction process, only the rules having exactly the N time delay between the last attribute in the antecedent part and consequent attribute are applicable in the method-1, while the method-2 sets the current time unit more flexibly.

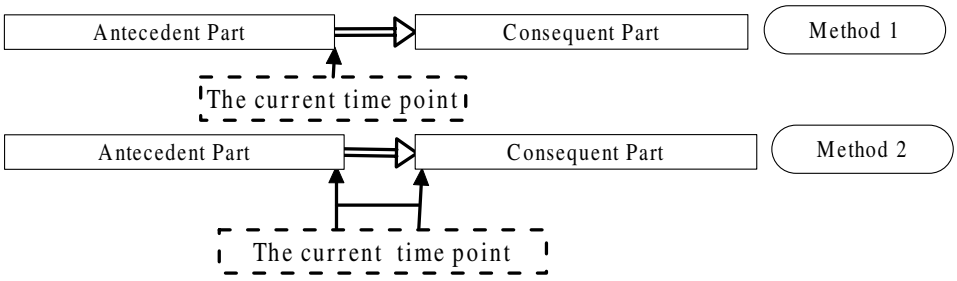


Figure 3.7: Two timing methods

For example, if we obtain the following time related class association rule:

$$(A_j(*)(t = p) = 1) \wedge \cdots \wedge (A_k(*)(t = q) = 1) \Rightarrow (A_c(*)(t = r) = 1),$$

the time difference between the last attribute of the antecedent part and consequent attribute is defined as Prediction Span(PS), thus we can define $PS = r - q$. If we want to predict the traffic density of N time units later, we call this prediction N-step Prediction, e.g., we want to predict the traffic density of 1 time unit later, this is called 1-step prediction.

Naturally, if we want to predict the traffic density of N time units later, we have to use the time related rules which satisfy: $PS \geqslant N$. Method-1 uses only the time related rules satisfying $PS = N$ to predict the traffic density of N time units later, while Method-2 uses all of the time related rules satisfying $PS \geqslant N$ to predict the traffic density of N time units later.

Instead of choosing only the rules with high quality, all of the matched rules with traffic data can contribute to the prediction.

The ratio of the number of matched rules to the total number of rules of each class of the consequent attributes is calculated, and the testing traffic data is classified to the class whose ratio is the highest. The concrete process is like the following:

(1). Calculate R_k: set of the suffixes of the rules in class k whose antecede attributes match the testing traffic data.

(2). Calculate $Credit_k$ in class k:

$$Credit_k = \sum_{r \in R_k} confidence_r, \tag{3.9}$$

(3). Calculate $Score_k$ in class k:

$$Score_k = \frac{Credit_k}{Total_k}, \tag{3.10}$$

where, $confidence_r$ is the confidence of rule r and $Total_k$ is the total number of rules in class k, that is, the fixed number of N_f in this chapter.

(4). Compare $Score_k$ and the class with the highest value becomes the winner for the testing traffic data.

3.3 Simulations

In this section, the effectiveness and efficiency of the proposed method are studied by a simple traffic simulation. Unlike other methods in the traffic prediction of recent years, we not only aim at predicting the traffic jam on a specific section of the road networks, but also, we are interested in providing the whole traffic prediction for all of the sections on the road networks so that the navigation system can refer to this information for the calculation of the optimal route.

For example, we can extract the time related class association rules using a large traffic database, which can provide a stable estimation of the traffic densities. And not only the traffic congestion/jam can be predicted, but also the sections with low traffic density can be predicted, which could be interesting since drivers might want to take a path with few cars. So, the uniqueness of the proposed method is that all of the traffic densities(Low/Middle/High) are predicted for all of the sections on road networks.

3.3.1 Traffic Simulator

Each section between two intersections in the road has two directions, and we assume each direction of the section represents different literals, i.e., attributes. The traffic simulator used in our simulations consists of the road model with 7×7 roads described in chapter 2, i.e., each section has the same length, all cars have the same speed for simplicity, and the shape of the total road is like a grid network[32].

Actually the cars on the map do not have the same speed. Every time unit, all the cars can move forward by length 1 if and only if there exist spaces before them, thus the actual speed of all the cars are influenced by the traffic lights or traffic jams. For

example, if a car encounters the red light or traffic jam, it has to wait until the red light period passes or all the hindrances before it are moved. Therefore, the cars have different speeds depending on the concrete traffic situations.

All the cars use the optimal route by Q value-based dynamic programming[33]. Although in the real situations, not all the cars use the same optimal algorithm in the traffic systems, but, in recent days, most of the cars in Japan have the optimal route searching mechanism. Time shift in road setting in Fig.2.6 represents the time delay of the traffic lights between neighboring intersections.

Table 3.2: Example of OD (Origin/Destination)

O \ D	#$N1$	#$N2$	#$N3$	#$N4$
#$N1$	...	7	1	8
#$N2$	12	...	7	5
#$N3$	8	0	...	6
#$N4$	2	1	9	...

Table 3.3: Parameter setting for evolution

Items	Values
Number of judgment nodes	100
Number of processing nodes	10
Number of attributes	672
Number of time units	800
Number of generations per round	100
Sub attribute set size	100
Fixed number of rules per class	100
Threshold for multiple rules	3
Mutation range of time delay	5

The generation of cars is based on O/D (Origin / Destination) shown in Table 3.2. For example, in Table 3.2, the " #N1" is the name of a starting/end point, and the numerical value 12 in the table means that the car traveling from the point named "#$N2$" to the point named"#$N1$" has the traffic flow of 12 vehicles per time unit. The car traveling from the starting point to itself is forbidden here.

The parameter setting of the proposed data mining is shown in Table 3.3. Attributes correspond to the judgment node functions, and an attribute named "$W4N6$, $W4N7(Low)$" can be interpreted as the section "$W4N6$, $W4N7$" has low traffic. We have 7×8×2=112

sections in our simulator and each section has two directions, so, there exist $112 \times 2 = 224$ sections in total. What's more, each section has 3 categories (Low/Mid/High), thus we have $224 \times 3 = 672$ attributes including classes. Time units in Table 3.3 represents the number of total time units in our database.

The adaptive step size of criteria s_1, s_2 and s_3 are set at 0.9 based on our experimental studies.

3.3.2 Simulation results

In the first part of the simulations, the fixed number N_f of rules are obtained using the proposed method for each class of the consequent attributes. N_f is set at 100 in order to get the complete picture of the whole databases as much as possible and each round of the searching has the same number of generations, e.g., 100 [32].

We tested our method on 5 databases generated from our simulator using the same OD table. But, 5 databases are different due to the randomness of their generation. Q-value based dynamic programming is used as the routing algorithm of each car. Class association rules with χ^2 test were extracted by GNP. Descretization of continuous attributes is done using the thresholds, e.g., the middle threshold is 4 and high threshold is 7. All the continuous attributes are transformed to a set of attributes, whose attribute is 1 or 0.

In order to test the efficiency of the proposed time related class association rule mining using GNP, the obtained rules for all consequent attributes are stored in the corresponding rule pools. Fig.3.8 shows the average number of rules obtained in the total rule pool using 5 databases versus round number.

In the conventional method without time delay addition mechanism, only two dimensional searching is used [32], which means no extra consequent points are checked during the searching for one candidate rule, as a result, the consequent part can be checked only after the transition of all antecedent part is finished. Thus, one searching just generates one possible association rule.

Results in Fig.3.8 shows that the proposed time delay addition mechanism extracts the important time related association rules from the database efficiently as the round goes on, since the proposed method can generate several possible association rules simultaneously during one searching.

An example of the rule extracted is like the following:

$$(W4N5, W4N6, Low(t = 0) = 1)$$
$$\wedge\ (W4N5, W3N5, High(t = 9) = 1)$$
$$\Rightarrow (W3N5, W3N6, High(t = 11) = 1)$$

The above rule means that if the section on the map named "$W4N5, W4N6$" has low traffic at time 0, and the section named "$W4N5, W3N5$" has high traffic at time 9, then,

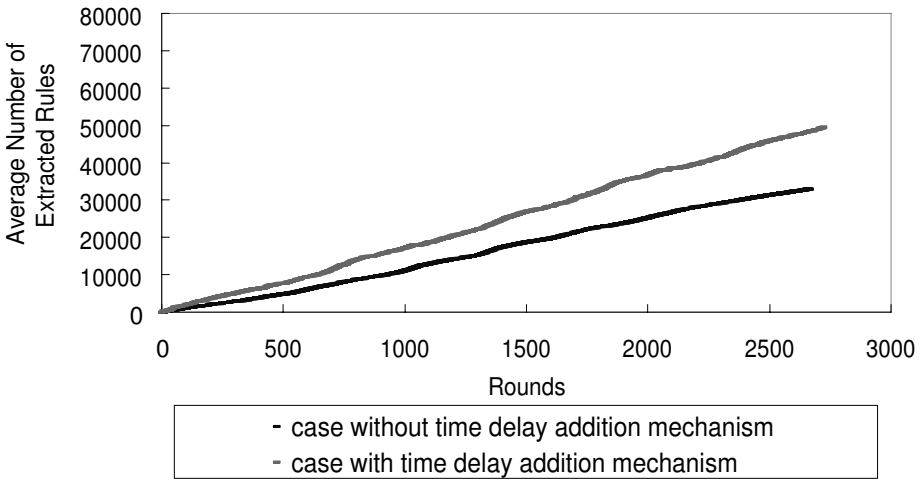


Figure 3.8: Number of the rules obtained using 5 databases

the consequent attribute, i.e., the section named "$W3N5, W3N6$" will have high traffic density at time 11.

Since all the extracted rules satisfy the conditions of importance, all of them have the qualification for the classifier. Therefore, we do not need to store, retrieve, prune or sort a large number of rules for classification as conventional methods.

In the second part of the simulations, the extracted rules are used for estimating the class of the testing traffic data, as a result, to which class the time-related data belong is determined. Two different kinds of timing mechanism are used in the simulations.

Firstly, we used 5 databases which are the same as the rule extraction.

The class association rules for one step prediction are generated based on each database, then method-1 is applied. The accuracy is defined in the following: if the traffic prediction result of the section at time t is "Low" and the real traffic of this section at time t is exactly "Low", then, the accuracy is 100%. The Low/Middle/High accuracy means the accuracy when the real traffic is Low/Middle/High, respectively. The results of the accuracy averaged over all 224 sections by method-1 is shown in Fig.3.9. It is found from Fig.3.9 that the middle accuracy is rather high compared to the low and high accuracy as expected. Method-2 is also examined in a similar way, and its results are shown in Fig.3.10. Fig.3.10 shows that almost the same prediction accuracy is obtained as Fig.3.9, but the accuracy of method-2 is a bit higher than method-1.

Next, in order to test the time related class association rules for one step prediction using the different databases from the ones used for rule extraction, 4-fold validation was carried out for 5 different data, and its average accuracy over all 224 attributes is

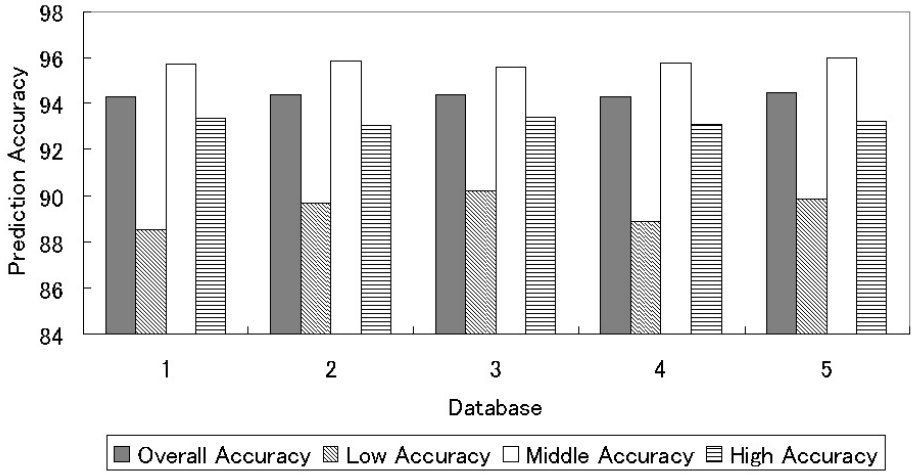


Figure 3.9: Result of Accuracy by Method-1

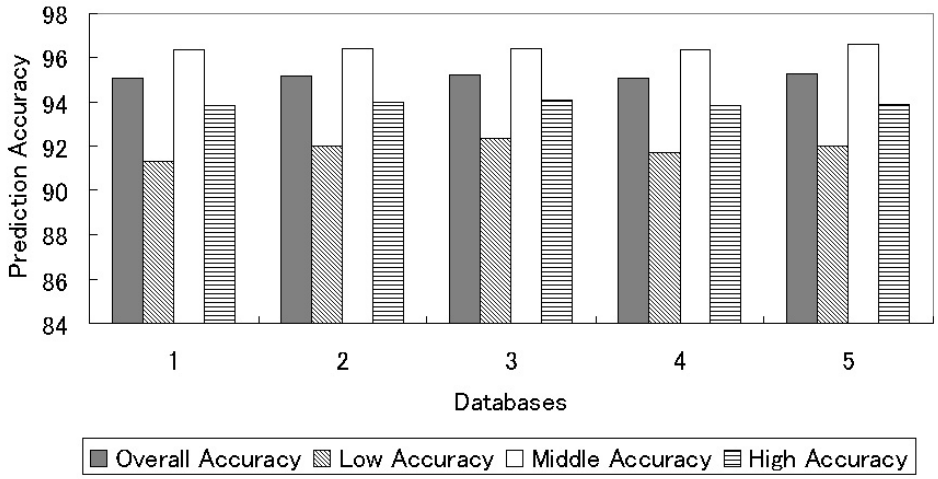


Figure 3.10: Result of Accuracy by Method-2

shown in Table 6, where D represents database index and M1 and M2 represent the method-1 and method-2, respectively. The comparison between two methods on the overall average accuracy using cross validation is shown in Fig.3.11.

It is found from Fig.3.11 that both of the two timing methods can get a relatively accurate prediction on the traffic of all of the 224 sections on the map, which means that the proposed method is useful in the traffic prediction problem. It is also found that the timing method-2 is slightly more accurate than method-1.

Table 3.4: Result of cross validation

	Overall		Low		Middle		High	
D	$M1$	$M2$	$M1$	$M2$	$M1$	$M2$	$M1$	$M2$
1	85.11	85.42	72.38	73.17	88.17	88.37	83.01	83.21
2	85.12	85.54	72.63	73.85	88.48	88.36	82.79	83.41
3	86.09	86.46	74.03	75.32	88.29	89.10	84.22	84.55
4	85.82	86.13	73.46	74.93	88.81	88.85	83.67	84.06
5	86.03	86.27	74.41	74.76	89.14	89.25	83.41	84.11

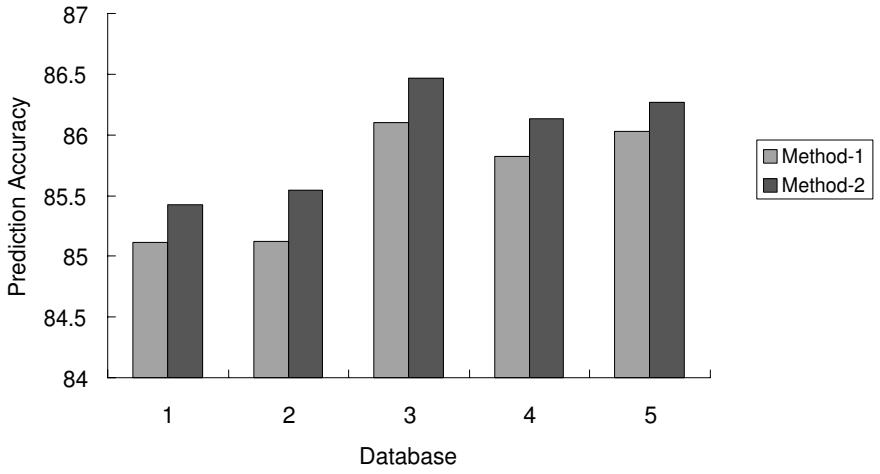


Figure 3.11: Comparison of overall accuracy between two methods

And also longer step prediction is explored by studying the 2-step and 3-step prediction. Results are shown in Fig. 3.12 using both method-1 and method-2. We can see from Fig.3.12 that the increase of the steps does not affect the overall accuracy so much.

Method-1 becomes more accurate than method-2 in 3-step prediction in Fig.3.12, because the matching condition of method-1 is more severe than method-2, therefore, method-1 can not give prediction result for every class. Actually, a small number of matched rules are produced in method-1 as shown in Fig.3.13. The average number of usable association rules for prediction in method-1 and method-2 and each prediction step shows that as the number of prediction steps increases, the number of rules used for both timing methods decreases, however the number of rules for method-1 is decreased more rapidly than method-2.

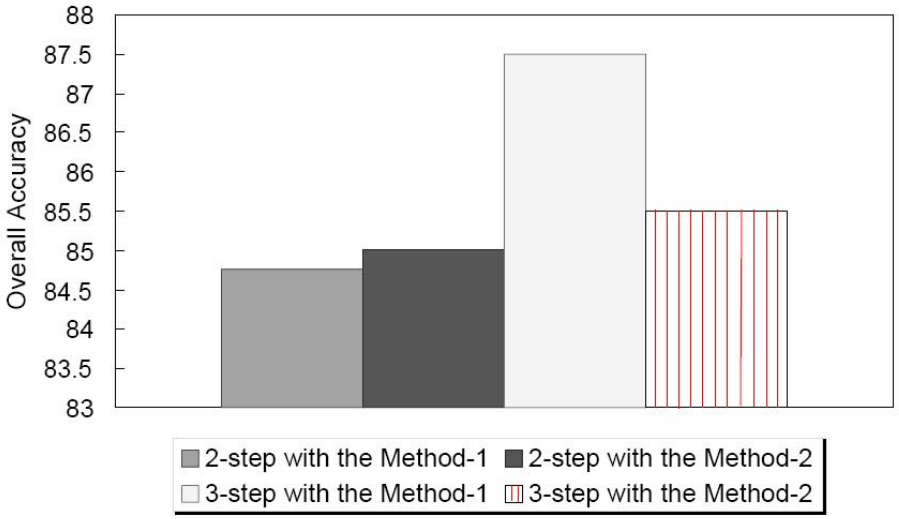


Figure 3.12: Overall accuracy of 2 step and 3 step prediction

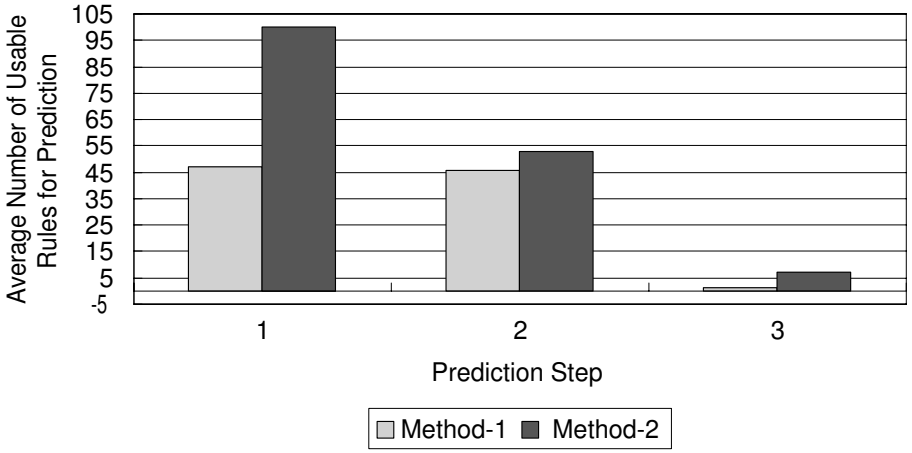


Figure 3.13: Average number of usable rules for prediction per class

The proposed method cannot extract all the rules meeting the given definition of importance since we use the fixed number of rules for each class of the consequent attributes, but the result shows that the ability to extract important rules is sufficient enough for our purposes.

The proposed method uses the evolutionary computation, where time delays are included to use the time series data from real traffic networks easily. Although training

GNP from scratch needs comparatively long time(around 2 days under the environment of $IBM\ X32$ with $512M$ Memory), the evolutionary based method is easy to adapt to new environments by retraining GNP for a small number of generations using new training data. What's more, even though the training time of the GNP-based model is computationally expensive, generally the training is done off-line, which means the trained GNP-based prediction model would be used for on-line prediction in real-time applications as any other statistical models.

3.4 Conclusions

In this chapter, a method of class association rule mining using Genetic Network Programming with time series processing mechanism has been proposed. The proposed method can extract important time-related association rules for each class of the consequent attributes efficiently. These rules are used to decide to which class the time-related traffic data belong using two different kinds of timing mechanisms. From simulation, it has been cleared that method-2, which is more flexible than method-1, is likely to have a bit higher prediction accuracy than method-1.

But in order to study the effectiveness and efficiency of the proposed method, the traffic data from just a simple road simulator were used, which is not sufficient enough for confirming the effectiveness and efficiency of the proposed method, although the basic algorithms for dealing with large scale traffic systems such as AAM, EMS, time delay addition mechanism, self-adaptive mechanism and classification timing method have been studied.

Further improvement including the study of the applicability of the proposed method to real traffic systems using the large scale traffic simulator is done, which has around 8000 sections in a joint research with a car company.

Chapter 4

Multi-Branches and Full-Paths(MBFP) Generalized Association Rule Mining

In order to prevent the congestions or provide the useful information for traffic management systems, when and where the traffic congestion or heavy traffic will occur should be predicted, then, the navigation system can choose the route avoiding the sections with potential congestion and give higher priority to the sections with potential low traffic. Many traffic density prediction methods belong to the traffic congestion prediction using regression analysis.

A Time Related Association Rules Mining with GNP in traffic prediction has been already proposed in the previous chapters, however the method just uses simple if-then type judgment connections, which is not efficient. More sophisticated method is now proposed named Simple Transition Route Searching(STRS), where the judgment logic is no longer simple Yes/No-connections, instead, the Low/Middle/ High connections are introduced in STRS method.

The proposed STRS method naturally leads to the problem of how to explore all the possible combinations of the Low/Middle/High connections among attributes. In order to realize this MBFP model, two basic methods are used named Transition Route Searching(TRS) and Transition Route Memory(TRM).

The basic feature of the proposed mechanism is like the following:

- The GNP structure of rule representation is generalized to multiple branches instead of simple "Yes-side" and "No-side" transitions. Thus, it could be applied to generate candidate association rules in a more generalized and efficient way.
- Multi-Branches and Full-Paths(MBFP) searching mechanisms are used to find all the potential combinations of the attributes using GNP, which can improve

the efficiency of the proposed method.

- Two different kinds of MBFP searching mechanisms are proposed and analyzed. Transition Route Searching mechanism(TRS) first determines the searching transition routes, then, examine the transitions according to the route using database, so the form of judgments is something like "The attribute is High?", while Transition Route Memory(TRM) mechanism realizes multi-judgments using the form like "What is the value of the attribute?".

- The proposed method uses an evolutionary approach to obtain association rules, however, unlike Pittsburgh approach and Michigan approach [5],[6], which represent the rules as individuals or a part of an individual, GNP is used as a tool to pick candidate rules. Therefore, the aim of the evolution is not to find the best GNP individual, but to pick up an enough number of rules to carry out the classification effectively and efficiently.

This chapter is organized as follows: In section 4.1, the algorithm of the time related class association rule mining and classification using Generalized GNP with Multi-branches and Full-Paths is described. Section 4.2 shows the simulation environments, conditions and experimental results. Finally, section 4.3 is devoted to conclusions.

4.1 Time Related Class Association Rule Mining Using Generalized GNP

4.1.1 Outline of the Proposed Method

The whole procedure of the proposed algorithm is shown in Fig.4.1. The first step is the time related rule extraction step. In the rule extraction procedure, two kinds of iterations are included, i.e., the outer iteration represents the class association rule mining, which means to extract the time related class association rules of all the consequents with Attribute Accumulation Mechanism(AAM) and Extraction Mechanism at Stages(EMS)[25], while the inner iteration represents the basic procedure of the time related data mining, which uses Generalized GNP individuals to generate the possible candidate rules, then checks the counts of the possible candidate rules using Simple Transition Route Searching(STRS), Multi-Branches and Full-Paths(MBFP) searching including Transition Route Searching mechanism(TRS) and Transition Route Memory(TRM) mechanism which will be explained later.

How to obtain these important counts based on the databases effectively and efficiently is the kernel issue of the proposed method, and this process is called searching

mechanisms. Only after obtaining the counts based on the time related databases, the important criteria of the candidate rules can be calculated and hence the important interesting rules can be finally extracted using the criteria. As long as important counts can be obtained efficiently and effectively, important candidate rules can be properly picked up as association rules, thus the searching mechanism is the fundamental mechanism for the whole rule extraction process.

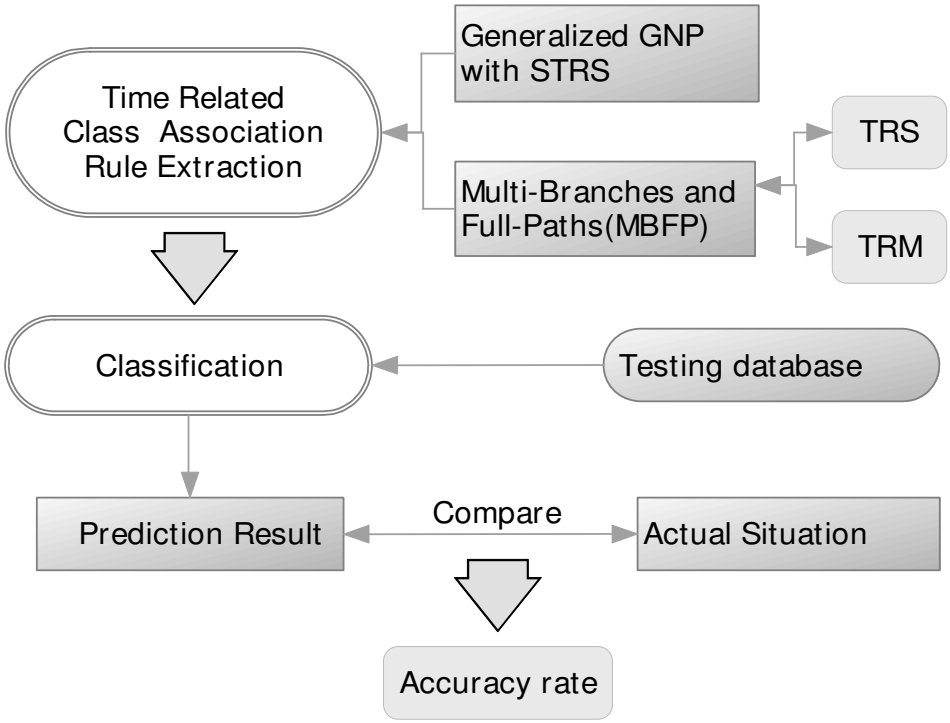


Figure 4.1: Basic steps of the proposed algorithm

The second step is to use the extracted rules to predict the future events using the matching degree between the traffic data and antecedent part of the rules in each class, and the class with the highest matching degree will be assigned to the traffic data[32]. If the classification result is the same as the actual situation, then the prediction is correct, otherwise incorrect, thus the accuracy of the prediction can be obtained in this way.

4.1.2 Time Related Class Association Rule

The following is a formal statement of the problem of mining association rules. Let $A = \{A_1, A_2, \ldots, A_k\}$ be a set of items or attributes, i.e., it can represent each section on

the traffic networks in the traffic density prediction problem. Let G be the total number of time units in the time related traffic volume database. Each time unit is associated with a unique identifier whose set is called $TimeID$. Let D=$\{D_1, D_2, \ldots, D_k\}$ be a set of time units of the event occurrence of the corresponding attribute, i.e., if the event of attribute A_i occurs at time unit 3, then D_i is denoted as $D_i = 3$.

Definition 1. *Time Related Attribute: The sequence of time related attribute A_i is defined as A_t =$\{A_{1(D_1)}, A_{2(D_2)}, \ldots, A_{k(D_k)}\}$, where D_k is the time unit when the event of A_k occurs, $A_k \in A$ and $D_k \in D$.*

The continuous value of the time related database has been already discretized to three different levels: *Low*, *Middle* and *High* (briefly, *L*, *M* and *H*). In general, a set of items in A_t with its corresponding value level is called time transition. A time transition is now defined as follows:

Definition 2. *Time Transition: Time related attribute set A_t with its corresponding Low/Middle/High level is defined as time transition TT=$\{A_1(V_1)_{(D_1)}$,$A_2(V_2)_{(D_2)}$, $\ldots$, $A_k(V_k)_{(D_k)}\}$, where, $V_k \in V = \{Low, Middle, High\}$, $A_k \in A$ and $D_k \in D$.*

Definition 3. *Sub Transition: A time transition STT is called a sub time transition of the time transition TT, if and only if it constitutes a sub sequence starting from the first attribute of TT.*

Fig.4.2 shows an example of the time transition. Supposing time transition TT_x=$\{$ $A_1(High)_{(0)}$, $A_2(Middle)_{(1)}$, $A_3(Middle)_{(2)}\}$, the time transitions in the database starting from time unit 0000 and 0004 are two examples of the time transition, and the time transitions starting from 0002 and 0006, however, just satisfy a part of the above time transition. Time transitions starting from 0002 and 0006 are in fact the sub transitions of time transition TT_X.

Based on the Def. 1 and Def. 2, a time related association rule is an implication of the "$X \Rightarrow Y$" where $X \in A_t$, $Y \in A_t$. While, X is called antecedent and Y is called consequent of the time related association rule.

Definition 4. *Time Related Class Association Rule: Let $A_i(*)(t = p)$ be an attribute in a database at time p. $A_i(*)$ represents $A_i(Low)$/$A_i(Mid)$/$A_i(High)$. The time related class association rule mining extracts the following association rules:*
$A_j(*)(t = p) \wedge \cdots \wedge A_k(*)(t = q) \Rightarrow A_c(*)(t = r)$,
where, $A_c(*)(t = r)$ indicates the class of the consequent attribute.

Here, $p \leq q \leq r$, and the first t always equal 0 and other time units are the relative time shifts from the first attribute.

Each time transition has its own associated measure of statistical significance called support, confidence and chi-squared value. These values are calculated based on the

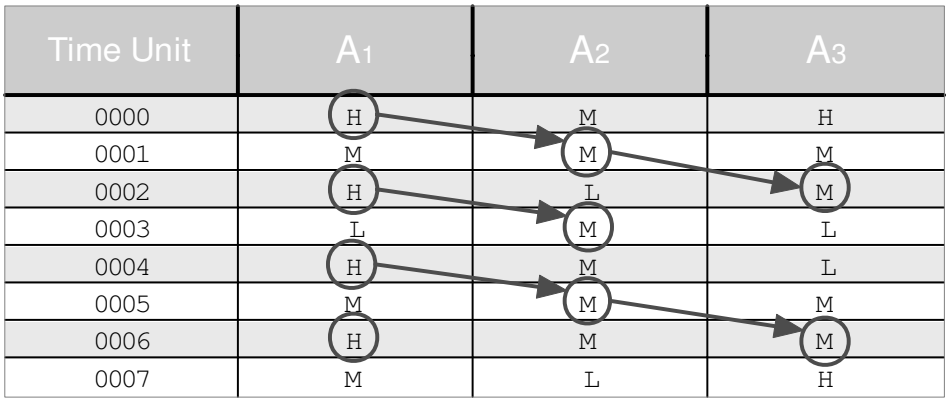


Figure 4.2: Time Transition

counts of the time transitions obtained from the searching mechanism. If the number of time transitions containing X in database G equals t, and the total number of time transitions in G is N, then $support(X) = t/N$. The rule $X \Rightarrow Y$ has a measure of its strength called confidence defined as the ratio of $support(X \cup Y)/support(X)$.

Calculation of the chi-squared value [24] of the rule $X \Rightarrow Y$ is described as follows. Let $support(X) = x$, $support(Y) = y$ and $support(X \cup Y) = z$. If the events X and Y are independent, we can get $support(X \cup Y) = xy$. The chi-squared value is calculated using x, y, z and N as follows:

$$\chi^2 = \frac{N(z - xy)^2}{xy(1 - x)(1 - y)} \tag{4.1}$$

This has 1 degree of freedom. If it is higher than a threshold value (3.84 at the 95% significance level, or 6.63 at the 99% significance level), the independence assumption will be rejected.

After calculating the criteria of time transitions, if the significance level of the time transition is important enough, which means this time transition shows the important association between time related event sequences, then it can be picked up as an association rule. As a result, each time transition in fact represents an candidate rule in the proposed method.

4.1.3 Generalized GNP for Time Related Class Association Rules Mining

The proposed method extracts important time related association rules using GNP supposing that the connections of nodes are represented as candidate association rules. The generalized GNP has a proper number of branches in the judgement nodes in stead of simple “Yes-side” or “No-side” transition in the previous method [25].

The basic structure of the generalized GNP network structure for class association rule mining is shown in Fig.4.3. Each judgement node of GNP is in charge of checking whether the corresponding attribute - Low, $Middle$ or $High$ is satisfied. For example, given a ”$A_1 - High$” condition, then it corresponds to the High-side of the connection from node A_1, and ”a, b and c” are the counts of the transitions moving to the corresponding judgement result($Low/Middle/High$) of attribute A_1.

The time delay for $Low, Middle$ or $High$ connection is the same value in each judgement node and evolves in the same way as other parameters and structures of GNP.

Simple Association Rule Mining using Generalized GNP, Simple Transition Route Searchings(STRS)

As a natural extension of the previously proposed method [25][32], a route with the fixed number of n attributes(user-defined) starting from the processing node is used as the possible antecedent parts, and it is combined with the current consequent class to generate the time transitions.

All the items of the consequent part are placed in Consequent Table(CT)[25] as shown in Fig.4.3. This means that GNP individuals are evolved to extract the class association rules for each class of the attributes, respectively, i.e., for each traffic density level of each section of the road networks, and this procedure is repeated for all the consequent items.

When extracting the rules, the basic procedure for Simple Transition Route Searching using Generalized GNP(STRS) is like the following: firstly, the node transition moves to the connecting another judgement node from the current judgement node one after another randomly. Then, a time transition(candidate rule) of length n(user-defined) can be obtained. After that, the criteria of this time transition will be checked according to the database.

After one turn of the database searching, the counts for the given time transition and all of its sub time transitions can be automatically obtained, as a result, the support,

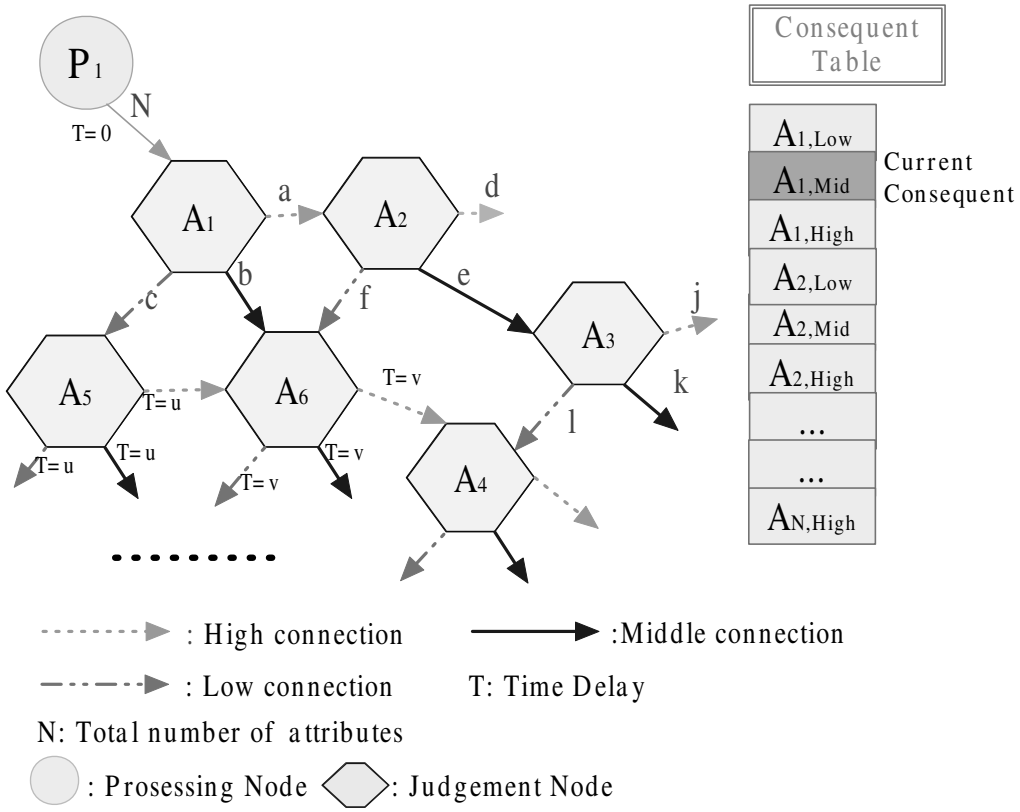


Figure 4.3: Basic structure of class association rule mining using Generalized GNP-STRS

confidence and chi-squared value can be calculated for the candidate rule and sub candidate rules. If the criteria are higher than their thresholds, then the time transitions and its sub time transitions become association rules and will be stored in a rule pool during the generations.

The total number of moving to each branch from the processing node at each judgment node is calculated for every processing node, which is a starting point for calculating association rules as described above. In Fig.4.4, N is the total number of the searches, and a, b and c are the number of searches moving to $A_1(Mid)$, $A_2(Low)$ and $A_3(Mid)$ at each judgment node of A_1, A_2 and A_3. While the count c corresponds to the count for the whole time transition, the count a and b can be used for the sub time transitions. The measurements of candidate rules are calculated by these numbers.

For example, as represented in Fig.4.4, if the node transition moves from A_1 to A_2 and from A_2 to A_3 randomly, and the class of the current consequent attribute A_c is Middle, then following rule is obtained,

If "$A_1(Middle)(t = 0) \wedge A_2(Low)(t = 2)$
$\wedge A_3(Middle)(t = 4)$", then "$A_c(Middle)(t = 6)$" considering the time delay $T = 2$, which can also be evolved. Sub candidate rules are also constructed as follows:
"$A_1(Middle)(t = 0) \Rightarrow A_c(Middle)(t = 2)$",
"$A_1(Middle)(t = 0) \wedge A_2(Low)(t = 2) \Rightarrow$
$A_c(Middle)(t = 4)$"
In order to calculate the criteria of the given candidate rule, its corresponding time transition should be checked in the database starting from time unit "0000", "0001", ..., until the end of the database. If the data satisfies the judgement of the time transition, continue to the next judgement and increase the count of the corresponding connection, otherwise, start all over again from the next time unit as shown in Fig.4.4. Even if the data doesn't satisfy the whole time transition when starting from a certain time unit, the satisfied parts can be used for criteria calculation of the sub time transitions.

The most different point between STRS and conventional method [25] is that STRS generates the time transition route randomly for each processing node, so even using the same GNP structure, the same processing node can obtain different time transition routes generation by generation, which can reduce the overlap of the extracted rules and hence increase the rule extracting efficiency.

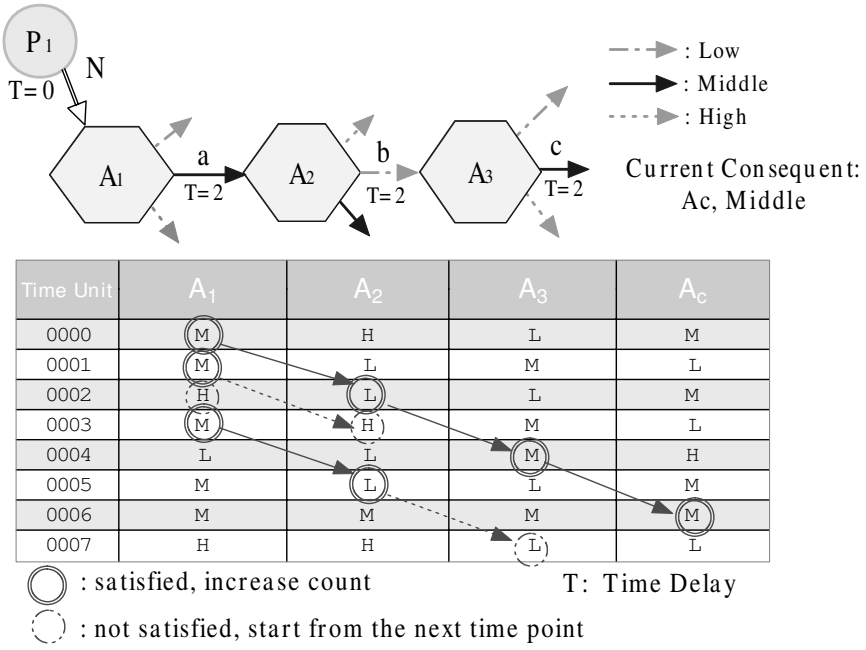


Figure 4.4: Two dimensional searching method

Multi-Branches and Full-Paths(MBFP) GNP Method

In STRS method, the time transition is randomly decided to obtain the possible time transitions, and the obtained transition is one of the possible combinations of the attributes, which could be used in the conventional way of GNP-based data mining [25].

However, it is not efficient and does not completely utilize the potential reusability of the directed graph structure, thus more efficient methods have been proposed here to obtain all the possible combinations of n(user-defined) judgment nodes starting from the processing node using two different MBFP methods.

There exists two different ways on how to check the counts for all the possible time transitions starting from the processing node. The first one is a relatively naive extension of STRS method, which is called MBFP-TRS.

A Transition Route Searching mechanism(TRS) based method first determines the route of transitions using GNP structure, then carry out the searching according to the concrete data in the database using two-dimensional search[25].

STRS is obviously one kind of TRS based method, since it randomly generates one possible time transition and checks the counts according to the concrete data in the database. Actually, STRS just explores one possible time transition starting from the processing node, while there could have 3^{n-1} possible time transitions with the length of n.

Although MBFP attempts to record the counts for all the possible time transitions from one processing node, a problem emerges as shown in Fig.4.5, where there are two different time transitions from P_1 to $C(Low)$ such as α=$A(Low)_0, C(Low)_{t_1}$ and β=$A(High)_0, D(Middle)_{t_1}, C(Low)_{(t_1+t_3)}$.They both go through the $C(Low)$ connection, thus, the count of $C(Low)$ in Fig.4.5 will be confused with each other for different time transitions. Here, attributes are denoted by A, B, C and so on. The problem is which time transition this count corresponds to?

Simple memory structure to record the passed sub time transitions and their corresponding counts can solve the problem, however, at the expense of huge overlap checking during the searching. Therefore, a counting structure of Time Transition tree(TT tree) is proposed, which indicates all the possible time transitions starting from one processing node as a tree structure(Fig.4.5). The same node C in Generalized GNP structure now becomes located in different time transitions, hence the counts can be recorded correctly without remembering the history of the passing routes. Traveling through the tree to any of the nodes will automatically give the information of the passed routes, and also, the counts for different time transitions can be stored separately.

TT tree is just a temporary searching structure for rule generation, and after all

the counts are recorded, and importance criteria are evaluated, it can be destroyed immediately. Individuals are encoded using GNP structure, while GNP can generate a variety of different TT trees for different processing nodes.

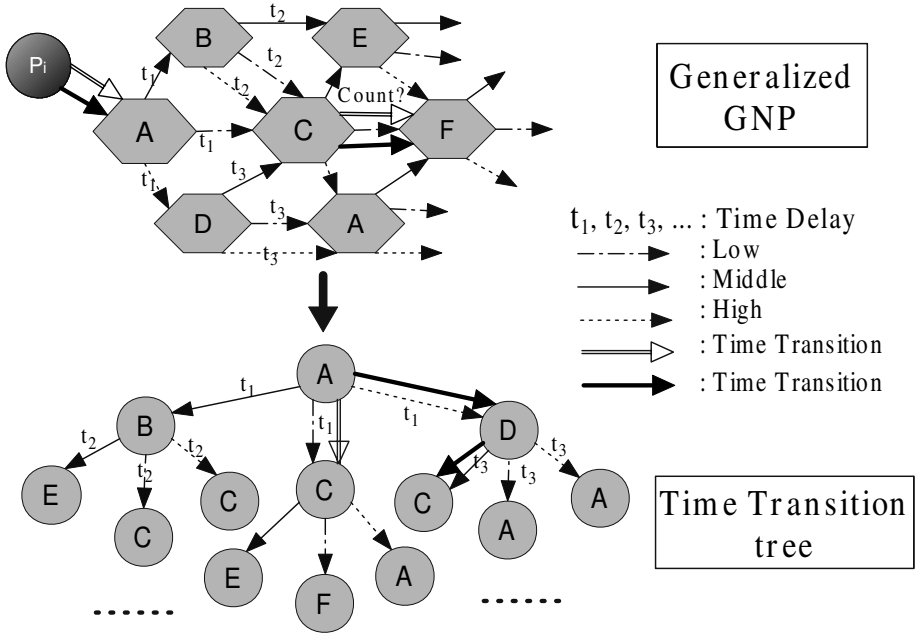


Figure 4.5: MBFP-TRS mechanism

MBFP-TRS method is also an TRS based method, and it aims at exploring all the possible time transitions for one processing node. It is an natural extension of STRS method, and it is only different from the STRS method that MBFP-TRS method checks all the possible time transitions starting from a processing node by traveling through the tree in a root-first order.

MBFP-TRS uses the same searching method as STRS mechanism when checking the time transition according to the concrete data in the database. Although it checks all the possible time transitions in a root-first order, the basic methodology is not improved, thus it is a naive extension of STRS method.

In order to further improve MBFP algorithm, instead of using the ”time transition → database” thinking logic, a backwards structure is proposed like ”database → time transition”.

This is a Transition Route Memory(TRM) mechanism, which means that the concrete transitions of the attributes in GNP are determined using the database. In other words, ”What is the value of the attribute?” type judgement nodes are used in MBFP-TRM.

The conventional method does have to decide firstly the time transition, e.g.,
TT_x={$A(High)_{(0)}$, $B(Middle)_{(1)}$},
then the conventional method will check the database in the following: start from database tuple 0000, check whether attribute A is high at 0000 and whether attribute B is Middle at 0001, if the condition is satisfied, increase the corresponding count, otherwise, start the same checking process from the next time unit 0001 checking whether attribute A is High at 0001 and whether attribute B is Middle at 0002, and get the final counts after studying all the database, which is the one turn of checking in one time transition.

However, TRM does not decide first the time transition as the conventional methods, instead, transit the generalized GNP according to the concrete database as follows:

For example, given a TT tree by GNP structure as shown in Fig.4.6, first attribute A is checked using the database at time unit 0000, and since the value in the database is Middle, the middle connection of TT tree will be selected, which means the count a on the middle connection of TT tree will be increased by one and the next attribute becomes attribute B. Since the time delay between attribute A and attribute B is 1 as shown in TT tree, the value of B at 0001 is checked as the next attribute, the value of B is Low, so we have now checked time transition {$A(Middle)_{(0)}$, $B(Low)_{(1)}$}. And as a result, the count b on that connection also increases by one and the next attribute becomes C.

After checking the time transitions starting from 0000, the time transition starting from 0001 will be checked in the same way. We should notice that the time transition starting from 0001 is different from the time transition starting from 0000 because of the different attribute values of the database. It means that different time transitions are checked during the one turn of checking the database. After checking all the time transitions from all the records starting 0000, 0001, 0002, ..., using the TT tree structure, if some of the time transitions in the TT tree do not appear in the database, then their corresponding counts will be automatically zero.

When using TRS based MBFP mechanism, only one time transition and its sub time transitions are checked during one turn of the database check, so if we use TRS based approach, $3^{(n-1)}$ turns of the database searching is necessary for one processing node, where n is the user-defined time transition length.

When using TRM based MBFP mechanism, all of the possible combinations of the time transitions can be checked by only one database searching, which saves the calculation time and the efficiency of the data mining is improved dramatically.

The counts recorded in the TT tree is fundamental data used for the calculation of the support, confidence and chi-squared value, which represent the interesting level of the obtained time transition, e.g., the support counts how many records in the database satisfy the time transition, confidence calculates the rate of if antecedent occurs, then

the consequent will also occur and chi-squared estimates the inner-relationships among the antecedent and consequent parts.

Only the time transition with high support, confidence and chi-squared values can be considered as interesting time transitions and its corresponding rules becomes extracted as association rules and they will be stored in the rule pool. The high threshold of the interesting level guarantees that the extracted rules are all important and interesting, and since MBFP method can check a lot of time transitions during the one database check, it has much more chance of generating rules during the one database check, thus a larger number of important association rules can be obtained by MBFP method.

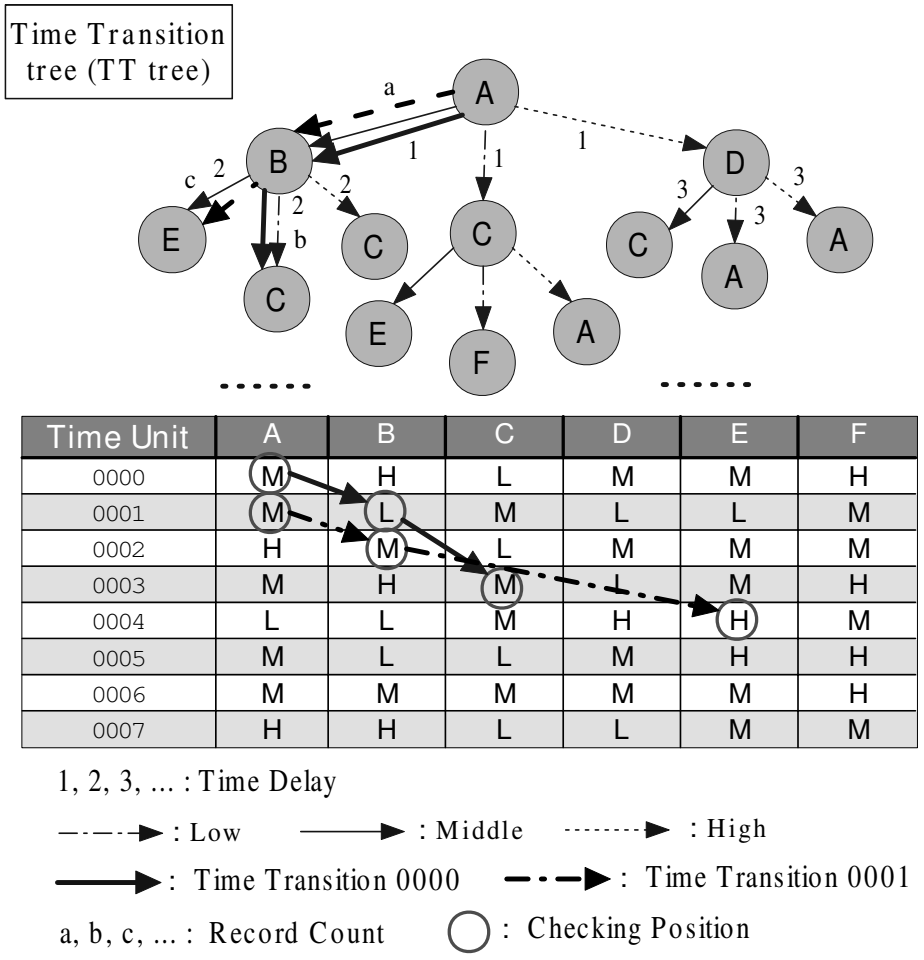


Figure 4.6: MBFP-TRM mechanism

In conclusion, two different kinds of time related association rule mining methods

using Generalized GNP have been proposed, one is a simple extension of the previous method called STRS, and in order to further improve the algorithm and increase the efficiency, MBFP searching methods, that is, a naive extension of STRS named MBFP-TRS and a more sophisticated mechanism named MBFP-TRM have been proposed.

4.2 Simulation

In this section, the effectiveness and efficiency of the proposed MBFP methods are studied by a simple traffic simulation. Unlike other methods in the traffic prediction of recent years, the proposed MBFP methods not only aim at predicting the traffic jam on a specific section of the road networks, but also interested in providing the whole traffic prediction for all of the sections on the road networks so that the navigation system can refer to this kind of information for the calculation of the optimal route of the road networks.

The time related association rules can be extracted for classification using a large traffic database, which can provide a stable estimation of the traffic density. And not only the traffic congestion/jam can be predicted, but also the sections with low traffic density can also be predicted, which could be also interesting since drivers might want to take a path with few cars. So, the uniqueness of the proposed method is that all of the traffic volumes ($High/Middle/Low$) are predicted for all of the sections on road networks.

4.2.1 Optimal Route Algorithm

In order to use more realistic traffic data for generalized association rule mining, the Q-value based optimal routing algorithm is adopted[33]. Fig.4.7 shows the procedures on how the optimal route is calculated and updated by using the Q values in the simulator.

The detailed steps are as follows:

1. Initialize the traveling time of all sections.

$$t_{ij} = d_{ij}/\upsilon_a, \tag{13}$$

where,

t_{ij}: the traveling time from intersection i to intersection j,

d_{ij}: distance from intersection i to intersection j,

υ_a: average speed of cars.

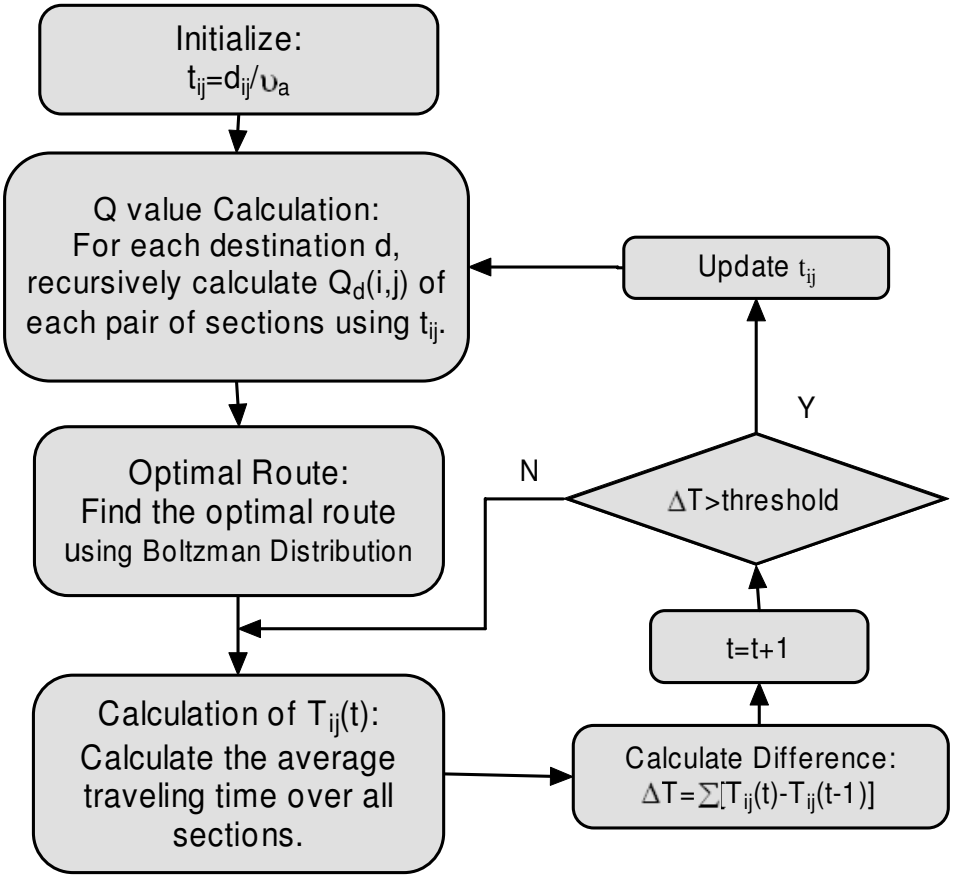


Figure 4.7: Optimal route calculation with Q values

2. For each destination d, Q values of all of the intersection pairs can be obtained by iterating the following equation:

$Q_d^m(i,j) \longleftarrow t_{ij} + \min_{k \in A(j)} Q_d^{m-1}(j,k)$,

$Q_d(i,j) = \lim_{m \to \inf} Q_d^m(i,j)$,

where,

$A(j)$: set of suffixes of intersections directly movable from intersection j,

$Q_d(i,j)$: optimal traveling time to destination d, when the car bound for destination d moves to intersection j

at intersection i.

$Q_d^m(i, j)$: $Q_d(i, j)$ in the m^{th} iteration.

3. The following probability $P_d(i, j)$ using $Q_d(i, j)$ is used to navigate the car bound for destination d.

$$P_d(i, j) = \frac{e^{-\frac{Q_d(i,j)}{\tau}}}{\sum_{j \in A(i)} e^{-\frac{Q_d(i,j)}{\tau}}}, \tag{16}$$

where,

τ: temperature constant, τ =10 is used.

$P_d(i, j)$: probability for the car bound for destination d to move to intersection j at intersection i as the next intersection.

4. Calculate the average traveling time difference ΔT between the current time and one step previous time over all sections.

5. When ΔT exceeds the threshold $\overline{\Delta T}$, revise t_{ij} and go back to step 2, else go back to step 4, $\overline{\Delta T}$ = 10000 is used in the simulation.

4.2.2 Comparison between Generalized GNP with Simple Transition Route Search(STRS) and Conventional GNP method

In simulation 1, the number of rules stored in the rule pool is compared between the proposed method with Generalized GNP with STRS and conventional method with Conventional GNP using only the Yes-side and No-side transitions. Each round has the same number of generations of 50 and the chosen set size for AAM is 100[25].

Fig.4.8 shows the total number of rules obtained in the rule pool versus round number. In the conventional method, GNP individuals have only the Yes-side and No-side connections, while Generalized GNP individuals with STRS have multi-branches like Fig.4.3. We can see from Fig.4.8 that the proposed method can extract important class association rules efficiently, when compared with the conventional one.

Fig.4.9 and Fig.4.10 shows the number of rules extracted per each round. Although we can obtain around 500 rules in some rounds, the average number of rules extracted per round is not high. There exist some rare cases where the consequent class does not occur so often in the entire database. In these cases, it is difficult to obtain association rules that satisfy the required criteria, so the proposed methods gradually decrease the criteria of interesting rules[25][34]. As a result, it is found that a good number of the rules are finally obtained per each round in the proposed method.

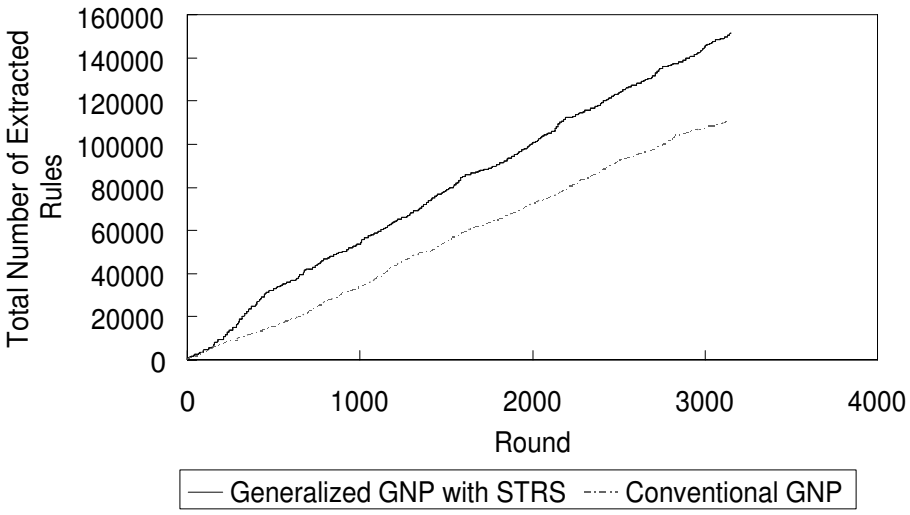


Figure 4.8: Total number of rules extracted.

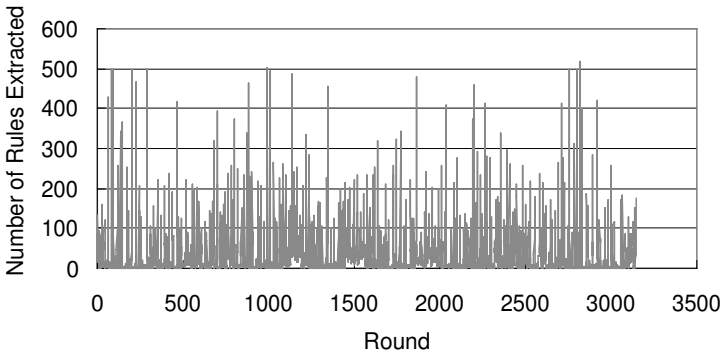


Figure 4.9: The number of rules extracted per each round in Conventional GNP

Fig.4.11 shows the average number of rules extracted per round. The reason for Generalized GNP with STRS can get a larger number of rules than Conventional GNP is that Generalized GNP with STRS provided more exploration ability than Conventional GNP, that is, the Generalized GNP with STRS can generate different candidate rules based on its $High/Middle/Low$ transitions, while Conventional GNP has only

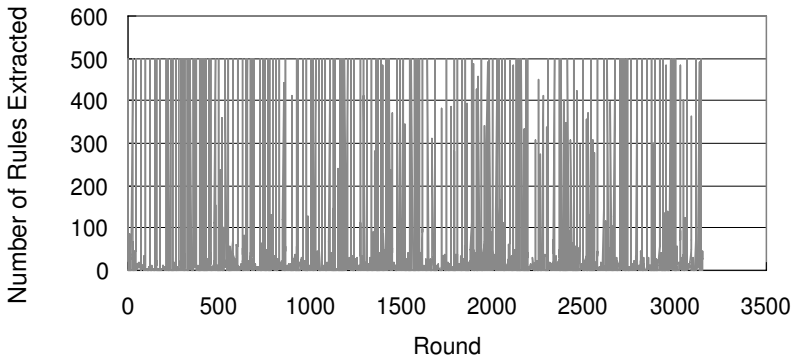


Figure 4.10: The number of rules extracted per each round in Generalized GNP with STRS

one-way of Yes-side transitions. As a result, Generalized GNP with STRS shows better efficiency than Conventional GNP. And also it shows that there are a number of rounds where no rules are extracted, and in such cases the self-adaptive mechanism adjusts the criteria of χ^2_{min}, sup_{min} and $conf_{min}$ gradually round by round. But, the average number of rules extracted per round is not very high, even though the proposed method can extract around 500 rules in some rounds.

4.2.3 Multi-Branches and Full-Paths(MBFP) algorithms and STRS

In this simulation, the number of rules stored in the rule pool is compared among the proposed Generalized GNP with Multi-Branches and Full-Paths(MBFP), the Generalized GNP with STRS mechanism and Conventional GNP. Each round has the same number of generations of 50 and the chosen set size for AAM is 100 [25][34].

Fig.4.12 shows the total number of rules obtained in the rule pool versus round number. In the conventional method, GNP individuals have only the Yes-side connection, while Generalized GNP individuals have multi-branches like Fig.4.3. We can see from Fig.4.12 that the proposed MBFP methods can extract important class association rules more efficiently, when compared with the conventional ones.

Since MBFP-TRS and MBFP-TRM methods both explore all the possible time transitions for each processing node, and they are only different in the searching method, they show almost the same efficiency of rule extraction.

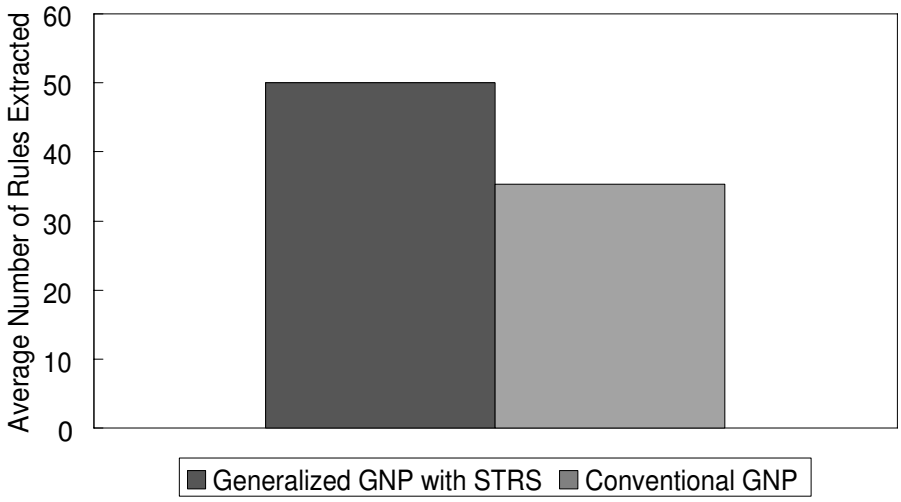


Figure 4.11: Average number of rules extracted per round

Since all the possible time transitions are explored, a comparative high average number of rules can be obtained in MBFP algorithms. As a result, it is found from Fig.4.13 that a proper number of the rules are obtained efficiently by the proposed MBFP methods.

Generalized GNP with MBFP mechanisms provided more exploration ability than Generalized GNP with STRS, that is, Generalized GNP with STRS can generate candidate rules based on the random transition to $Low/Middle/High$ sides, while Generalized GNP with MBFP can extracts all the possible transitions, thus it stresses the exploration aspect more strongly than the Generalized GNP with STRS and Conventional GNP.

As a result, Generalized GNP with MBFP methods shows better efficiency than other GNP methods. However, the efficiency is not only related to the average number of rules extracted, but also related to the real computation cost of the algorithm. The computation time efficiency is shown in Table 4.1. Conventional GNP shows the best time cost, however, in an inefficient way of rule extraction, and GNP with STRS shows almost the same time cost as Conventional GNP, but also with a comparatively lower efficiency in rule extraction. Both MBFP methods obtain a good number of rules, but with higher time cost as shown in Table 4.1, and GNP with TRS shows a extremely high time cost compared with other methods, which proves that TRS based method doesn't show the appropriate performance in MBFP algorithms.

GNP with TRM shows better results since it can obtain more number of rules while consuming the similar execution time as STRS method and Conventional GNP. The

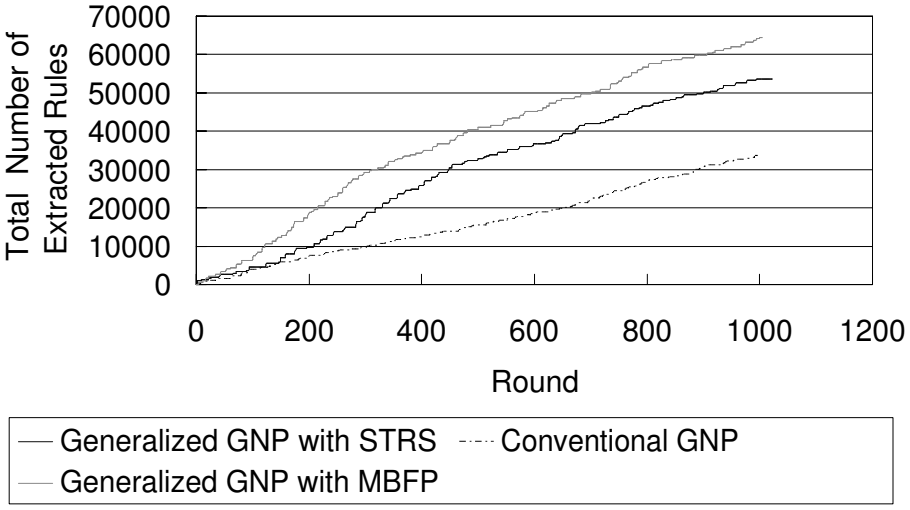


Figure 4.12: Comparison of the total number of rules extracted.

process of rule extraction is not a linear procedure, and the generated rules might be overlapped with the already obtained rules in the evolutionary process, thus the result shows that the GNP with TRM can extract rules more efficiently than GNP with STRS and Conventional GNP.

What's more, since the algorithm shows that GNP with TRM can extract a large number of rules using only the one turn of the database checking, it could be applied more efficiently to the database in real-world environments, especially in huge real-time traffic databases, where the access to the database is time consuming.

However, under different experimental environments, for example, under short fixed calculation time limits, the GNP with STRS may be also a good choice to generate a relatively appropriate number of rules in a short time.

Table 4.1: Time Cost Comparison

Method	Time per generation(Milliseconds)
GNPwithTRS	9278.06
GNPwithTRM	243.19
GNPwithSTRS	238.20
ConventionalGNP	230.13

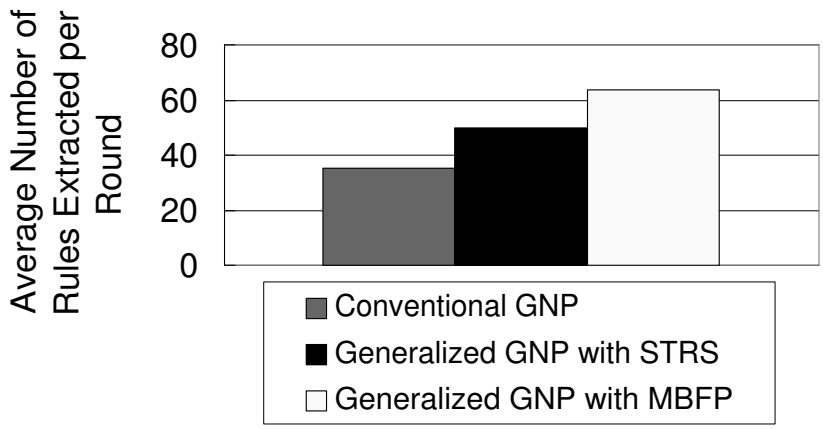


Figure 4.13: Comparison of average number of rules extracted per round.

4.2.4 Self-decrease Criteria and Prediction Accuracy

As described in section 4.3.6, when there do not exist an enough number of association rules satisfying the current criteria, the thresholds for evaluating the importance of the rules should be decreased based on the self-decrease rate in order to extract enough number of N_f rules for each class.

How to choose the proper self-decrease rate is very important, since it influences the final testing prediction accuracy as shown in Table 4.2 and Table 4.3. The accuracy is defined in the following: if the traffic prediction result of the section at time t is "Low" and the real traffic of this section at time t is exactly "Low", then, the accuracy is 100%. The $Low/Middle/High$ accuracy means the accuracy when the real traffic is $Low/Middle/High$, respectively. The results of the training and testing average prediction accuracy over all 224 sections for 800 time units are shown in Table 4.2 and Table 4.3, respectively.

It can be seen from Table 4.2 and Table 4.3 that the more the criteria for important association rules decreases, the lower the prediction accuracies become.

4.2.5 Longer Prediction Steps

Longer step prediction is explored by studying the 2-step, 3-step and 4-step prediction, where the notation of n-step means the prediction of the traffic density at n time units later. It's results are shown in Fig. 4.14, where the rules extracted by self-decreasing rate of 0.9 are used. It is shown from Fig.4.14 that the prediction accuracy

Table 4.2: Average Training Accuracy for different self-decrease rate(Training)

self-decrease rate	Prediction Accuracy			
	Overall	*Low*	*Middle*	*High*
0.95	89.74	69.79	91.32	86.53
0.90	88.67	69.50	90.65	85.84
0.85	87.79	69.49	90.00	81.44
0.80	87.37	58.02	89.77	77.34
0.75	86.72	55.60	89.60	76.97
0.70	83.19	52.72	88.62	76.74

Table 4.3: Average Prediction Accuracy for different self-decrease rate(Testing)

self-decrease rate	Prediction Accuracy			
	Overall	*Low*	*Middle*	*High*
0.95	85.01	65.76	88.44	84.01
0.90	82.26	64.98	85.31	80.81
0.85	81.72	60.19	82.26	71.83
0.80	80.97	59.70	82.01	70.41
0.75	80.67	58.02	73.92	73.68
0.70	76.04	54.75	83.39	67.93

decreases as the prediction step increases, but the increase of the steps does not affect the overall accuracy so much, so the proposed method can do relative stable prediction even if the prediction step increases.

The ratio of the usable rules to the total number of rules in each prediction step are shown in Fig.4.15, which describes that how the number of the usable rules for prediction decreases by the increase of the steps, considering the condition that the time delay between the antecedent part and consequent part is bigger or equal to the prediction step. In another word, as the prediction step increases, the number of the usable rules decreases.

The proposed method cannot extract all the rules meeting the given definition of importance since it uses the fixed number of rules for each class of the consequent attributes, but the result shows that the ability to extract important rules is sufficient enough for the purposes.

Since GNP data mining aims at picking up important rules during the evolution and storing them in a separate pool, not to obtaining the optimal individuals using GNP structure, so even if the rules are obtained by the training under a certain environment, when the new database(environment) arrives, GNP can be trained again for a small

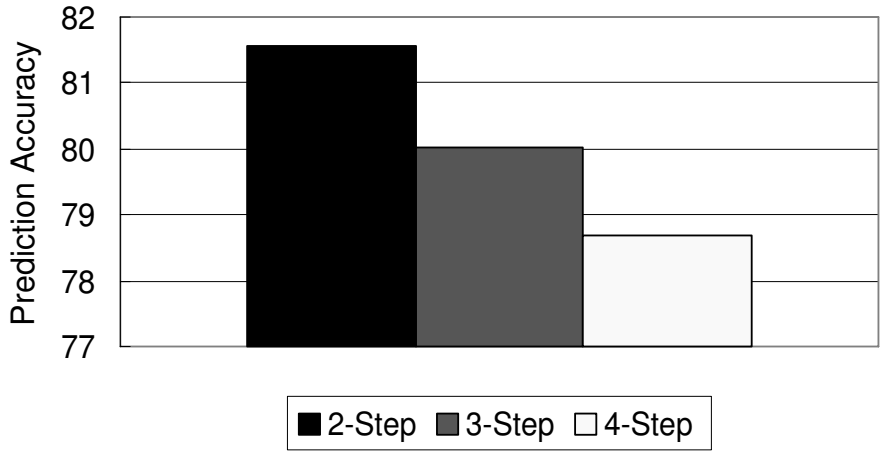


Figure 4.14: Overall accuracy of 2-step, 3-step and 4-step prediction

number of generations using the new environment and an enough number of rules are obtained.

Under changing environments, the rule extraction model could be used for retraining GNP, as a result, the proposed method can adapt to changing environments without changing the model framework.

The proposed method uses the evolutionary computation, where the time delays are also evolved easily using the time series data from real traffic networks. Although training GNP from scratch needs comparatively long time(around 1 days under the environment of *LensanceDT* with 1024*M* Memory), the evolutionary based method is easy to adapt to new environments by retraining GNP for a small number of generations using new training data. What's more, even though the training time of the GNP based model is computationally large, generally the training is done off-line, which means that the trained GNP-based prediction model would be used for on-line prediction in real-time applications as any other statistical models.

4.2.6 Optimal Route with Traffic Prediction

The next step is to use the prediction for the optimal route calculation in order to obtain better routes. For example, if the navigation system knows section *A* in the route will have a high traffic volume during the traveling, the navigation system can chose another optimal route to avoid being trapped in the heavy traffic.

The future traffic information influences the optimal route mechanism in the opti-

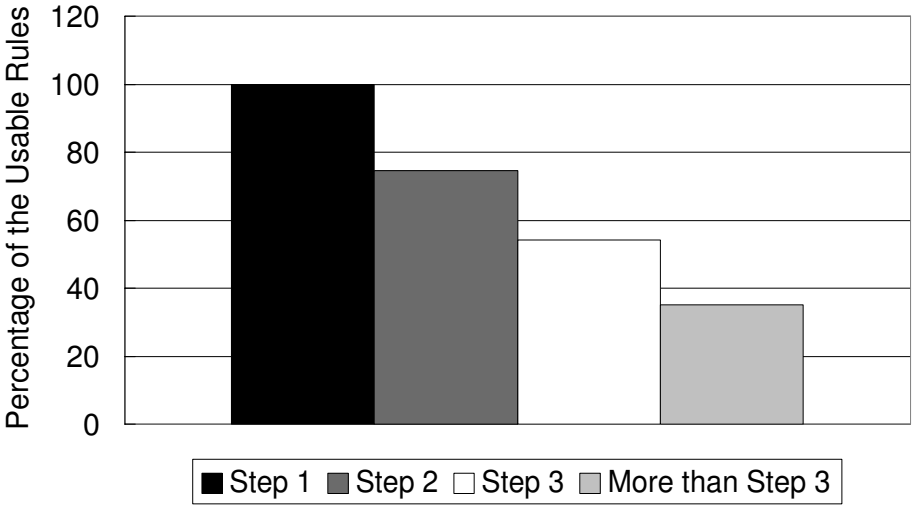


Figure 4.15: Average percentage of usable rules in each bucket

mal route selection phase shown in Fig.4.16. After the convergence of the Q value of each section for the corresponding destination, at every crossing point(intersection) for the current route, the future traffic volume of the candidate route is considered as the following:

Q values at the intersections of the current route are revised as follows when the car arrive at the intersection i at time t:

$Q_d(i,j) \leftarrow Q_d(i,j) - \gamma_1$, If $V_{ij}(t+1) = Low$,

$Q_d(i,j) \leftarrow Q_d(i,j) + \gamma_2$, If $V_{ij}(t+1) = High$,

where, $V_{ij}(t+1)$ represents the traffic volume prediction from intersection i to intersection j at time t, γ_1 and γ_2 are the parameters for the reward and punishment by future traffic prediction, respectively, and γ_1=γ_2= 5 are used.

In order to study the efficiency of the traffic volume prediction, the greedy method is applied to the vehicle after the above update of Q values, which means the vehicle will choose the route with the minimum Q-value.

Comparison is done between using updated Q values by traffic prediction and using Q values based on the current traffic.

Three ODs are considered for calculating the routes with or without prediction, which are shown in Fig.4.16.

As shown in the Table 4.4, Q-value based optimal route calculation with traffic prediction can reduce the traveling time of the optimal route since it considers the future traffic situations when choosing the next section on the optimal route.

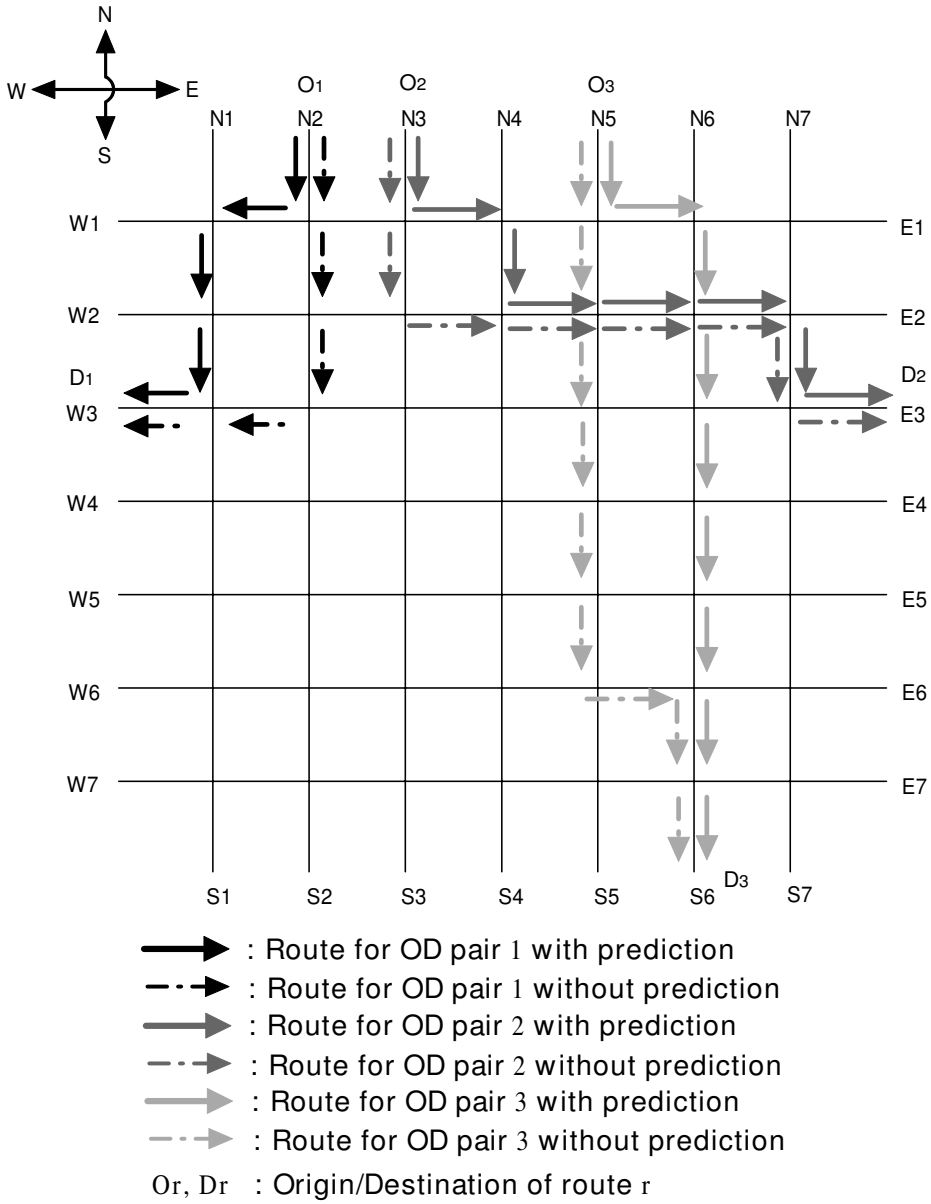


Figure 4.16: Comparison of the routes with or without prediction.

4.3 Conclusions

In this chapter, the time related association rule mining methods have been proposed using Generalized Genetic Network Programming. STRS firstly proposed uses the random mechanism to decide the time transition route. Since STRS doesn't fully

Table 4.4: Average Traveling time of optimal Route(time unit)

OD pair	without prediction	with prediction	Improved Rate(%)
1	61	51	19.6 %
2	215	187	13.0 %
3	155	114	26.5 %

explore all the possible time transitions from each processing node, two different MBFP methods have been proposed to improve the efficiency of the rule extraction, where the GNP with TRM outperforms the GNP with TRS, since it can obtain a sufficient number of rules and relatively low computation cost. The proposed methods can extract important time-related association rules for each class of the consequent attributes efficiently. These rules are used to decide to which class the time-related traffic data belongs. It has been cleared from the simulations that the proposed methods can be used for traffic density prediction.

Self-adaptive criteria are also included in the proposed method as explained in chapter 3. When dealing with various situations in the training process, the criteria can be changed automatically, as a result, when an enough number of rules can not be obtained for certain sections, the evolutionary program can adjust the parameters to obtain enough rules.

In order to study the effectiveness and efficiency of the proposed method, the traffic data obtained from a simple road simulator were used, which is not sufficient enough for totally confirming the effectiveness and efficiency of the proposed method, although the basic algorithms for dealing with large scale traffic systems have been studied such as AAM, EMS. And the obtained rules can provide useful information for the Q-value based optimal route calculation, thus, finally improve the effectiveness of the optimal route algorithm for the navigation system.

Now, further studies should be carried out about the applicability of the proposed method to real traffic systems using a large scale traffic simulator.

Chapter 5

Accuracy Validation and Large Scale Simulator

5.1 Introduction

Vast amount of traffic data are currently available using various components of the intelligent transportation system(ITS) [1]. Historical and current traffic databases provide the fundamental basis for the traffic prediction system relies on the collected database to predict future traffic conditions and hence to plan the management appropriately.

The proposed mechanism uses an evolutionary based method to extract interesting association rules, while these rules are stored and combined with a classifier model which predicts the traffic of the road networks, thus the proposed method can deal with various real-time traffic situations and show good mining performances under the changing environments.

The time related association rules mining with Genetic Network Programming(GNP) [9][10] in traffic prediction has been already proposed in the previous research [32][29], so this chapter is devoted to the further study of the performance of the time related association rules mining and traffic prediction algorithm using real time large scale simulator SOUND/4U. In this chapter, an accuracy validation mechanism is also proposed and combined with the evolutionary based rule extracting process which leads to generate more general time related association rules.

The proposed method has the following features for the time related association rule mining, time related classification and traffic prediction.

- The applicability of the proposed prediction mechanism has been verified in the large scale simulator of SOUND/4U using a real world map.

- The accuracy validation mechanism is used to overcome the overfitting problem and more importantly, the prediction accuracy is used as one of the criteria in evolutionary process, thus more general association patterns are obtained.
- The accuracy validation mechanism can obtain information from both the training and validation databases without rule mining for all of the two databases.
- The proposed method does not prune the already extracted association rules, and all of the extracted association rules contribute to the final prediction using a partial match classifier.

This chapter is organized as follows: section 5.2 is devoted to introduce the proposed algorithm. Section 5.3 shows the simulation environments, conditions and several experimental results in large scale simulator SOUND/4U. Final section 5.4 is devoted to conclusions.

5.2 Time Related Association Rule Mining using GNP and Accuracy Validation

5.2.1 Accuracy Validation and Evolution

In order to verify the association patterns produced by the rule mining algorithms, the basic approach is to separate the data into training and testing databases, i.e., the daily collected datasets can be systematically divided to different groups, e.g., the data of one day for the training and the data of another day for the testing.

However, not all the patterns found in training database are valid, i.e., abnormal changes or technical exceptions cause the noises in the common traffic database. It is common for the data mining algorithms to extract patterns which are not suited to the general data set, causing the overfitting problem. To overcome the overfitting problem and to extract more general vehicle traffic association rules on the traffic network, extra validation databases are used in this chapter.

The interesting rules are not only stored in the rule pool but also are validated by the validation database. As a result, both the conventional criteria of the association rules and the validation accuracy are considered in the rule extraction phase, thus the evolution process can concentrate on extracting more general rules which adapt to real-time traffic databases and avoid the over-fitting to the training database during the evolution process.

Actually, the validation accuracy is introduced to the fitness function of rule extraction in order to extract more general association rules.

For interesting rule r, the validation accuracy is defined as follows:

$$V(r) = \frac{N(r)}{N_a(r)},$$

where,
$N_a(r)$: the number of validation data which satisfy the antecedent part of rule r.
$N(r)$: the number of validation data which satisfy rule r.

Here, the satisfication means that all the attributes of the antecedent part of rule r match with validation data. Validation Accuracy tries to find out all the data that satisfy the antecedent part of the time related rules, and what percentage of them will satisfy the consequent, so it has the range of [0,1].

After the validation of the interesting rules generated by a GNP individual, their validation accuracies are taken into the fitness function, hence the proposed algorithm can guide the evolution to generate GNP individuals which can extract more general rules adaptable to real-time databases.

The information contained in the training database and validation database is considered simultaneously in the proposed method, thus it can obtain more general rules to improve the overall accuracy. The proposed method can obtain general association rules over training and validation databases with only one round of rule mining in the training, so another round of rule extraction about validation database is unnecessary, which is time consuming.

The rules which contain more than N_{thre} kinds of attributes are so called multiple rules, where N_{thre} is a user-defined threshold [25]. The following $\alpha_{mult}(r)$ is added to the fitness function to increase the diversity of the evolution process by including many different kinds of attributes in the rules.

$\alpha_{new}(r)$ in the following fitness function is necessary in the evolutionary rule extraction since individuals might pick up the same association rules, thus really new rules can be obtained by introducing $\alpha_{new}(r)$[25].

Therefore, the fitness function of the GNP individual is now defined as:

$$F = \sum_{r \in R} \{\chi^2(r) + \beta V(r) + 10(n_{ante}(r) - 1) + \alpha_{new}(r) + \alpha_{mult}(r)\}.$$

The symbols are as follows:
R : set of suffixes of important extracted association rules which satisfy the importance requirements of chi-squared, support and confidence values.
$\chi^2(r)$: chi-squared value of rule r.
β : coefficient of validation accuracy.

$n_{ante}(r)$: the number of attributes in the antecedent of rule r.

$\alpha_{new}(r)$: constant defined by

$$\alpha_{new}(r) = \begin{cases} \alpha_{new}, & \text{if rule } r \text{ is new} \\ 0, & \text{otherwise} \end{cases}$$

$\alpha_{mult}(r)$: constant defined by

$$\alpha_{mult}(r) = \begin{cases} \alpha_{mult}, & \text{if rule } r \text{ has more than} \\ & N_{thre} \text{ kinds of attributes} \\ 0, & \text{otherwise} \end{cases}$$

$\chi^2(r)$, $V(r)$, $n_{ante}(r)$, $\alpha_{new}(r)$ and $\alpha_{mult}(r)$ are concerned with the importance, accuracy, complexity, novelty and diversity of rule r, respectively.

It can be seen from the above description that GNP individuals are defined as a tool to pick up candidate rules, then the proposed method does not aim at obtaining the optimal individual itself, but aims at developing the GNP which produces as many rules as possible. So, the individuals which can obtain many new important association rules with multiple attributes will have high fitness values.

In each generation, GNP individuals are replaced with the new ones by the selection policy and other genetic operations. Four kinds of genetic operators are used, i.e., uniform crossover, mutation for functions, mutation for connections and mutation for time delays of judgement nodes, respectively[25].

5.2.2 Classification

The proposed method does not prune the extracted association rules, instead, all of the rules are taking part in the final decision of classification according to their different reliability levels and validation accuracies by using a partial matching classification model[35]. This mechanism simulates the group discussion process, where the opinion of the group with the highest credibility level becomes adopted.

Firstly, classify the extracted association rules in the pool into consequent classes. Every attribute has three consequent classes, i.e., attribute A_c is classified into the following consequent classes, $A_c(Low)$, $A_c(Middle)$, A_c
$(High)$.

Then, the association rules in each class are used to study whether the testing data satisfy the antecedent attributes of rules. The testing data are called satisfied if they satisfy the antecedent attributes of the rules.

Unlike the validation process, the following partial matching strategy is carried out for calculating the matching of rule r with testing data d.

$$M_k(r) = \frac{N_k(d, r)}{N_k(r)},$$

where,
$N_k(d, r)$: the number of matched attributes in the antecedent part of rule r in class k with testing data d.
$N_k(r)$: the number of attributes in the antecedent part of rule r in class k.

The credibility of the rules in every class is calculated considering the confidence, matching degree and validation accuracy. The testing data is classified into the class whose credibility is the highest. The concrete process is like the following:

(1) Calculate R_k: Calculate the set R_k of suffixes of rules in class k.

(2) Calculate $Credit_k$ in class k:

$$Credit_k = \sum_{r \in R_k} M_k(r)(confidence(r) + \alpha V(r)),$$

where, α is a weight of the prediction accuracy.

(3) Calculate $Score_k$ in class k:

$$Score_k = \frac{Credit_k}{|R_k|},$$

where, $|R_k|$ is the fixed number of N_f in this chapter[25][35].

(4) Compare $Score_k$ and the class with the highest score becomes the winner for the consequent attribute A_c, e.g., if $A_c(Low)$ has the highest score, then A_c is classified into Low.

5.2.3 Outline of the Mining Method

The whole procedure of the proposed algorithm is shown in Fig.5.1. The first procedure is the rule extraction process. In the rule extraction procedure, GNP individuals are evolved to generate candidate rules, then the criteria of the support, confidence and chi-squared values of the candidate rules are calculated by the searching method based on the time related training databases.

Secondly, each extracted interesting rule is validated according to its prediction accuracy using the validation database, and the prediction accuracy is used as a part of the fitness functions as shown in Fig.5.1. Thus, whether the candidate rules are really

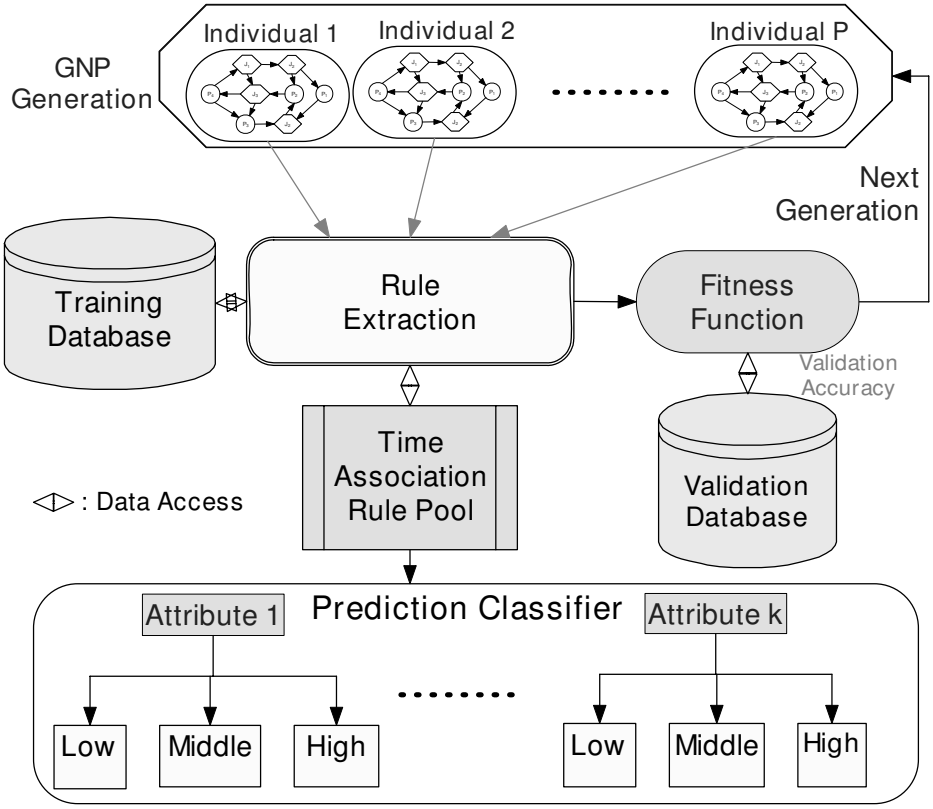


Figure 5.1: Basic framework of the proposed algorithm

interesting rules or not are determined[25]. As a result, only the interesting, but enough rules are finally stored in the rule pool.

The final step is to use the extracted rules to predict the traffic density level, i.e., traffic density of each section of the road network in the future based on the credibility of the rules using the matching degrees between the traffic data and antecedent part of the rules in each class, and the class with the highest score is assigned to the traffic data.

If the classification result is the same as the actual situation, then the prediction is correct, otherwise incorrect, as a result, the accuracy of the prediction can be obtained.

5.3 Simulation

5.3.1 Objectives

In this section, the effectiveness and efficiency of the proposed method are studied by traffic simulations which are done using a large scale traffic simulator. The time related association rule can be used for traffic density prediction. Actually, if we find that the current traffic data satisfies antecedent part of time related rules, then the future traffic density can be predicted using the consequent part of association rules.

Unlike other methods in the traffic prediction of recent years, the proposed method not only aims at predicting the traffic congestion on a specific section of the road networks, but also is interested in providing the whole traffic prediction for all of the interesting sections on the road networks so that the navigation system can refer to the information for the optimal route calculation of the road networks.

5.3.2 Simulator

Real-time simulator SOUND/4U, which is a fully-customizable macroscopic free-flow traffic simulator aiming at providing efficient traffic control and management of the urban-level large scale traffic network, is used. The SOUND/4U simulates the real-time traffic density and traveling speed of vehicles on the VICS[36] systems based on the OD(Origin/Destination) information.

The simulation is carried out using the traffic network of Kurosaki area in Kitakyusyu, Japan. As shown in Fig.5.2, the road model is based on the real traffic network and there are a huge number of sections on the traffic network. The traffic conditions are shown in the Table 5.1:

Table 5.1: Parameter setting for simulations

Items	Values
Number of sections	7941
Number of intersections	4243
Number of traffic lights	142
Data collection interval	1(minute)
Total execution time	2(hr)
Number of OD points	20
Number of OD pairs	100

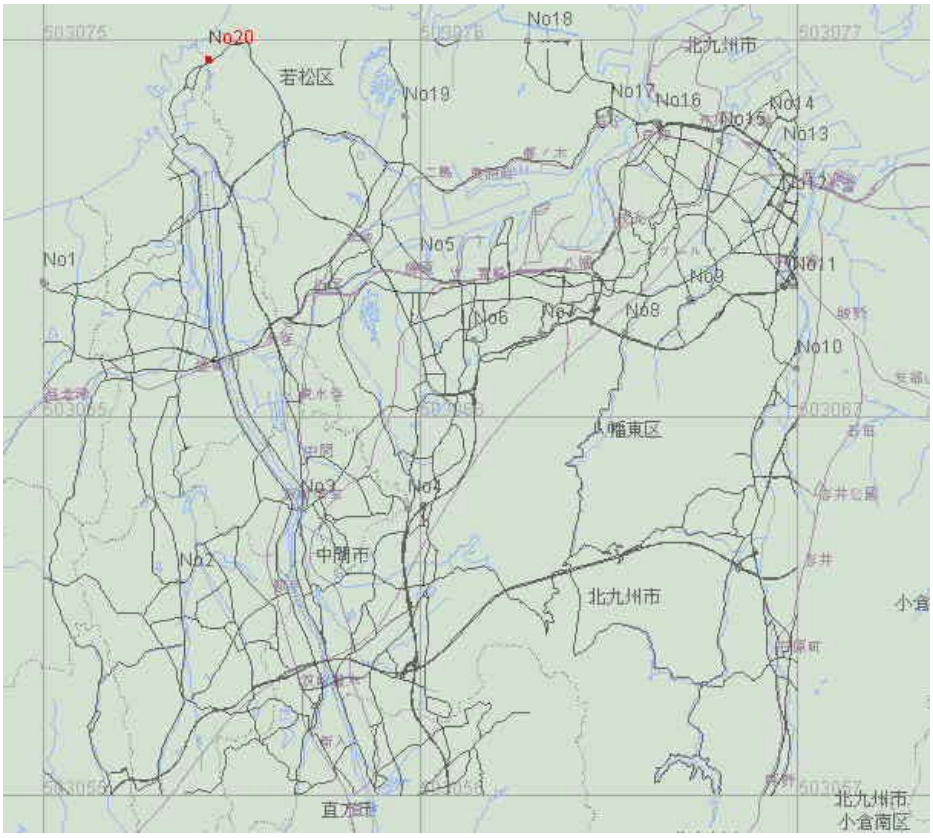


Figure 5.2: Road model used in simulations

The real-time traffic density of section s, represented as $T(s)$ is calculated as follows:

$$T(s) = \frac{N_r(s) + N_{in}(s) - N_{out}(s)}{C_s \times L_s}$$

The symbols are as follows:
L_s: length of section s.
C_s: capacity of section s, e.g., $C_s = 1$ for sections with one lane, $C_s = 2$ for sections with two lane, likewise.
$N_r(s)$: the number of cars that remain on section s from the last time point.
$N_{in}(s)$: the number of cars traveled to section s from other sections at the current time point.
$N_{out}(s)$: the number of cars traveled from section s to

other sections at the current time point.

After calculating the traffic density $T(s)$ for each section s, N_{in}(user-defined) number of sections are chosen as the interesting sections, here in the simulation, $N_{in} = 500$ sections with heavier traffic densities were selected and their corresponding traffic densities at every time point were discretized to *Low*/*Middle*/*High*.

Judging by common sense, the high threshold for traffic density is defined as follows: for sections with one lane , if there exist 2 or more cars in 10 meters, then it is a high traffic situation. On the other hand, the middle threshold is defined as follows: for sections with one lane, if there exist more than 0.5 and less than 2 cars in 10 meters, it is a middle traffic situation. The remaining is discretized to low traffic situation.

The cars are randomly generated based on the values of the pre-defined OD(Origin/ Destination) pairs, and the OD value for every OD pair is changing during the execution imitating the real traffic situation, for example, as shown in Fig.5.3.

There are two routing algorithms in the SOUND/4U, one of them is the deterministic routing which just chooses the route with the smallest cost, and another one is based on the probabilistic logit model, e.g., supposing the current section is s, the probability P_k of chosing section k as the next section is defined as follows:

$$P_k = \frac{exp(-\theta \cdot C_k)}{\sum_{i \in I} exp(-\theta \cdot C_i)},$$

where, C_i represents the time cost of section i, I represents the set of all the possible sections from the current section s, and the parameter θ is a logit value. If the logit value is approaching infinity the algorithm becomes the greedy method, while if the logit value is very small, the routing algorithm tends to choose the next section randomly.

5.3.3 Simulation Result

The parameter setting of the proposed evolutionary data mining is shown in Table 5.2, and the optimal parameters are obtained by several trails[25]. The total number of rules stored in the rule pool is shown in Fig.5.4. Each round has the same number of generations of 50 and the chosen set size for AAM is 100[25].

In order to check the effectiveness of the extracted rules, we tested the classification accuracy of the proposed method using the classifier described in section 3.5. The prediction accuracy is defined as follows: if the result of the traffic prediction of the section at time t is "*Low*" and the real traffic of this section at time t is exactly "*Low*", then the accuracy is 100%. The *Low*/*Middle*/*High* accuracy means the accuracy when the real traffic is *Low*/*Middle*/*H igh*, respectively.

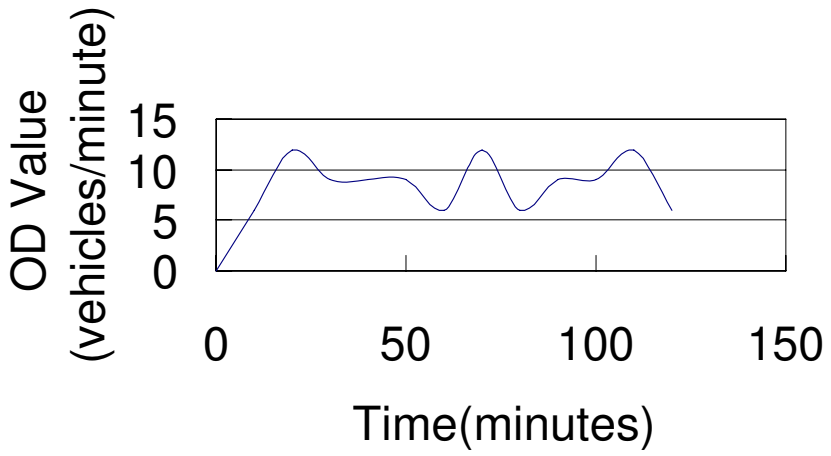


Figure 5.3: Change of Origin/Destination values in simulations

Table 5.2: Parameter setting for evolution

Items	Values
Number of judgment nodes	100
Number of processing nodes	10
Number of attributes	500
Number of consequents	1500
Number of time units	120
Minimum confidence value	0.9
Minimum chi-squared value	6.63
Minimum support value	0.08

Table 5.3 shows that the deterministic routing algorithm with class association rule mining obtains almost the same good prediction accuracy in *Low*, *Middle* and *High* situations under the testing database in the case of one time step prediction, while the probabilistic routing has small prediction accuracies in *Middle* traffic situations, where Deter. and Prob. mean the deterministic and probabilistic routing , respectively.

Table 5.3 also shows that the method with Accuracy Validation(AcV) described in Section 3.4 can improve the traffic prediction accuracy. As the logit value θ decreases from 1.0 to 0.6 in the case of No AcV, which means the increase of the random factors in the car routing, the degradation of the prediction accuracy largely increases, while the method with AcV fairly maintains better performances. The *Middle* traffic situation of the traffic network can not be predicted accurately in the probabilistic routing

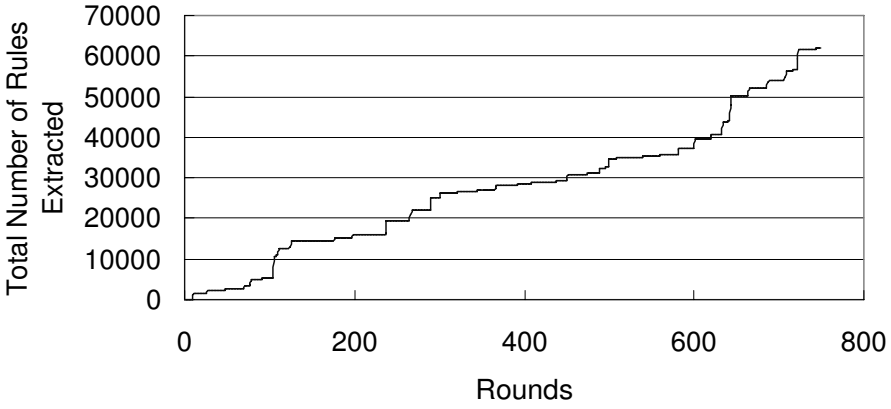


Figure 5.4: Total number of rules extracted.

because the *Middle* can be easier to be mistaken to *Low* and *High* situations especially in the probabilistic routing.

However, the proposed method can still provide the real important accurate information to navigation systems, for example, which section will have high traffics (congestion) and which section will have low traffics (vacant) are given by the proposed method.

Table 5.3: Average Prediction Accuracy under Testing Database

Method			Prediction Accuracy			
			Overall	*Low*	*Middle*	*High*
Deter.		AcV	93.52	94.02	89.20	95.08
		No AcV	93.51	93.95	89.14	94.88
Prob.	1.0	AcV	82.49	83.60	63.79	87.95
		No AcV	81.59	83.56	63.59	87.29
	0.8	AcV	82.01	84.37	63.57	87.25
		No AcV	80.94	82.27	63.44	87.07
	0.6	AcV	76.04	83.67	50.24	82.32
		No AcV	72.33	78.21	53.26	81.69

Table 5.4 and Table 5.5 show the percentage comparison between the actual situations and prediction results in the case of the probabilistic routing with the logit value of 1.0, where the percentage on the diagonal represents the accurate prediction rate, and other percentages show the wrong prediction rates, i.e., although the actual traffic

Table 5.4: Result of Testing with AcV(%)

Actual Situation	Prediction Result		
	Low	Middle	High
Low	85.84	05.27	08.89
Middle	23.84	57.71	18.45
High	07.06	03.23	89.72

Table 5.5: Result of Testing without AcV(%)

Actual Situation	Prediction Result		
	Low	Middle	High
Low	83.57	07.62	08.81
Middle	24.85	56.57	18.57
High	08.38	04.49	87.13

is *Low*, the prediction says it is *Middle*. It can be seen from Table 5.4 and Table 5.5 that the method with Accuracy Validation(AcV) has better performances in reducing the rate of the wrong prediction.

The evolutionary based rule mining method needs long off-line training time(around 1 day under the environment of *LensanceDT* with 1024M Memory). On the other hand, since the AcV mechanism only validates the new generated rules, the time efficiency is not increased so much as shown in Table 5.6. Separatly using the training and validation database, and proper combination of two rule pools obtained by these databases may also improve the overall prediction accuracy, however it needs to double the off-line training time, thus it is out of the consideration. For more detailed time efficiency study, readers could refer to[35]. Therefore, AcV mechanism provides a new way to use the information from both the training and validation database simultaneously without training large databases and without worrying about how to combine two rule pools, which could be obtained by training data and validation data.

Table 5.6: Time Cost Comparison

Method	Time per generation (milliseconds)
GNPwithAcV	250.72
ConventionalGNP	238.20

Although training GNP from scratch needs comparatively long time, the evolution-

ary based method is easy to adapt to new environments by retraining GNP for a small number of generations using new training data. What's more, even though the training time of the GNP based model is computationally expensive, generally the training is done off-line, which means that the rules obtained by the trained GNP-based prediction model would be used for on-line prediction in real-time applications as any other statistical models.

The ratio of the usable rules to the total number of rules in each prediction time step is shown in Fig.5.5, which describes that how the number of usable rules for prediction decreases by the increase of the prediction time steps. It is caused by the fact that the time delay between the antecedent part and consequent part should be bigger or equal to the prediction time step[25].

Longer time step prediction, i.e., the n-time step prediction with decreased usable rules is studied in Fig.5.6, where n-time step means the prediction of the traffic density at n time steps later. Fig.5.6 shows the one to ten time step prediction using the logit value equals to 1.0. It is shown from Fig.5.6 that the prediction accuracy decreases as the number of prediction time step increases, but the increase of the number of prediction time steps does not affect the overall accuracy so much, so the proposed method can do relative stable prediction even if the number of prediction time steps increases.

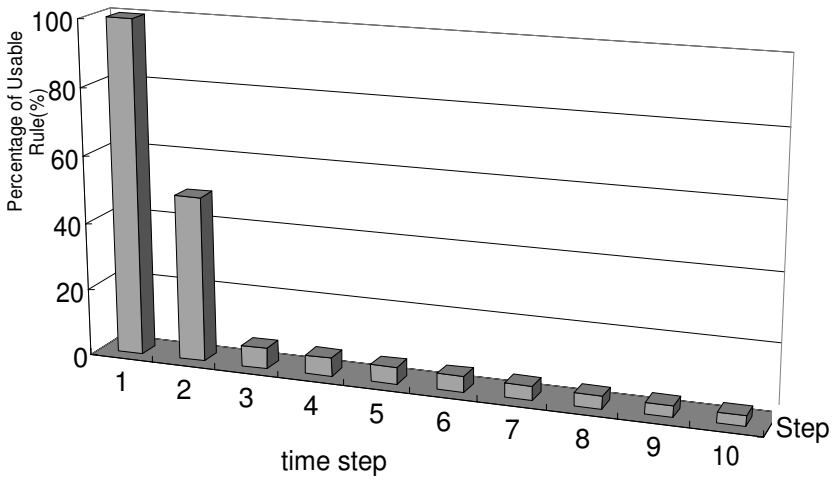


Figure 5.5: Average percentage of usable rules for each prediction time step

The proposed method cannot extract all the rules meeting the given definition of importance, since it uses the fixed number of rules for each class of the consequent attributes[25][37][38], but the result shows that the ability to extract important rules

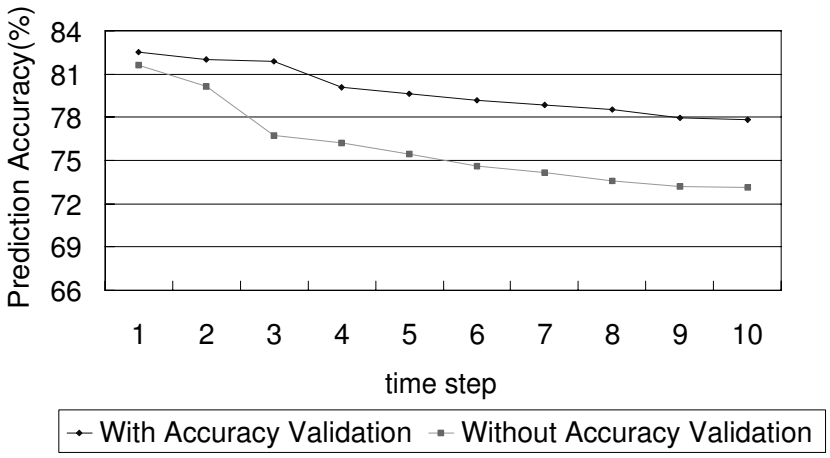


Figure 5.6: Overall accuracy of n-time step prediction

is sufficient enough for the traffic prediction purposes. The mechanism of accuracy validation maintains stable performances as shown in Fig.5.6.

5.4 Conclusion

In this chapter, an association rule mining method using GNP with Accuracy Validation mechanism has been proposed. It is also clarified from simulations that the proposed method can extract important time-related association rules for each class of the consequent attributes efficiently. What's more important is that these rules can be used to predict the traffic density in the road networks accurately using the mechanism with accuracy validation. Further improvements of the proposed method will be studied in terms of applying the proposed method to real world navigation systems.

Chapter 6

Traffic Prediction using Time Related Association Rules and Vehicle Routing

6.1 Introduction

In this chapter, how the traffic information affects the vehicle routing is studied, in particular, how the road traffic prediction affects vehicles' performance in routing algorithms is analyzed. In order to do that, the performances of two groups of vehicles are provided, while one group uses the rules extracted from the traffic database with prediction information and another group always chooses the route using the current shortest cost without prediction.

Conventional association rules are not enough to predict the future traffic situations in real time systems, which means that the association rules should be time related like the following: "If section X on the traffic map has high traffic density at time $t = 0$(current time), then section Y will also has high traffic density at $t = 10$(10 time steps later)."

Since mining real time data efficiently and effectively is too intricate when using conventional methods, especially for the time related sequential database in dynamic systems, e.g., traffic systems, the proposed mechanism uses an evolutionary based method to extract interesting association rules, while these rules are stored and can be used to predicts the traffic of the road networks, thus the proposed method can deal with various real-time traffic situations and show good mining performances under the changing environments.

The time related association rules mining with Genetic Network Programming(GNP) [9][10] in traffic prediction has been already proposed in the previous research [39][34], so in this chapter, the time related association rules mining with the combination of traffic prediction and the routing algorithms is mainly studied using a real time large

scale simulator SOUND/4U.

This chapter is organized as follows: In section 6.2, the feature of time related data mining with fixed prediction step using GNP is explained. Section 6.3 discuss how to combine the extracted prediction information with routing algorithms. Section 6.4 is devoted to introduce the simulation model and represent experimental results. Final section 6.5 is devoted to conclusions.

6.2 Time Related Association Rule Mining using GNP with Fixed Prediction Step

6.2.1 Outline of the Proposed Method

The whole procedure of the proposed algorithm is shown in Fig.6.1. The first phase is the time related rule extraction and prediction phase. In the rule extraction procedure, time related association rules are extracted using an evolutionary algorithm of GNP based on the past accumulated traffic database, and the extracted rules are stored altogether in the rule pool, thus when a new current data arrives, the future traffic situations can be predicted by matching the current data with time related association rules[25] as shown in Fig.6.1.

Secondly, the future traffic prediction at a certain future time unit is combined with the current routing algorithm, and there exist two methods of for the combination. One is to update the traffic information by prediction for the routing algorithm without actually obtaining the traffic information from the traffic network, and another one is to consider the predicted traffic when calculating the current cost of each section on the traffic network. Either method requires the prediction results, which will be explained in detail later in Section 4.

Using the information provided by the prediction phase, more proper cost of each section on the map can be estimated. In the routing phase, the Dijkstra routing algorithm [40] is used for finding the optimal route, and finally the average traveling time of vehicles using the proposed method can be obtained.

Although Dijkstra algorithm is used to investigate the effectiveness of the proposed result, in fact, the proposed mechanism of using the prediction information can be used to any routing algorithm.

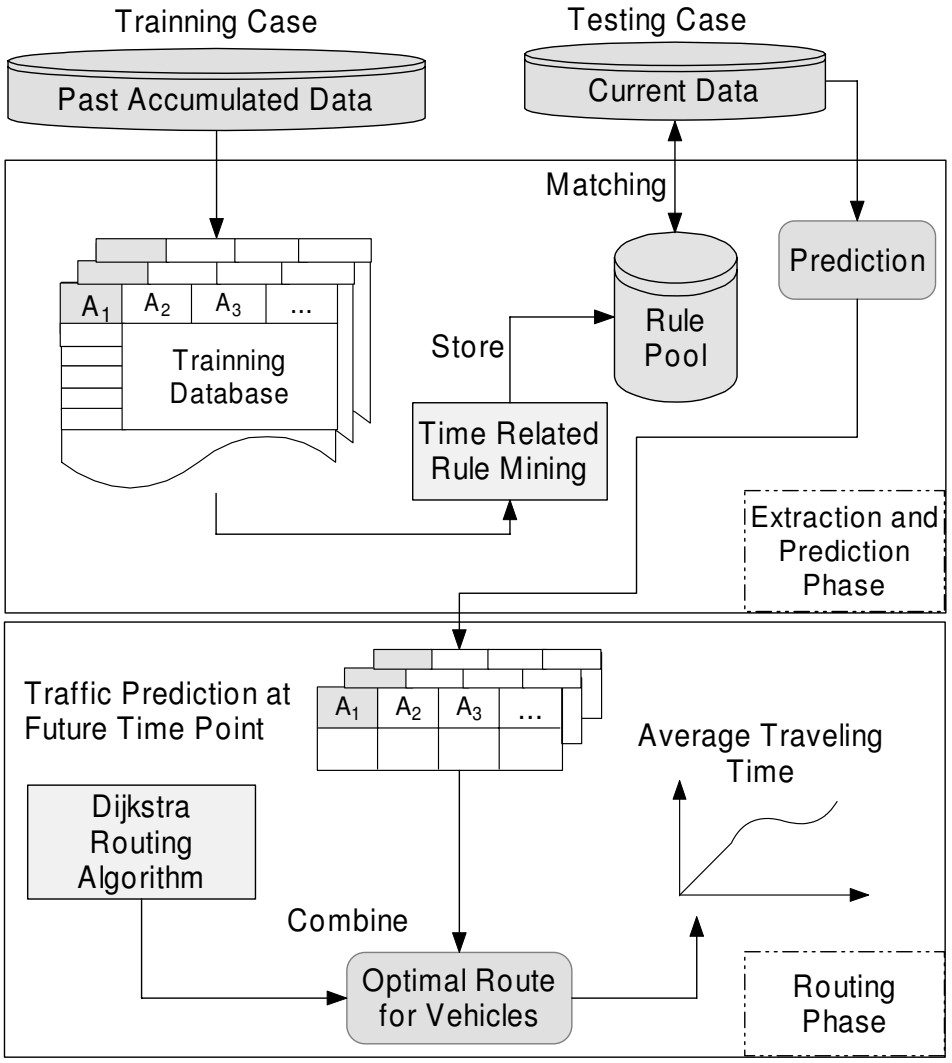


Figure 6.1: Basic steps of the proposed algorithm

6.2.2 Generalized GNP for Time Related Class Association Rules Mining

The proposed method uses GNP supposing that the connections of nodes are represented as candidate association rules. The generalized GNP has a proper number of branches, (e.g., *Low*/*Middle*/*High*) in the judgement nodes. The generalized GNP network structure for class association rule mining is shown in Fig.6.2. The time transition with the fixed number of user-defined n attributes starting from the processing node is used as the possible antecedent part, and it will be combined with the object conse-

quent class to generate the candidate rules, e.g., the shading part of GNP individual in Fig.6.2 constitute the antecedent part of candidate rule *r*.

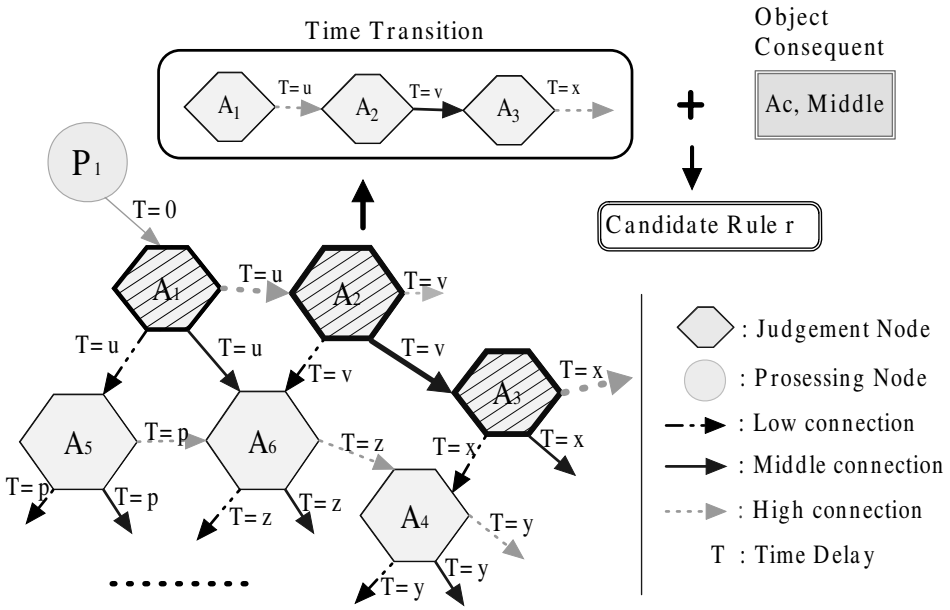


Figure 6.2: Generalized GNP structure

After generating candidate rules using GNP individual structure, these candidate rules should be checked on how frequently the corresponding events happen in the training database in sequence, then the corresponding counts will be used for calculating support, confidence and chi-squared value[35].

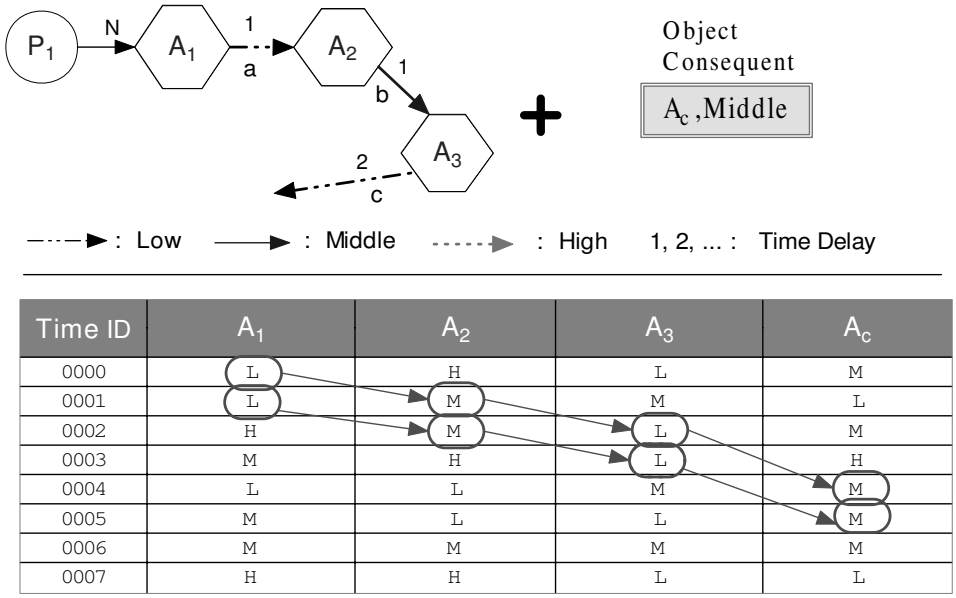
Therefore, the examination for the counts should consider both the attribute dimension and time dimension concurrently. Actually, the process for calculating the criteria of candidate rules becomes two dimensional by considering the time delays.

For example, As shown in Fig.6.3, the transition starts from database tuple 0000, and which traffic level attribute A_1 has is checked at 0000 and after that which traffic level attribute A_2 has is checked at $0000 + 1 = 0001$, since A_1 has *Low* level, *Low* levle connection of A_1 is connected to A_2 and the time delay between two attributes is 1, . . . , and the same process continues likewise until the end of the candidate rule. Then, increases the count of *a*, *b* and *c* of *Low* level branch of A_1, *Middle* level branch of A_2 and *Low* level branch of A_3 by one, respectively, since the above time transition was satisfied in the database. Secondly, the transition starts the similar procedure checking from the next tuple 0001, e.g., checks which traffic level A_1 has at 0001 and which traffic level A_2 has at 0002, . . . , increase the counts likewise, and finally we can get all

the counts after studying all the time transitions in the database, which is the one turn of checking the candidate rules. For more detailed explanation of this process, readers could refer to [35].

Actually, in Fig.6.3, N is the total number of searches, and a, b and c are the number of transitions satisfying $A_1(Low)$, $A_2(Middle)$ and $A_3(Low)$ at judgment node of A_1, A_2 and A_3, respectively. While the count c corresponds to the count of the time transition, i.e., the candidate rule, the count a and b can be used for the sub time transitions, i.e., the sub candidate rules.

Now, the important association rules are defined as the ones which satisfy the minimum chi-squared, support and confidence threshold, i.e., χ^2_{min}, sup_{min} and $conf_{min}$, respectively. Only the important association rules can be considered as interesting and stored in the rule pool.



Time ID	A_1	A_2	A_3	A_c
0000	L	H	L	M
0001	L	M	M	L
0002	H	M	L	M
0003	M	H	L	H
0004	L	L	M	M
0005	M	L	L	M
0006	M	M	M	M
0007	H	H	L	L

Figure 6.3: Two dimensional searching method

6.2.3 Fixed Prediction Time Step

Extracted important rules are explored for studying the n-time step prediction in this section, where n-time step prediction means the prediction of the traffic density at n time units later, e.g., $n = 3$ means to predict the traffic density 3 time units later.

The structure of the time related association rules represents if-then type time sequential association relations and if the antecedent part of the rules satisfy the testing

data, then it is probable that the consequent part will also occur at some time units later, therefore, the proposed method regards the antecedent part as the events already happened and the consequent part as the future events.

In another words, when using the extracted association rules, the time unit of the last attribute of the antecedent part should before the current time unit, and the consequent part will be the prediction of the future traffic.

Definition 5. *Prediction Step: Let $A_i(*)$ $(t = p)$ be an attribute in the database at time unit p. $A_i(*)$ represents $A_i(Low)$/$A_i(Middle)$/$A_i(High)$. Given the following association rule r:*

$A_j(*)(t = p) \wedge \ldots \wedge A_k(*)(t = q) \Rightarrow A_c(*)(t = s)$,

then the possible prediction range PR is defined as $s - q$ in association rule r.

Based on Def. 5, the time related association rule can be used for a n-time step prediction if and only if $PR \geqslant n$ is satisfied. Generally speaking, as the number n increases in the prediction process, PR becomes larger, so the usable number of rules decreases, which makes the prediction accuracy worse in the rule prediction phase.

In order to overcome this problem, the proposed method focuses on the extraction of the rules exclusively for prediction time step n, which means, during the rule extraction and prediction phase, the time delay for the last attribute of the antecedent part of the candidate rule is fixed to $PR = n$ as shown in Fig.6.4.

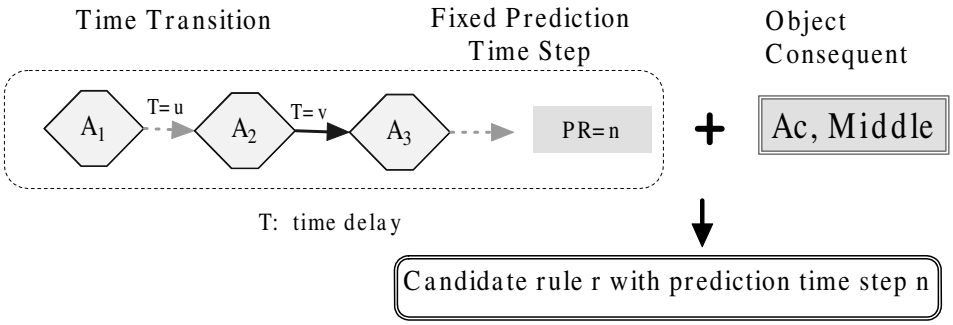


Figure 6.4: Candidate rules with fixed prediction time step

6.3 Routing Algorithm using Prediction

The obtained time related association rules can provide the estimation and prediction of the future traffic density for the sections in traffic networks[25]. How to provide this information to the routing algorithms of vehicles is discussed in this section.

The vehicle routing uses Dijkstra search algorithm [40]. Its cost function of section s at time unit t is defined as follows:

$$Cost(s,t) = C_d \cdot g(s) + C_t \cdot f(s,t)$$

where, g is the constant traveling time depending on the distance of section s, f is the current traveling time of section s at time t, C_d and C_t are the coefficients for distance and traveling time, respectively.

Dynamic real time traffic network systems are so complex that both updating the traveling time information and future traffic prediction are needed for the intelligent transportation system(ITS).

In real applications, although there exists frequent traveling time information updating, however, due to the rapid change of the traffic situation on the traffic network, if the future traffic information is provided and utilized, the routing algorithm could be improved.

Estimated future traveling time $f(s,t+n)$ is calculated as follows:

$$f(s,t+n) = f(s,t) + P$$

where, P is a constant time penalty defined as:

$$P = \begin{cases} pL, & \text{if predicted traffic density level is Low at time unit t+n} \\ pM, & \text{if predicted traffic density level is Middle at time unit t+n} \\ pH, & \text{if predicted traffic density level is High at time unit t+n} \end{cases}$$

where, $pL \leq pM \leq pH$ represent the corresponding time penalty for the predicted traffic density level Low, Middle or High at time unit $t+n$, respectively.

Two methods of applying the future traffic information to the routing algorithm is proposed. The first method is to use the prediction of $f(s,t+n)$ directly for determining the optimal route by Dijkstra search algorithm, where n is the future time unit to predict the traffic and $f(s,t+n)$ represent the predicted traffic density level of Low, Middle or High respectively. This method uses the future traffic information simply as additional traffic information updates when the traffic information is not updated frequently enough. For example, suppose there is a traffic routing system updating the current traveling time for each section every 60 minutes. As shown in Fig.6.5, vehicles can only obtain new traffic information at time unit 60 minutes and time units 120 minutes, when the running time of the system is 120 minutes, while it is supposed that the prediction of the traffic density at time unit 30 and 90 minutes is done based on data accumulated at initial time unit 0 and time unit 60 minutes, respectively. As a result, the prediction information make it possible to provide more frequent traffic updates.

Another method is a look-ahead method, where it combine the current update infor-

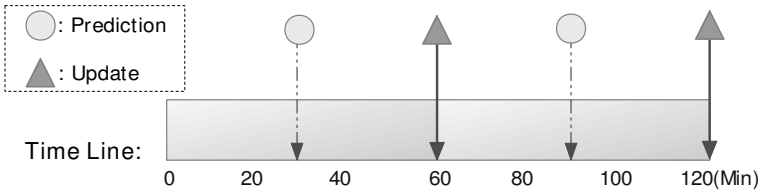


Figure 6.5: Time line for prediction and update of traffics

mation and predicted future traffic information for the routing algorithm, which means when vehicles decide the optimal route, they consider not only the current traveling time for the candidate routes, but also the future traffic information by changing the cost function as follows:

$$Cost(s,t) = C_d \cdot g(s) + C_t \cdot [(1-\gamma) \cdot f(s,t) + \gamma \cdot p(s,t+n)]$$

where $\gamma \in [0,1]$ is a look-ahead parameter considering the future traffic situation, and $p(s,t+n)$ is the predicted traveling time at time unit $t+n$, which is calculated as follows:

$$p(s,t+n) = C_p \cdot \frac{\sum_{s \in S(*)}\{f(s,t+n)\}}{|S(*)|}$$

Here C_p is a coefficient and $S(*)$ is the set of sections corresponding to $*$, which is determined by whether section s at time $t+n$ is *Low*, *Middle* or *High* by prediction, e.g., if section s will have *High* traffic density at time unit $t+n$, then $S(*) = \{\ s \mid s$ has *High* traffic density $\}$, which means $p(s,t+n)$ is the average traffic density of *High* at time $t+n$. $|S(*)|$ represents the number of sections with the traffic density of $*$.

This look-ahead method actually adds the future traffic to the current cost, and coefficient C_p is used to normalize the range of the predicted traffics, while look-ahead parameter γ defines how much the current routing algorithm considers the future traffic, in another words, if γ increases, the routing algorithm depends on the future prediction more, otherwise, the routing algorithm concentrates on the current traveling time more, especially, if $\gamma = 0$, it is the same as the conventional method.

6.4 Simulation

In this section, the effectiveness and efficiency of the proposed method are studied by traffic simulations. The proposed prediction method not only aims at predicting

the traffic congestion on a specific section of the road networks, but also is interested in providing the whole traffic prediction for all of the interesting sections in the road networks so that the navigation system can refer to this kind of information for the calculation of the optimal route of the road networks.

As a result, the obtained prediction results are combined with Dijkstra routing algorithm to provide the testing vehicles with the future traffic information, where the effectiveness of this information is studied considering the average traveling time of testing cars.

6.4.1 Simulator

Real-time simulator SOUND/4U, which is a fully-customizable macroscopic free-flow traffic simulator aiming at providing an efficient traffic control and management of the urban-level large scale traffic network, is used. The SOUND/4U simulates the real-time traffic density and traveling speed of vehicles on the VICS[36] systems based on the OD(Origin/Destination) values.

The simulation is carried out using the traffic network of Kurosaki in Kitakyusyu, Japan. As explained in the simulation part of chapter 5. The traffic conditions and routing parameters are shown in the Table 6.1:

Table 6.1: Parameter setting for simulation

Items	Values
Total execution time	2(hr)
Number of OD points	20
Number of OD pairs	100
Routing Algorithm	Dijkstra
Prediction Time Step	30(minute)
Time Coefficient C_t	1.0
Distance Coefficient C_d	1.0
Coefficient C_p	10
Time Penalty pL	30(second)
Time Penalty pM	80(second)
Time Penalty pH	120(second)

6.4.2 Simulation Results in Rule Extraction

The parameter setting of the proposed evolutionary data mining is shown in Table 6.2.

Table 6.2: Parameter setting for evolution

Items	Values
Number of judgment nodes	100
Number of processing nodes	10
Number of attributes	500
Number of consequents	1500
Number of time units	120
Minimum confidence value	0.9
Minimum chi-squared value	6.63
Minimum support value	0.08

In order to check the effectiveness of the extracted rules, we tested the prediction accuracy of the proposed method. The prediction accuracy is defined as: if the traffic prediction result of the section at time t is "*Low*" and the real traffic of this section at time t is exactly "*Low*", then the accuracy is 100%. The *Low*/*Middle*/*High* accuracy means the accuracy when the real traffic density is *Low*/*Middle*/*High*, respectively.

Although training GNP needs comparatively long time, the evolutionary based method is easy to adapt to new environments by retraining GNP for a small number of generations using new training data. What's more, even though the training time of the GNP-based model is computationally expensive, generally the training is done off-line, which means that the rules obtained by the trained GNP-based prediction model would be used for on-line prediction in real-time applications as any other statistical models.

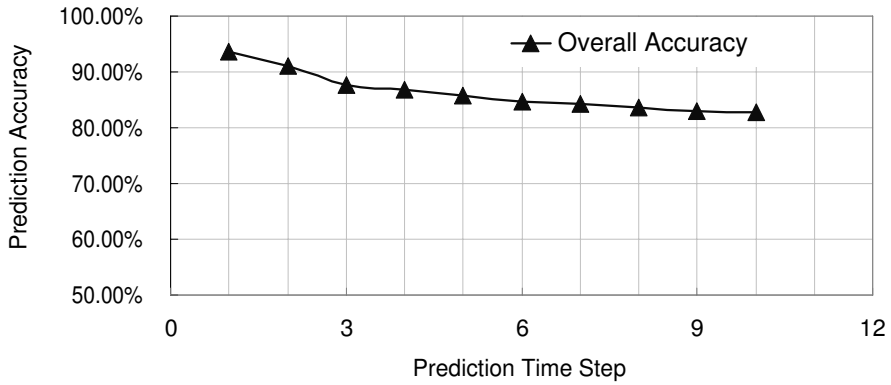


Figure 6.6: Prediction result using conventional rule extraction

As shown in Fig.6.6, as the number n increases in the n-time step prediction, the prediction accuracy decreases. It is natural since the time delay between the last attribute of the antecedent part and consequent part should be bigger or equal to the prediction time steps, so the number of usable rules decreases as the number of prediction time step increases.

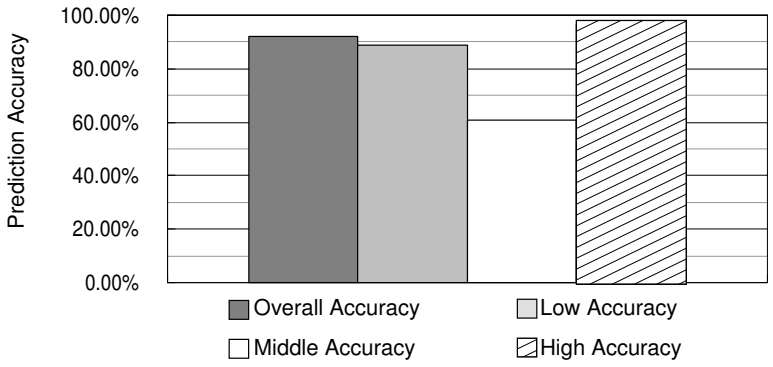


Figure 6.7: Prediction result using fixed time step rule extraction (n = 30)

On the other hand, the fixed time step method can concentrate on extracting the rules for the object prediction time step, which is 30 in the simulations. Fig.6.7 shows how the fixed time step method performs to predict the future traffic density of 30 time units(minutes) later. The prediction result shows that the fixed time step method can provide a relatively stable performance even under the long prediction time step of 30.

6.4.3 Simulation Results in Routing

Since the experiments in section 5.2 already shows that the proposed fixed time step rule mining method can extract enough association rules, and these rules can provide the accurate traffic prediction information, the next step is to study how the extracted traffic prediction information helps to improve the routing algorithms. The simulation is studied in a relatively large area considering a long traveling route as shown in the Fig.6.8.

In order to study the performance of different routing strategies, different testing groups of vehicles are compared using different routing algorithms. There considered three testing groups of vehicles as follows:

- Ignorant Group
  The ignorant group has no update traffic information and also no traffic predic-

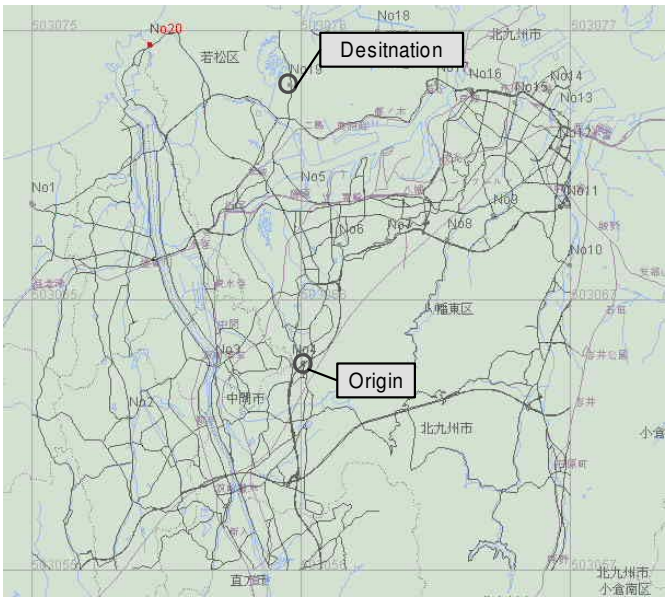


Figure 6.8: The map for routing algorithm

tion abilities, thus the vehicles just use the constant routing during the simulation period using $g(s)$ in Dijkstra routing method.

- Update Group
  The update group can get the new traffic information at the traffic update time, thus the vehicles can update the optimal route based on the new cost function $f(s,t)$ of Dijkstra method explained in Section 4.

- Prediction Group
  The prediction group can also get the new traffic information $f(s,t)$, and based on the obtained new information, future traffic situations can be predicted using time related association rules. Given the update information $f(s,t)$ and future traffic situation $f(s,t+n)$, we studied the following two prediction groups:

  - Prediction Group-1: to use the prediction of $f(s,t+n)$ directly for determining the optimal route.
  - Prediction Group-2: to combine the current update traffic information and predicted future traffic information when calculating the cost function as described in Section 4.

Fig.6.9 and Fig.6.10 show the average traveling time(ATT) of Ignorant Group, Update Group, and Prediction Group-1 and the mean value of ATT for all the testing cars

on the traffic network. It can be seen that Ignorant Group has the worst performance, since no information is updated, while both the Update Group and Prediction Group-1 shows an improvement of the traveling time after the traffic information update at time unit 60 minutes.

Results shows that the prediction is accurate at corresponding time units, thus provide additional update chances for the routing algorithm without actually obtaining the information from the traffic network, as a result, it can improve the whole routing algorithm as shown in Fig.6.10.

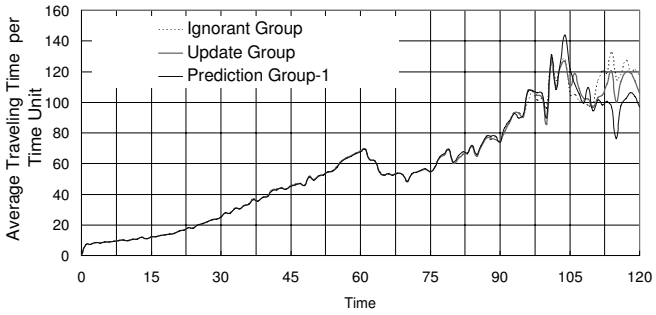


Figure 6.9: Average traveling time(ATT) in Prediction Group-1

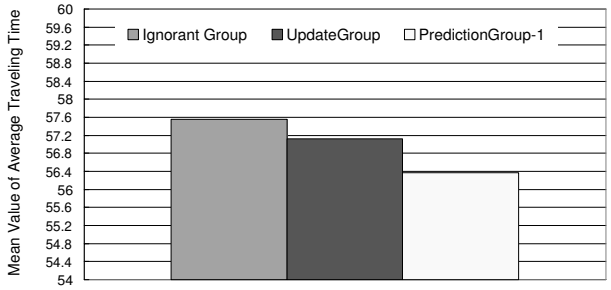


Figure 6.10: Mean Average traveling time(ATT) over time units in Prediction Group-1

Fig.6.11 and Fig. 6.12 show the performance of Prediction Group-2 considering different look-ahead parameter, which is γ = { 0.3, 0.4, 0.5 } by comparing the ATT of Ignorant Group and Update Group. By adding the future traffic to the current cost function, the performance of the testing cars can be improved as shown in Fig.6.12. The look-ahead parameter using γ = 0.5 shows the best performance. Results show

that the predicted future traffic information for updating the cost function can benefit the current routing algorithm.

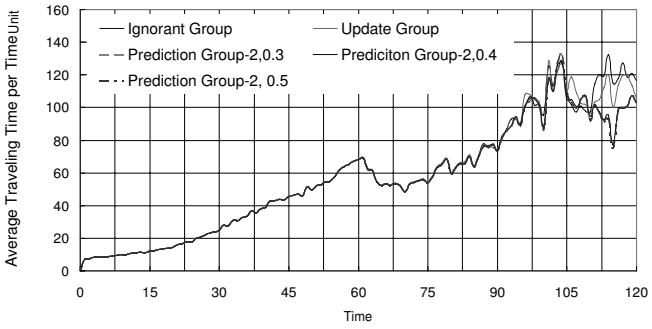


Figure 6.11: Average traveling time(ATT) in Prediction Group-2

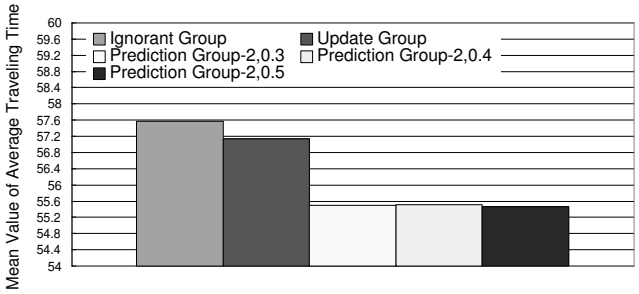


Figure 6.12: Mean Average traveling time(ATT) over time units in Prediction Group-2

6.4.4 Small Area Simulation

In order to investigate the performance of the proposed method in more detail considering the real route changes, a relatively small region on the traffic network is checked with a closer view as shown in Fig.6.13.

Fig.6.13 shows that the original Dijkstra routing algorithm choose the route of Ignorant Group not awaring of the traffic information. However, the sections of the route of Ignorant Group route have serious traffic congestions during the simulation period, thus the vehicles of Update Group choose an new route after the traffic information update as shown in the left side of Fig.6.13.

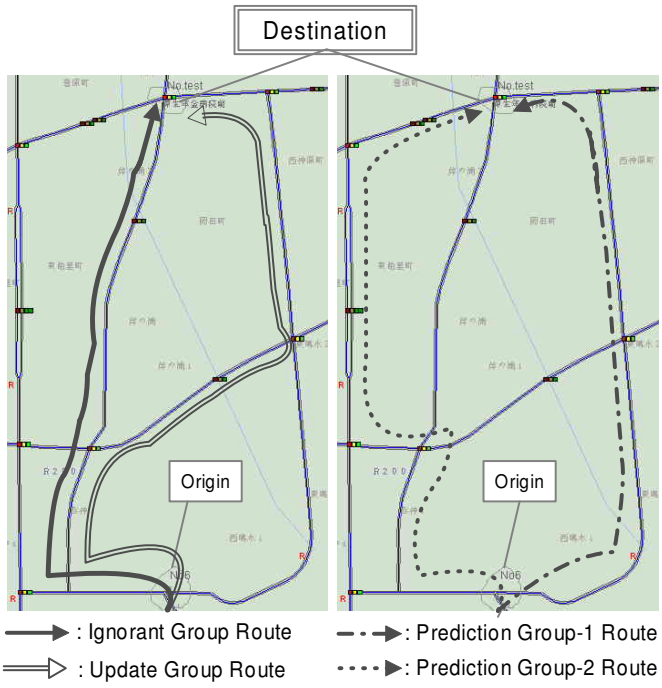


Figure 6.13: Route from Origin to Destination in different groups

Traffic density prediction result of this small area is shown in Fig.6.14 using the fixed prediction time step $n = 10$, which means to predict every 10 minutes starting from time unit 10. The traffic density distribution of *Low*/*Middle*/*High* is clearly shown in the Fig.6.14. As a result, it is shown from Fig.6.14 that the future traffic prediction result can accurately represents the real future traffic situation.

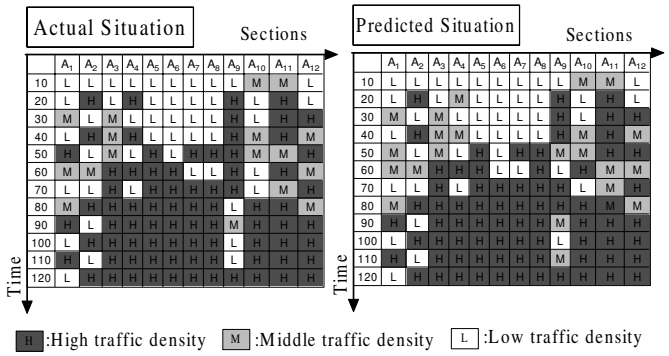


Figure 6.14: Actual and predicted traffic density distribution

Based on the traffic information accumulated at the update time units, Prediction Group utilized the predicted traffic information in two different ways, and chose their corresponding routes as shown in the right side of Fig.6.13.

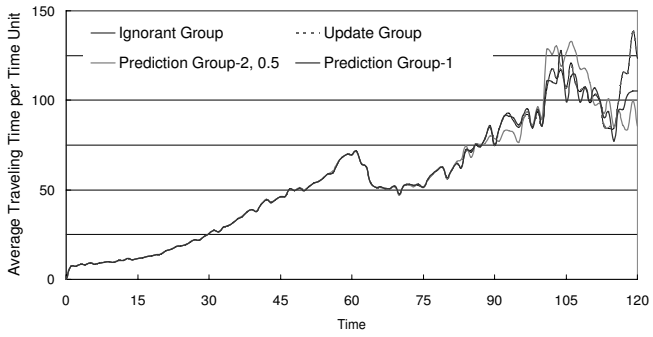


Figure 6.15: Average traveling time of small area in different groups

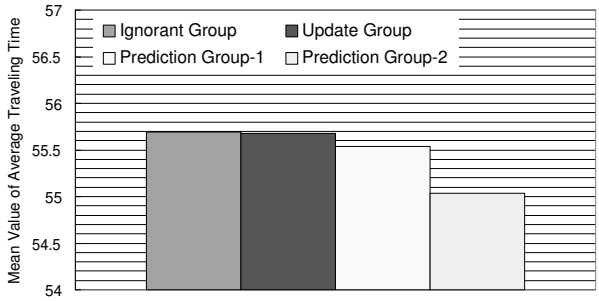


Figure 6.16: Mean Average traveling time of small area over time units in different groups

Comparisons between the performance of Update Group and Prediction Groups are shown in Fig.6.15 and Fig.6.16, where Prediction Group-1 just uses the prediction information at time unit 30 minutes and 90 minutes for additional update and Prediction Group-2 uses the look-ahead parameter of $\gamma = 0.5$. Simulation shows that the both Prediction Groups outperform the Update Group in this test area, where the route shown in the right side of Fig.6.13 is used.

The route of Ignorant Group has such serious traffic congestion that the average traveling time in this small route is even higher than the test case in the large area of

Section 5.3. This situation can also be seen from Fig.6.14, where more than 50% of the sections have *High* traffic situations. Under this heavy traffic density situation, the prediction can largely improve the ATT for the optimal routes as shown in Fig.6.16.

In conclusion, simulation results show that the proposed prediction method can extract important association rules and predict the future traffic density accurately, as a result, the performance of the vehicles of Prediction Groups are improved by using the predicted information. Simulation results are studied using Dijkstra routing algorithm, however, the proposed prediction method can be easily extended to other routing algorithms by simple amendment of the cost function.

6.5 Conclusions

In this chapter, an association rule mining method using GNP with fixed prediction time step has been proposed. From simulations it is clarified that the proposed method can extract important time-related association rules for each class of the consequent attributes efficiently. Furthermore, it is also proposed and clarified that these rules are used to predict the future traffic densities which are combined with the routing algorithms to improve the performance.

Chapter 7

Conclusions

In this research, a time related class association rule mining method is used to extract interesting associations between traffic time related databases, and represents those extracted relationships as association rules.

The searching process for time related association rules is basically two-dimensional, considering both attribute dimension and time dimension simultaneously. The proposed method use an evolutionary optimization mechanism of GNP as a tool to pick up interesting candidate rules, thus the aim of the evolution is not to find the optimal individual, instead, GNP evolves the individuals to extract as many association rules as possible.

Time related association rules are analyzed to constitute an classifier, which can provide future traffic density information to the navigation systems for vehicles in traffic networks, hence finally help vehicles adapt to the constantly changing environments of the traffic network and reduce traffic congestions in the traffic system.

In chapter 2, a method of association rule mining using Genetic Network Programming with time series processing mechanism and attribute accumulation mechanism has been proposed. The proposed method can extract important time-related association rules efficiently. Extracted association rules are stored temporarily in Small Rule Pool(SRP) and finally in Big Rule Pool(BRP) all together through rounds of generations. These rules are representing useful and important time related association rules to be used in the real world. A simple road simulator has been built and the effectiveness and usefulness of the proposed prediction algorithm has been examined. The results showed that the proposed method extracts the important time-related association rules in the database efficiently and the attribute accumulation mechanism improves the performance considerably. These rules can be useful in time-related problems, for example, traffic prediction.

After obtaining effective time related association rules, the next problem is how to apply them to the traffic prediction problem. Chapter 3 proposed a mechanism of using

class association rules, which means the attribute in consequent part is restricted to the corresponding class for each rule. Consequently, the mechanism of EMS are proposed to extract rules about the current attribute class. The extracted time related class association rules are tested by constituting an simple classifier, which is applied to the traffic density prediction of the simple simulator introduced in chapter 3. Result shows that the proposed prediction method can predict future traffic situations effectively.

In order to fully utilize the potential ability of Generalized GNP and more importantly to improve the rule extracting efficiency of the proposed rule extracting process, a new logic of the searching process called TRM is proposed which can extract all the possible rules starting from one processing node of GNP individual by using only one turn of the database scan, and this mechanism significantly improves the rule extracting process as shown in chapter 4.

Further improvements of the proposed method also includes the Accuracy Validation(AV) mechanism explained in chapter 5, which will validate extracted important rules using different validating databases, thus self-adaptively adjust the evolutionary process to generate more general rules, which shows better robustness and stable performance in the simulation of a large scale simulator, SOUND/4U.

Finally, to achieve the objective and motivation of the proposed mechanism, the rule extraction phase has been adjusted by using fixed prediction step. And based on the rule pool of the fixed prediction step, the prediction of the future traffic has been combined with a classical routing algorithm. Simulation results showed that by providing future traffic information, the average traveling time for the testing vehicles can be improved, which proves that the proposed method can deal with the traffic prediction combined with the optimal route search problem fairly well.

Appendix A

Genetic Network Programming(GNP)

Evolutionary Algorithm (EA) is a stochastic heuristic method for optimization. Since the 1960s, there has been increasing interests in imitating living things to develop powerful algorithms for difficult optimization problems.

EA is now the general term for several computational techniques which are based on the evolution of biological life in the natural world. Individuals in EA are solutions of the problems, where the fitness function is defined in order to evolve the individuals. Only the fitter ones in the generation can survive to the next generation and produce offspring by mutation and crossover imitating the evolution process in the nature. Using this kind of selection and evolving process, the fitter individual(solution) for the problem can be obtained and applied to the problem to solve.

In this section, Genetic Network Programming(GNP)is briefly introduced [9],[10], [26]. GNP is a sort of evolutionary optimization techniques(EA), which evolves arbitrary directed graph programs as solutions(individuals). Because of the strong expression ability of the directed graph structures, GNP has the ability of partially observable processes, where any kind of judgement nodes and processing nodes are used.

The structure of Genetic Network Programming(GNP) individual is briefly introduced in this section. GNP is an extended method of Genetic Algorithm (GA) [5][6], and it uses directed graph structures as solutions [9][10]. The GNP individual can also have compact structures because of the reusage of the nodes in GNP. Additionally, GNP can find solutions of the problems without bloating compared with Genetic Programming (GP) [7][8], because of the fixed number of nodes in GNP.

The genotype expression of GNP nodes is also shown in Fig.A.1. This describes the gene of node i, then the set of these genes represents the genotype of GNP individuals. NT_i describes the node type, $NT_i = 0$ means node i is the start node, $NT_i = 1$ represents node i is the judgment node and $NT_i = 2$ represents node i is the processing node. ID_i is an identification number, for example, $NT_i = 1$ and $ID_i = 1$ means node i is

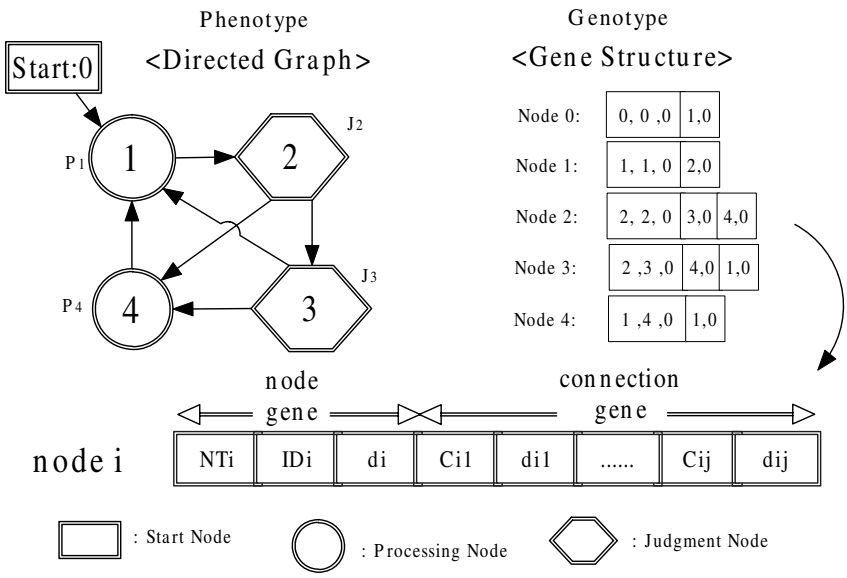


Figure A.1: The basic structure of GNP individual

J_1(Judgment node with ID 1). C_{i1}, C_{i2},..., denote the connections from node i. d_i and d_{ij} are the delay time required to execute the processing of the nodes and transition between nodes.

Each individual of GNP represents a solution of the problem, and the judgement node is in charge of judging situation, while the processing node does the real processing. Once GNP is started up, firstly the execution starts from the start node; consequently, the next node to execute is determined according to the connection and if-then judgment results of the current node. After the execution of the task, each individual gets the fitness value of itself depending on the accomplishment of the task.

As one of the applications of GNP, the proposed method uses the evolutionary approach using GNP to obtain association rules, however, unlike other genetic data mining method such as Pittsburgh approach and Michigan approach [5],[6], which represent the rules as individuals or a part of an individual, GNP is used as a tool to extract candidate rules. Therefore, the aim of the evolution is not to find the best GNP individual, but to pick up an enough number of rules to carry out the classification effectively and efficiently.

A.1 Evolution Process

The flow chart of the GNP evolution process is shown in Fig. A.2. The first step is to initialize the population by randomly generating Judgement and Processing nodes of each individual in the initial generation. Then, execute each individual(solution) in the problem domain(environment) to obtain the fitness value. This process is the **evaluation** of the current generation.

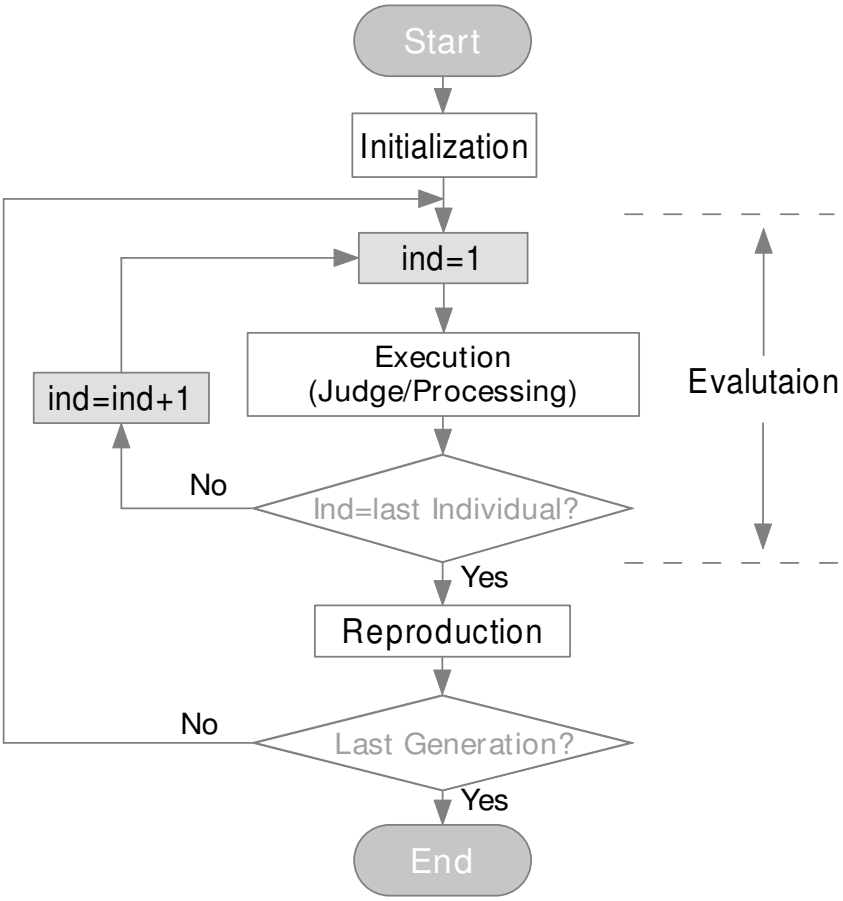


Figure A.2: The flow chart of GNP evolution

After the evaluation, we select the individuals to reproduce new individuals for the next generation. Generally, the individuals with higher fitness values have the higher chance to be selected for the reproduction. Most EA systems avoid selecting only the fittest individual in reproduction, but rather a random (or semi-random) selection among the fitter individuals is carried out, thus the diversity of the whole evolution can be maintained.

There are basically two methods of reproduction: **mutation** and **crossover**. In evolutionary algorithms, **mutation** is a genetic operator used to maintain genetic diversity in the population from one generation to the next generation. It is analogous to biological mutation which mutate the gene structure of individuals. The purpose of mutation in EAs is to help the algorithm to avoid local minima by preventing individuals from becoming too similar to each other.

Generally, mutation is to change some of the nodes in GNP individual, randomly, and connections and delays are also changed by mutation operator in GNP.

As shown in Fig.A.3, the based operation of mutation has following operations:

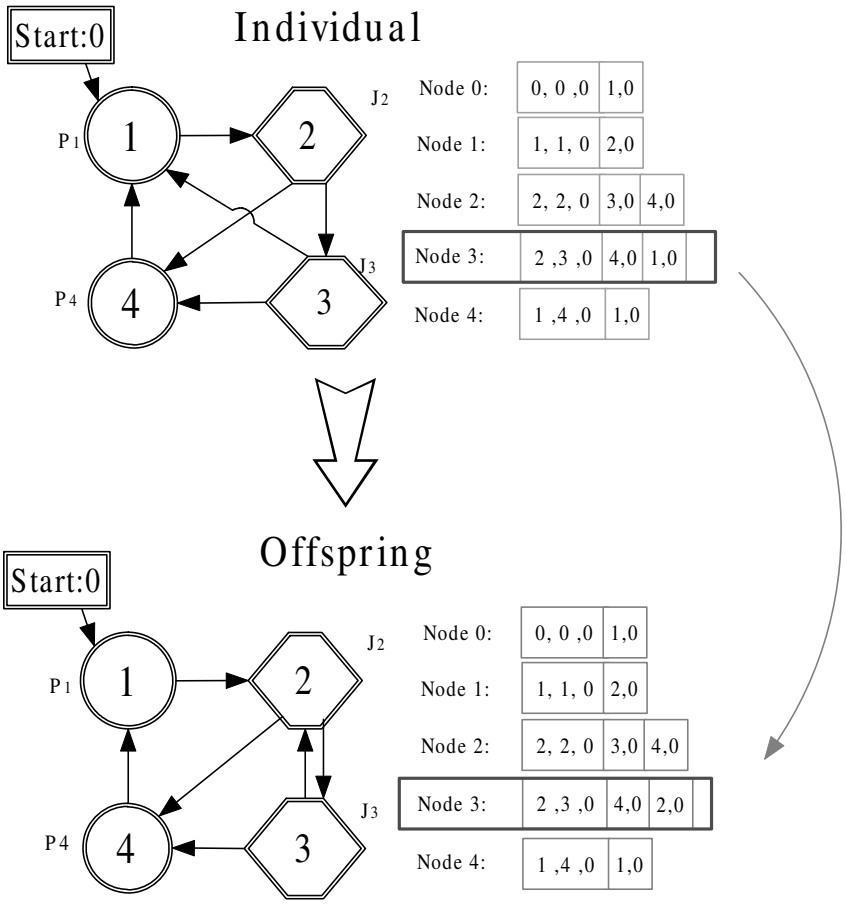


Figure A.3: The basic procedure of GNP mutation

- Use any selection method to choose an individual for mutation

- In one point crossover as shown in Fig. A.3, randomly choose a mutation point by probability of P_m.
- Change the mutation parts randomly as shown in Fig.A.3 to generate new offspring.

Crossover is also a genetic operator used to change individuals from one generation to the next generation. It is analogous to biological crossover, upon which genetic algorithms are based. Two parents exchange their gene and produce two offspring. The crossover operation is as follows:

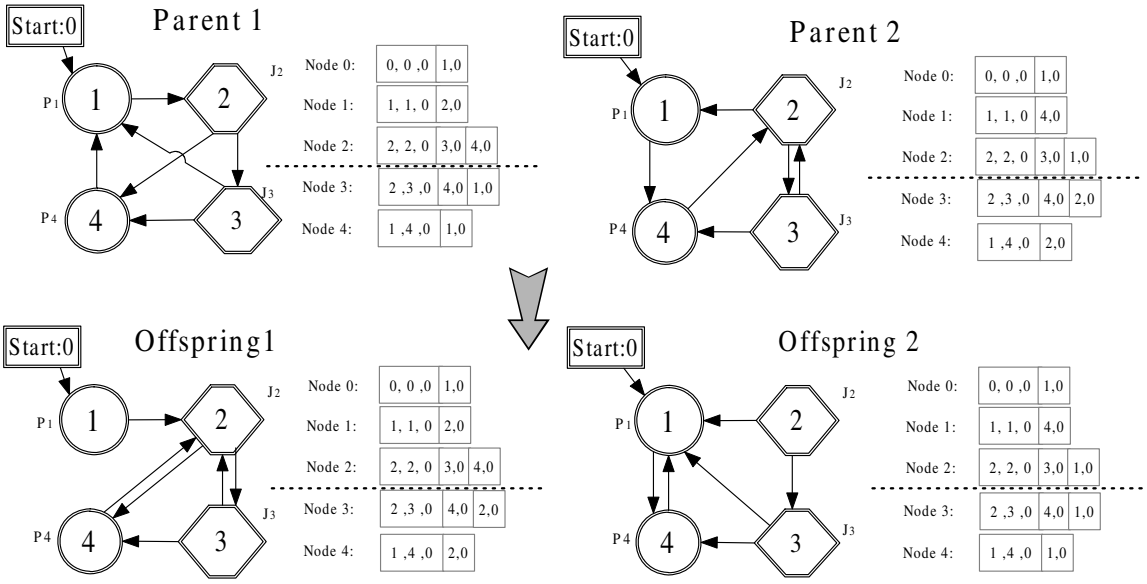


Figure A.4: The basic procedure of GNP crossover

- Use any selection method to choose two parents for crossover
- In one point crossover as shown in Fig.A.4, randomly choose a crossing point by probability of P_c.
- Exchange the parts divided by the crossing point as shown in Fig.A.4 to generate new offspring.

Using the above reproduction method, GNP generates the new population for the next generation. Thus, GNP evolves individuals using the above evaluation and reproduction sequence generation by generation.

Appendix B

Data Mining

B.1 Introduction

Data Mining, as the extraction procedure of implicit, previously unknown, and potentially useful information from databases, has been able to grasp the attention of many fields in scientific research, businesses, banking sectors, intelligence agencies and others from the early days of its inception. However, the application of data mining was not so easy as it is appears to be. People understand that data mining is a valuable tool, but may not know where to start or how to apply it to their businesses. The rapid growth of various tools and software during the recent years enable it to be used more and more widely than ever before.

Recent developments in the computing and electronics technology, especially in sensor devices and distributed systems, are leading to an exponential growth in the amount of data stored in the database. It has been estimated that this amount doubles every 20 years. For some applications, this data are doubling their size every 10 month.

As a result, the clear demand for sophisticated data mining tools is increasing to support decision-making applications based on huge amount of achieved databases. Data mining tools can be used to automatically find important patterns and also, tries to find patterns able to predict the behavior of specific attributes or features, whose particular process is also called predictive Data Mining.

Data Mining is used by businesses to improve its marketing performance and to understand the buying patterns of clients. Attribute Analysis, Customer Segmentation and Cross Selling are the important ways through which data mining is showing the new techniques in which businesses can multiply their revenue.

Data Mining can also be used in the banking sector for credit card fraud detection

by identifying the patterns involved in fraudulent transactions. It is also used to reduce credit risk by classifying a potential client and predicting bad loans.

Methods of data mining not only show their penitential capabilities in the business field, they are also drawing more and more attentions in all aspects of application fields, e.g., biology, environmental monitoring, satellite and medical images, security data and web, government data management and other decision-making applications.

Computational tools or solutions based on intelligent systems are being used with great success in data mining applications. Nature has been very successful in providing clever and efficient solutions to different sorts of challenges and problems. Data mining methods inspired on the biological process can be found in a large number of applications.

B.2 Association Rules

Data mining consists of attempting to discover novel and useful knowledge from data, trying to find patterns among databases that can help in intelligent decision making. Association rule mining is one way of representing the "useful knowledge" which could be extracted from the database.

Association rule mining tries to discover important associations and potential relations from the database and represents them as association rules. An association rule has the form of $(X \Rightarrow Y)$, where X represents antecedent and Y represents consequent. The association rule "$X \Rightarrow Y$" can be interpreted as: the set of attributes satisfying X is likely to satisfy Y.

The following is a formal statement of the problem of mining association rules. Let $A = \{A_1, A_2, \ldots, A_k\}$ be a set of events, called items or attributes. Let G be a large set of transactions, where each transaction T is a set of items such that $T \subseteq A$. Each transaction is associated with a unique identifier whose set is called TID. We define that a transaction T contains X, which is a set of some items in A, if $X \subseteq T$. An association rule is an implication of the "$X \Rightarrow Y$" where $X \subseteq A$, $Y \subseteq A$, and $X \cap Y = \emptyset$. As a result, X is called antecedent and Y is called consequent of the association rule. In general, a set of items is called an itemset. Each itemset has its own associated measure of statistical significance called support. If the number of transactions containing X in G equals t, and the total number of transactions in G is N, then we say that $support(X) = t/N$. The rule $X \Rightarrow Y$ has a measure of its strength called confidence defined as the ratio of $support(X \cup Y)/support(X)$. Calculation of the chi-squared value of the rule $X \Rightarrow Y$ is described as follows. Let $support(X) = x$, $support(Y) = y$, $support(X \cup Y) = z$ and the number of database tuples equals N. If the events X and Y are independent, we can get $support(X \cup Y) = xy$. Table B.1

Table B.1: The contingency of X and Y

	Y	$\neg Y$	$\sum_{row}$
X	Nxy Nz	$N(x-xy)$ $N(x-z)$	 Nx
$\neg X$	$N(y-xy)$ $N(y-z)$	$N(1-x-y+xy)$ $N(1-x-y+z)$	 $N(1-x)$
$\sum_{col}$	Ny	$N(1-y)$	N

(N: the number of tuples (= $|TID|$))

is the contingency of X and Y ; the upper parts are the expectation values under the assumption of independence, and the lower parts are observational values.

Now, let E denote the value of the expectation under the assumption of independence, and O is the value of the observation. Then, the chi-squared value is defined as follows:

$$\chi^2 = \sum_{AllCells} \frac{(O-E)^2}{E}, \tag{1}$$

We can calculate the chi-squared value using x, y, z and N of Table B.1 as follows:

$$\chi^2 = \frac{N(z-xy)^2}{xy(1-x)(1-y)}. \tag{2}$$

This has 1 degree of freedom. If it is higher than a threshold value (3.84 at the 95% significance level, or 6.63 at the 99% significance level), we should reject the independence assumption.

The time related association rule is used to represent the sequence pattern between attributes in the database in the GNP-based data mining method. Let $A_i(*)(t = p)$ be an attribute in a database at time p and its value is binary values of 1 or 0(after discretization). Here, $A_i(*)$ represents $A_i(Low)/A_i(Middle)/A_i(High)$. The proposed method extracts the following association rules:

$(A_j(*)(t = p) = 1) \wedge \cdots \wedge (A_k(*)(t = q) = 1) \Rightarrow$
$(A_m(*)(t = r) = 1) \wedge \cdots \wedge (A_n(*)(t = s) = 1)$

(briefly, $A_j(*)(t = p) \wedge \cdots \wedge A_k(*)(t = q) \Rightarrow$
$A_m(*)(t = r) \wedge \cdots \wedge A_n(*)(t = s)$)

Here, $p \leq q \leq r \leq s$, and the first t always equals 0, other time points are the relative time shifts from the first attribute. For example: $A_1(Low)(t = 0) \wedge A_2(Low)(t = 6) \Rightarrow A_3(High)(t = 22)$ means that A_1 is Low at time 0 and A_2 is Low at time 6, then A_3

becomes High at time 22. These kinds of rules could find time related sequential relations between attributes and would be used, for example, in prediction problems.

B.3 Time Transition and Time Related Association Rule Definition

The proposed method extracts important associations between attributes(sections on the traffic map) using association rule mining, and these associations are represented by time related association rules, whose accuracy and understandability are ensured by the proposed method.

The following is a formal statement of the problem of mining time related association rules. Let $A = \{A_1, A_2, \ldots, A_k\}$ be a set of items or attributes, i.e., it can represent each section on the traffic networks in the traffic density prediction problem. Let G be the time related traffic density database. Each time unit is associated with a unique identifier whose set is called $TimeID$. Let D=$\{D_1, D_2, \ldots, D_k\}$ be a set of time units of the event sequence occurrence, i.e., if attribute A_i of the event sequence occurs at time unit 3, then D_i is denoted as $D_i = 3$.

Definition B.1. *Time Related Attribute: The event sequence of time related attributes is defined as* $A_t = \{ A_{1(D_1)}, A_{2(D_2)}, \ldots, A_{k(D_k)} \}$*, where* D_k *is the time unit when the event of* A_k *occurs,* $A_k \in A$ *and* $D_k \in D$.

The continuous value of the time related database has been already discretized to three different levels: $Low, Middle$ and $High$ (briefly, L, M and H). In general, a set of items in A_t with its corresponding value levels is called time transition. A time transition is now defined as follows:

Definition B.2. *Time Transition: Time related attribute set* A_t *with its corresponding Low/Middle/High levels is defined as time transition* $TT = \{ A_1(V_1)_{(D_1)}, A_2(V_2)_{(D_2)}, \ldots, A_k(V_k)_{(D_k)} \}$*, where,* $V_k \in V = \{Low, Middle, High\}$.

Definition B.3. *Sub Transition: A time transition* STT *is called a sub time transition of the time transition* TT*, if and only if it constitutes a sub sequence starting from the first attribute of* TT.

Based on the Def.B.1 and Def.B.2, a time related association rule is an implication of the rule "$X \Rightarrow Y$" where $X \in TT$, $Y \in TT$. While, X is called antecedent and Y is called consequent of the time related association rule.

Definition B.4. *Time Related Class Association Rule: Let $A_i(*)$ $(t = p)$ be an attribute in a database at time unit p. $A_i(*)$ represents $A_i(Low)$/$A_i(Middle)$/$A_i(High)$. The time related class association rule mining extracts the following association rule:*

$A_j(*)(t = p) \wedge \ldots \wedge A_k(*)(t = q) \Rightarrow A_c(*)(t = s)$,

where, $A_c(*)(t = s)$ indicates the class of the consequent attribute.

Here, $p \leq q \leq s$, and the t of the first attribute always equal 0 and other time units are the relative time shifts from the first attribute.

Each time transition has its own associated measure of statistical significance called support, confidence and chi-squared value. These values are calculated based on the counts of the time transitions obtained from the searching mechanism. If the number of time transitions containing X in database G equals t, and the total number of time transitions in G is N, then $support(X) = t/N$. The rule $X \Rightarrow Y$ has a measure of its strength called confidence defined as the ratio of $support(X \cup Y)/support(X)$.

After calculating the criteria of time transitions, if the significance level of the time transition is important enough, which means the transition shows the important association between X and Y, then it can be picked up as an association rule. As a result, each time transition in fact represents an candidate rule in the proposed method.

References

[1] I. Kaysi, M. Ben-Akiva and H. Koutsopoulos, “An Integrated Approach to Vehicle Routing and Congestion Predictions for Real-time Driver Guidance”, Transportation Research Records, 1408, Transportation Research Board, Washington D.C., pp. 66-74, 1993.

[2] Hussein Dia, “An object-oriented neural network approach to short-term traffic forecasting”, *European Journal of Operational Research*, Vol.131, Issue 2, 2001.

[3] P. J. Lingras and P. Osborne, “Effect of noise on regression and neural network predictions”, In Proc. of the Conference of Canadian Society of Civil Engineers, pp. 331-339, Sherbrooke, Quebec, June, 1997.

[4] D. Joksimovic, M. Bliemler and P. Bovy, “Optimal toll design problem in dynamic traffic networks with joint route and departure time choice”, In Proc. of the 84th Annual Meeting of the Transportation Research Board, pp. 61-72, Washington, DC., 2005.

[5] J. H. Holland, “Adaptation in Natural and Artificial Systems”, *Ann Arbor: University of Michigan Press*, 1975.

[6] D. E. Goldberg, “Genetic Algorithm in search, optimization and machine learning”, Addison-Wesley, 1989.

[7] J. R. Koza, “Genetic Programming, on the programming of computers by means of natural selection”, *Cambridge, Mass., MIT Press*, 1992.

[8] J. R. Koza, “Genetic Programming II, Automatic Discovery of Reusable Programs”, *Cambridge, Mass.: MIT Press*, 1994.

[9] 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.

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

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

[12] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", In Proc. of the 20th VLDB Conf., pp.487-499, 1994.

[13] S. Brin, R. Motwani and C. Silverstein, "Beyond market baskets: generalizing association rules to correlations", In Proc. of the 1997 ACM SIGMOD Conf., pp.265-276, 1997.

[14] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen and A.I. Verkamo, "Finding Interesting Rules from Large Sets of Association Rules", In Proc. of Third Int'l Conf. Information and Knowledge Management, pp.401-408, 1994.

[15] A. Savasere, E. Omiecinski and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", In Proc. of 1995 Int'l Conf. Very Large Data Bases, pp. 432-443, 1995.

[16] J. S. Park, M. S. Chen and P. S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules", In Proc. of 1995 ACM SIGMOD Conf., pp.175-186, 1995.

[17] X. Wu, C. Zhang and S. Zhang, "Efficient Mining of Both Positive and Negative Association Rule", *ACM Transactions on Information Systems*, Vol.22, No.3, pp.381-405, 2004.

[18] R. Andrews, J. Diederich and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks", *Knowlege-Based Systems*, Vol. 8, No. 6, pp.373-389, 1995.

[19] A. B. Tickle, M. Orlowski, J. Diederich, "DEDEC: A methodology for extracting rule from trained artificial neural networks", In Proc. of the AISB'96 Workshop on Rule Extraction from Trained Neural Networks (AISB'00), pp.90–102, Brighton, 1996.

[20] A. K. H. Tung, H. Lu, J. Han and L. Feng, "Efficient Mining of Intertransaction Association Rule", *IEEE Transactions on Knowledge and Data Engineering*, Vol.15, No.1, pp.43-56, 2003.

[21] S. Wu and Y. Chen, "Mining Nonambiguous Temporal Patterns for Interval-Based Events", *IEEE*

Transactions on Knowledge and Data Engineering, Vol.19, No.6, pp.742-758, 2007.

[22] P. S. Kam and A. W. C. Fu, "Discovering Temporal Pattern for Interval-Based Events", In Proc. of Second Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK'00), pp. 317-326, 2000.

[23] C. W. Omlin and C. L. Giles, "Extraction of Rules from Discrete-time Recurrent Neural Networks", *Neural Networks*, Vol. 9, No. 1, pp.41-52, 1996.

[24] K. Shimada, K. Hirasawa and J. Hu, "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.

[25] H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, "Time Related Association Rules mining and its Application to Traffic Control", In Proc. of FAN symposium, pp. 97-102, Nagoya, 2007.

[26] K. Hirasawa, T. Eguchi, J. Zhou, L. Yu and S. Markon, "A Double-Deck Elevator Group Supervisory Control System Using Genetic Network Programming", *IEEE Trans. on Systems, Man and Cybernetics, Part C*, Vol. 38, No. 4, pp. 535-550, 2008.

[27] H. Zhou, W. Wei, M. K. Mainali, K. Shimada, S. Mabu and K. Hirasawa, "Class Association Rules Mining with Time Series and Its Application to Traffic load Prediction", In Proc. of SICE Annual Conference 2008, pp.1187-1192, Tokyo, Japan, 2008.

[28] H. Zhou, S. Mabu, M. K. Mainali, X. Li, K. Shimada and K. Hirasawa, "Generalized Association Rules Mining with Multi-Branches・Full-Paths and Its Application to Traffic Volume Prediction", In Proc. of ICROS-SICE International Joint Conference 2009, pp. 147-152, Fukuoka International Congress Center, Japan, 2009.

[29] H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Backward Time Related Association Rule mining with Database Rearrangement in Traffic Volume Prediction", In Proc. of IEEE International Conference on Systems, Man, and Cybernetics, pp. 1047-1052, San Antonio, USA, 2009.

[30] H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Generalized Time Related Sequential Association Rule Mining and Traffic Prediction", In Proc. of IEEE Congress on Evolutionary Computation 2009, pp. 2654-2661, Trondheim, Norway, 2009.

[31] B. Liu and W. Hsu and Y. Ma, “Integrating Classification and Association Rule Mining”, In Proc. of ACM International Conf. on Knowledge Discovery and Data Mining, pp. 80-86, 1998.

[32] H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, “Time Related Association Rule Mining with Attribute Accumulation Mechanism and Its Application to Traffic Prediction”, In Proc. of IEEE Congress on Evolutionary Computation 2008, pp. 305-311, Hong Kong, China, 2008.

[33] M.K. Mainali, K. Shimada, S. Mabu, K. Hirasawa, “Optimal Route of Road Networks by Dynamic Programming”, In Proc. of International Joint Conference on Neural Networks, pp. 3416-3420, 2008.

[34] H. Zhou, S. Mabu, W. Wei, K. Shimada and K. Hirasawa, “Time Related Class Association Rule Mining and Its Application to Traffic Prediction”, *IEEJ Transactions on Electronics, Information and Systems*, Vol. 130, No.2, pp.289-301, 2010.

[35] H. Zhou, S. Mabu, X. Wang and K. Hirasawa, “Multi-Branches and Full-Paths(MBFP) Generalized Association Rule Mining and Classification in Traffic Volume Prediction”, *IEEJ Transactions on Electrical and Electronic Engineering* (to be published).

[36] K. Tamura and M. Hirayama, “Toward realization of VICS - Vehicle Information and Communication System”, In Proc. of the Vehicle Navigation and Information Systems Conference, pp. 72-77, 1993.

[37] H. Zhou, S. Mabu, X. Li, K. Shimada and K. Hirasawa, “Generalized Rule Extraction and Traffic Prediction in the Optimal Route Search”, In Proc. of IEEE World Congress on Computational Intelligence, pp. 2625-2632, Barcelona, Spain, 2010.

[38] H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, “Time Related Association Rules Mining with Attributes Accumulation Mechanism and its Application to Traffic Prediction”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 12, No. 5, pp. 467-478, 2008.

[39] E. W., Dijkstra, “A note on two problems in connection with graphs”, *Numerische Mathematik*, 1959.

Acknowledgements

This thesis could not be finished without the help and support of many people who are gratefully acknowledged here. My deepest gratitude goes first and foremost to Professor Hirasawa , my supervisor, for his constant encouragement and patient guidance. He has walked me through all the stages of writing this thesis. Without his consistent and illuminating instruction, this thesis could not have reached its present form. He has offered me valuable ideas, suggestions and criticisms with his profound knowledge in forensic language and rich research experience.

Secondly, I would like to express my great respect and gratitude to Prof. Koyanagi, Prof. Matsumaru of Waseda University and Prof. Kawabe of Kyushu University, who make efforts to provide me lots of helpful comments and advise as the co-examiners.

Thirdly, I would like to express my heartfelt gratitude to Doctor Shimada, who led me into the world of data mining. His meticulous guidance and invaluable suggestions are indispensable to the completion of this thesis.

I am also grateful to Dr. Mabu, whose patient attitude and kindness helped me a lot to the completion of my doctor courses. With his extraordinary patience and consistent encouragement, he gave me help by providing me with necessary materials, advice and inspiration. Thanks are also due to my friends in our laboratory, who never failed to give me great encouragement and suggestions. Special thanks should go to Ms. Wei Wei, Ms. Yu Lu, Ms. Yu Shanqing, Ms. Chuan Yue and Mr. Fengming Ye for their encouraging me when I had problems carrying out the research and writing this thesis.

Last my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years. I also owe my sincere gratitude to my friends and my fellow classmates who gave me their help and time in listening to me and helping me work out my problems during the difficult course of the thesis.

Research Achievements

Journal Paper

(1) H. Zhou, S. Mabu, X. Wang and K. Hirasawa, “Multi-Branches and Full-Paths (MBFP) Generalized Association Rule Mining and Classification in Traffic Volume Prediction”, IEEJ Transactions on Electrical and Electronic Engineering (accepted).

(2) X. Li, S. Mabu, H. Zhou, K. Shimada, K Hirasawa, “Genetic Network Programming with Estimation of Distribution Algorithms for Class Association Rule Mining in Traffic Prediction”, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 14, No. 5, pp. 497-509, 2010/07.

(3) H. Zhou, S. Mabu, W. Wei, K. Shimada and K. Hirasawa, "Time Related Class Association Rule Mining and Its Application to Traffic Prediction", IEEJ Transactions on Electronics, Information and Systems, Vol. 130, No.2, pp.289-301, 2010/02.

(4) H. Zhou, S. Mabu, W. Wei, K. Shimada and K. Hirasawa, “Traffic Flow Prediction with Genetic Network Programming (GNP)”. Journal of Advanced Computational Intelligence and Intelligent Informatics Vol. 13, No. 6, pp.713-725, 2009/11.

(5) 嶋田香、王路涛、周輝宇、平澤宏太郎, “交替型遺伝的ネットワークプログラミングによる2つの属性グループの相関ルールの抽出”, Journal of Signal Processing (信号処理), Vol. 13, No. 3, pp. 267-278, 2009/05.

(6) H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, “Time Related Association Rules Mining with Attributes Accumulation Mechanism and its Application to Traffic Prediction”, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 12, No. 5, pp. 467-478, 2008/07.

(7) W. Wei, H. Zhou, K. Shimada, S. Mabu and K. Hirasawa, "Comparative Association Rules Mining Using Genetic Network Programming (GNP) with Attributes Accumulation Mechanism and its Application to Traffic Systems", Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 12, No. 4, pp. 393-403, 2008/07.

International Conference Paper

(1) H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Time related Association Rule Mining with Accuracy Validation in Traffic volume Prediction with Large Scale Simulator", 2010 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2249-2255, Istanbul, Turkey, 2010/10.

(2) X. Li, S. Mabu, H. Zhou, K. Shimada and K. Hirasawa, "Analysis of Various Interesting Measures in Classification Rule Mining for Traffic Prediction", SICE Annual Conference 2010, pp.1969-1974, Taipei, Taiwan, 2010/08.

(3) X. Wang, S. Mabu, H. Zhou and K. Hirasawa, "Time Related Association Rules Mining with Attributes Accumulation Mechanism Applied to Large-scale Traffic System", SICE Annual Conference 2010, pp.2637-2641, Taipei, Taiwan, 2010/08.

(4) H. Zhou, S. Mabu, X. Li, K. Shimada and K. Hirasawa, "Generalized Rule Extraction and Traffic Prediction in the Optimal Route Search", IEEE World Congress on Computational Intelligence, pp. 2625-2632, Barcelona, Spain, 2010/07.

(5) X. Li, S. Mabu, H. Zhou, K. Shimada and K. Hirasawa, "Genetic Network Programming with Estimation of Distribution Algorithms for Class Association rule Mining in Traffic Prediction", IEEE World Congress on Computational Intelligence, pp. 2673-2680, Barcelona, Spain, 2010/07.

(6) H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Backward Time Related Association Rule mining with Database Rearrangement in Traffic Volume Prediction", IEEE International Conference on Systems, Man, and Cybernetics, pp. 1047-1052, San Antonio, USA, 2009/10.

(7) X. Li, S. Mabu, H. Zhou, K. Shimada and K. Hirasawa, "Genetic Network Programming with Estimation of Distribution Algorithms, and its Application to Association Rule Mining for Traffic Prediction", ICROS-SICE International Joint Conference 2009, pp.3457-3462, Fukuoka International Congress Center, Japan, 2009/08.

(8) Y. Wang, S. Mabu, H. Zhou, X. Li, K. Shimada, B. Zhang and K. Hirasawa, "Time Related Association Rules Mining for Traffic Prediction based on Genetic Network Programming combined with Estimation of Distribution Algorithms", ICROS-SICE International Joint Conference 2009, pp. 3468-3473, Fukuoka International Congress Center, Japan, 2009/08.

(9) H. Zhou, S. Mabu, M. K. Mainali, X. Li, K. Shimada and K. Hirasawa, "Generalized Association Rules Mining with Multi-Branches·Full-Paths and Its Application to Traffic Volume Prediction", ICROS-SICE International Joint Conference 2009, pp. 147-152, Fukuoka International Congress Center, Japan, 2009/08.

(10) H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Generalized Time Related Sequential Association Rule Mining and Traffic Prediction", IEEE Congress on Evolutionary Computation 2009, pp. 2654-2661, Trondheim, Norway, 2009/05.

(11) W. Wei, H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Traffic Flow Prediction with Genetic Network Programming", SICE Annual Conference 2008, pp.670-675, Tokyo, Japan, 2008/08.

(12) H. Zhou, W. Wei, M. K. Mainali, K. Shimada, S. Mabu and K. Hirasawa, "Class Association Rules Mining with Time Series and Its Application to Traffic load Prediction", SICE Annual Conference 2008, pp.1187-1192, Tokyo, Japan, 2008/08.

(13) W. Wei, H. Zhou, K. Shimada, S. Mabu and K. Hirasawa, "Comparative Association Rules Mining using Genetic Network Programming with Attribute Accumulation Mechanism and Its Application to Traffic Systems", IEEE Congress on Evolutionary Computation 2008, pp. 292-298, Hong Kong, China, 2008/06.

(14) H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, "Time Related Association Rule Mining with Attribute Accumulation Mechanism and Its Application to Traffic Prediction", IEEE Congress on Evolutionary Computation 2008, pp. 305-311, Hong Kong, China, 2008/06.

Domestic Conference Paper

(1) X. Wang, S. Mabu, K. Shimada, H. Zhou and K. Hirasawa, "Time Related Association Rules Mining with Attributes Accumulation Mechanism Applied to Large-scale Traffic System", 第 54 回システム制御情報学会研究発表講演会 SCI'10, pp.259-260, 京都, 2010/05.

(2) X. Li, H. Zhou, S. Mabu, K. Shimada and K. Hirasawa, "Classifier for Traffic Prediction Using Genetic Network Programming with Estimation of Distribution Algorithms", 計測自動制御学会システム・情報部門学術講演会 2009 (SICE SSI 2009), pp. 286-291, 横浜, 2009/11.

(3) H. Zhou, W. Wei, K. Shimada, S. Mabu and K. Hirasawa, "Time Related Association Rules mining and its Application to Traffic Control", 第 17 回インテリジェント・システム・シンポジウム, pp. 97-102, 名古屋, 2007/08.

(4) W. Wei, H. Zhou, K. Shimada, S. Mabu and K. Hirasawa, "Comparative Association Rules Mining using Genetic Network Programming (GNP) and its Application to Traffic Content", 第 17 回インテリジェント・システム・シンポジウム, pp. 103-108, 名古屋, 2007/08.

Book Chapter

(1) H. Zhou, K. Shimada, S. Mabu and K. Hirasawa, “Sequence Pattern Mining”, Foundations of Computational Intelligence Volume 4: Bio-Inspired Data Mining, Part I, Chapter 2, pp. 23-48, Springer Verlag 2009.