Rule-Based System for Finding Shortest Traveling Time

By

Mr. Noppasak Wannavorrawong

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Information Technology
Assumption University

October, 2003

Rule-Based System for Finding Shortest Traveling Time

By

Mr. Noppasak Wannavorrawong

Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Science in Information Technology Assumption University

October , 2003

The Faculty of Science and Technology

Master Thesis Approval

Thesis Title Rule-Based System for Finding Shortest Traveling Time

By Mr. Noppasak Wannavorrawong
Thesis Advisor Dr. Jirapun Daengdej
Academic Year 1/2003

The Department of Information Technology, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **IT7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology.

Approval Committee:

(Dr. Jirapun Daengdej)
Advisor

(Asst.Prof.Dr. Thiraphong Charoenkhunwiwat)
Committee Member

(Asst.Prof.Dr. Thotsapon Sortrakul)
Committee Member

(Asst.Prof.Dr. Surapong Auwatanamongkol)
Commission of Higher Education
University Affairs

Faculty Approval:

(Asst.Prof.Dr. Thotsapon Sortrakul)
Program Director

(Asst.Prof. Dr. Supavadee Nontakao)
Dean

October / 2003

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

The Faculty of Science and Technology

Thesis Approval

Thesis Title	Improve Case-Based Retrieve Technique
By	Noppasak Wannavorrawong
Thesis Advisor	Jirapun Daengdej
Academic Year	2003

The Department of Computer Science Faculty of Science and Technology, Assumption University approved this final report for the **Twelve** Credits course **IT 7000 Master Thesis.** Submitted as partial fulfillment in the requirements for a degree in Master of Science and Information Technology.

The approval Committee members:

(Jirapun Daengdej) Advisor	(.....) Co-Advisor
(.....) Committee	(.....) MUA Representative

Faculty Approval:

(.....) Director	(.....) Dean

_____ / SINCE1969
Month/Year

ACKNOWLEDGEMENTS

The writer desires to express his most sincere appreciation and gratitude to Dr. Jirapun Daengdej for the encouragement and advice through the course of this study and for his help in the preparation and writing of this thesis.

Rule-Based System for Finding Shortest Traveling Time

Noppasak Wannavorrawong

Faculty of Science and Technology

Assumption University, Bangkok, Thailand

E-mail: noppasakw@hotmail.com

Abstract

In this paper I explored the techniques to improve and retrieve performances that solves content problem in a case-based reasoning (CBR). It focuses on a part of Bangkok's domain map. I also describe a hybrid between case-based reasoning (CBR) and simple rule-based reasoning to analyst and rank the retrieved data in case-based database. These are evaluated by comparing the traveling time and processing time of proposed technique with alternative techniques. The result of proposed technique represents suitable paths for each situation, and the meaning of a suitable path is "the path that uses shortest traveling time".

1. Introduction

1.1 Definition of CBR

Roger Schank [1] founded the Case-Based Reasoning (CBR). It is one of the machines learning Case-based reasoning used in solving a problem, which is done by using an old solution that is stored in the case base [2]. The process of CBR refers to a cognitive analogy to previous cases. Therefore, the basis of the previous cases in deciding how to solve the problem, are not unique. It, however, depends on range of cases in the case database and retrieve algorithm.

CBR consists of database that stores previous cases and solutions and indices to retrieve a previous case and store a new case. The solutions stored in case-based database may either be a successful solution or a failure solution. Algorithm on the other hand measures similarities and adapt recalled cases.

The case is the feature describing a problem where content may include text, number, symbols and multimedia. The case representation may be the records in database or unstructured text.

Name	Age	Graduate	Job
A	25	Bachelor	Sale
B	30	Bachelor	Sale

Table 1-1: The record in case-based database (structured text)

The problem: The computer can't start.

Solve problem: -Turn on the computer

- Check the power supply

Figure 1-1: The unstructured text

The type of case consists of

1. Metadata is the un-indexed feature. It is not used for retrieval purposes. It simply describes the background information of a case.
2. Data is indexed features. It is used for retrieval.

1.2 How CBR works?

The CBR process is not the same as other machines learning, basically because the CBR process is like a human mind solving a problem. A human being uses previous experience stored in the brain as a basis to make a decision on how to solve a problem and in the same manner that it keeps the new experience in the brain. In comparison with other machine-learning example the neural network, the neural network must retain training set for the new problem when that training set cannot solve the problem. Therefore in this situation, the processing time that CBR used is lower than the neural network. The next paragraph describes the detail of CBR process.

When the target problem is identified, CBR recalls a previous case and a successful solution that is similar with the target problem. If CBR finds a previous case and a successful solution, CBR adapts the recalled previous case and adjust between previous case and the target problem. When CBR adjusts the cases successfully, CBR then revise a new solution. If the new solution is not acceptable, CBR will again adjusts a previous case and the target problem. When CBR accepts a new solution, CBR stores a target problem and a new solution into the case database, to be used as reference in solving future target problem. However, there are times when CBR stores the new solution even if adaptation is a failure. The failure and the success solutions help the person adjust the target problem. When the person knows

that each adaptation techniques may either be a failure or a success, the person will know to adjust the previous case and the target problem efficiently.

The competence of CBR is better than other machine learning dynamics in an environment that cannot use static algorithm or static rule for solving a problem.

The cycle of Case-base reasoning is Retrieve- Reuse – Revise – Retain. There are [4] cycle details:

1. Retrieve the most similar case
2. Reuse the information to solve the problem
3. Revise the proposed solution
4. Retain the experienced used for future problem solving

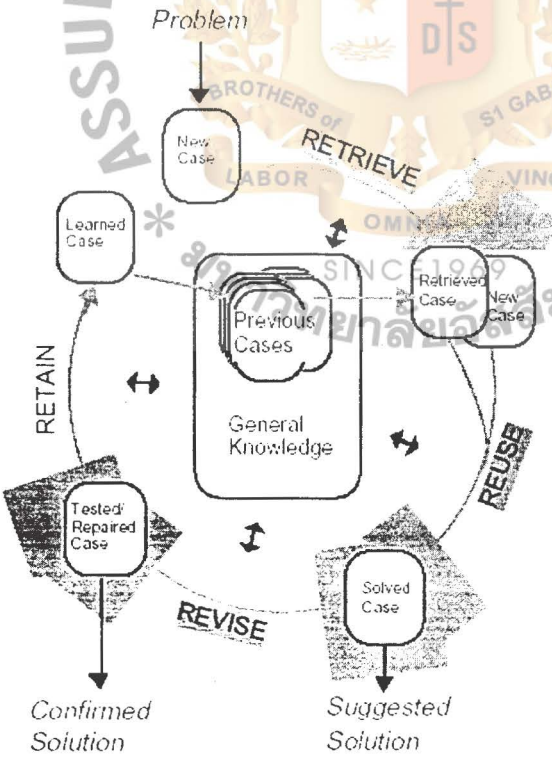


Figure 1-2: CBR Cycle [4]

1. Retrieval process is the first process in case-based reasoning. Case-based reasoning begins by retrieving a problem and solution in case-based database, which retrieval information is similar with the target problem.

2.Reuse process is adaptation process between problem, solution in case-based database and the target problem. This process primarily use human to decide adaptation because many problems are dynamic problems. Non-human adaptation doesn't succeed for many situations, see example below

In dynamic situation (e.g. patient case analysis) correct result is needed. If the reuse process produces a incorrect result, the patient could be in danger.

Therefore this process must use human mind in deciding how to adapt between the target problem and the problem with solution in case-based database. At present many researches produce the technique to assist case-based reasoning for adaptation e.g. hybrid other artificial intelligence to help case-based reasoning.

3.Revise process is checking the result from previous process. If the result of reuse process is not acceptable, case-based-reasoning will go back to previous process again. When the result of reuse process is acceptable, the process sends the target problem and the result of reuse process to retain process.

4.Retain process is keeping the target problem and solution in case-based database regardless the solution is a failure or successful. This process may maintain the index in case-based database.

The figure 1-3, 1-4,1-5, 1-6 and 1-7 describe the example of CBR process.

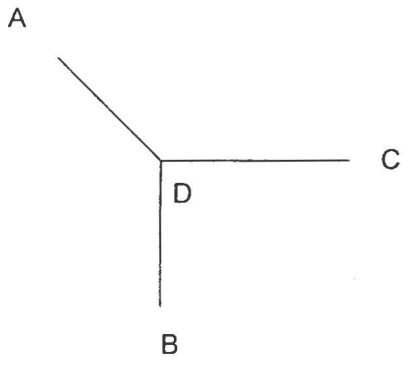


Table 1-3: Case-based database

Table 1-4: The new problem

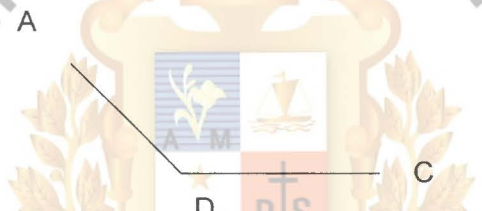


Table 1-5: The retrieved information

Table 1-6: The results of case after reuse and revise process

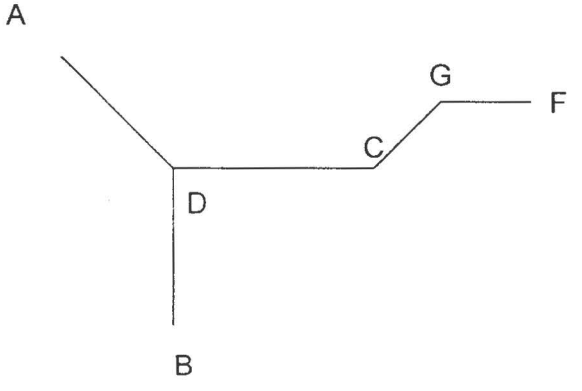


Table 1-7: The case-based database after retrain process

Figures 1-4 illustrate a case and solution that is stored in case-based, and figures 1-5 display the target problem. CBR retrieves cases and solutions in order to solve target problem. (Figure 1-6) After CBR retrieves a case, CBR adjusts target problem in order to produce a new solution (Figure 1-7) then CBR check new solution. When the new solution is suitable, CBR stores it in the case-based. (Figure 1-8)

In the previous section, this paper describes the characteristics and the competence of CBR as follows

1. CBR can solve complex problem.
2. CBR is simplified to maintain knowledge.
3. Because CBR use human to revise new solution therefore the new solution is acceptable by users.

1.3 Applications of CBR

Today there are many organizations that use CBR method as guideline to help them make decisions. One of the organizations that use CBR applications is General Electric (GE). The General Electric (GE) has used many CBR applications to make decisions. Some of the examples are as follows:

1. GE's Aircraft Engine Monitoring system. It is used for monitoring aircraft engines to detect abnormal behavior before it malfunctions.

2. Call Center Automation. It is used for customer service that consists of the following:

- Call Taker Education: helps customer's service center of GE in providing answers to customer's inquiries.
- Web Self-Service: the call center tool is web-based application that is available for customers. The web self-service tool is made on the www.geappliances.com.
- E-mail Response Automation: The tool stores replies to frequently asked questions, so that they can be reused in answering further similar questions. The case select reply system uses rule-based, in order to search the text in e-mail messages.

Only case-based reasoning cannot solve problem efficiently because case-based reasoning contains several problems. These problems affect the competence of case-based reasoning therefore many researches tried to hybrid several techniques in order to improve the competence of case-based reasoning. Problems of case-based reasoning are discussed in the next section.

Hewlett Packard hybrids are several artificial intelligence such as rule-based reasoning, case-based reasoning, qualitative reasoning, Bayesian networks, neural networks and model-based reasoning, all are used to diagnose any error in network system.

The diagnosis is determined by data instead of goals. The rule in rule-based reasoning consists of two parts, the first part describes the most commonly occurring symptom/fault associations, and the other part contains meta-rules, giving the control knowledge about when and how to use different diagnostic strategies. After receiving the filtered and correlated events, the diagnostic system works as follows. It first uses rule-based reasoning to diagnose the underlying faults based. If this fails, it uses the meta-knowledge described in the rule to provide some clues for choosing case-based

reasoning or model-based reasoning to identify the error. If the case-based reasoning is chosen and the result is not acceptable, because there are no cases resembling the current situation in the case library or the proposed solution is unacceptable, it will then try the model-based reasoning to diagnose the error. The model-based reasoning uses Bayesian networks to choose the components for testing based on their probabilities after it has obtained the suspect set from the fault detection process. During the candidate removal, model-based reasoning may use qualitative reasoning to find the cause of the normal and the fault behaviors of some components of which quantitative models are difficult to acquire or too complicated to describe. Thus, in this hybrid approach, the behaviors of managed components, of which quantitative models are difficult to acquire, are described qualitatively. After the faults have been diagnosed, the solution will be evaluated, and then stored as a new case together with its solution steps and the current contextual information.

Cunningham and Smyth describe CBR solving problem as the "The traveling salesman's problem", for it provides a quick and good quality solution. They reuse old components in order to reduce size of case base database [8].

1.4 Problems of CBR

Although CBR is easy to create, there are still a lot of problems as it is difficult to use and maintain.

The main problem of CBR is "Adaptation task". There is only little research that has been undertaken on this problem. [12] At present, the technique adaptation case to solve problem is not guaranteed. Many CBR still use human mind to reuse a case in solving a problem because some problem domains are very complicated for adaptation, diagnosing a patient is an example.

Research tried to hybrid other artificial intelligence with CBR in order to improve the adaptation process, an example is shown below

D. Hastings and L. K. Branting and J. A. Lockwood approach uses model-based reasoning in order to assist the case-based reasoning for adaptation. [15]

The retrieve process contains a problem with similar case. The concern of CBR in the retrieve process is as follows:

"What technique can find the most similar case with target problem?"

For each situation, the retrieval technique may differ. Sometimes retrieval technique wants only a case equal to the target problem. There are also times when CBR wants retrieval case to cover the target problem, and the retrieval technique does not only want a case equal to the target problem.

For revise process, the problem maybe the same with the reuse problem as computers cannot revise new solutions by itself. This process must use expertise to work.

The problem of retain process covers the following:

1. How to decide to choose the retain technique in order to store the new solution and problem into case-based database.
2. A data in CBR is structure representation, the problem occurs when CBR generates index which may occur as an "index problem".
3. The problem concerns a large quantity of data, which are case-based and uses fixed storage. It however reduces the performance of CBR, because it cannot store a new problem and solution and will also affect the retrieval time.

The index problem's concern amounts to deciding what types of indexes to use for future retrieval, and how to structure the search space of indexes.

Noise problem or Inconsistent-case problem is a situation wherein false information is stored in the database therefore affects the systems competence. Furthermore, the accuracy of the system is affected when there is noise stored in the database.

The cause of noise consists of

- Users error to input information. An example is the estimated price of used car domain, the used car price is 300,000 bahts but the user has mistakenly input the 30,000 bahts.
- Information in the section contradicts with other sections.

To process time is also one of the problems of CBR, it uses time higher than other machine learning techniques e.g. model-based reasoning, when the domain needs the accuracy and speed for processing e.g. fault diagnosis in machine, model-based is still better than CBR. The model-based calculate input by formula then displays the result however CBR retrieves data in a database, adapts a case and revises it. This problem is concerned with the quantity of information in case-based database. In case the case-based database contains a lot of information, this situation decreases the retrieval efficiency and affect the other process (e.g. retrain process) that this paper describes later.

If a data in CBR is unstructured text, it is hard to retrieve and maintain and in case of redundancy problem, it is hard to identify redundant cases.

Sample Case:

CASE NAME: The printed page is black.

POSSIBLE CAUSE: The printed page is black due to an unseated toner cartridge, Reset the toner cartridge and reprint the document.

CASE SOLUTION:

To reset the toner cartridge:

1.) Turn off the laser printer.

2.) Open the top by pressing button to release latch.

Figure 1-7: Example of the unstructured text in CBR. [13]

In figures 1-7, CBR cannot retrieve case efficiency because CBR does not know what case is similar to the target problem. In the case of retrieval case, CBR cannot retrieve the cases that cover the target problem.

Content problem of CBR is the situation where in the case-based database stores a lot of information that reduces retrieval efficiency of CBR. The content problem affects the people who decide on how to adjust a previous case and the target problem.

In this section, each problem is considered as the content problem. The next section describes in detail the problems that this paper considers.

1.5 Considered Problems

In the previous section, the content problem occurs when a lot of information is stored in case-based database. When retrieving a previous case in case-based database, a lot of information should be retrieved which affects the adaptive person. For adaptation process, the person gets confused to adjust the previous case and the target problem when the quantity of previous case is large.

The example of considered problem is described below

Start From	To Target path
60	14

Table 1-2: The target problem

No.	Way
1	60->42->44->39->34->24->14->
2	60->42->44->39->34->24->26->14->
3	60->42->44->39->40->35->34->36->22->24->26->14->
4	60->3->39->34->36->22->24->14->
5	60->3->39->34->36->22->24->26->14->
6	60->3->39->34->24->14->
7	60->3->39->34->24->26->14->
8	60->3->39->40->35->34->36->22->24->26->14->
9	60->3->39->40->35->34->36->22->24->14->
10	60->3->39->40->35->34->24->26->14->
11	60->3->4->36->22->24->14->
12	60->3->4->36->22->24->26->14->
13	60->3->4->36->34->24->14->
14	60->3->4->36->34->24->26->14->
15	60->3->4->36->34->35->28->26->24->14->
16	60->3->4->36->34->35->28->26->14->
17	60->3->39->34->24->22->5->6->8->11->14->
18	60->3->39->44->45->40->35->34->36->4->5->22->24->26->14->
19	60->3->39->34->36->4->5->22->24->14->
20	60->3->39->34->24->22->36->4->5->6->8->11->14->

.

.

.

183	60->42->44->45->40->39->34->35->28->26->24->14->

Table1-3: The retrieval case (The content problem)

In the case of adaptation by experts, the quantity of retrieved information is important. As an example, if the quantity is low, experts cannot adjust the previous case efficiently. However, if the quantity of retrieval case is very large, experts could have difficulties in adapting the case. An example is described below:

In some domain the person who adjusts the case might not be an expert. They may not have enough knowledge in making decision on how to adjust the previous case and the target problem efficiently.

For instance the Microsoft help, this program is an example of CBR, when users want to know how to solve windows problem, they input the target problem then this program retrieves a lot of previous cases. Many users are not experts therefore they don't know what previous case is suitable for them. Sometimes they may test all previous cases in order to find new solution.

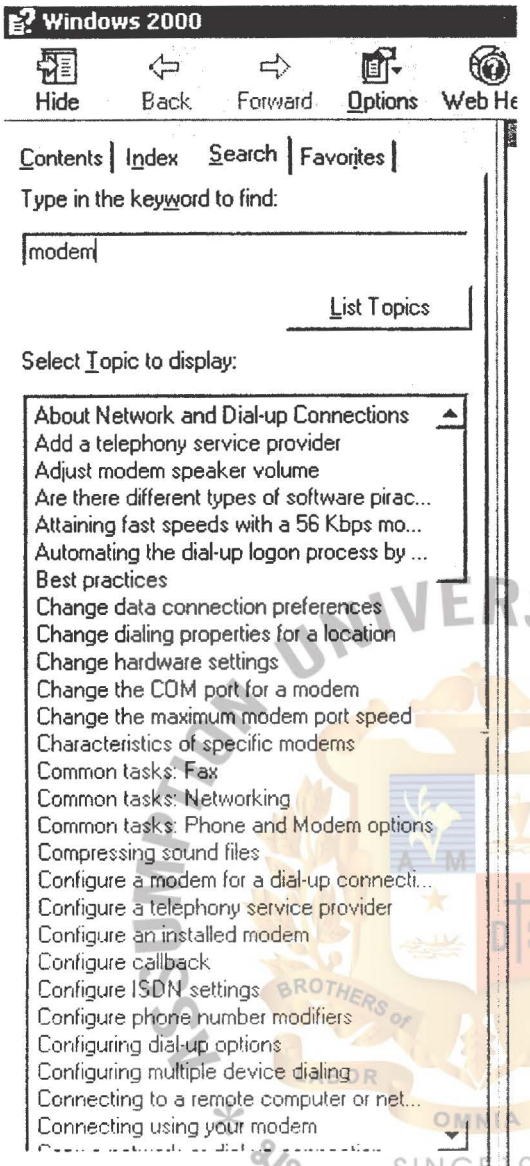


Figure 1-7: Microsoft Help program

In case the system retrieves a lot of cases, the person who determines the problem may use random method in order to choose a previous case. For this method, they will not know if the selected case is suited for the problem until they use it. Sometimes they can find the suitable solution and sometimes they cannot.

In this section, this paper describes the detail of considered problem. The content problem is concerned with the amount of previous case. It is also concerned with the person who adjusts the previous case and the target problem.

The content problem has many research concerns that have to be improved and this will be shown in the next section of this paper.

2. Literature Reviews

As this paper is focused on content problem in case-based database, it considers more research in case-based maintenance to reduce the quantity of cases that could be retrieved from case-based database.

2.1 Remember to forget

Smyth and Keane developed the strategy concerning case-base deletion policies [5] they divided the format of a case into three categories: First, the Pivotal case is a case where reachability set contains just the case itself. Deleting a pivotal case reduces the competence of the system. Second, the Auxiliary case is a subset case in a pivotal case. Deleting an auxiliary case does not reduce the competence of a system. Third, the Support case, these are groups of cases that cover other cases in a group. Deletion of members in the support group does not reduce competence of the system. However, deletion of pivotal case or deletion of the entire group reduces the competence of system.

Furthermore, the factors affecting competence of the system are coverage and reachability. Coverage is defined as a set of problem that a case can solve and reachability on the other hand is defined as a set of case that can solve a target problem.

Definition of Coverage and Reachable

Adaptable(x, y) = case 'x' retrieve and adapt to provide solution for case 'y'

A case-base $C=\{c_1,c_2,\ldots,c_n\}$ $c \in C$

Coverage(c)={c' ∈C:Adaptable(c,c')}

Reachable(c)={c' ∈C:Adaptable(c',c)}

The Footprint Deletion Policy

This policy provides a means of ordering deletion cases. The auxiliary cases are lowest important, this case has no competent contribution. Next, the support cases are spanning cases. Finally, the pivotal cases are high important.

The footprint deletion strategy is simply a deletion technique, in which auxiliary case is removed first, followed by deletion of support case. This technique keeps only a pivotal case. Figure 2-1 describes deletion algorithm, as shown below

```
DeleteCase(Cases):
If there are auxiliary cases then
    SelectAuxiliary(AuxiliaryCases)
ElseIf there are support cases then
With the largest support group
    SelectSupport(SupportGroup)
ElseIf there are spanning cases then
    SelectSpanning(SpanningCases)
ElseIf there are pivotal cases then
    SelectPivot(PivotalCases)
EndIf
```

Figure 2-1: The Footprint Deletion Alogrithm [5]

This paper describes the example of this technique bellow. This paper uses the estimate used car's price domain in order to describe the example of this technique.

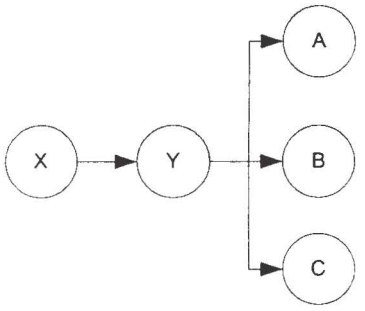


Figure 2-2: Example of data storing in case-based database.

In figure 2-2, case Y, A, B and C are auxiliary case of case X because system retrieves case X and adapt in order to solve case. Y, A, B and C.

Figure 2-3: Example of a Case using Footprint deletion policy

In figure 2-3 shows the result of footprint deletion policy, the remainders are case X.

2.2 Remember to add

Zhu and Yang illustrate the policy additional case. They argue the remember to forget technique on the following grounds:

1. Nothing is mentioned why auxiliary cases should be deleted.

2. How the quality of resulting case is ensured after the deletion is done.

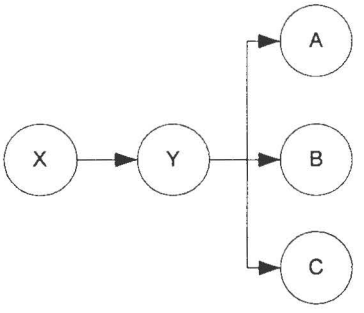


Figure 2-4: Case based structure graph.

In figure 2-4, remember to forget technique's classification scheme, pivotal case is case "X", spanning case is case "Y" and auxiliary case are case "A", "B", "C". When using footprint deletion technique, auxiliary will be deleted ("A", "B", "C"), competence of case based will be lost.

The neighborhood of a case is likewise being considered. This policy is good when neighborhood is large. The structure of case-based database represents the graph format. This technique does not only keep the highest benefit one can derive from a case but also a case with the largest neighborhood. [3] This technique keeps the highest benefit case into empty case-based database. This technique considers the case coverage as the benefit due to the fact that benefit can define many notations. Remember to add technique approach is applicable in making travel plan between cities. Users want to minimize the cost of searching for each plan.

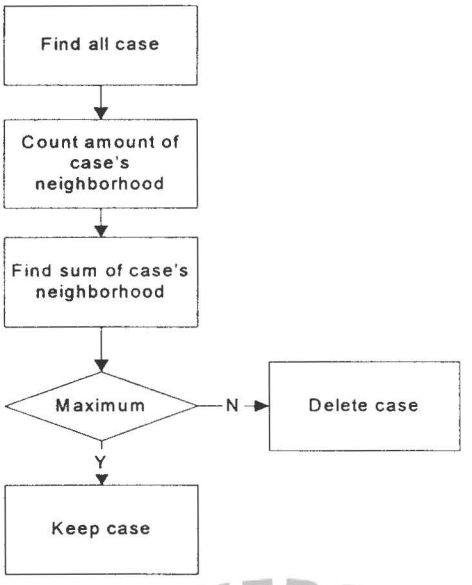


Figure 2-5: Diagram of remember to add policy process

$$\text{Coverage}(X1) = \frac{\sum_{x \in N(X1)} P(x)}{\sum_{x \in X} P(x)}$$

X = case space

X1 = subset of X

N(X1) = neighborhood of X1

P = frequency of x

x = subset of X

If case X1 has high coverage, case X1 has a high benefit too. This policy only stores the highest benefit of a case in new case-based database.

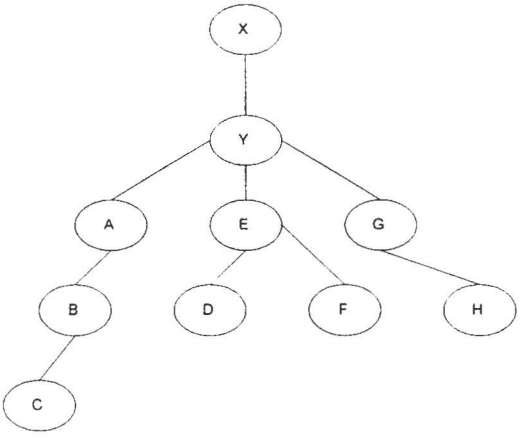


Figure 2-6: Example of case in case-based database

In figure 2-6, coverage value of (x,c), (x,d), (x,f) and (x,h) are

(x,c) =(1+1+3+1)/10

=.6

(x,d) =(2+3+1)/10

=.6

(x,f) =(2+3+1)/10

=.6

(x,h) =(1+3+1)/10

=.5

Hence the highest benefit of case are (x, c), (x, d) and (x, f)

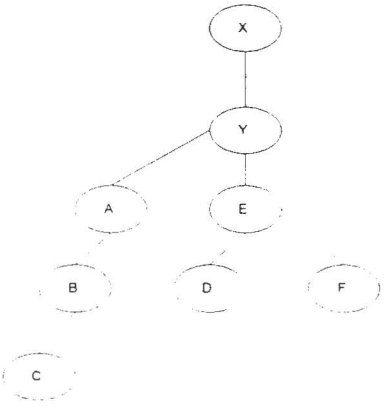


Figure 2-7: Example of case in case-based database after using remember to add policy

2.3 Why to remember

Leake and Wilson consider the performance of case base more than its competence. The assumption of this technique is "The case is good with high adaptation cost". They developed a relative performance (RP) metric to help measure the adaptation system. This technique therefore keeps the case showing high value in relative performance metric, otherwise this technique deletes cases when the value of relative performance is low. [6]

The RP technique is a tradeoff between competence and size of CBR. When RP is used for deleting cases, the range of problems is decreased.

In case of non- identical problems, the RP metric favors useful cases in high traffic areas. The RP include size of traffic area for decision-making; if it is high traffic area and adaptation cost is high, the result is to keep the case.

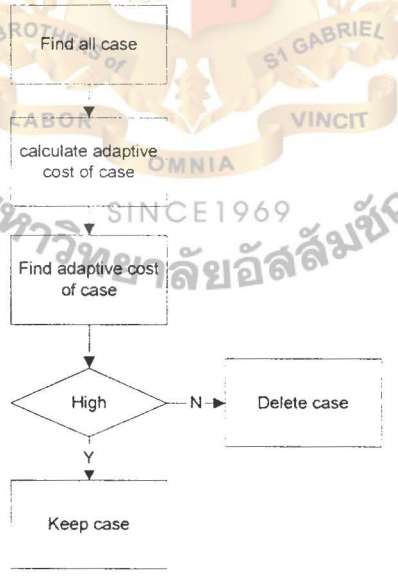


Figure 2-8: Diagram of why to remember policy process

2.4 Deleting cases that have not been used for a long time

Deleting unnecessary case in case-based library, some research focuses on deletion of information that is unused for a long time. Portinale and Torasso and Tavano applied deletion technique of deleting cases, which are not used longer than defined time. [10]

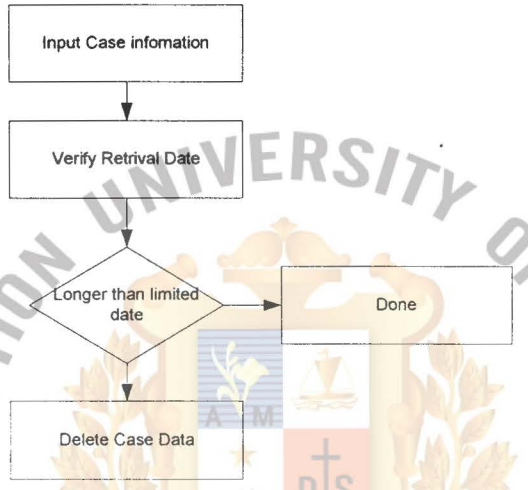


Figure 2-9: Diagram of deleting case unused for a long time.

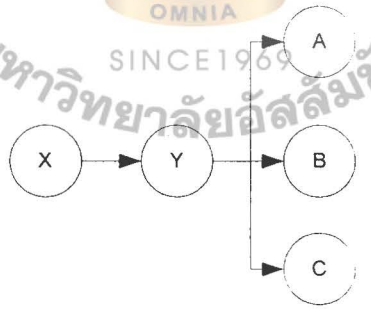


Figure 2-10: Example of Case after deletion

In figure 2-10, system defines the time below year 1995. If case "A" unused longer than defined time, it will be removed.

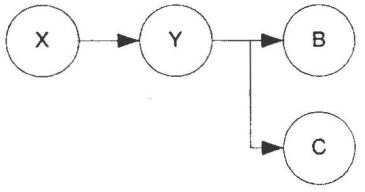


Figure 2-11: Example of Case before deletion

Not only case-based maintenance research considers improving content problem but also other researches consider this problem too. Next section describes other techniques that tried to improve this problem.

2.5 K-Nearest Neighbor techniques

The k-nearest neighbor technique is a classification technique. This technique is used for searching the records that is nearest to the problem. Many case-based reasoning use this technique to retrieve a case data in case-based database. The meaning of "K" is quantity of data that is closest to the problem , the quantity of data depends on how much the person wants. Some people want data that is equal to the problem. Example is searching path domain, searcher may want the information that is equal to target problem.

The advantages of k-nearest neighbor technique are defined as follows [13]

1. This technique has the ability to copy very complex target functions by a collection of less complex approximations.
2. This technique is easy to program.
3. No optimization or training is required for this algorithm
4. This technique is tough for noisy training data. Provided with sufficiently large training dataset, it has been shown to be quite effective. The concept here is that by taking the weighted average of *k* neighbors nearest to the query point, it can smooth out the impact of isolated training examples.
5. Information can be taken incrementally at run-time to the *k*-nearest neighbor.
6. Information is never lost because the training examples are stored explicitly.

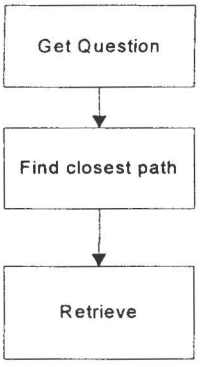


Figure 2-12: Diagram of KNN process

In table 2-2, the system retrieves the records, which is closest with the problem in table 2-1.

From	To
6	13

Table 2-1: The new problem

NO.	Path
	16->8->81->13
	26->13
	36->7>8>81->15->13

Table 2-2: The retrieved cases by k-nearest neighbor technique

Many researchers focus on k-nearest neighbor technique. Balaraman. Chakraborti and Vattaml [13] researched to improve this technique. They focus on directory assistance domain. In this domain, the information in case-based library is

unstructured text. They discovered that the error of KNN in this particular domain concerns strings of factors, which cover:

1. Incomplete input
    - Only first name or part of last name etc.
2. Incorrect or differently written input
    - Misspellings : Prefix, suffix, infix errors
    - Different terminology : Street for Road etc.
    - Ordinal forms: First or I'st etc.
    - Acronyms and expansions: St. for Street etc.
    - Different order of input: "Ram Mohan" or "Mohan Ram"
    - Different punctuation
    - Different levels of specification: For example, Occupation might be stated generally as Professional or at the other extreme as Hardware Design Analyst.

Moreover, they also tried to improve strings of error in the previous section, thus, they proposed a technique that could help tackle the problem. The process of this technique consists of

1. Stemming: reduce a derived or inflected word
2. Synonym re-writing and auto correction:
3. Stripping of noise words: reduce the words that don't add any meaning to the semantic content of a sentence.
4. Correcting the grammar used.

For searching process, they approach the "Graded Retreat" in order to search the information in case-based library. This technique tried to reduce the retrieval time. This technique separates to search the level and increasingly search level by level.

2.6 Shortest Path

The traveling salesman problem (TSP) is the method that tries to reduce cost of travel between each of them [9]. Dijkstra algorithm is one of TSP method that finds the shortest path from source node to terminate node. [7]

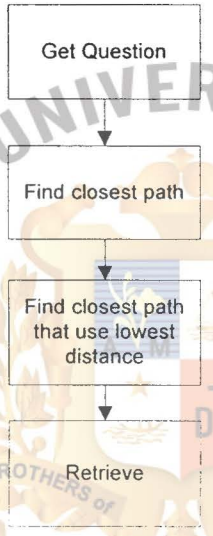


Figure 2-13: Diagram of dijkstra algorithm process

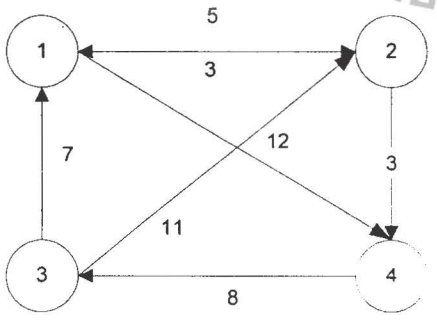
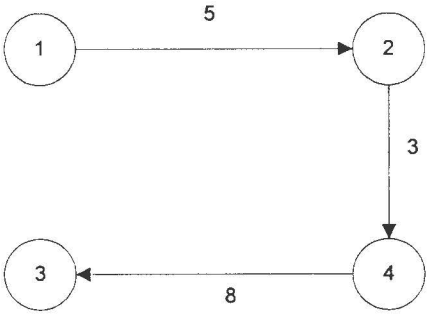
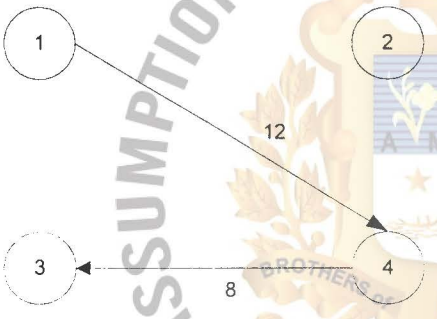


Figure 2-14: Example of Map before use Dijkstra's algorithm

In figure 2-14, the shortest path from 1 to 3 is 1->2->4->3 because it uses the distance lower than other path (the distance of path 1->2->4->3 = 16 and 1->4->3 = 20).



(a)



(b)

Figure 2-15: (a) The shortest path of map when compared with (b)

This paper describes the other research finding which is the shortest path.

Optimal Salesman Problem (OSP) [14]

The meaning of optimal salesman problem is described below

"The problem of finding a closed path passing through all the vertices with minimum weight is call the optimal salesman problem (OSP)"

This technique uses the matrix in order to remove the path that has high weight. Figure 2-16 shows the map before solving problem. First, this technique converts each path with a value in the matrix. In the case of filling up with a value in the matrix, if the map does not contain a path, this technique then fills a character into the matrix. After translation of each path into the matrix has finished, this technique obtains the minimum weight for each row in the matrix. Finally this technique translates the chosen value in matrix to the path.

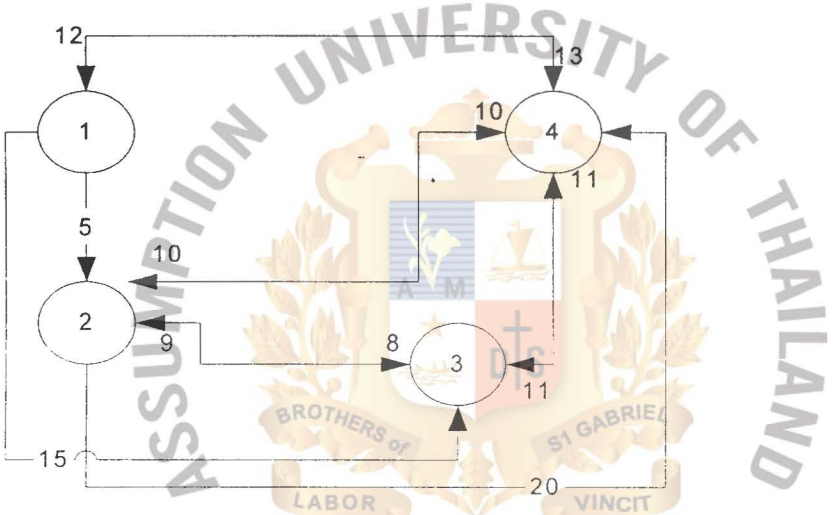


Figure 2-16: The map domain.

When translating to the matrix bellow:

M	5	15	13
M	M	8	20
M	9	M	11
12	M	11	M

When finished finding the minimum of each row and subtract each row, the result is shown below

M	0	10	8
M	M	0	12
M	0	M	2
1	M	0	M

For the result, use the Hungarian method to find the final result and the final result is shown below

M	0	10	6
M	M	0	10
M	0	M	0
0	M	0	M

When the use the Hungarian method is finished, the result are 1-2, 2-3, 3-4 and 4-1. Hence the minimum path shown in figure X.

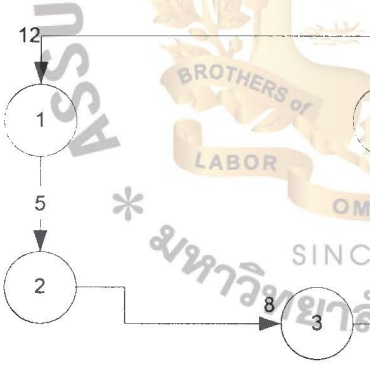


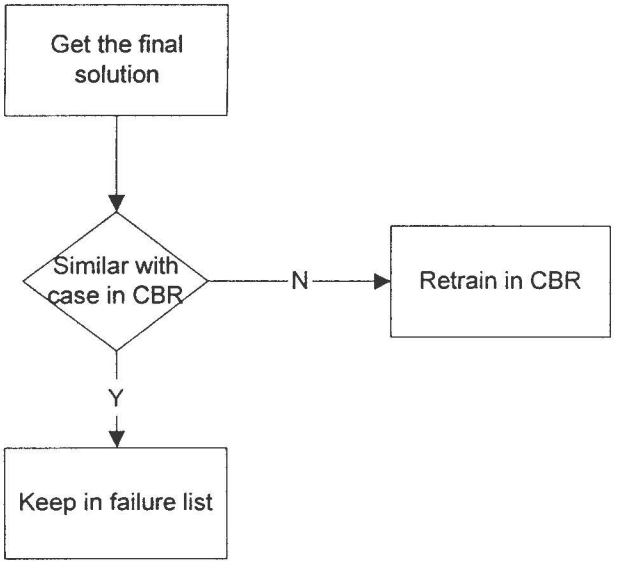
Figure 2-17: the map after solving OSP.

2.7 The ROBBIE System

The ROBBIE system is hybrid CBR and introspective reasoning for road map domain. ROBBIE uses hindsight to detect when a successful solution is based on the wrong case. After a successful plan has been generated, it uses a standard to determine which case retrieved most similar plan. If new solution is similar to another case, a retrieval failure is indicated. [11]

ROBBIE eliminates the wasted case; when CBR has completed resolving the problem, introspective compares the result with previous case in database. If the result is similar with the previous case, the system indicates that the result has failed then it would be kept in the failure list. In the retrieving process, if system cannot find case with a similar target problem, the system finds case in failure list.

(a)



(b)

Figure 2-18: (a) The retrieval process (b) The retaining process

This section describes many researches that tried to improve the content problem. The researches consist of

1. CBM technique is to decrease the quantity of case in case-based database.
2. Retrieval technique is to retrieve the previous case which is similar with the target problem in case-based database.

The next section describes the approaches for each technique in solving content problem for problem domain. This describes what problem remains after using each technique to solve problem.

3. Problem Domain

Bangkok is the metropolis of Thailand. This is the center of businesses and education. Therefore this city has a lot of transportation systems, which are very complex. CBR approach is to help a person make decision on how to go to their destination.

The problem of a domain is that the database contains a lot of data. It affects the users, because they can't choose their target path efficiently.

In figure 3-1, person could not identify what is the suitable path.

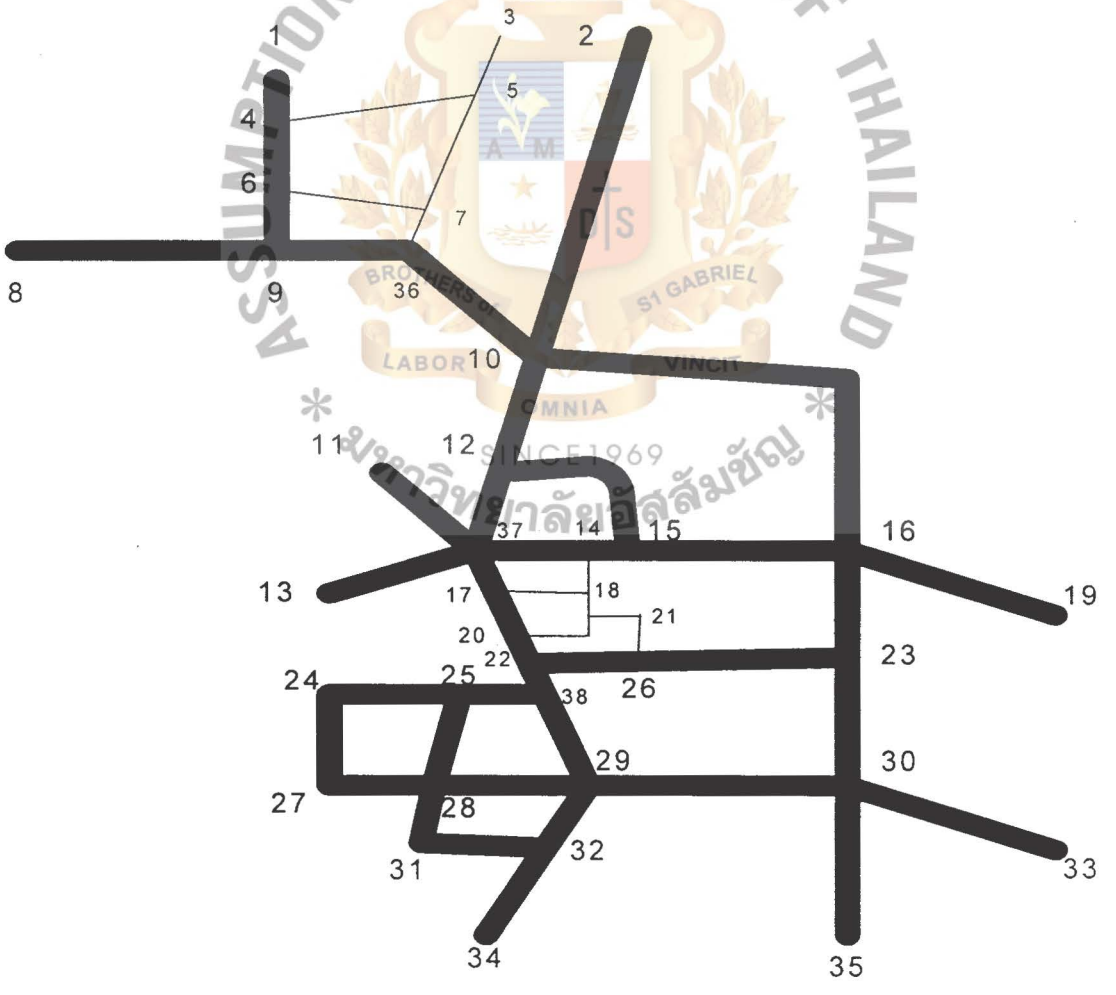


Figure 3-1: Road map of problem domain.

ID	Name
1	Bangkok-Patumtanee
2	Teevanont
3	Taspatum
4	Sukapiban1
5	Sukapiban 2
6	Ponsee
7	Ponsee
8	Bangbuatong - bangkuawat
9	Bangkok-Patumtanee
10	Taspatum
11	Jangwattana
12	Muantongtanee
13	Poomwate
14	Tonsoon
15	Muantongtanee
16	Leabkongpapa
17	Shuchartanuson
18	Shuchartanuson

Table 3-1: The number that replace pace name.

Figure 3-1. illustrates road paths in Bangkok, which this paper represents in number. instead of name of places. The distance, volume of traffic jam, status of road and size of road in problem domain are only assumptions. Case-based database already have any path in Bangkok. Reuse and revise cycle process by humans.

For footprint strategy, pivotal case in problem domain is path 8 because this path remains in case-based before other paths. This strategy keeps only pivotal case (path 8). case-based loose all competence because it can't retrieve any path in solving the target problem.

Remember to add policy and why to remember techniques, deletes some paths with a number of neighborhood and paths with low costs of adaptation e.g. path 3, 5 and 7. When case base retrieves path to find paths 3, 5 and 7, case-based may not find

this path because it was deleted by those techniques. Although why to remember technique (RP metric) can keep the high-traffic area of cases its not however enough for searching suitable path because condition of suitable path is not only the high traffic area but also there are other conditions (Status, Drivers, Time and etc.).

The stored problem for k-nearest neighbor technique is adaptation problem, which this paper describes below.

When drivers want to find path from 9 to 23, case base will retrieve any path that is nearest problem 9->23 e.g. 9-36-10-16-23, 9-36-10-12-37-17-20-22-23, 9-36-10-12-37-14-15-16-23, 9-36-10-12-15-16-23 etc. This technique only retrieves case closest to the target problem. The remaining problem is that users cannot adapt case efficiently because they do not have information to advise which one is the suitable path. If user selects path 9-36-10-12-37-14-15-16-23, these paths may be traffic jammed or may have longer distance than other paths.

For Dijkstra algorithm (shortest path), when drivers want to find path from 9 to 29. The algorithm only finds shortest path and the result of the said technique is path 9-36-10-12-37-20-22-38-29. This path may not be suitable in some situations. For traffic jam situation, this path will not be suitable because it is a main path with a lot of junctions therefore, it will use a lot of time to use this path.

ROBBIE system cannot be used for problem domain because the problem domain already contains each path in case-based database. This technique is good when case-based database does not contain each path and this research describes some weak points of this technique below:

"The reactive planner handles facets of the simulated world that cannot be predicted by the CBR system, such as traffic lights and blocked street."

This section represents problems, which techniques in the literature review cannot solve. This paper proposes new techniques, which could improve remained problem.

4. Proposed Technique

4.1 Retrieval Technique

For retrieval process, this paper uses "Find Target Path Technique" in order to retrieve data in database. This technique uses recursive solution in order to retrieve data from database that this technique uses for each retrieval technique (KNN, Shortest path).

This section describes the method of Find Target Path Technique. First, this technique finds start point in the table. When the start point is found, this technique then uses the stop point data in order to find start point in other records, then, the process is being repeated. When it finds the target point, this technique stores data in temp table. However, when it cannot find any data anymore, this technique stops. In cases this technique can't find the target, this technique displays "System can't find data" message. In figure 4-1 shows flow chart that describes method of find target path technique.

The database structure stores path information and is composed of 3 tables the path information table, the traffic-jam information and the temp table.

Table 4-1, 4-2 and 4-3 show the database to store path information. Table 4-1 describes the database to store information in each path that consists of start point, stop point, distance, type of road, road status and size of road. Table 4-2 describes the table that stores traffic jam area information. This table illustrates the path and traffic jam time. Table 4-3 describes the table that stores retrieved information.

CaseNo	Start Point	Stop Point	Distance	Type	Status	Size
51	1	4	0.6	N	G	L
10	2	10	4.7	N	G	L
7	3	5	0.8	N	B	S

Table 4-1: The table that keeps information about map.

CaseNo	TimeStart	TimeStop
10	7:30	8:30
10	15:00	16:00
12	7:30	8:30
12	15:00	16:00
14	7:30	8:30
14	15:00	16:00

Table 4-2: The table that keeps information about traffic jam area.

Way	Jam	Size	Type	Status	Distance	Mark	Time
9->10->11->15->23->32->33->30->	0	25.15	0	25.15	25.15	5	0.4193
9->10->11->12->17->15->23->32->33->30->	0	26.95	0	26.95	26.95	4	0.4493
9->10->20->22->23->32->33->30->	0	27.5	0	27.5	27.5	0	0.4585
9->10->12->17->15->23->32->33->30->	0	28.05	0	28.05	28.05	0	0.4676
9->10->11->12->13->17->15->23->32->33->30->	0	28.39	0	28.39	28.39	3	0.4734

Table 4-3: The temp table

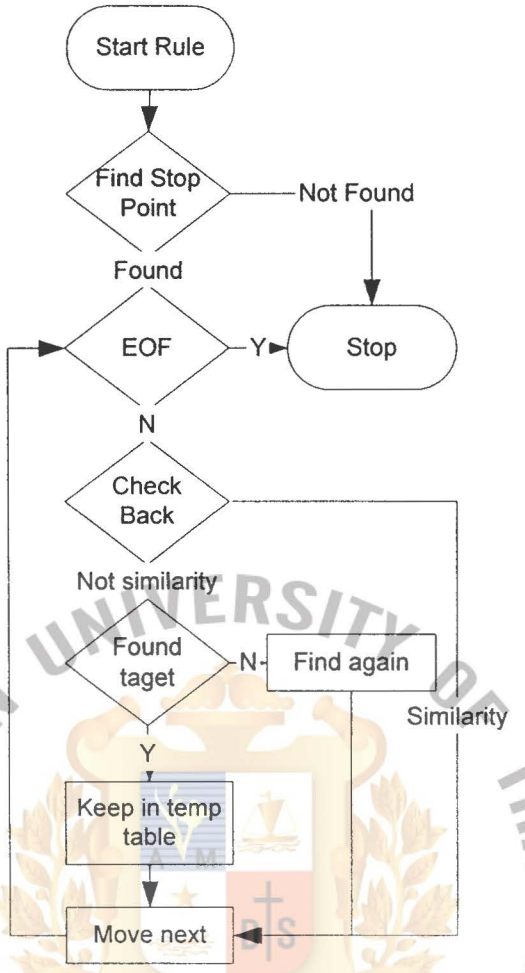


Figure 4-1: Find target path technique

4.2 Overview of Proposed Technique

The suitable path consists of many meanings, and the meanings depend primarily to the drivers. Example are "The path that uses shortest distance", "The path aren't traffic jam" and "The path with good status"

This paper sets the meaning of suitable path as "The path that uses shortest time from start point to target destination".

This paper proposes technique that will help improve the domain problem. The proposed technique applies simple rule base system in order to analyze retrieved

case and rank it. This paper expects that the proposed technique can solve the remaining problem.

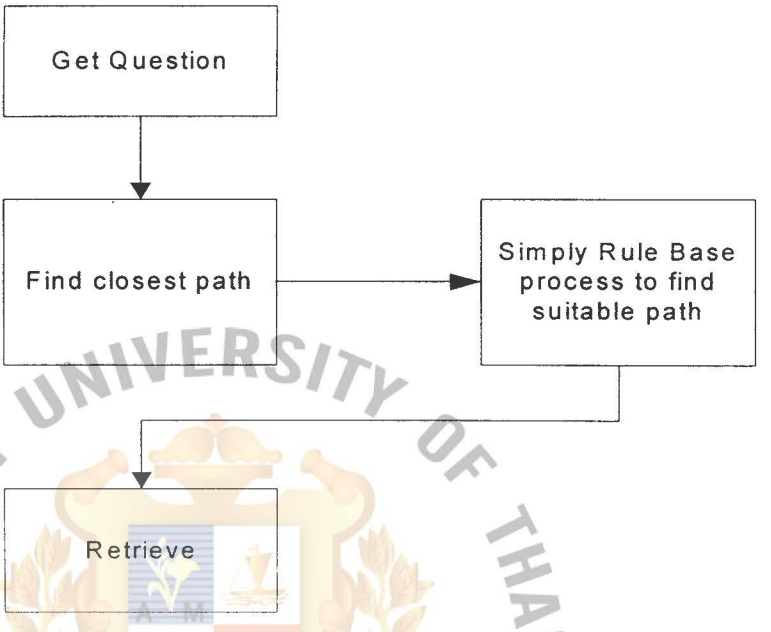


Figure 4-2: Diagram of proposed technique

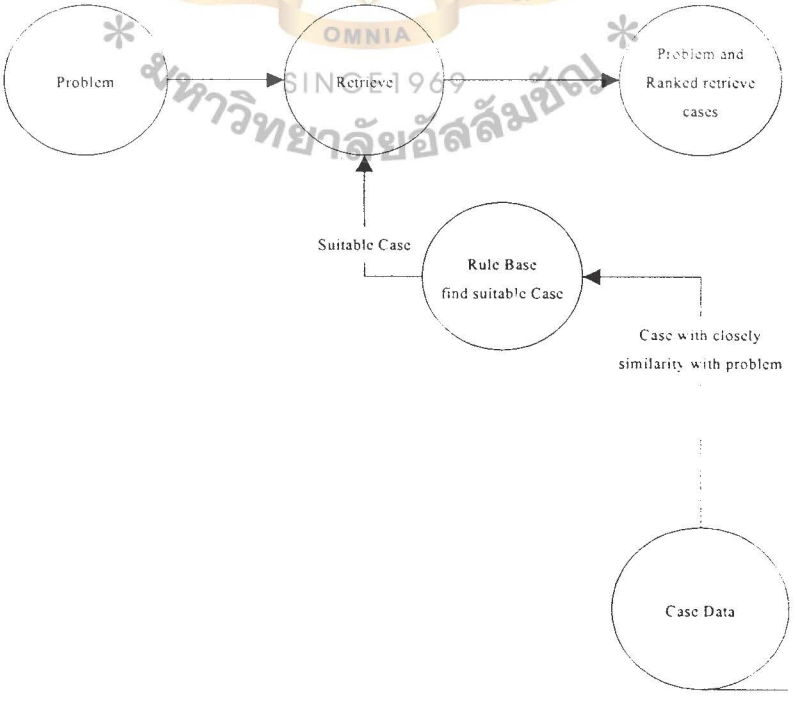


Figure 4-3: Architecture of Retrieve Technique

Figure 4-3, represents the architecture of proposed technique. After the system has finished retrieving the data (Keep Distance Technique), the rule-based system evaluates and ranks the retrieved cases in order to find the suitable path. The system was implemented by visual basic 6.

There are many factors affecting a driver, but not all factors affect every driver. Example is, a new driver cannot drive well in situation where that size of road is small but old driver or an experienced driver can drive well in the same situation. However, these factors may also affect various conditions of drivers, like some factors affect drivers' mind in the state of driving, an example is a driver driving in a mannered path, this affects the driving of any driver positively because that is the path of good status.

This paper considers the factors that affect the traffic situations and driving of a driver, such as:

1. Situation of traffic
    a. Time: the period that traffic jam
2. Condition of road
    a. Size of road: large road or small road
    b. Normal road or Express way
    c. Status of road: good or bad
3. Distance of path
    a. Short distance or Long distance
4. Driver
    a. New driver or Experience driver (Old driver)

This paper describes details of factors below.

Factor	Value	Description
Size of Road	L	Large size (more than 1 way per road)
	S	Small size (1 way per road)
Status of road	B	Bad status (under construction or holes)
	G	Good status (normal road)
Type of road	N	Normal way
	E	Express way

Table 4-4 : The factor that affect the suitable path

This paper assumes that the new drivers are not suitable to drive in small sized roads and roads of bad conditions. This is not however true for experienced drivers. The expressway factor is the suitable factor for all drivers. The traffic jam does not suit all drivers. For traffic jam factor, system has the table that stores information about time of traffic jam.

The proposed technique consists of 2 methods, the keeping distance method and the rule ranking method. Keeping distance method is using condition to check retrieved data, method keeps the distance value into the temp table if the condition is right. Rule ranking method is using rule-base to analyze and rank data in temp table, the system shows the path with highest mark after rule-base processing is finished.

This paper describes detail of proposed technique below:

1. Retrieve the path.
2. Rule-based check variable of path
    a. If variable of path matches with the condition of rule-based, rule-based keep the distance of path.
3. Store the information of this path to temp data.
4. Process 1 is repeated again until no data is found in the database.
5. When the previous process is finished, the other rule-based compare each path in temp data and rank it

Figure 4-4: Method of Proposed technique

4.3. Keep Distance Technique

The keep distance technique is a sub process to target path technique. This technique starts processing after the find target path technique has finished finding the target record.

When the system has finished using the stop point field in order to find the start point in other records, the keep distance technique uses rule base to choose each factor and this technique keeps the distance of path in temp table when the condition is right. If size of road is large, system will keep quantities distance into size variable. If status of road is good, system will keep quantities distance into status variable. If type of road is express way, system will keep quantities of distance into type variable. This technique finds starting time with traffic-jam information, when system finds traffic-jam time in traffic-jam information, it keeps quantities distance into traffic-jam variable. When this technique is finished keeping the distance, the find target

technique path process again. This technique has finished its process when the find target technique stops to find any target. Figure 4-5 describes the relationship between the keep distance technique and the keep distance technique. Figure 4-6, describes the method of keep distance technique.

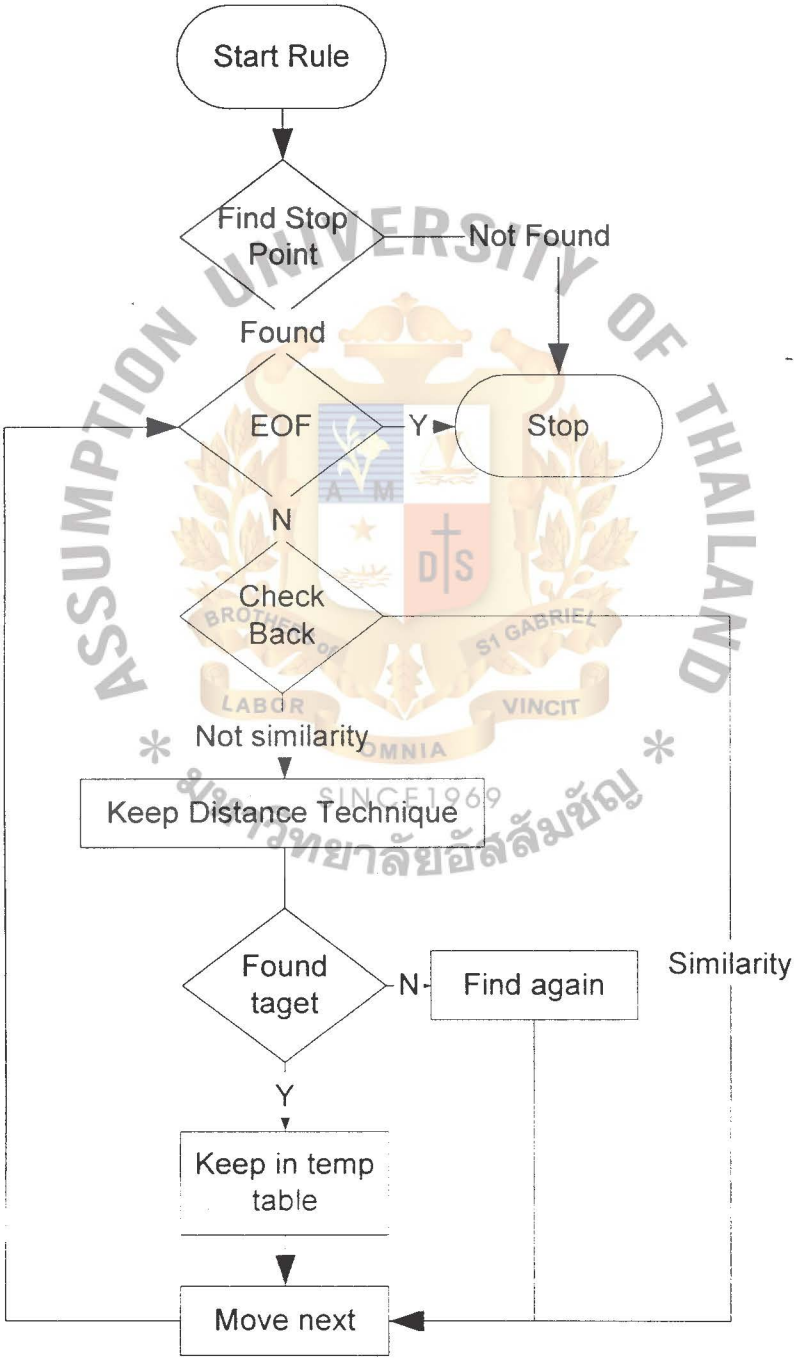


Figure 4-5: Keep Distance Technique

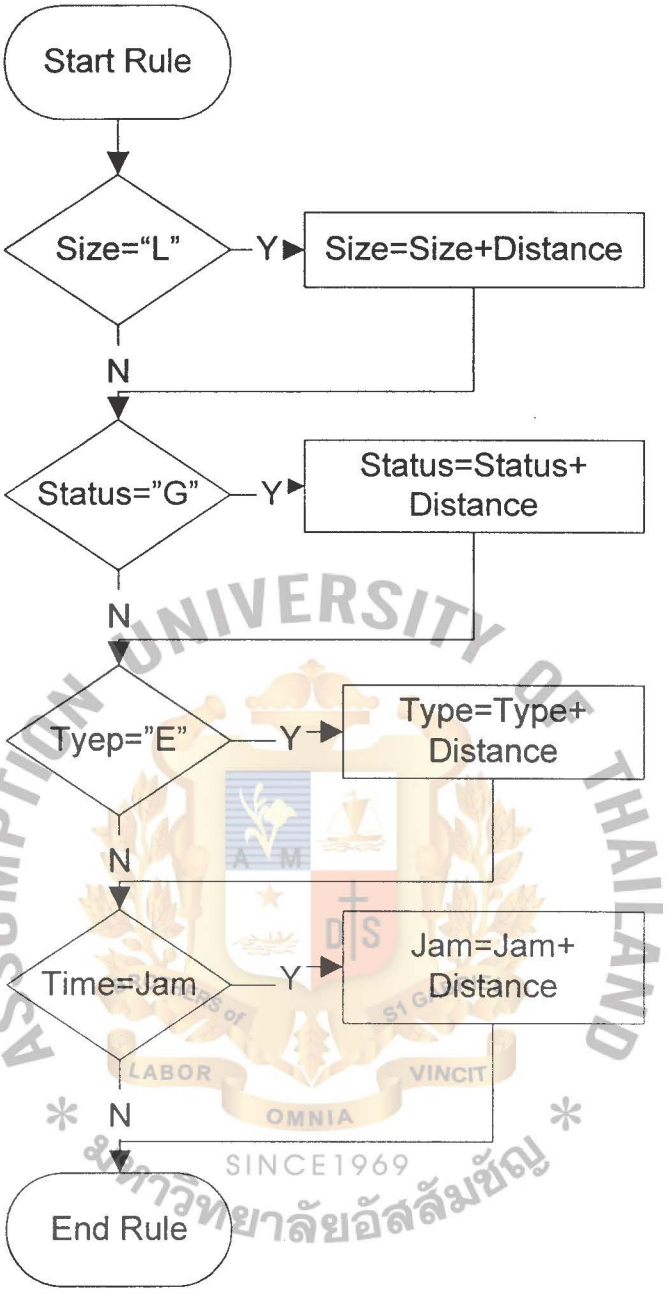


Figure 4-6: Rule process to assign value before keeping in tamp table

4.4. Comparing Technique

After keeping the path information in the temp table (figure 4-6), the system compares each path and displays the highest value of path in temp data. This technique compares each result 1 by 1. This technique starts comparing from a first record and stop comparing when it doesn't find a data. Figure 4-7 describes the overview of comparing technique. Figure 4-8, represents a rule process that compares value to each path. This paper assumes the variable A and B that the variables are a condition of each path (e.g. the traffic-jam distance, the large-size distance) in the temp data. Example: comparing the distance of path "A" and the distance of path "B". When the condition is correct, system gives a mark to the path variable (valA or valB). Example: if the distance of path "A" shorter than path "B", system gives a mark to variable "valA".

The rule-based comparing in figure 4-8 creates from assumption the driving speed of each driver in table 5-1.

When the distances of each path are not the same, the proposed technique could not compare correctly if it only compares the distance. Therefore this paper uses ratio formula between the status and the distance to compare two paths. Example: status of path A is 350 Metre and status of path B is 100 Metre and distance of path A is 600 Metre and distance of path B is 100 Metre, if proposed technique compares only the distance, comparison of paths will be incorrect. The status of path A is higher than path B. When it uses ratio formula to compare paths, the result of comparison is more correct than comparing only the distance (the status of path B higher than path A). An example of ratio formula that can be used to compare 2 paths is, when the system compares status distance between 2 paths, system uses formula below

$$\frac{status\_distance\_A}{all\_distance\_A} \text{ Compare with } \frac{status\_distance\_B}{all\_distance\_B}$$

The comparing method compares 2 paths and assigns mark in the right condition. In comparing conditions, the system compares the ratio of distance between two paths. Example: Comparing traffic-jam condition, if the ratio of path A higher than path B, system gives mark to variable of path A. when the condition is wrong, system doesn't give the mark to any variable, then the system process the next condition.

If the mark of one path is higher than the mark of other path, the system will swap the paths. If the marks of both paths are the same, the system keeps both paths in the list. When the rule process is finished, the system gets the next record in the temp data in order to compare again, till the desired data is found in the temp data.

When the overall process is finished, system finds the highest mark of path in temp table then retrieve it, if there are many paths with highest mark, system randomize the paths with the highest mark in descending order to retrieve only one result. For random method, this paper uses this method for any retrieval technique when the result of each technique contains same value. Example: in case the result of shortest distance path technique contains many paths that used shortest distance, this paper applies random method in order to show only one result.

Figure 4-8, each condition does not have equal mark because comparing technique separates each condition according to driver affectation. The condition of high importance affects the driver most and this condition occurs frequently. If condition is of high importance, the comparing technique gives high mark. Example: Status of road factor affects for new drivers but it doesn't happen at all times. It also doesn't affect the experienced drivers; therefore, this factor has a low mark. However

with traffic-jam factor, the traffic-jam factor has great effect for any driver and usually occurs frequently, therefore this factor has high importance (high mark).

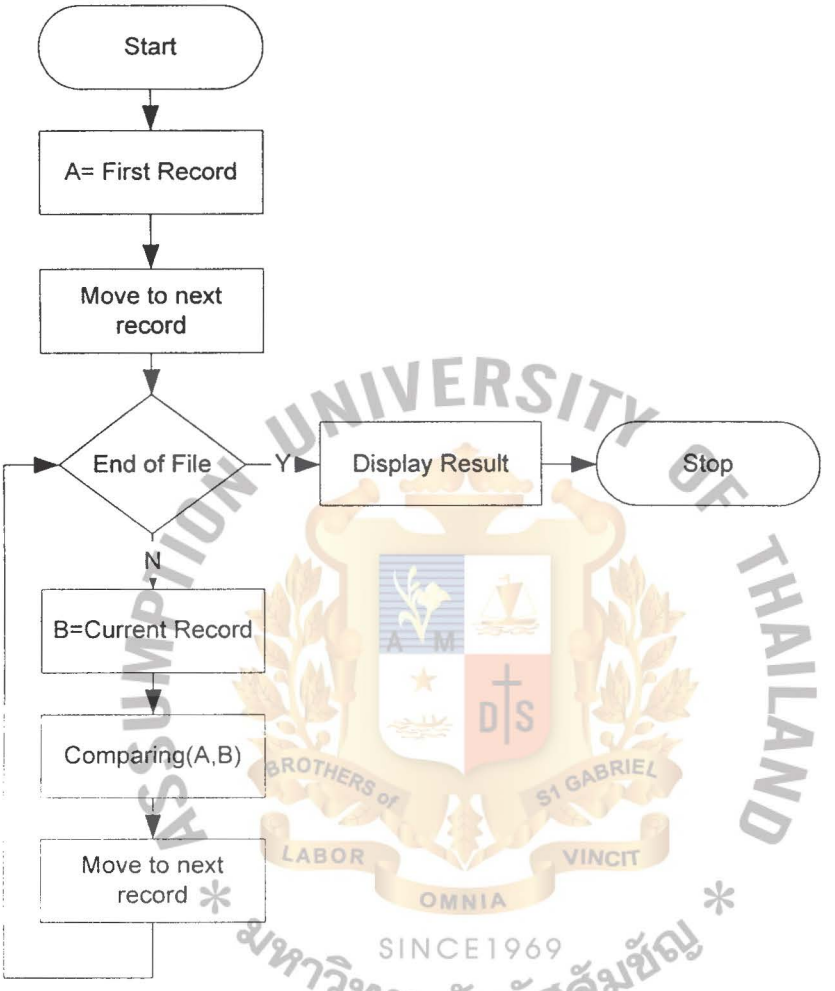


Figure 4-7: The overview of comparing technique

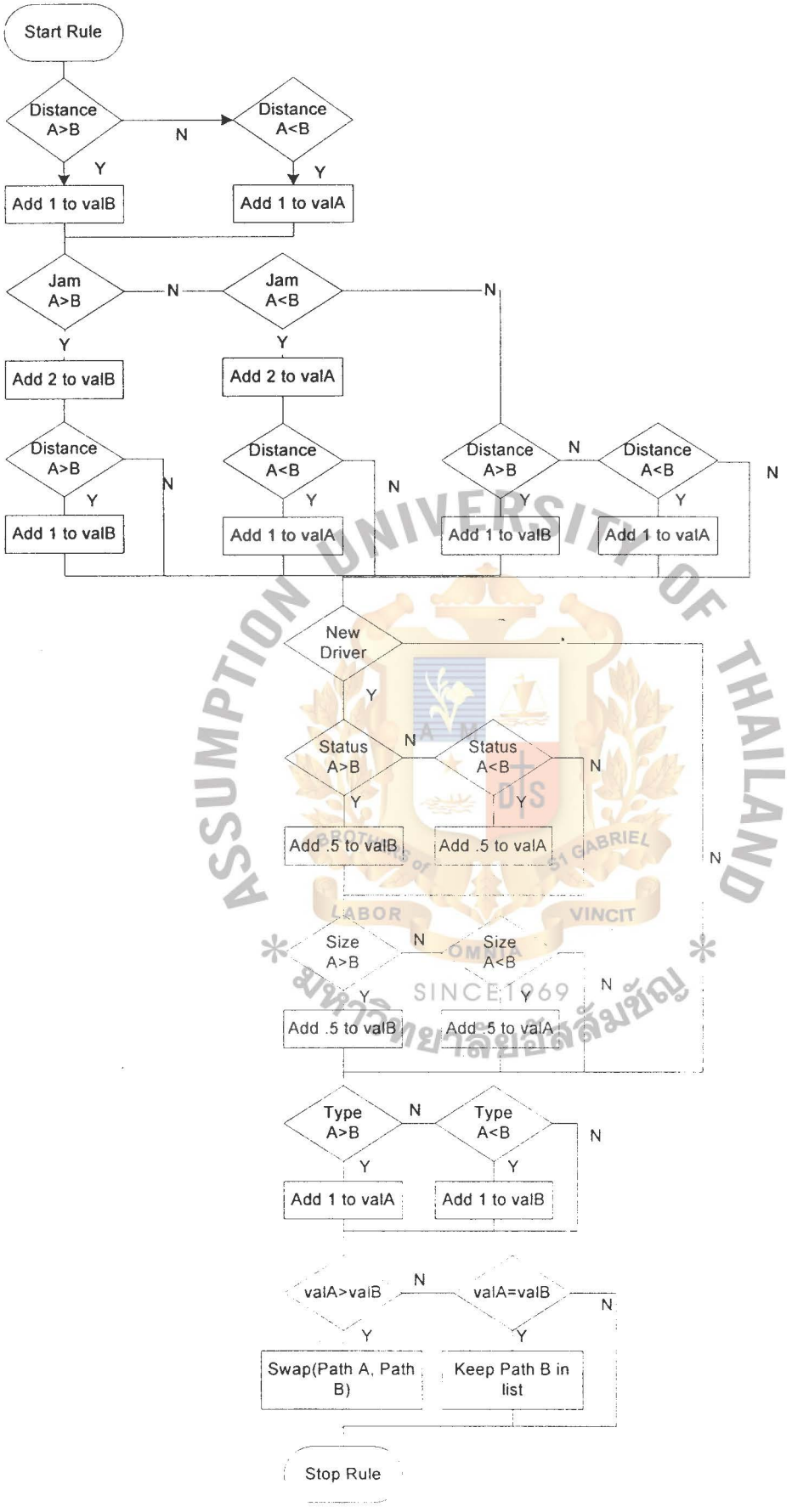
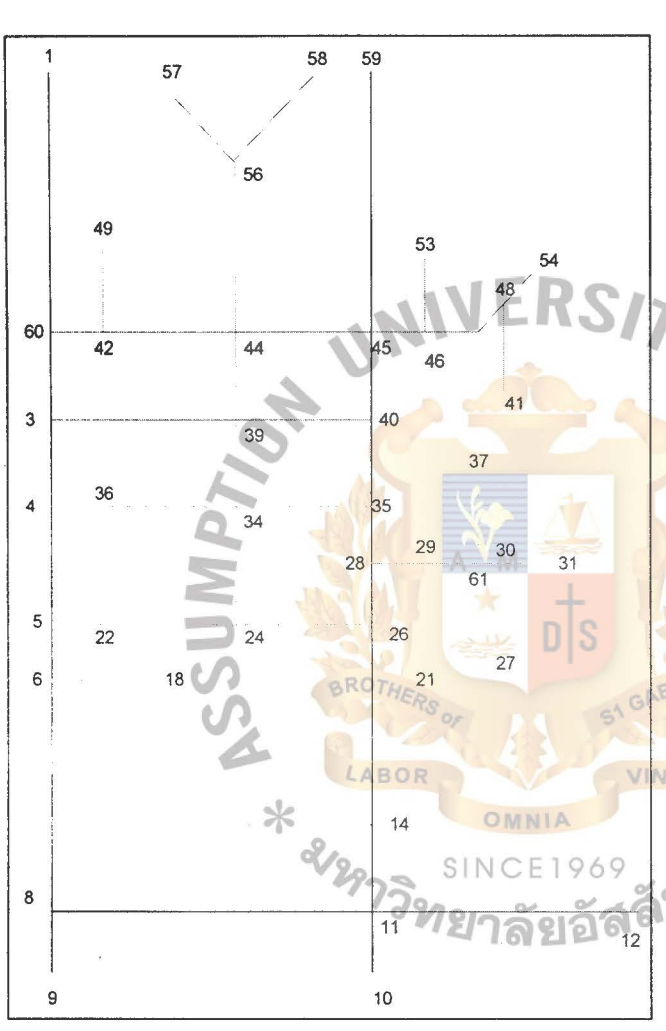


Figure 4-8: Rule process find suitable path in temp table

5.Evaluation

5.1 Case Study

For each map in evaluation section, this paper assumes to generate the path and condition e.g. distance, status, and size.



Small size of road

Large size of road

Bad status of road

Express Way

Figure 5-1: Map 1

The detail of map 1 consists of

1. Traffic jam areas are from 1to 9, from 8 to 11 and from 59 to 10.

   Traffic jam time are 8:00-9:00 and 15:00-18:00.

2. Any path is good status.

3. This map contains the roads of small size.

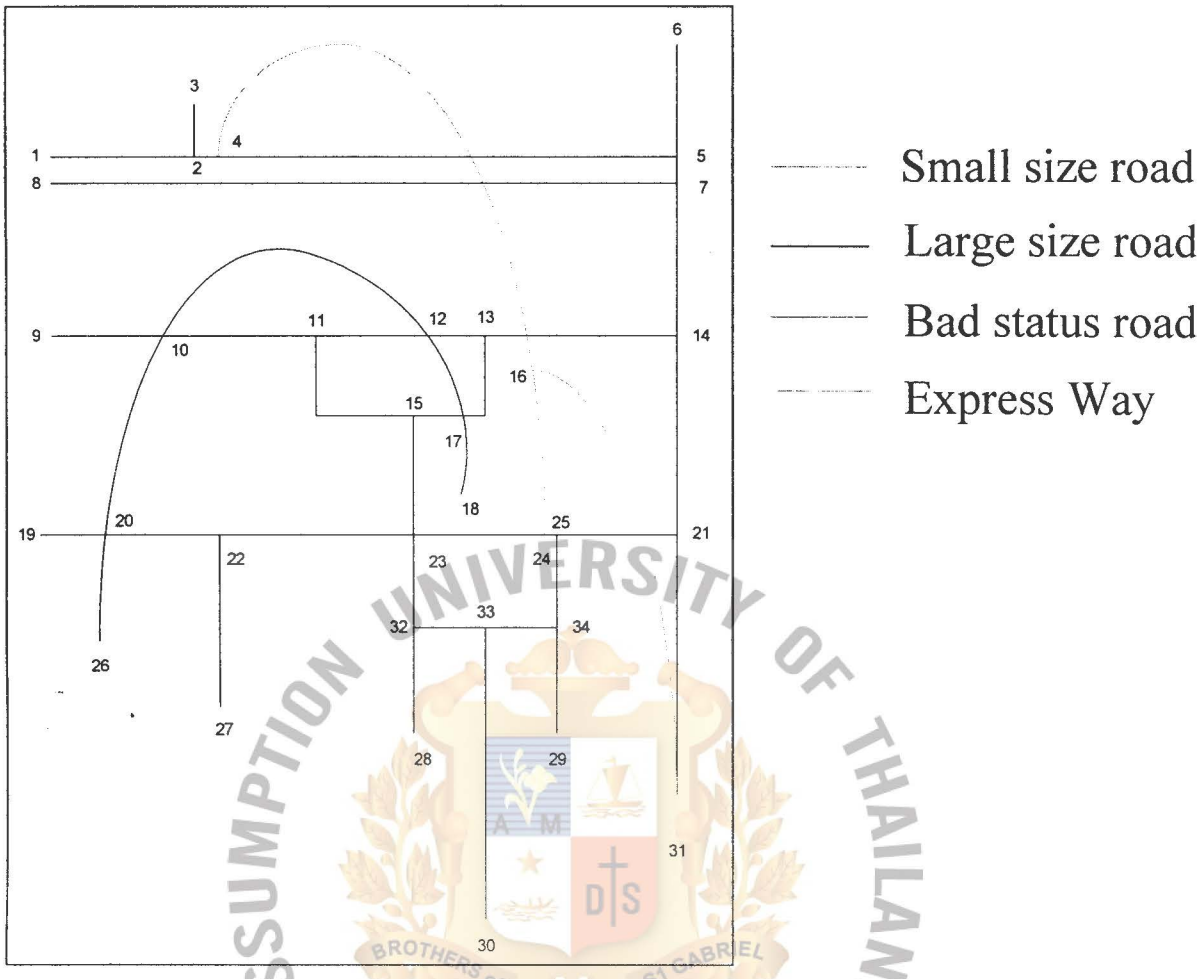


Figure 5-2: Map 2

The detail of map 2 consists of

1. No traffic jams area.
2. Any path is good status and large size.

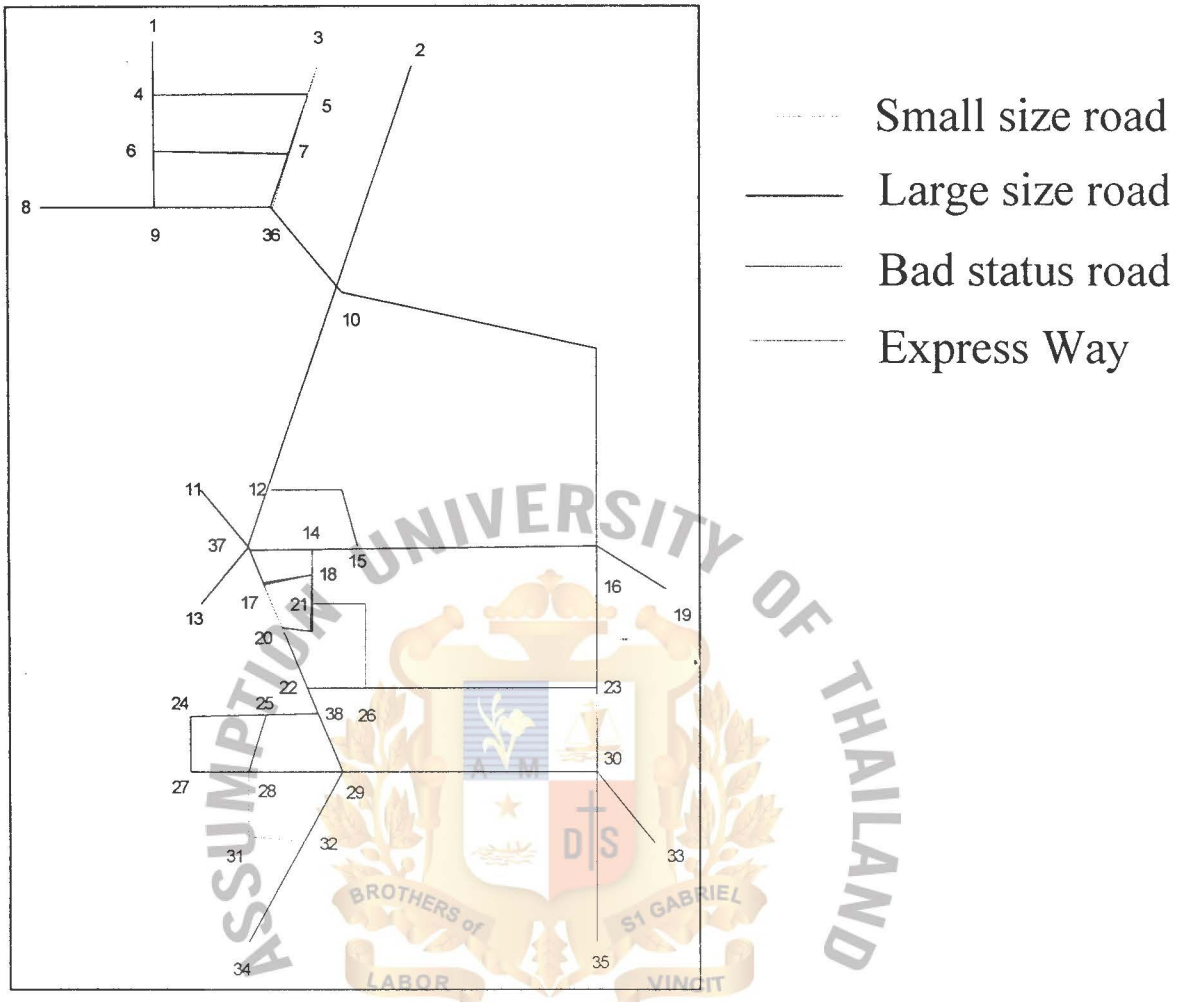


Figure 5-3: Map 3

The detail of map 3 consists of

1. Traffic jam areas are from 2 to 37, from 37 to 29 and from 29 to 34.

   Traffic jam time are 7:30-8:30 and 15:00-17:00.

2. This map consists of bad status roads.

3. This map consists of small size roads.

5.2 Evaluate Method

This paper evaluates the proposed technique by comparing the results between the proposed technique and other techniques (Shortest distance path, KNN). The time consumed by each driver is the factor used in considering whether the proposed technique is better than other techniques (Shortest distance path and KNN) or not. In KNN case, this paper uses the random method to choose one of the results that will be compared with other techniques. The formula to find traveling time is shown below

$$T = \frac{S}{V}$$

T = Traveling time

S = Distance from origin to destination

V = Velocity

This paper assumes the speed of cars in order to perform the evaluation, the metric of speed factor use Metre per seconds. The speed factor depends on each factor. Table 5-1 represents relationship between each factor and speed factor.

In table 5-1, this paper uses the ratio Metre per Kilometer, which is 3600 Metre per 1Kilometer.

In traffic jam situation, any driver can't drive fast, this paper assumes that new driver drives at speed of 5 kilometer per hour and experienced driver drives at speed of 6 kilometer per hour, this is because experienced driver knows how to drive efficiently in this situation while new driver doesn't have an experience how to efficiently drive in the same situation.

For speed of cars, this paper assumes the variation of speed found below

	New Drivers	Experience Drivers (Old Drivers)
Time: Traffic jam	5	6
Time: Traffic normal	60	60
Type of Road: Express way	80	80
Type of Road: Normal way	60	60
Status of Road: Good	60	60
Status of Road: Bad	5	6
Size of Road: Large	60	60
Size of Road: Small	30	40

Table 5-1: The value of speed for each situation

In case the path involves many situations (traffic jam, size large, status bad), any retrieval technique use the lowest speed in order to calculate the time used.

5.3 The Result of Evaluation

This paper assumes the input data in order to experiment below

	From	To	Time	Driver
Map 1	60	14	07 :50	N
Map 1	60	14	08 :40	N
Map 1	60	14	08 :40	O
Map 2	9	30	12 :00	N
Map 2	9	30	12 :00	O
Map 3	8	29	08 :00	O
Map 3	8	29	08 :00	N

Table 5-2: Input data.

In table 5-2, for driver column, variable "N" is new driver and variable "O" is experienced driver or old driver. The experiment starts at row 1 in table 5-4. Example is experiments map 1, start point from 60, target path is point 14, start time is 7:50 and driver is a new driver.

For comparing the relation of processing time and intersection, this paper uses computer AMD 950 MHZ, memory 256 MB in order to find the relation of them. In figure 5-4 shows the relation between processing time and intersection. When system route to pass each intersection, the processing time will be increasing.

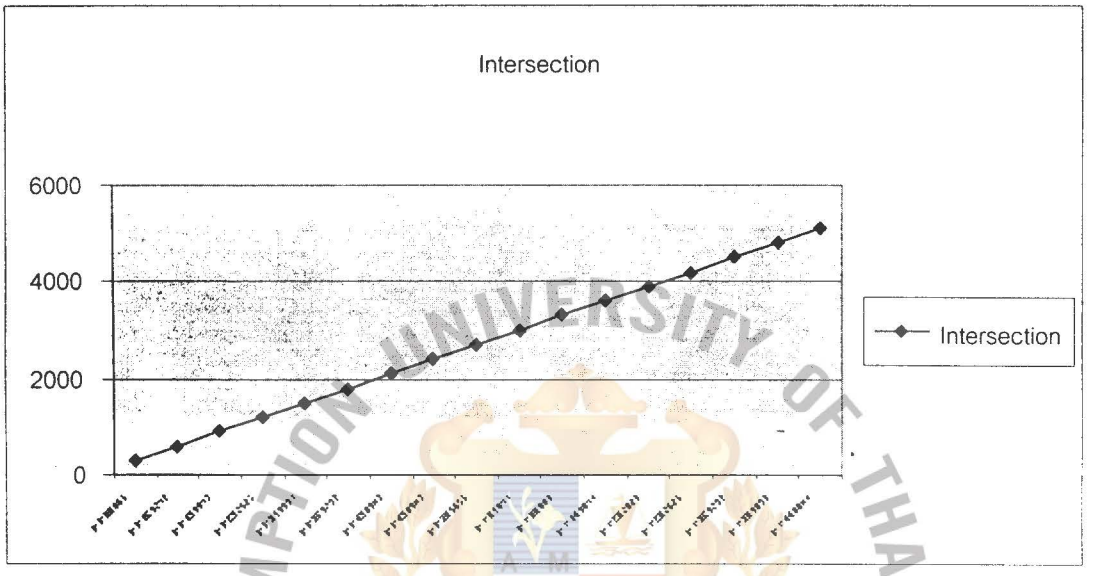


Figure 5-4: Comparing between intersections and processing time.

	Used Time						
	Map1(1)	Map1(2)	Map1(3)	Map2(1)	Map2(2)	Map3(1)	Map3(2)
KNN	0.8786	1.6801	3.4385	0.5285	1.0252	1.0157	1.1248
Short	0.4633	1.5867	1.3134	0.4193	0.4193	0.9416	1.1083
Proposed	0.4318	1.2268	0.8892	0.4193	0.4193	0.4734	0.5867

Table 5-3: Traveling time of each technique.

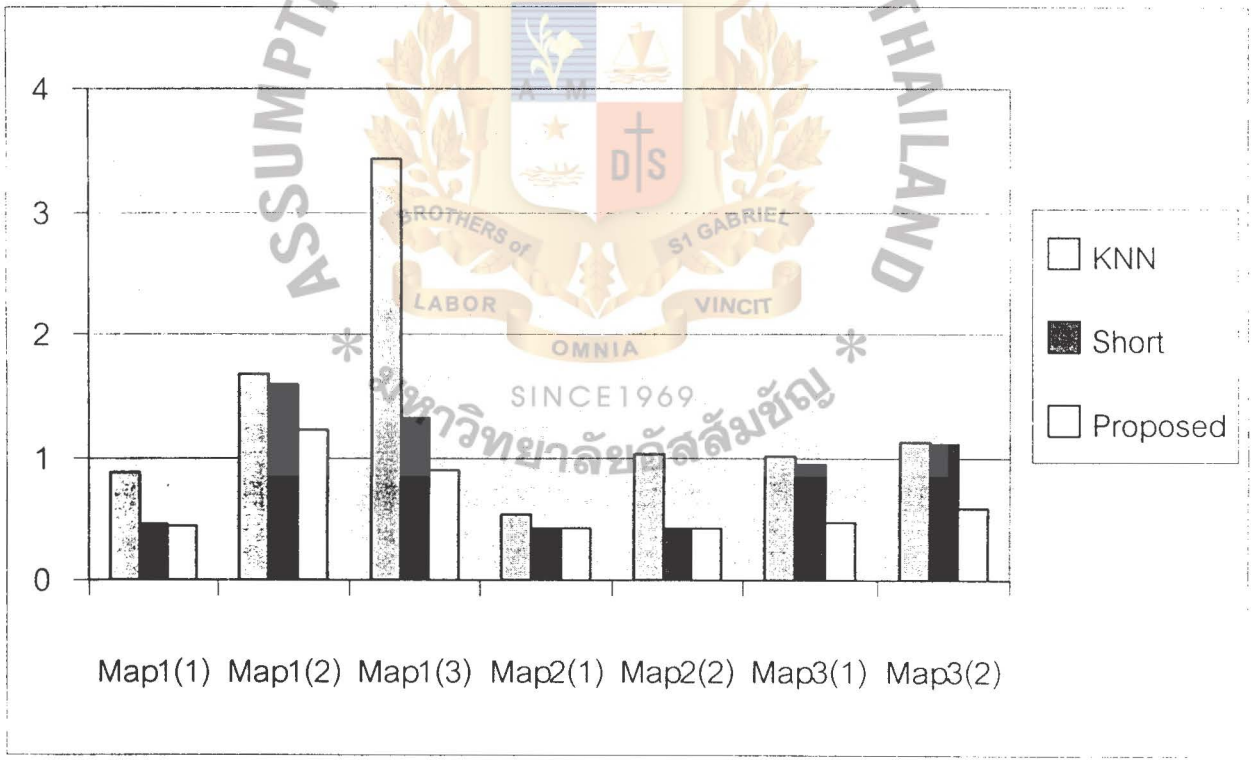


Figure 5-5: Traveling time of each technique.

	Processed Time						
	Map1(1)	Map1(2)	Map1(3)	Map2(1)	Map2(2)	Map3(1)	Map3(2)
KNN	1.0505	1.0303	1.0404	0.404	0.4141	2.0707	2.0808
Short	1.0303	1.0404	1.0404	0.404	0.404	2.0909	2.0808
Proposed	1.1616	1.1616	1.1616	0.4747	0.4848	2.3535	2.3232

Table 5-4: Process time of each technique.

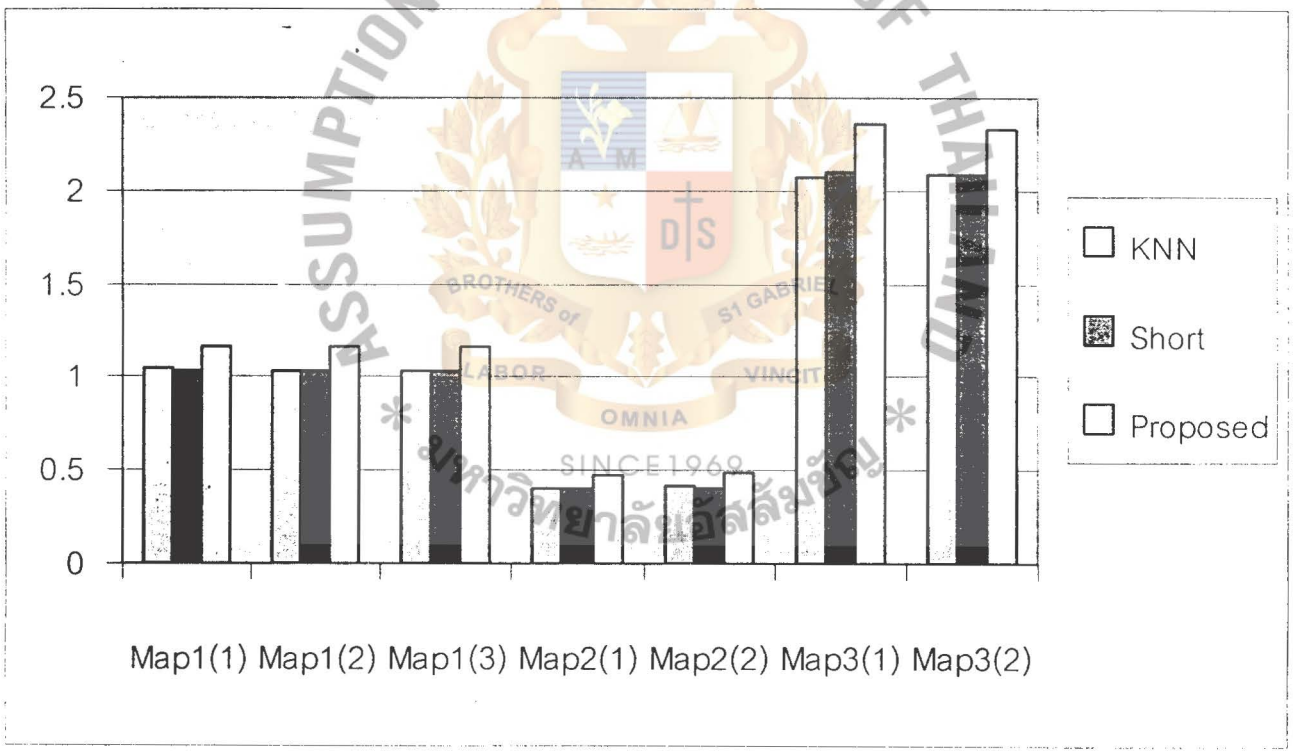


Figure 5-6: Process time of each technique.

This paper describes the result of experiment by time used below, which is in table 5-3, the ratio is 1 per 1,000 second.

For map 1, first experiment, the KNN technique has highest time used. The proposed technique's traveling time was 31.5 second lower than shortest distance path technique. The proposed technique's traveling time was around 4 second less than KNN technique.

For map 1, second experiment, any technique has traveling time higher than previous experiment because the starting time was traffic jam time. The KNN technique has slightly used more time than the shortest distance path technique. The proposed technique has lowest traveling time.

For map 1, third experiment, the KNN technique has traveling time twice over than the shortest distance path technique as the KNN technique has arranged randomly the paths with high traveling time. The traveling time of proposed technique was around 1,2268 second less than other technique (KNN, shortest distance path)

For map 2, first experiment, the KNN technique has slightly more traveling time than the proposed technique and the shortest distance path technique. The proposed technique has traveling time equal to the shortest distance path technique because map 2 doesn't involve condition that decreases driving efficiency.

For map 2, second experiment, the KNN technique has highest traveling time. The proposed technique has traveling time equal to the shortest distance path technique because map 2 doesn't contain condition that decrease driving efficiency.

For map 3, first experiment, the KNN technique's traveling time was 215.4 second higher than the shortest distance path technique. The proposed technique traveling time was approximately .5 second less than shortest distance path. Any technique has used more time than pervious experiment because the driver is a new

driver and map 3 involves bad status road and small size road, the status of road and size of road does affect new driver but does not affect experienced driver (old driver).

This paper describes the result of experiment by traveling time below. In table 5-4, this paper uses the minute to measure processing time.

For map 1, first experiment, the proposed technique has the highest processing time. The KNN technique has slightly more processing time than the shortest distance path technique.

For map 1, second experiment, the shortest distance path has slightly less processing time than the KNN technique, those techniques have processing time less than the proposed technique.

For map 1, third experiment, the shortest distance path technique has processing time equal to the KNN technique, those techniques have processing time less than the proposed technique.

For map 2, first experiment, the processing time of shortest distance path technique was .404 that is equal to the KNN technique. The proposed technique 's processing time was around .747 second higher than any other technique.

For map 2, second experiment, the KNN technique has slightly more processing time than the shortest distance path technique. Those techniques have less processing time than the proposed technique. In experiment of map 2, the processing time is less than other any experiment because this map includes few junctions and few roads.

For map 3, first experiment, the KNN technique has slightly more processing time than the shortest distance path technique. The proposed technique has the highest processing time.

For map 3, second experiment, the processing time of KNN technique has 2.0808 minute, which is equal to the shortest distance path technique. The processing time of proposed technique was 2.3232, the highest processing time.

Those experiments have highest processing time because the map 3 includes many roads and many junctions.

In case a map shows any positive condition (e.g. size is large, no traffic jam, status is good), the proposed technique retrieves a way that uses time equal to the shortest distance path technique.

On the other hand, when the map shows negative condition (e.g. traffic jam, size is small), the proposed technique retrieves a way that uses time lower than other techniques (shortest distance path, KNN).

In processing time, the proposed technique uses the processed time that is higher than other techniques because proposed technique uses rule process to rank each path after retrieving a way.

6.Conclusion

Every case in Case-Based database is important. Deletion technique can't resolve content problem because CBR can't find deleted case. This paper proposes to improve retrieve technique that can solve content problem. The proposed technique simply uses rule-based ranking in retrieving a case in order to find suitable path for all drivers in each situation. The time used for proposed technique is lower than other techniques when the map shows negative condition.

This paper describes the limitation of retrieval technique

Although proposed technique is better than other techniques, the processing time remains to be a problem because retrieval time is high. The retrieval technique uses only the small map. In case the maps are large size or contain a lot of intersections, system uses highest processing time to route to each path in order to find the target path. Another limitation of proposed technique is that some conditions are not realistic.

In future work, this paper expects to research filter technique to capture the size of map in order to reduce processing time and improve the conditions to be more realistic than what it has now.

6.Reference

[1] R. Schank: "Dynamic memory; a theory of reminding and learning in computers and people." Cambridge University Press. 1982.

[2] J. Kolodner: "Case-Based Reasoning; Morgan-Kaufman", San Mateo, California. 1993.

[3] J. Zhu and Q. Yang "Remembering to Add: Competence-preserving Case-Addition Policies for Case-Base Maintenance," In Proceedings of the Fifteenth In-
Ternational Joint Conference on Arti_cial Intelligence. Morgan Kaufmann, 1999.

[4] A. Aamodt and E. Plaza "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" AI Communications. IOS Press, Vol. 7: 1, pp. 39-59. 1994

[5] B. Smyth and M. Keane "Remembering to forget: A Competence-preserving case deletion policy for case-based reasoning systems" In international Join Conference on Artificial Intelligence, Volume 1, Pages 377-382, 1995

[6] D. Leake, and D. Wilson "Remembering Why to Remember: Performance-Guided Case- Base Maintenance," In Proceedings of 5 th European Workshop EWCBR 2000, Springer Verlag, pp 161-172.

[7] T. Cormen, C. Leiserson, R. Rivest and M. Hill Introduction to Algorithms 1990.

[8] P. Cunningham, B. Smyth "CBR in Scheduling: Reusing Solution Components," The International Journal of Production Research. pp 2947-2961.

[9] D. Applegate, V. Chvátal, and W. Cook "Finding Tours in the TSP," a preliminary chapter of a planned monograph on the TSP, 1998

[10] L. Portinale, P. Torasso and P. Tavano "Speed-up, quality, and competence in multi-modal reasoning.," In Proceeding of the Third International Conference on Case-Based Reasoning, pp 303-317

[11] S. Fox and D. Leake "Learning to Refine Indexing by Introspective Reasoning"

[12] R. Bergmann and W. Wilke, Towards a new formal model of transformational adaptation in case-based reasoning. In: L. Gierl and M. Lenz (eds.): Proc of 6th German Workshop on CBR, University of Rostock, 1998, pp. 43-52

[13] V. Balaraman, S. Chakraborti, S. Vattam, "Using CBR inexact search for intelligent retrieval of data", In Proceedings of International Conference, Knowledge Based Computer System (KBCS) 2000 conference.

[14] V. K Balakrishna, "Network Optimisation", Chapman & Hall, 1995.

[15] J. D. Hastings and L. K. Branting and J. A. Lockwood, "Case adaptation using an incomplete causal model", Lecture Notes in Computer Science, 1995, pp. 181-192