A Simulation-based Routing Selection Decision Support System

Yung-Hsin Wang
Department of Information Management
Tatung University, Taiwan
ywang@ttu.edu.tw

Chun-Ting Chen
Government Network Service Dept.
Chunghwa Telecom Co., Ltd., Taiwan
koly@cht.com.tw

Abstract

The method chosen for determining the "best" routing selection has a great effect both upon theoretical rules and the real problems met in our business life. When it comes to theoretical research, a good routing selection algorithm can not only trim costs and raise the efficiency of the system but also reduce the waste of system resources. Making a correct routing selection decision could help us deal properly with the routing problems such as that for the delivery service or machine arrangement in the factory. On the other hand, simulation methods are often used to solve complex system problems that cannot be solved by applying simple rules of mathematics. The objective of this study is to build a Simulation Based Decision Support System (SBDSS) to improve the quality of routing selection decisions. In order to raise the quality and performance of decision-making, simulation is used to assist the traditional decision support system and support the traditional rule base to create a more flexible decision support tool. Using simulation method as the kernel of our decision support system has benefit of replacing the fixed decision rules and getting more meaningful decision suggestion. This study also uses a real case of a travel agency as the experimental model to verify our system's performance. Simulation results show that the SBDSS architecture is effective and valuable for solving the routing selection problem.

1. Introduction

Routing selection is a key activity that affects the return on investment of both commercial industries and research organizations. Outstanding routing selection can have serious effects on organizational cost and efficiency. As a result, routing selection algorithms have been developed in numerous categories, such as delivery industries' road map problems, curriculum-scheduling methods, and even message forwarding on the Internet.

Routing selection problem can be defined as how to select the most economic route for a vehicle which makes multiple stops after departing from a central location and returns to the origin on completion of a series of tasks at those stops. The routing selection problem in this research is based on the classic Traveling Salesman Problem (TSP) [2] and Vehicle Routing Problem (VRP) [3] [4]. The TSP focuses on the minimization of travel costs for a single traveler with a flexible itinerary. Assume that a traveling salesman must visit several cities to expand his business; starting at one of the cities, one must visit all the others exactly once and return to the starting point [6]. The objective is to select the most economic routing. The VRP, in a situation constrained by either limited loading capacity or limited mileage, or both, creates a vehicle delivery routing plan that moves goods from warehouse to customer and returns the empty vehicle to the warehouse. The route plan must satisfy the needs of customers, minimize mileage and/or traveling time and allow for flexibility when any of these variables changes in priority [10].

The difference between VRP and TSP is primarily in the linkages between agents, destinations and limiting factors. In the VRP, any agent can usually service any destination. The TSP tends to use only one agent and destinations cannot be assigned to another agent. The VRP problem rapidly becomes more complex when multiple agents are introduced and limiting factors increase in number or variability. The most commonly defined optimum solutions for these problems require them to minimize costs in a multiple agent and multiple limiting factors situation. Conflicting or even mutually exclusive goals for those optimum solutions are inevitable in most business environments. It is the methods for resolution of these conflicting goals, which present the most fertile field for research.

The method chosen for deciding the "best" routing selection algorithm has a great effect both upon theoretical rules and the real problems we meet in our business life. When it comes to theoretical research, a good routing selection algorithm can not only trim costs and raise system efficiency but also reduce the waste of system resources. Making correct selection of routing algorithm is critical in most processes when materials or information must be delivered between multiple locations with multiple possible paths. For example, the routing for a vehicle must consider the optimum path that takes into account total mileage, road and traffic conditions, speed limits and congestion on alternate roads, and time constraints to arrive at particular points. Since some of these factors are often time variant, the problem can become very complex. Thus, creating rules and algorithms to provide optimum information to the decision makers at the right time under different environments will enhance the competitive advantage of the enterprise and help control unnecessary expenses.

Simulation methods are often used to solve complex system problems that cannot be solved by applying simple rules of mathematics [5] [7] [8] [11]. Problems with multiple variables, particularly those are inherently random, can often be solved only by the simulation method. Simulation allows researchers to analyze the differences in system behavior, which occur when multiple input parameters can vary simultaneously and/or randomly. A good simulation will test combinations of variables that are prohibitively expensive, in terms of time, money and/or number of iterations, to create in the real world. Simulation can also be used to analyze systems which cannot be physically accessed and give researchers the tools to analyze the impact of changes on these systems without the costs or risks associated with real system changes when the results cannot be predicted in advance. A basic property of simulation is that it can compare all possible results of all possible input values. With state of the art simulation tools, this can be done quickly and accurately.

Therefore, we attempt to use simulation approach as the kernel of the routing selection to develop a so-called Simulation Based Decision Support System (SBDSS). A travel agency is to be used as the experimental model to verify system's performance. The routine business of the travel agency often includes field assignments. Although the travel industry has made great effort on the office automation, travel agencies still have many procedures and services that must be handled personally, such as visa and passport, paper works, tickets delivery, and billing. Due to competitive pressure, the concept of "Customer First" is a business necessity for a travel agency to succeed. In order to promote the level of customer service, the field agents must plan their activities in the most time efficient manner while keeping costs minimum to the agency and the customer. The number of volatile, contradictory and conflicting requirements of this sort of scheduling made this business an ideal candidate for simulation.

As has been indicated, the routing selection problem of this study is defined on the basis of Traveling Salesman Problems (TSP) and Vehicle Routing Problem (VRP). We added several time factors to these traditional routing selection problems; i.e., processing time, traveling time, and time-limits, creating a system that is as close to the real situation as possible. Therefore, the routing problems we focused on are not pure TSP or pure VRP. The design of our research problem is based on a fusion of these two types of routing problems and adds consideration of additional parameters. The problem can be described as follows.

Assuming that there are several "Goal Points" distributed through the city, we have a "Start Point" from which several "Agents" depart on various assignments. Each assignment has multiple goal points and the various agents may cooperate by sharing works. An agent spends a period of time (Traveling time) traveling to a goal point, and then the agent must remain (Processing time) at the goal point while performing the task assigned to that goal point. After completion of the task, the agent will depart the goal point. And the agent must check to see if there are any additional goal points to be processed for any assignment and, if so, which of the remaining assignments' goal points should be the next. Each goal point must be processed within its time-limit. Any goal point not processed in time will become an incomplete goal point. For example, for a point with an average processing time of 0.5 hours and a time-limit of 3 hours, if no agent can process that goal point within 3 hours of the start of the routine, either because the agents are too busy or because there are too few agents, it will become an incomplete goal point and the assignment is incomplete too.

Each goal point has four basic attributes, including x coordinate, y coordinate, time-limit, and average processing time. Agents pass these goal points one by one continuously (Figure 1). Different sequences of passing the goal points mean different routings, and our problem is which routing can achieve our management goals and which is/are the optimum routing (there may be equal alternatives). The optimum changes when different weights are applied to various management goals. The priority goal might be the "shortest path" (save travel costs) or the "highest complete rate" (high service performance). Because there can be more than one agent, as the numbers of agents and goal points increase, the problem becomes more complex.

Note that the objective of this study is to implement a decision support system for routing selection through the design of the SBDSS architecture. The system is built to adapt to the actual routine of the field workers in a travel agency and the performance is evaluated. We hope to use the simulation method as an auxiliary for the traditional decision support system to obtain high quality routing decisions and help maximize customer satisfaction and minimize operation costs. However, we did not focus on the optimum routing selection algorithm. That is, the research does not necessarily provide an optimum solution for a problem but aims for flexible decision support architecture.

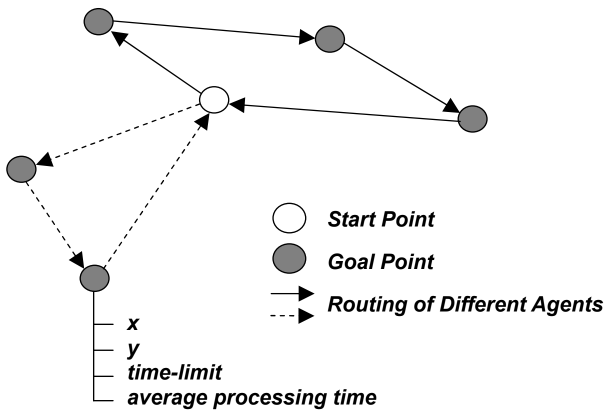

Figure 1. Problem definitions

2. Simulation Process and SBDSS Overview

2.1 Simulation Process

To build a model through simulation software or programming language that can operate on the computer by use of logic, probability and statistics algorithms. The model will be varied in accordance with each type of simulation software, but the logic design must conform to the target system behavior. Animation of the simulation model helps users to believe the validity of the model. Animation is also almost indispensable when communicating the outcome of a simulation to the non-technical audience.

A valid model can be used to simulate. In most cases, a complete set of the relative parameters and information that correspond to the model are necessary for simulation. For example: when modeling the questions: “What is the average processing time of a single processor?” or “What is the random number distribution of an entity source?” Obviously, these parameters and information will influence the result of the simulation directly; therefore, we must be very careful while determining the proper values. In a physical existent system, we usually can collect these parameters and gather information using the knowledge and experience of the system staff or use information that is generated by the system itself.

The purpose of simulation is to understand and analyze the system. Therefore, simulation must be able to produce the results and statistics that users care about in order to provide a basis to improving system performance.

2.2 Simulation Based Decision Support System

Shahraray and Maeschke [9] provided a simulation based decision support system operating architecture for scheduling problems common to the manufacturing industry. In manufacturing, the decision makers usually have to follow a different operating strategy to achieve management objectives. For example, assume that we have several orders with different expiration dates and handling processes in a job shop. The manager's objectives are to fulfill the orders on time, keep a low level of Work-In-Process (WIP), a high level of machine utilization and a minimum of processing time all at the same time. In essence, each employee should be busy at all times, each machine should be fully utilized and all work should move through the shop in minimum time. These objectives are conflicting in nature. Even the most skilled workers perform at different speeds. Different machine processes run at different rates. The numbers of orders driving the system is totally outside the manager’s control and usually arrive at random times and in random quantities [1]. Because of these variables, it is difficult, if not impossible, to create an operating strategy which satisfies all these objectives at the same time. Yet, these are the factors which contribute the most variables to costs and which must be controlled in order to run the enterprise profitably. Therefore, an appropriate operating strategy is very important to the manufacturing industry.

The SBDSS architecture is presented as a possible solution to the manufacturing strategy problem [1] [9]. It provides decision makers an optimum operating strategy based on the situation of the factory, the characteristics of the product and the goals of management. The architecture of Figure 2 shows three main functional modules in the SBDSS:

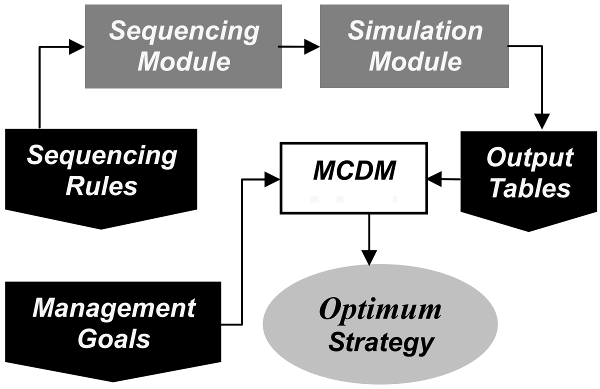

Figure 2. An SBDSS architecture

(1) Simulation Module: The main function is to simulate and compare all the alternatives quickly after the sequencing module selects the sequencing rules. The result is saved for the foundation of later analysis.
(2) Sequencing Module: It provides the decision makers appropriate sequencing rules based on factory status, product characteristics and management goals.
(3) Multiple Criteria Decision Making (MCDM) Module: The main task of the module is to analyze and evaluate the results of the simulation and reference the management goals that were set to provide the decision suggestions. The MCDM module is the explanation module; it provides goals related information to the decision makers.

The SBDSS architecture proposed can provide decision suggestions in multiple criteria situations. It uses criteria scores to present the results and provides information to support the decision makers in selecting operation strategies. This architecture supplies reference material for the components of the SBDSS and also verifies the effectiveness of the SBDSS.

3. System Modeling and Simulation Design

3.1 System Architecture Design

3.1.1 System Requirement Analysis

The first step in our system design is to define the requirements. The main functions are as follows:

1. Problem Defining Function
2. Problem Saving Function
3. Problem Describing Function
4. Simulation Environment Setting Function
5. One Time Simulation Function
6. Standard Simulation Function
7. Experimental Simulation Function
8. Result Display Function
9. Decision Support Function

3.1.2 Architecture Design

According to the requirements mentioned above, we design the system architecture as shown in Figure 3.

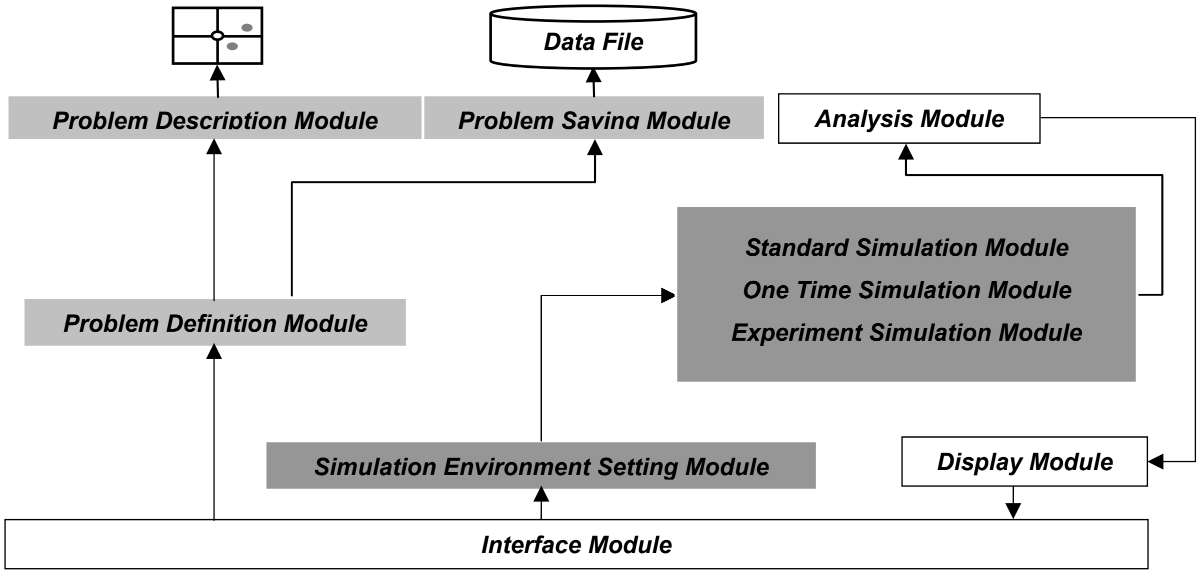

Figure 3. System architecture design

3.2 Modeling

3.2.1 Basic Operation

Building the simulation environment was the most important step in system design phase. We designed the simulation environment as a plane where x and y denotes the horizontal and vertical coordinate axis. According to the research problem, we defined the data structure of a single point as Table 1.

Table 1. Structure of the point data

```
// Point_Data class was defined to be the basic
// structure of the saved point data
public class Point_Data {
    public double   X;
    public double   Y;
    public double   Due_Hour;
    public double   AP_Time;
    public boolean  Processed;
    public boolean  Complete;
}
```

The variable X and Y are defined to record the coordinate while *Due_Hour* and *AP_Time* are used to record the time-limit and average processing time of a point. Besides that, Processed and Complete are two flags for simulation control. They are used to determine the status of a point in simulation.

We defined (0, 0) as the origin point that the coordinates of all the destinations must be defined as the relative value to it. In our design, first, all agents travel from the origin point to a chosen goal point. Then, the agent will stay at that goal point until the process of the point has been done. Finally, the agent will choose another goal point in accordance with its routing selecting algorithm.

An obvious error will occur while two or more agents forwarding to the same destination. To avoid that, we use the Processed flag to determine if the goal point is available. When an agent chooses a point as the next goal, the flag will be set to True. It will prevent other agents from choosing the same destination as long as all agents choose from the goal points with "Processed" equals "False." Under this design, we can avoid unnecessary collision that might cause meaningless results.

3.2.2 Agents and Algorithms

The algorithm is, the rule that agents use to choose next goal point, the critical factor of the system performance. Under a good plan of algorithms, the frequency of the simulation can be reduced; the system performance can be improved, and the foundation of the suggestion will be more powerful. The main purpose of the research is to build the SBDSS architecture; therefore, we didn't focus on optimizing the routing selection algorithms. To expand the sort of algorithm or enhance the routing selection method might be the future work of the research.

In our system design, we build three simple algorithms, including shortest path first, shortest processing time first, and shortest time-limit first. The shortest path first selects the nearest destination from all the unprocessed goal points; the shortest processing time chooses the goal point that needs shortest average processing time, and the shortest time-limit first selects the most urgent one.

The model assumes that the algorithms of the agents were given by users or system before simulation start and would never change until the end of simulation. Different numbers of agent assigned with different algorithms forms different alternatives of the simulation work. Then system will test all the alternatives and find out the most appropriate one and give the suggestion. In general, the procedure of the agent can be presented as Figure 4.

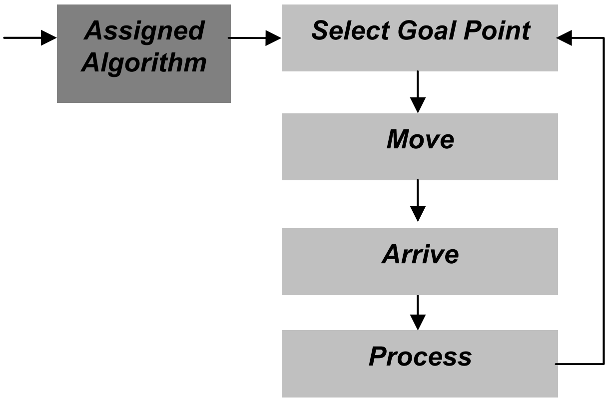

Figure 4. Logical procedure of the agent

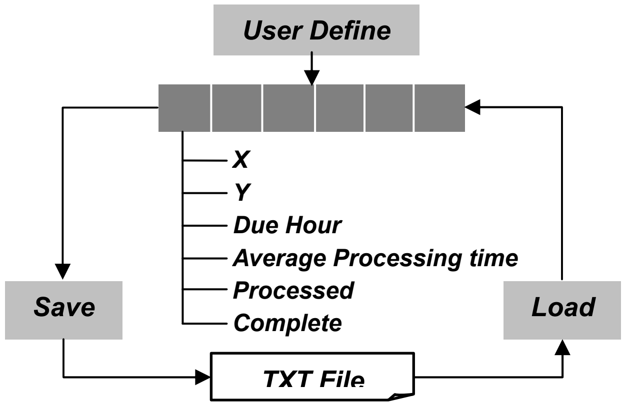

Figure 5. Saving and loading the point Information

3.2.3 Random Variables

Another critical design of modeling is the distribution of random variables. In our research, two main random variables are traveling time and processing time. Since different kinds of problem may have different characteristics, the general purpose simulation software usually provide various random number generator functions. We design three basic random number generator functions, including normal distribution, exponential distribution, and uniform distribution.

3.2.4 Individual Path Match Mechanism

In most situations, the mean value of processing time distribution is determined by average processing time defined by users, and average traveling time is defined by s/v, where s and v denotes the distance of the path and the average speed of agents. In addition, in order to make up for the defect that system environment cannot describe the city roads well; the individual path match mechanism was designed to record the average traveling time of specific path that are defined by users. While agents decide the next goal point, the path between current point and goal point will be checked. If the path has been defined, the random number generator will generate the random t raveling time in compliance with the distribution and user-defined value; otherwise, agent will count the traveling time in accordance with the distance and the average speed.

3.3 Operating Design

3.3.1 Operation of the Saved File

Saved file, in the system, is used to store the information of all goal points defined by the users. The file type is text file (*.txt). While saving, the system writes the necessary values of each point in sequence from object array declared base on the structure shown in Table 1 into the assigned file. The necessary values include X coordinate, Y coordinate, time-limit, average processing time, and two flags. On the other hand, the system will read these values into the object array by parsing the strings extracted from the text file line by line while loading a file. The operations are depicted in Figure 5.

3.3.2 Operation of Individual Path Match Mechanism

The basic data structure of individual path match mechanism is similar to that of point information. Table 2 shows the declaration of this structure.

Table 2. Structure of individual path match mechanism

```
// The class was defined to save the user-defined
// average traveling time of particular paths
public class Individual_Path_Average_Traveling_Time {
    double  Start_X;
    double  Start_Y;
    double  Goal_X;
    double  Goal_Y;
    double  APT;
}
```

The individual path match structure is also declared an object array to record each user-defined average traveling time data. Note that not all defined value is necessary to be used. Since we cannot predict the actual routings of agents because of random process of the simulation work, we can only input the self-defined path average traveling time as more as possible. The input data are not promised to work in simulation. For example, the path A to C is defined with average traveling time equals to half an hour. While an agent take the path form A to B and then from B to C, the pre-defined path value will not affect the simulation.

3.3.3 Operation of One Time Simulation Function

The display of the simulation result differs from one time simulation mode to standard simulation mode. One time simulation mode, using pre-defined algorithm match, runs one time only and presents the result to the users. The main purpose of this mode is for observation. In this mode, the step information will be added to a text area after each step. We designed “Step by Step” and “Continuous” options that users can easily control the simulation advance. In Step by Step option, simulation is driven by user click; Continuous option will drive the simulation continuously until end of simulation. Figure 6 shows the design of one time simulation operation.

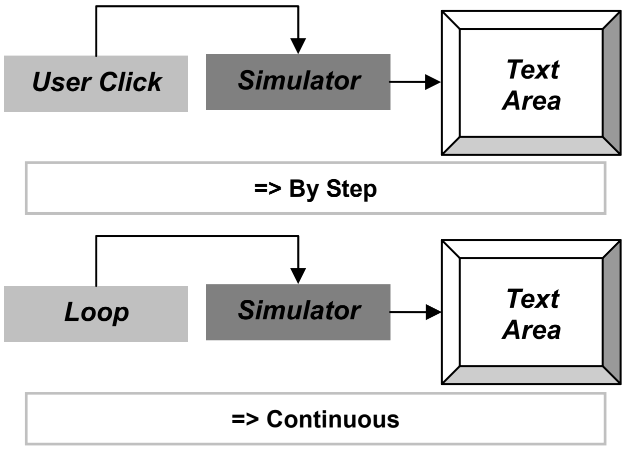

Figure 6. Operation of one time simulation

3.3.4 Operation of Standard Simulation Function

In the standard simulation mode or experiment mode, system tries every possible algorithm matches, puts into simulation, and generates the result information. After that, users can select the proper suggestions agree with management goals. In this situation, simulation must be executed many times in these modes. Therefore, the display method of one time simulation is not suitable for this case.

Considering that system must record huge amount of result data continuously, there must have a huge temp to store these data. We use two temp files to solve the problem. After end of each time of simulation, system writes the result data into two string buffers. The result buffer records some important result data, including simulation time, total processing time, total traveling time, and complete rate, etc. And the routing buffer records all the routing data of each time of simulation. Then the data in these two buffers will be written into two temp files after all simulation works have been done. The temp file of result data mainly used to be the source of a result table that is displayed to the users, and the temp file of routing data exists for retrieve of the detail routing information. The operation is illustrated as Figure 7.

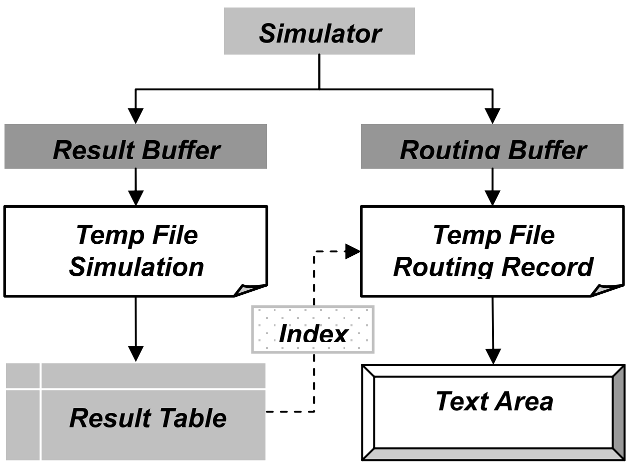

Figure 7. Operation of standard simulation

Writing into a file involves the I/O process. To CPU processing time, too many I/O processes leads to inefficiency and uneconomical resource waste. Therefore, we leave result data in the string buffers and write into temp files only once at the end of all simulation works. In so doing, the system performance can be improved effectively, but it will obviously increase the load of main memory. However, for improving the system performance, the waste of main memory is reasonable and acceptable.

3.4 Simulation Design

The design of simulation operation is the kernel of system design work. Discrete event simulation plays an important role for the simulation concept in the system, and is also the most critical methodology in our research.

Discrete event simulation advances by event happening. Every event has event type and event time that event type defines what kind of event should happen and event time defines when it does. Simulation control process always chooses the event with minimum event time from event list, that is, the nearest event that should happen. And then, according to its event type or other attributes, corresponding processes should be done. In most cases, an event often arranges another event into event list while processing and that will ensure the simulation can go forward until the end. For example, an arrival event should arrange a departure event, and a departure event should arrange another arrival event. Note that “end of simulation” is just a pseudo event. The operation of discrete event simulation model of this research can be described as Figure 8.

The departure event processes are: First, the agent will select next goal point in accordance with assigned algorithm. Then, a random value of traveling time will be generated according to individual path match records or simply to the value calculated by distance and average speed. Finally, a new arrival event will be added into the event list, representing the event that agent arrives at next goal point. On the other hand, the arrival event processes are: First, a random value of processing time will be generated according to the average processing time of that point. Then, another departure event will be inserted into the event list for another departure event with event time equals to current time plus random processing time.

The simulation control processes are in charge of deciding which event to be happened next and time advance. The event, which has the minimum event time, will be selected as the next event, and once the event had happened, it will be deleted from the event list. That is, the next nearest event will be selected naturally and elder events will never happen again.

Another data structure seems to be very important to the simulation work is that of the event list. Event list is the basis of simulation advance that works as the schedule of the events. The data structure of the event list is as shown in Figure 9. Note that in addition to event type and event time, the event unit records which agent is supposed to execute the event process in a multi-agent environment.

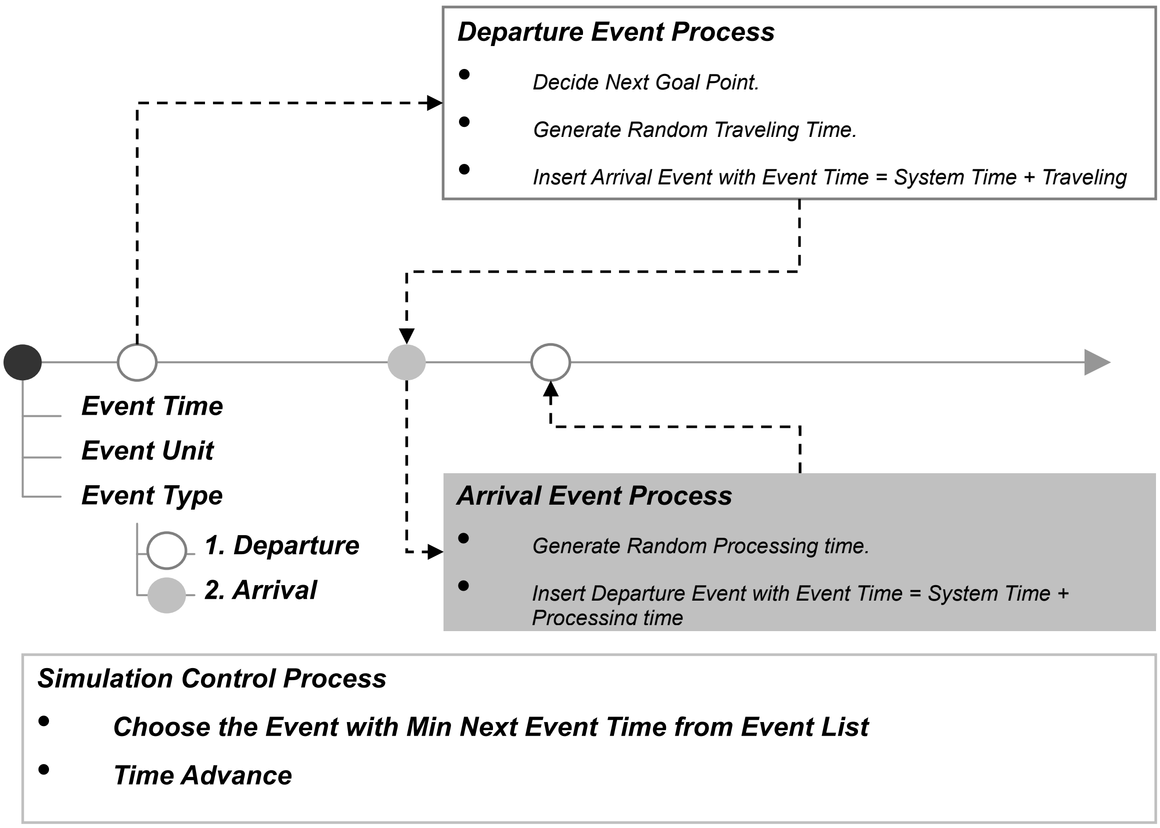

Figure 8. Discrete event simulation model of this study

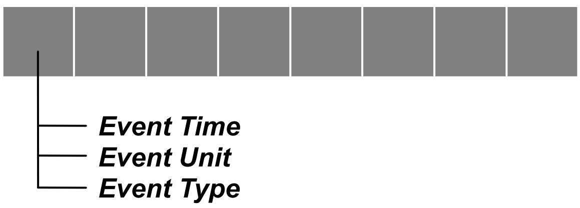

Figure 9. Data structure of the event list

4. System Experiment and Evaluation

4.1 Sampling

Before experimenting, the distributions of traveling time and processing time should be confirmed. The main purpose of sampling is to collect data for goodness of fit analysis in next phase. Obviously, the sampling must focus on a specific path and a specific process routine, so that we are supposed to select a particular destination as the observing object. Here we choose the BOCA (Bureau of Consular Affairs) Taipei as the object since that this destination is most often to be arranged into the processing schedule. The service of BOCA Taipei includes passport issuance, document authentication, country entry and exit, alien regulations, foreign labor management, military service and household registration, etc. The field workers were asked to record traveling time and processing time every time when they were assigned to go to the BOCA. We spent more than three months collecting data from Dec. 23, 2003 to Feb. 27, 2004. These data were collected and recorded by the field workers of the Glory Travel Agency.

The path that we recorded the traveling time is from travel agency to BOCA Taipei or from BOCA to travel agency, that is, only if the BOCA was the first or the last destination of a task, the traveling time was recorded.

On the other hand, in order to obtain the reasonable data, the sampling of the processing time must be limited to focus on a specific task at BOCA. We selected the passport application as the sampling object; thus, only when the field workers deal with the passport application at BOCA, the processing time was recorded.

4.2 Goodness-of-Fit Test

A goodness-of-fit test is used to find out the proper distribution of traveling time and processing time. First, we divide the source data of traveling time into 5 groups and count the frequencies for each group, and the data of processing time is done in the same way. We then use the chi-squared test with $\alpha = 0.05$ to test if each of the two variables (traveling time and processing time) is a normally distributed variable.

The result shows that there's lack of evidence to prove that the two variables are not normally distributed. Therefore, we used normal distribution to generate random numbers of the traveling time and processing time in our experiment.

4.3 Experimental Design

The experiment lasted for a month from March 1, 2004 to March 31, 2004, and the experimental result is shown in the next section. Our design is to divide the days into 4 periods (one week for each period except the last one) as experimental group and control group.

We set period 1 and 3 as the control group; that is, in these periods, field workers were asked to do the jobs as they usually did. And in most situations, they decide their routings only by their intuition or simple discussion. Period 2 and 4 were set to be the experimental group. In these periods, the field workers made their decisions in accordance with the suggestions of our system.

The fieldwork of the Glory Travel Agency is always dynamic; that is, only a little part of the work is stable routine. A new task may be inserted into the schedule at any moment. Therefore, the field workers always wait for a period of time, and then get tasks their manager assigned. The manager needs to consider how many destinations are going to be processed, how they distribute, and how many workers are free to be assigned right away. After that, the manager subjectively decides how many workers go out for tasks. In brief, there is more than one assignment in a single day. For example, they may have several tasks in the morning and another in the afternoon; it is impossible to leave all the works to the afternoon.

The setting of parameters is another problem. Since each field worker has different ways to deal with their tasks, it is hard to set parameters that everyone will strongly agree with. For this reason, we asked them to set a near-average value that each one of them can accept all the parameters they need to handle. By the way, there are four field workers in the Glory Travel Agency, and each of them has other duties in the company while they are waiting for the work assigned.

4.4 System Performance Evaluation

The performance evaluation is done by testing the differences between experimental group and control group. First, we must define an evaluating statistic value for evaluation. The performance index is defined as: $s = w \times t / n$, where $\boldsymbol{w}$ denotes the number of field works, $\boldsymbol{t}$ denotes the complete time (in minute), and $\boldsymbol{n}$ denotes the number of destinations. The statistic value of each record in every period is calculated and shown in Table 3.

In the situation that there are too little destinations, the decisions made by human brains may not have obvious differences with computer's suggestions. So we ignored the records that have destinations less than 4 and put others together into Table 4.

To analyze the data we firstly sort it for both groups and then transform it into a polygon chart as shown in Figure 10 where we can easily identify the differences between the two groups. The improvement of the field workers' performance proves the effectiveness of the system.

Table 3. Performance index values of each record

The Control Group				The Experimental Group			
Period 1		Period 3		Period 2		Period 4	
Date	Value	Date	Value	Date	Value	Date	Value
3/1	60.5	3/15	67.8	3/08	46.29	3/22	48.67
3/1	77.33	3/15	92	3/08	50.4	3/22	42
3/1	51	3/15	67.5	3/08	23.5	3/23	54.5
3/2	50	3/16	72.57	3/09	46	3/23	42
3/2	62.8	3/16	66	3/09	39.33	3/24	32
3/3	47	3/17	71.67	3/09	49.5	3/24	46.15
3/3	49.2	3/17	68.4	3/10	50	3/25	39.75
3/3	51	3/18	78.67	3/10	28	3/25	45
3/4	75	3/18	54.22	3/11	59	3/25	64
3/4	45	3/18	65	3/11	49.8	3/26	59.33
3/5	69.77	3/19	74.91	3/11	71	3/26	58.4
3/5	49.5	3/19	54.86	3/12	58.4	3/29	65.14
				3/12	42	3/29	42
						3/29	34
						3/30	46.5
						3/30	52
						3/31	54.4
						3/31	49

Table 4. Performance index of the chosen records

The Control Group				The Experimental Group			
Period 1		Period 3		Period 2		Period 4	
Date	Value	Date	Value	Date	Value	Date	Value
3/1	60.5	3/15	67.8	3/08	46.29	3/22	42
3/1	51	3/15	67.5	3/08	50.4	3/23	54.5
3/2	50	3/16	72.57	3/09	39.33	3/24	46.15
3/2	62.8	3/17	71.67	3/09	49.5	3/25	39.75
3/3	49.2	3/17	68.4	3/10	50	3/25	45
3/4	75	3/18	54.22	3/11	59	3/26	59.33
3/5	69.77	3/19	74.91	3/11	49.8	3/26	58.4
		3/19	54.86	3/12	58.4	3/29	65.14
				3/12	42	3/30	46.5
						3/30	52
						3/31	54.4
						3/31	49

Finally, we use the t test to compare the two population means. The hypotheses are as follows.

$$H_0: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

The report generated by the SPSS shows that $t = 4.55$ (equal variances not assumed) does not fall into the 95% confidence interval, [7.12379, 18.91336]. Thus, we reject H_0: $\mu_1 = \mu_2$. In other words, there are significant differences between the experimental and control group. This result also represents the performance of our simulation based architecture.

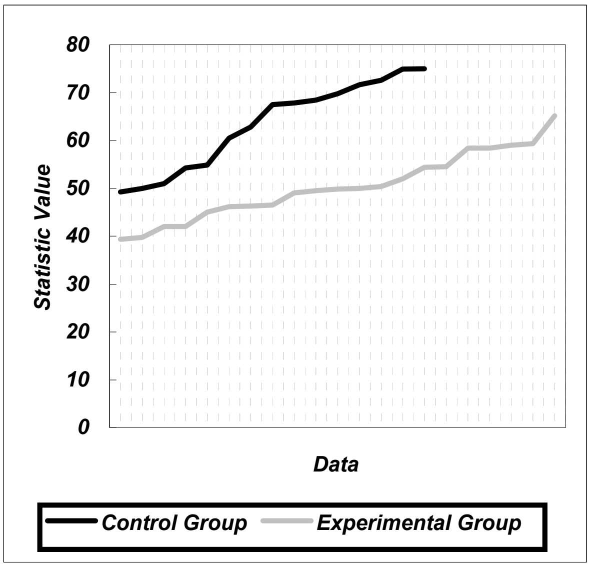

Figure 10. Polygon chart of sorted statistic values

5. Conclusion

Routing selection problems affect our daily life very much in many ways. The method chosen for determining the "best" route selection has a great effect both upon theoretical rules and the real problems we meet in our business life. A good routing selection algorithm can not only trim costs and raise the efficiency of the system, but also reduce the waste of system resources. In this research, we implemented a simulation based decision support system to solve time-considered routing selection problems. According to experimental result of the case travel agency experiment, we proved that the suggestions provided by our system are more effective than that by human intuition. Although we cannot prove optimum, the system does improve the quality of decision effectively.

Simulation methods are often used to solve complex system problems that cannot be solved by applying simple rules of mathematics. Problems with multiple variables, particularly those that are inherently random, can often be solved only by the simulation method. The routing selection problem is such a good application example. In this study, the routing problem is defined as time consideration one which involves random processing time, random traveling time, and time limitation. Obviously, there's no existed mathematical rules can be used to solve this kind of complex problem. Through the experiment, we also presented that the simulation approach is suited to solve the routing selection problems.

This research still has some spaces for improvement on system design. One of our future works is described in what follows. Three kinds of different algorithms were built in this system. In most routing selection deciding processes, taking use of only three simple algorithms is not very convincing. Therefore, the future works should consider more complex routing selection algorithms and that would make the system suggestion more meaningful.

References

[1] Andersson, M. & Olsson, G. "A Simulation Based Decision Support Approach for Operational Capacity Planning in a Customer Order Driven Assembly Line," In *Proceedings of the 1998 Winter Simulation Conference*, 1998, 935-941.

[2] Flood, M.M. "The Traveling Salesman Problem," *Operations Research*, 1956, Vol. 4, 61-75.

[3] Fisher, M. "Vehicle Routing," in M. Ball, T. Magnanti, C. Monma and G. Nemhauser (eds.), *Network Routing, Handbooks in Operations Research and Management Science*, North-Holland, Amsterdam, 1995, 1-33.

[4] Golden, B.L., Assad, A., Levy, L. & Gheysens, F.G. "The Fleet Size and Mix Vehicle Routing Problem," *Computer & Operations Research*, 1984, Vol. 11, 49-66.

[5] Law, A.M. & Kelton, W.D. *Simulation Modeling and Analysis*, 3rd Edition, McGraw-Hill, 2000.

[6] Papadimitriou, C.H. & Steiglitz, K. "Some Complex Results for the Traveling Salesman Problem," In *Proceedings of the Eighth Annual ACM Symposium on Theory of Computing*, Pennsylvania, United States, 1976, 1-9.

[7] Pegden, C.D., Sadowski, R.P. & Shannon, R.E. *Introduction to Simulation Using SIMAN*, 2nd Edition, McGraw-Hill, New York, 1995.

[8] Schriber, T.J. "The Nature and Role of Simulation in the Design of Manufacturing Systems," in Wichmann, K.E. (ed.), *Simulation in CIM and Artificial Intelligence Techniques*, The Society for Computer Simulation, San Diego, California, 1987, 5-18.

[9] Shahraray, M.S. & Maeschke, M. "SBDSS – A Simulation based decision support system," In *Proceedings of the 22nd Winter simulation conference*, New Orleans, Louisiana, United States, 1990, 664-668.

[10] Thangiah, S.R., Shmygelska, O. & Mennell, W. "An agent architecture for vehicle routing problems," In *Proceedings of the 2001 ACM Symposium on Applied Computing*, Las Vegas, Nevada, United States, 2001, 517-521.

[11] Turban, E. & Aronson, J.E. *Decision Support Systems and Intelligent Systems*, Sixth Edition, Prentice-Hall International, Inc., Upper Saddle River, N.J., 2001.