A Knowledge-Based Approach for Railway Scheduling

Kiyotoshi Komaya and Toyoo Fukuda

Industrial Electronics & Systems Development Laboratory
Mitsubishi Electric Corporation
8-1-1, Tsukaguchi-Honmachi Amagasaki, Hyogo, 661, JAPAN

Phone: +81-6-497-7639
Fax: +81-6-497-7727

ABSTRACT: *This paper proposes a new problem solving architecture for knowledge based integration of simulation and scheduling, and describes two knowledge-based systems for railway scheduling using it, DIAPLAN and ESTRAC-III. For railway scheduling, the most important point in experts' problem solving processes is that they integrate simulation with scheduling in an intelligent manner. To emulate these processes on a computer, our architecture consists of four major components — partial simulation, basic command, tactical knowledge, and strategic knowledge. The partial simulation and basic command are employed for simulation of train movements in a subsystem, and tactical knowledge is employed for local scheduling. Strategic knowledge is employed to emulate experts' reasoning processes by performing partial simulation and local scheduling repeatedly in a specific order. Since these components are defined independently and layered hierarchically in our architecture, we can develop a knowledge-based system in a step-by-step manner by defining four components — partial simulation, basic command, tactical knowledge, and strategic knowledge in that order.*

DIAPLAN and ESTRAC-III are designed to support experts in planning and restoration of railway systems, respectively. Through the experience of applying both systems to large and complicated railway systems operated in the real world, it has been confirmed that both systems can support experts efficiently and effectively.

AI TOPIC: Case Study (Railway Systems)

DOMAIN: Railway Scheduling

LANGUAGE: C

STATUS: Development

EFFORT: 6 person-years

IMPACT: A new problem solving architecture is proposed for railway scheduling. Its effectiveness has been confirmed through the experience of developing DIAPLAN and ESTRAC-III.

A Knowledge-Based Approach for Railway Scheduling

Kiyotoshi Komaya and Toyoo Fukuda

Industrial Electronics & Systems Development Laboratory
Mitsubishi Electric Corporation
8-1-1, Tsukaguchi-Honmachi Amagasaki, Hyogo, 661, JAPAN

Abstract: *This paper proposes a new problem solving architecture for knowledge based integration of simulation and scheduling, and describes two knowledge-based systems for railway scheduling using it, DIAPLAN and ESTRAC-III. For railway scheduling, the most important point in experts' problem solving processes is that they integrate simulation with scheduling in an intelligent manner. To emulate these processes on a computer, our architecture consists of four major components — partial simulation, basic command, tactical knowledge, and strategic knowledge. The partial simulation and basic command are employed for simulation of train movements in a subsystem, and tactical knowledge is employed for local scheduling. Strategic knowledge is employed to emulate experts' reasoning processes by performing partial simulation and local scheduling repeatedly in a specific order. Since these components are defined independently and layered hierarchically in our architecture, we can develop a knowledge-based system in a step-by-step manner by defining four components — partial simulation, basic command, tactical knowledge, and strategic knowledge in that order.*

1 Introduction

The operation of railway systems requires solving various types of scheduling problems, such as timetable preparation and resource allocation. Operations Research techniques are conventionally applied to these problems. However, it is difficult to solve real world railway scheduling problems by OR techniques, since they are essentially large-scale combinatorial problems and have many conflicting criteria (e.g., [1],[2],[3]). Railway scheduling problems, therefore, still depend heavily on flexible information-processing and decision-making abilities of human experts.

The new approach using Artificial Intelligence techniques has been applied to real world scheduling problems, such as factory-scheduling problems (e.g., [4],[5]). It is based on a human expert's knowledge and emulates his problem solving process on a computer. We have also studied AI techniques for more than seven years to apply to railway scheduling problems [1],[2],[6]. Based on the experience of building prototype expert systems and investigating experts' reasoning processes of problem solving, we propose a new problem solving architecture for transportation scheduling, especially for railway scheduling. This architecture is designed for knowledge based integration of simulation and scheduling, and consists of four major components — partial simulation, basic command, tactical knowledge, and strategic knowledge. The partial simulation and basic command are employed for simulation, and tactical knowledge is employed for scheduling. And strategic knowledge is employed to manage the order of integrating simulation with scheduling. Since these components are defined independently and layered hierarchically, we can develop a knowledge-based system in a step-by-step manner by defining four components — partial simulation, basic command, tactical knowledge, and strategic knowledge in that order.

We have developed two knowledge-based systems, using our architecture, DIAPLAN* and ESTRAC-III** [7]. DIAPLAN prepares a complete timetable from given initial conditions for a train. ESTRAC-III prepares a rescheduling plan in the case of disturbed train traffic, within a practical computation time. Through the experience of applying both systems to large and complicated railway systems operated in the real world, it has been confirmed that both systems can support experts efficiently and effectively.

This paper is divided into the following four sections. Section two surveys railway scheduling problems and describes two important features extracted from experts' reasoning processes. Based on this observation, we propose a new problem solving architecture in section three. And four major components of our architecture are also described in detail in this section. Section four overviews two knowledge-based systems, DIAPLAN and ESTRAC-III, and shows illustrative examples of their applications. Finally, section five summarizes our architecture and two knowledge-based systems.

---

* Train Diagram Planning Support System

** Expert System for Train Traffic Control

2 Problem definition

The most difficult part of developing a knowledge-based system consists of understanding and modeling human experts' reasoning processes in the real world problem solving. We started by taking a survey of railway scheduling problems which had been solved by experts.

2.1 Railway Scheduling Problems

In railway systems, the timetable containing the arrival and departure times of all trains at each station is the most essential schedule for train operation. The timetable is manually prepared by experts, though the train operation is automatically regulated by a computer system which holds the timetable in its memory. The experts prepare the timetable in two phases — planning and restoration. In planning, the experts prepare a set of timetables which involves the timetable for weekdays, Sundays, special holidays, and so on, and they also assign resources such as trains and their operators for each timetable. For most commuter lines in Japan, these timetables are prepared every two or three years by several experts, and it takes about one year or more. On the other hand, restoration is carried out by experts, called dispatchers, when an accident or trouble disturbs train traffic. The dispatchers are required to prepare a rescheduling plan for the working timetable of that day within a short time, since the restoration work is urgent. In both cases, planning and restoration, the experts must prepare a timetable which satisfies several criteria within the physical limits. Three components of this problem: a timetable, criteria, and physical limits are as follows:

Timetable

- <u>For all trains</u>
  - Arrival time at each station
  - Departure time at each station
  - Platform at each station

Criteria

- <u>For planning</u>
  - Optimization of passengers service
  - Minimization of operation costs
- <u>For restoration</u>
  - Minimization of the effects of disturbance
  - Minimization of additional operation costs
  - Minimization of inconvenience to passengers

Physical limits

- <u>For all train types</u> (such as an express and a local)
  - Minimum running time between two stations
  - Minimum stopping time at each station
- <u>For all stations</u>
  - Minimum headway time for arrival
  - Minimum headway time for departure
  - Minimum time of turning back at the terminal
  - Maximum limits of trains running between two stations

For this problem, no optimizing algorithms exist, since it is a large-scale integer problem. For example, there is 2^1 combinations for the overtaking operation at each station, that is, either an express train overtakes a local one or not. We could therefore create 2^{nk} combinations in a simple line running n express trains and including k stations where an express train overtakes a local one. And this problem has many conflicting criteria. For example, an increase of service frequency results in high-quality service, but also high operation costs. Furthermore, some criteria such as the optimization of passengers service are difficult to deal with numerically. Consequently, we cannot solve this problem in a straightforward algorithmic way.

2.2 Problem Solving Process

Based on this observation, we selected a knowledge-based systems approach for this problem. And we studied the experts' preparation processes for the timetable in cases of both planning and restoration, with the cooperation of several railway companies. From this experience, we found that experts use a two-dimensional chart of time and distance in their work; this is called a train diagram. The train diagram, as shown in Fig.1, is a domain-specific chart showing the arrival and departure time at each station. The train movement between two stations is drawn as a straight line. So the train diagram is supposed to be equivalent to the timetable in this field.

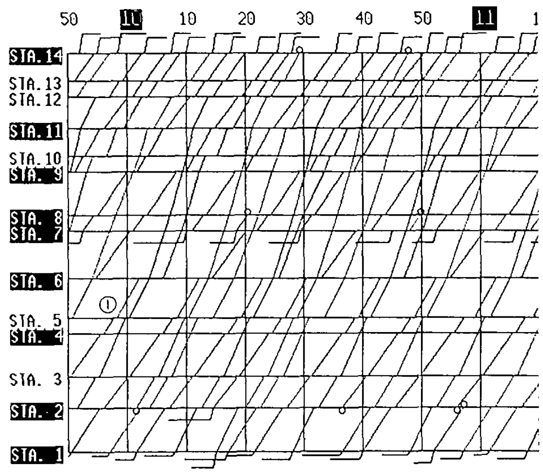

Figure 1. An example of train diagram.

On the train diagram, the experts break the problem down into relatively simple subproblems, and they select and solve a subproblem one after another. Solving the subproblem involves the following two tasks. One is simulation of train movements in a subsystem. The other is local scheduling in which the timetable of simulated trains is set and adjusted. By repeatedly selecting and solving a subproblem, a complete timetable can be constructed incrementally. The experts can efficiently carry out this repetition to prepare a complete timetable. This is the core expertise for problem solving in this case, and it has two important features as follows:

(1) The experts manually simulate and redraw train movements on the train diagram. It is a different way from the computer simulation technology. Since the experts require the arrival and departure time of each train at each station in their simulation, the discrete-event simulation technology is applicable for it. In the discrete-event simulation, the event means the arrival or departure of a train at a station, and is supposed to be a unit of simulation. The experts, however, are used to handling not an individual event but events in a subproblem together, because such a group of related events is more helpful than an individual event to understand train movements in their mind. Therefore, they consider such a group of related events as a unit in their simulation.

(2) The experts select and solve a subproblem in an intelligent manner, while the event is handled by incremental steps of a simulated clock in the discrete-event simulation. Furthermore, they consider only the necessary part of problem space. In restoration, for example, the experts select and solve only subproblems which are related to the delayed train operation, and never consider subproblems including the train being operated on time. In the case of conventional discrete-event simulation, however, all events in a certain period are handled sequentially according to the order of time. Therefore, the movement of trains which operate on time is also simulated.

3 Problem solving architecture

We propose a new problem solving architecture for transportation scheduling, especially for railway systems. The architecture is designed for knowledge based integration of simulation and scheduling; this is based on the observation of human experts' reasoning processes described in the previous section.

In our architecture, we suppose that transportation systems are composed of three types of agents — stops, vehicles, and a control center. The agents are supposed to be objects which contain data and procedure for their own operation, and can communicate with each other to obtain information about each other's state. Transportation systems usually contain many stops and vehicles but only one control center. The stop is a place where the vehicle can stop or pass. Stops are connected with each other by a route which has its own direction and maximum capacity. The vehicle moves along the route, and the control center manages the whole system according to the scheduled timetable. Our problem is to prepare a timetable for both planning and restoration.

To emulate human experts' problem solving processes, we design the system architecture consisting of four components shown in Fig.2 — partial simulation, basic command, tactical knowledge, and strategic knowledge. The partial simulation and basic command are employed for simulation of train movements in a subsystem. Tactical knowledge is employed for local scheduling, and strategic knowledge is employed to manage the order of solving subproblems.

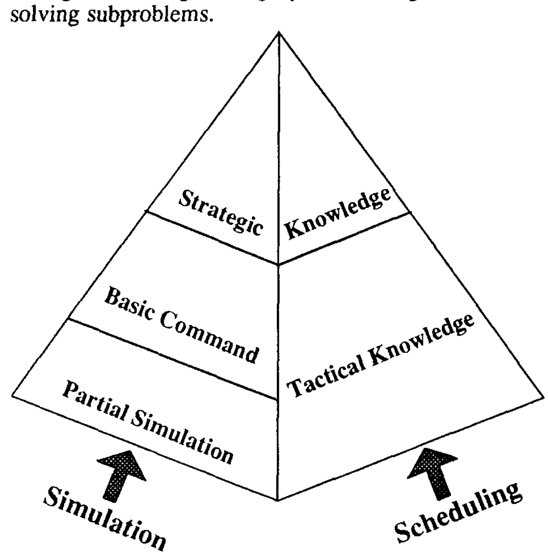

Figure 2. Our problem solving architecture.

3.1 Partial Simulation

The experts are used to handling not an individual event but some related events together in their simulation. We call this partial simulation. Partial simulation must be defined for each system. In railway systems, we define partial simulation as the handling of arrival and departure of a train at a station, such as a_1 and a_2 in Fig.3. If any overtaking is involved, the partial simulation is the handling of arrivals and departures of both the passed train and the passing train, such as d_1, d_2, d_3, and d_4 in Fig.3. Consequently, there are four partial simulations, A, B, C, and D, in Fig.3.

3.2 Basic Command

In our architecture, we suppose that one partial simulation is executed by receiving one basic command. The agent which receives the basic command judges whether partial simulation is feasible to execute or not through information exchange. If the partial simulation is judged feasible, then it is executed, and the result is sent back. The basic command also must be defined for each system.

In railway systems, we suppose that the control center sends a basic command to a station (called Receiving-Station from here on) and receives its result. And the partial simulation for that basic command is executed at a station (called Executing-Station from here on). We define three basic commands as follows:

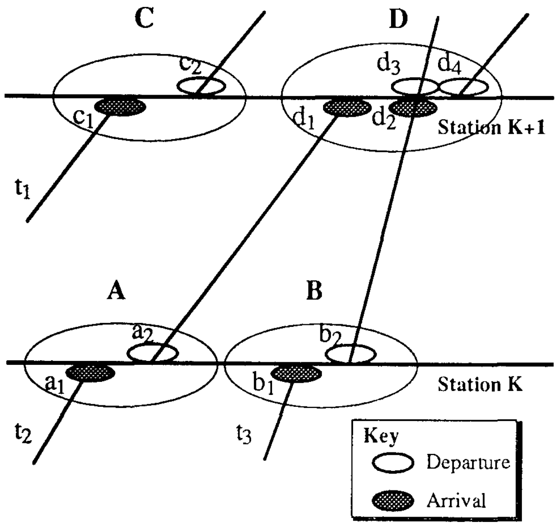

Figure 3. An example of partial simulation.

<u>First Command:</u> The first command is employed to make the train departing from the Receiving-Station move to the next station. For example, when the first command is sent to station K in the conditions shown in Fig.4, Receiving-Station K picks out train t_2 to move to the next station and partial simulation B, which includes train t_2, is selected to be executed at station K+1. The Executing-Station is therefore the station next to the Receiving-Station for this command. Partial simulation B is feasible to execute if the following conditions are satisfied.

(1) Partial simulation B is the next one to be executed at the Executing-Station.

(2) All trains included in partial simulation B can arrive at the Executing-Station, that is, they have departed from the Receiving-Station or can start at the Executing-Station.

This command has three results — YES, NO, and CONDITIONAL-YES. YES means that the partial simulation is executed, and the next possible command for this result is first one to the Executing-Station. NO means that the partial simulation is not executed, and the next possible command is either second one to the Receiving-Station or first one to the station preceding to the Receiving-Station. CONDITIONAL-YES is returned when condition (1) is not satisfied, and the next possible command is second one to the Receiving-Station.

<u>Second Command:</u> The second command is employed to execute the next partial simulation at the Receiving-Station. For example, when the second command is sent to station K+1 in the conditions shown in Fig.4, partial simulation B is executed at station K+1. The Executing-Station is therefore equal to the Receiving-Station for this command. Partial simulation B is feasible to execute if the same condition as condition (2) for the first command is satisfied.

This command has three results — YES, NO, and CONDITIONAL-YES. YES and NO have the same meanings as for the first command. The next possible command for YES is either first one or second one to the Receiving-Station, while that for NO is second one to either the station next or preceding to the Receiving-Station. CONDITIONAL-YES means that the partial simulation for the second command cannot be executed, but the last arrival and departure of the Receiving-Station operate on time. This result is an additional one, and the control center sends the third command to the Receiving-Station whenever it receives this result.

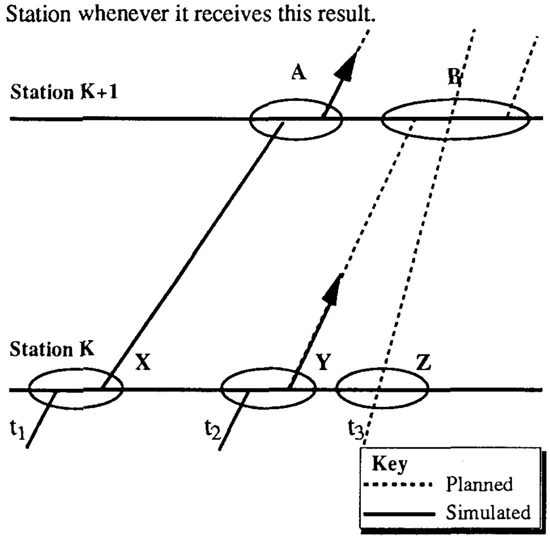

Figure 4. The relation of partial simulation and basic command.

<u>Third Command:</u> The third command is employed to start the simulation of train movement at the station preceding to the Receiving-Station. For example, when the third command is sent to station K+1 in the conditions shown in Fig.4, Receiving-Station K+1 picks out train t_3 to start the simulation, and partial simulation Z, which includes train t_3, is selected to be executed at station K. The Executing-Station is therefore the station preceding to the Receiving-Station for this command. Partial simulation Z is feasible to execute if the following conditions are satisfied.

(1) Partial simulation Z is the next one to be executed at the Executing-Station.

(2) Trains included in partial simulation Z are not simulated.

This command has two results — YES and NO. YES and NO have the same meanings as for the first command. The next possible command for YES is either first one to the Executing-Station or second one to the Receiving-Station, while that for NO is second one to the Executing-Station.

Finally, the list of the next possible commands for each result of three basic commands is given in Table 1.

First command:

result	next possible command
YES	first command to EX
NO	second command to RE
	first command to preceding to RE
COND	second command to RE

Second command:

result	next possible command
YES	first command to RE
	second command to RE
NO	second command to next to RE
	second command to preceding to RE
COND	third command to RE

Third command:

result	next possible command
YES	first command to EX
	second command to RE
NO	second command to EX

COND: CONDITIONAL-YES
RE: Receiving-Station
EX: Executing-Station

Table 1. The next possible commands for each result.

3.3 Tactical Knowledge

Tactical knowledge describes how to solve a local scheduling problem. In our architecture, we employ if-then rules to represent tactical knowledge. These rules consist of the conditional part of representing local traffic conditions and the procedural part of setting or adjusting the timetable, and they belong to each stop and vehicle. Tactical knowledge must be prepared for each problem.

To develop the two knowledge-based systems described in the next section, we have extracted several types of if-then rules from experts. Figure 5, for example, represents a linguistic expression of if-then rule for setting an overtaking operation. These if-then rules are automatically generated for each station and train from the physical limits or the scheduled timetable by a computer. And they can be modified easily to meet the specific requirements of each railway system and various operating conditions during rush hours or slack periods, since they are defined for each station and train. These rules are therefore suitable for representing domain-specific knowledge about local scheduling.

"**If** the difference in departure times of an express train and a local one is less than T seconds,
Then an express train overtakes a local one at a station."

(a) for each station

"**If** the difference in departure times of an express train and the preceding one is less than T seconds,
Then an express train overtakes the preceding one at a station."

(b) for each train

Figure 5. A linguistic expression of if-then rule.

3.4 Strategic Knowledge

Strategic knowledge describes to which station and in what order the three basic commands should be sent. Using strategic knowledge, one basic command is determined among the next possible commands for each result of the preceding one. By receiving a basic command, partial simulation is executed and then local scheduling using tactical knowledge is performed if necessary. Thus, partial simulation and local scheduling are performed repeatedly in a specific order which is managed by strategic knowledge. Strategic knowledge, therefore, plays a dominant role in emulating human experts' reasoning processes. Strategic knowledge belongs to the control center and must be defined for each problem.

Problem solving processes based on our architecture are summarized below. The control center has strategic knowledge and uses it to send basic commands to the station. The station receiving a basic command exchanges information with other stations and trains, and judges whether partial simulation can be executed or not. It then returns the result to the control center. When partial simulation can be executed, it is done, and local scheduling is performed by use of tactical knowledge if necessary. Thus, partial simulation and local scheduling are performed repeatedly in the same order as done by experts.

The process of building a knowledge-based system using our architecture is as follows. First of all, when the transportation system is given, we can define partial simulation for it. Once partial simulation is defined, we can investigate conditions for executing partial simulation, and select and execute it interactively. From this work, we can define basic commands. Using basic commands, experts execute a partial simulation one after another and actually solve a local scheduling problem in a trial-and-error manner. In this process, we can extract tactical knowledge and strategic knowledge. In this way, starting from the first model in which only partial simulation is defined, the second model defining basic commands is developed. Tactical knowledge and strategic knowledge can be defined using the second model. This is due to one of the most important features of our architecture — all four components are independent and hierarchically arranged, as shown in Fig.2.

4 Knowledge-based systems

We have developed two knowledge-based systems, DIAPLAN and ESTRAC-III. DIAPLAN and ESTRAC-III are designed to support experts in planning and restoration, respectively. Both systems employ partial simulation and basic commands described in the previous section, for simulation of train movements in a subsystem. And we have extracted tactical knowledge and strategic knowledge through interviews with experts and from case studies of planning and restoration, with the cooperation of certain railway companies. They are both written in the computer language C and run on the 32-bit engineering workstation.

4.1 DIAPLAN

DIAPLAN prepares a complete timetable from given initial conditions for a train, such as a starting station and time, a terminal, and a type. In this system, local scheduling means setting a timetable of simulated trains at the station next to the Executing-Station. The critical point of local scheduling is setting an overtaking operation, because experts perform it considering the sum of loss waiting time of passed and passing trains and the number of passengers at each station. The experts also perform partial simulation and local scheduling repeatedly considering the order of setting an overtaking operation.

DIAPLAN employs tactical knowledge to set an overtaking operation, which is represented by two sets of if-then rules, as shown in Fig.5 (a) and (b), respectively. One set belongs to the station and represents the minimization of loss waiting time. The other set belongs to the train and represents the specific requirement arising from the number of passengers at each station. Table 2 shows the list of a basic command which is determined among the next possible commands by strategic knowledge. Using this table, DIAPLAN performs partial simulation and local scheduling repeatedly to prepare a complete timetable.

We have applied DIAPLAN to a large and complicated railway system operated in the real world. Figure 1 shows its train diagram from 9:50 to 11:10. This line operates six types of trains with an average total of 40 trains running in both directions outside the rush hours and 700 trains in a day. Trains can overtake at nine (displayed inversely) of the 14 stations. To improve clarity, the down-bound trains are not shown in this figure. For several initial conditions, it has been confirmed that DIAPLAN can prepare a complete timetable within a practical computation time, for example, a timetable of 50 trains in 3 s. And DIAPLAN has proven to be able to prepare the same timetable as that of human experts preparing manually.

First command:

result	next command
YES	first command to EX
NO	first command to preceding to RE
COND	second command to EX

Second command:

result	next command
YES	first command to RE
NO	second command to preceding to RE

Table 2. The next command for each result in DIAPLAN.

4.2 ESTRAC-III

ESTRAC-III prepares a rescheduling plan for the working timetable in the case of disturbed train traffic, such as arrival and departure delays. In this system, the local scheduling means adjusting timetable of simulated trains. Tactical knowledge about adjusting timetable is divided into two sets of if-then rules. One set belongs to the train and governs shifting an overtaking operation, changing the linkage of trains at the terminal, switching the timetable of two trains, and cancelling a train service. The other set belongs to the station and covers shifting an overtaking operation, changing the platform to be used, and adding a train.

We have applied ESTRAC-III to a large and complicated railway system. For typical disturbances, it has been confirmed that ESTRAC-III prepares a rescheduling plan in 5 s, which includes many different methods of schedule adjustment in combination. And its plan is functionally equivalent to that of human experts working manually. Figure 6 and 7, for example, show the train diagram of simulation result and a rescheduling plan by ESTRAC-III, respectively, in the case that train 1 is 10 minutes late departing from station 5 in Fig.1. Table 3 shows the complete list of schedule adjustments for the rescheduling plan. Applying these schedule adjustments, total delay time (i.e., the sum of delays of each train at each station) is reduced to 4:19:10 from 8:30:40, which is employed by dispatchers to evaluate a rescheduling plan in the first approximation.

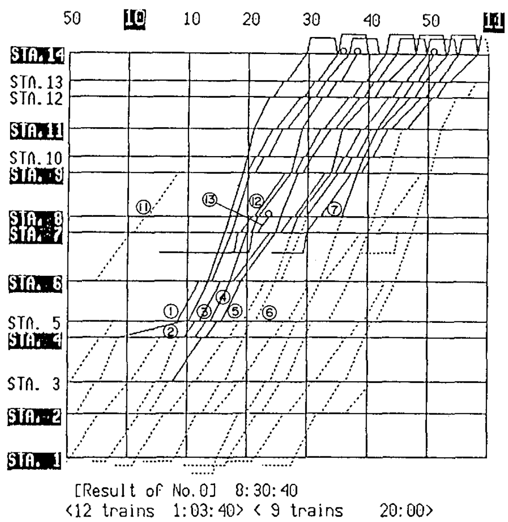

Figure 6. A train diagram for simulation result.

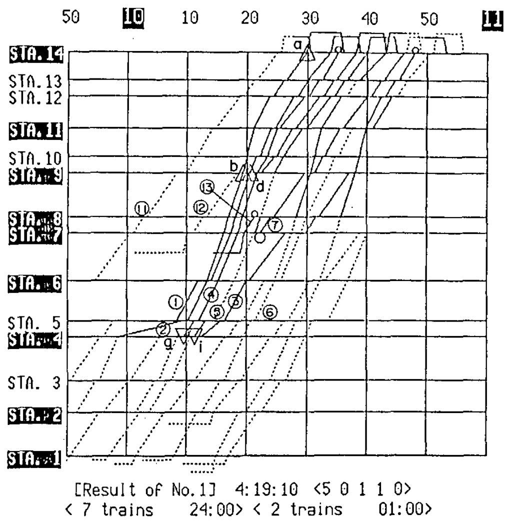

Figure 7. A train diagram for a rescheduling plan.

Shifting overtaking **b** from station **7** to **9**
Shifting overtaking **d** from station **7** to **9**
Shifting overtaking **a** from station **9** to **14**
Changing the linkage of trains **11** and **2** at station **14**
Shifting overtaking **g** from station **6** to **4**
Shifting overtaking **i** from station **7** to **4**
Switching train **3** and train **7** at station **7**

Table 3. Results of schedule adjustments.

5 Conclusion

This paper describes a new problem solving architecture for transportation scheduling. The architecture is designed to emulate experts' problem solving processes in which both simulation and scheduling are integrated in an intelligent manner. Using our architecture, we have developed two knowledge-based systems for railway scheduling, DIAPLAN and ESTRAC-III. Both systems have been applied to large and complicated railway systems operated in the real world and their effectiveness has been proven. DIAPLAN is now scheduled to be used in a certain railway company in the first half of 1991, and ESTRAC-III is also scheduled to be used in another railway company in 1993.

Through the experience of building two knowledge-based systems, it has been confirmed that our architecture can successfully emulate experts' problem solving processes of knowledge based integration of simulation and scheduling. We believe that our architecture can be applied to other transportation systems, and this will be a future subject of ours.

References

[1] S.Araya, K.Abe, and K.Fukumori, "An optimal rescheduling for online train traffic control in disturbed situations," Proc. 22nd IEEE Conference on Decision and Control, pp.489-494, 1983.

[2] S.Araya and K.Fukumori, "ESTRAC-II: An expert system for train traffic control in disturbed situations," Proc. 6th European Conference on Artificial Intelligence, pp.23-32, 1984.

[3] P.Lévine and J.-C.Pomerol, "Railcar Distribution at the French Railways," IEEE EXPERT, Vol.5, No.5, pp.61-69, 1990.

[4] K.G.Kempf, "Manufacturing Planning and Scheduling: Where We Are and Where We Need To Be," Proc. 5th CAIA, pp.14-19, 1989.

[5] M.S.Fox, "AI and Expert System Myths, Legends, and Facts," IEEE EXPERT, Vol.5, No.1, pp.8-20, 1990.

[6] K.Komaya and T.Fukuda, "An Expert System for Train-Traffic Control," MITSUBISHI ELECTRIC ADVANCE, Vol.43, pp.25–28, 1988.

[7] K.Komaya and T.Fukuda, "ESTRAC-III: An expert system for train traffic control in disturbed situations," IFAC Control, Computers, Communications in Transportation, pp.147–153, 1989.