

# Identification of causes of delays in urban railways

A. Yamamura<sup>1</sup>, M. Koresawa<sup>1</sup>, S. Adachi<sup>1</sup> & N. Tomii<sup>2</sup>

<sup>1</sup>*Tokyo Metro Co. Ltd., Japan*

<sup>2</sup>*Chiba Institute of Technology, Japan*

## Abstract

In urban railway lines, it is usually observed that once a small delay happens, it widely spreads out to many other trains. There is a compelling need to identify small delays that very often happen and influence widely and to take an effective countermeasure to decrease those delays by adjusting timetables for example. In this paper, we introduce an algorithm to identify delays that very often happen and influence widely from daily train traffic record data. The key idea of the algorithm is to seek for the causes of delays by tracing back a critical path on a PERT network constructed from train traffic record data. We show the details of the algorithm together with our experience of its application to real world data.

*Keywords: delay, urban railways, PERT, critical path, train traffic records.*

## 1 Introduction

A lot of attentions have been paid recently for the small delays which chronically happen in urban railways. In fact, in the three major cities in Japan (Tokyo, Nagoya and Osaka), troubles whose duration was less than 10 minutes happened almost 2,000 times in 2007 and the percentage of such troubles was more than 60% [1]. In [1], it is shown that the major causes of such small difficulties were rescue of sick passengers, problems at level crossings, congestion and so on.

During the busiest hours in urban railways, a lot of trains are operated densely and once a delay occurs, even if it is small, the delay easily propagates to other trains. The delay that happens first is called a primary delay and the delays caused by the primary delay are called secondary delays.

There are two choices to prevent delays from spreading out. One is to elaborately make timetables so that delays of trains do not propagate. In other



words, to make timetables more robust. In order to get a robust timetable for urban railways where trains are running densely, we should carefully determine dwell times, intervals between trains and running times [2].

The other is to try to avoid occurrence of primary delays. It is very difficult or almost impossible to avoid all the primary delays. But it is considered to be very effective if we could reduce primary delays which very often happen and give wide influence to many other trains. The first thing we have to do is to identify such primary delays. If we can identify such primary delays, we could take effective countermeasures to reduce those primary delays by modifying timetables (to adjust dwell times, to give some supplement to running times, to adjust buffer times etc.), improving deployment of station staff, negotiating with neighboring railway companies with which direct operation of trains are done and so on.

In this paper we focus on an algorithm to identify primary delays which very often happen and give wide influence to many other trains using train traffic record data.

## 2 Urban railways in Japan and propagation of delays

### 2.1 Urban railways in Japan

There is a massive amount of demands for railways in big cities in Japan. For example, in Tokyo area (the capitol of Japan), about 38 million people use railways a day in average [3]. In order to satisfy such a big demand, trains are operated very densely. In many railway lines in Tokyo, trains which consist of eight to ten (sometimes even 15) cars are running every two to three minutes on a double track line. This means that 20 to 30 trains are running per hour per direction (Figure 1 shows a typical timetable of a railway line in the Tokyo area. As you can see, trains are running every couple of minutes on this line).

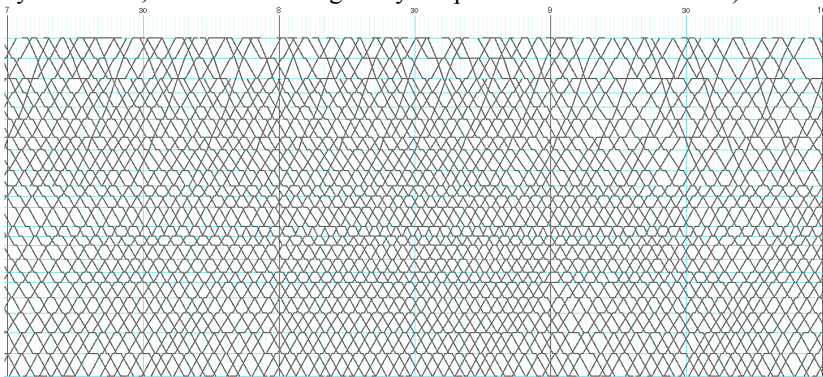


Figure 1: Diagram of an urban railway line (a part – three hours).

Still, trains and stations are very congested as shown in Figure 2. Sometimes, congestion rates of trains during peak hours exceed 150% or much more.



Figure 2: Trains and stations are congested.

## 2.2 Propagation of delays

As described earlier, in urban railway lines, small delays very often happen. The main causes of the small delays are:

- (1) A passenger becomes sick and in order to rescue him/her, departure of the train is delayed.
- (2) Due to congestion, something (a bag, an umbrella etc.) is caught by doors of a train and in order to release that stuff, all the doors of the train are opened again. Thus, departure of the train is delayed.
- (3) More passengers than expected get on/off a train and the departure is delayed because of increase of dwell times.

In a railway line where trains are operated very densely, if one train is delayed, the train's delay gradually increases as shown in Figure 3 (Figure 3 is the *chromatic diagram* we have developed to view situation of delay propagation

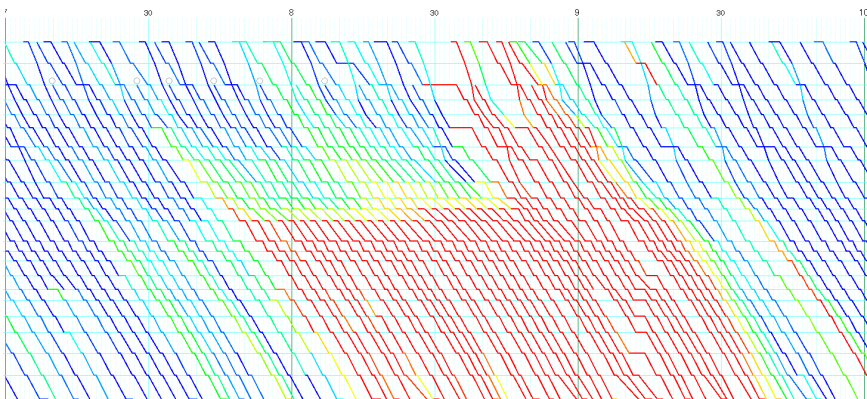


Figure 3: Delays gradually increase and spread out.

at a glance [2]. Each train segment is coloured reflecting its delay. If the train is on time, the colour is indigo. The colour becomes from indigo to blue, green, yellow, orange and red as the delay increases). From Figure 3, we can learn that how a delay of a train increases and how the delay is propagating to other trains. This is mainly because of the reason (3). That is, if one train is delayed, the interval at the next station between the train and its preceding train becomes larger and more passengers than usual are waiting for the train. Thus, the dwell time of the train tends to become longer and the delay increases. This phenomenon is repeated and the delay gradually increases like a snow-ball.

At the same time, delays easily propagate to other trains and the propagated delay again propagate to other trains (these delays are called knock on delays). We would like to show the mechanism of delay propagation using Figure 4. In Figure 4, thin lines depict planned times and bold lines depict actual times. In order to realize dense train operation, dwell times of trains are set to be necessary minimum. In many cases, dwell times of trains are set to be 20 to 60 seconds [4]. Thus, if there happens something unexpected as described above, dwell times of trains easily increase. The succeeding train cannot arrive at the station because the track is still occupied by the preceding train. This phenomenon is repeated by succeeding trains at subsequent stations and delays propagate to many other trains. In addition, because trains are compelled to stop before it arrives at a station, intervals of trains become larger. Thus, as shown in Figure 4, delays gradually spread out to the whole network.

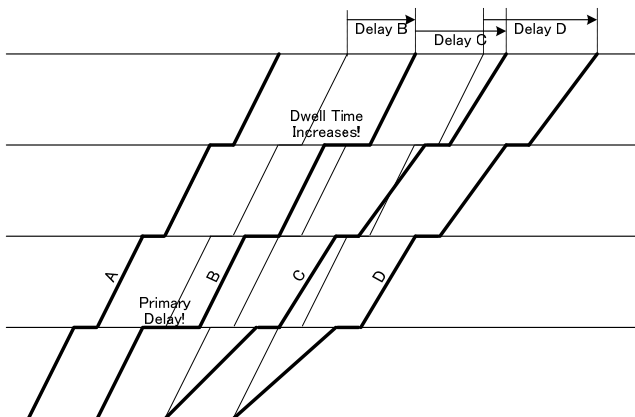


Figure 4: Mechanism of propagation and increase of delays.

### 3 Train traffic record data

In our research, we use train traffic record data. Train traffic record data are collected daily by a train operation control system from track occupation data. Arrival and departure times of all the trains at all the stations are contained. The granularity of arrival and departure time is one second. Thus, we can grasp all the behaviour of all the trains for a long time in detail.

## 4 Related works

Conte and Shöbel [5] introduced a graphical method to identify dependencies among delays. Arrival and departure delays of trains are associated with random variables. In contrast, our approach is an algorithmic one and we do not assume probabilistic distribution of delays. Flier *et al.* [6] presented algorithms that efficiently find systematic dependencies in large-scale railway delay data. But their algorithm is not network oriented but deals with the dependencies only in a station. Van der Meer *et al.* [7] presented a method to predict a train running time using track occupation data. But they do not deal with dependencies among delays. Cule *et al.* [8] introduced an algorithm to find patterns in the delays by using an episode mining technique. But they are not interested in finding the causes of secondary delays in a dense traffic.

## 5 Identification of a cause of a delay – basic idea

### 5.1 Identification of a cause of a delay

In this section, we introduce the basic idea of the algorithm to identify the primary delay which is considered to be a cause of a given secondary delay. By applying the algorithm repeatedly for train traffic record data of a long term, we can know primary delays which very often happen and cause secondary delays for many other trains.

### 5.2 Principles of train operation when trains are delayed

If a train is delayed, a driver tries to regain the delay as much as possible. Thus, if a train is delayed:

- An actual running time between two stations might be smaller than the running time of the planned timetable because some supplement is usually included in the planned running time. The lower bound of actual running times must be the technically minimum running time.
- An actual interval between two consecutive trains might be smaller than the interval in the planned timetable because some buffer time is usually included in the interval between trains. The lower bound of actual train intervals must be the minimum headway.
- An actual dwell time also might be smaller than the dwell time in the planned timetable. Factors which determine the lower bound of actual dwell times are, time needed for passengers to get on and off, time needed for station staff to give an indication to a train to depart and so on.
- An actual turn back time also might be smaller than the turn back time in the planned timetable. The lower bound of actual turn back times is determined by the time necessary for crew to take over, the time for all passengers to get off and so on.



### 5.3 Estimating causality

In order to identify a cause of a delay, we need to find causality and trace back the causality toward the origin. In this paper, we introduce the following ideas to estimate an existence of causality.

Case 1: If the following conditions hold, we regard the delay is caused by the departure delay of the preceding train and we trace back causality to the departure of the preceding train.

- The delay is an arrival delay.
- The interval between the delayed arrival and the departure of the preceding train is equal to the minimum headway.
- The departure of the preceding train is delayed.

Case 2: If the following conditions hold, we regard that the delay was caused by the departure delay of the train and trace back causality to the departure.

- The delay is an arrival delay.
- The actual running time of the train from the previous station is equal to the technically minimum running time.

Case 3: If the following conditions hold, we regard that the delay was caused by the arrival delay of the train and trace back causality to the arrival.

- The delay is a departure delay.
- The actual dwell time of the train is equal to a certain threshold.
- Arrival of the train is delayed.

Case 4: If the following conditions hold, we regard that the delay was caused by the arrival delay of the train and trace back causality to the arrival.

- The delay is a departure delay of a train starting from the station.
- The actual turn back time of the train is equal to a certain threshold.
- Arrival of the train at the station is delayed.

Case 5: If the following conditions hold, we regard that the delay was caused by the departure delay of the conflicting train and we trace back causality to the departure of the conflicting train.

- The delay is an arrival delay.
- The actual time interval between the arrival of the train and the departure of a train that uses a conflicting route is equal to the minimum crossover time.
- Departure of the conflicting train is delayed.

Case 6: If the following conditions hold, we regard that the delay was caused by the arrival delay of the conflicting train and we trace back causality to the arrival of the conflicting train.

- The delay is a departure delay.
- The actual time interval between the departure of the train and the arrival of a train that uses a conflicting route is equal to the minimum crossover time.
- Arrival of the conflicting train is delayed.

## 6 An algorithm to identify causes of a given delay

### 6.1 Basic procedures

We first show a basic algorithm to identify causes of a given delay, which we have developed based on the discussions described in section 5.

In a railway line where trains are running very densely and connections between trains are not considered, it is reasonable to think that propagation of delays occurs following physical conditions. If there are no buffer times, delays propagate. So, basically we can conclude that there exists causality between two events if there is a delay and there is no buffer between them. Thus, we can reach the cause of a delay by tracing back causality from a primary delay. An image of this procedure is shown in Figure 5.

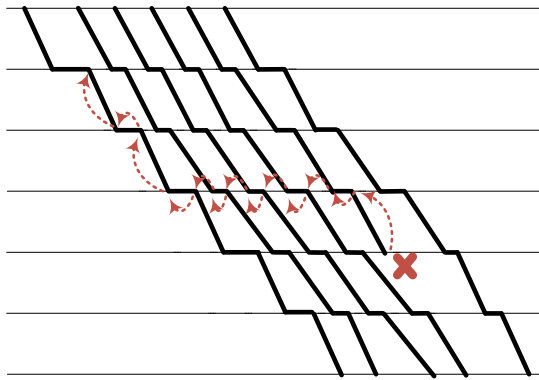


Figure 5: Tracing back the causality – an image.

In order to systematically trace back causality, we introduce a network which we call “a train operation record network.” A train operation record network (TORN) is a directed acyclic graph, which can be considered to be a sort of PERT (Program Evaluation and Review Technique) network. PERT based modelling of train operation was first introduced by Abe and Araya [9] to simulate train operation very quickly by calculating the longest paths to nodes and augmented by Tomii and Ikeda [10]. Tomii *et al.* [11] introduced an efficient rescheduling algorithm focusing on critical paths of the PERT network. In these works, PERT based modelling was used to predict future train operation. But in this paper, we use the model to express the past data of train operation to analyse train operation in the past.

A TORN is made from train traffic record data of one day. Nodes of a TORN are arrival events and departure events of trains. Arcs express chronological dependencies between nodes. Several types of arcs exist as described later. Each arc has a weight which indicates duration of time between its start node and its end node.

We can identify the cause of a given delay by tracing back critical paths on a TORN from a primary delay whose origin we want to seek for. A critical path is a path which contains no supplement in its arcs (we describe the details later).

## 6.2 Train operation record network

We will describe the details of Train Operation Record Network.

- Departure node: One departure node exists corresponding to a departure event of a train. A departure node contains a train number, station name and an actual departure time.
- Arrival node: One arrival node exists corresponding to an arrival event of a train. An arrival node contains a train number, station name and an actual departure time.
- We prepare an arrival node and a departure node for a passing train as well.
- Train running arc: A train running arc exists from a departure node of a train to an arrival node of the train of the next station. Its weight is a technically minimum running time.
- Dwell arc: A dwell arc exists from an arrival node of a train at a station to a departure node of the train of the same station. Its weight is the minimum dwell time. If the train is a passing train, the weight is 0.
- Turn back arc: A turn back arc exists from an arrival node of a train at a station to a departure node of the train at the station if the train turns back at the station.
- Headway arc: A headway arc exists from a departure node of a train at a station to an arrival node of the succeeding train at the station. Its weight is a minimum headway between trains.
- Crossover arc: A crossover arc exists from a departure (or an arrival) node of a train at a station to an arrival (or a departure) node of another train at the station if there is a crossover relationship between two trains.

We will show an example of a TORN in Figure 7, which was constructed from the diagram of Figure 6. Dotted lines depict planned times and solid lines depict actual times. Characters in nodes in Figure 7 are, a train number, a station name, indication of arrival or departure (“a” means an arrival, “d” means a departure) and an actual time.

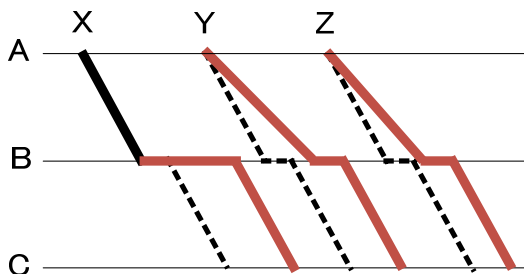


Figure 6: A train diagram.



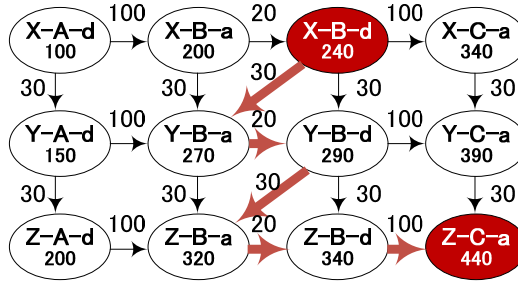


Figure 7: A TORN- train operation record network and a critical path.

### 6.3 Trace back critical paths

If an arc satisfies the following condition, we call the arc is “a critical arc.”

weight of the arc = actual time of the end node of the arc – actual time of the start node of the arc. (1)

If all the arcs of a path are critical arcs, we call the path is “a critical path.”

The aim of the algorithm is rephrased as to trace back a critical path from the node on a TORN which corresponds to the given secondary delay. If we cannot trace back a critical path anymore, we output the current node as the cause of the secondary delay. The bold line in Figure 7 depicts a critical path. This means that by tracing back a critical path from the arrival node of Train Z at Station C, we can learn that the cause of this delay is the departure delay of Train X at Station B.

### 6.4 Problems of the basic procedure

In actual train operation, however, our trial to trace back critical paths by applying the condition of formula (1) does not work well. This is because the condition of formula (1) assumes that all the trains run ideally and this assumption is in a sense too strict. That is:

- Train traffic record data are obtained from a train traffic control system by collecting track occupation data from signaling system. So, usually, there exist some differences between actual arrival / departure times and the data in train traffic record data.
- Technically minimum running times usually contain some supplement because they are round off by a unit of (e.g.) five seconds and they are calculated assuming environments such as voltage of catenary, acceleration and deceleration etc. stricter than actual. So, actual running times are sometimes smaller than technically minimum running times. This is also true for headways and cross over times.
- In case of headways, the situation is a bit more complicated. If a train's departure is delayed, the succeeding train is compelled to stop temporarily

before it arrives at the station because the track is still occupied by the train. This makes actual interval between the train and the preceding train longer than the minimum headway, although the driver had tried to regain a delay so that the train can run with the minimum headway. In this case, we should consider that the arrival delay of the train is caused by the preceding train. So, we should trace back the critical path from the departure node of the preceding train to the arrival node of the train even when the headway arc does not satisfy formula (1).

- As described earlier, dwell times tend to become longer if the train is delayed and the interval is larger than planned. In this case, it is natural to consider that the departure delay is caused by the arrival. So, we should trace back the critical path from the departure node to the arrival node even when the dwell arc does not satisfy formula (1).

## 7 Revised algorithm to identify causes of delays

Based on the discussions in 6.4, we revise the basic algorithm as follows:

- Instead of using technically minimum values, we set the weights of train operation record network based on train traffic record data. More precisely, we calculate  $x$  percentile of the running time, the headway, the crossover time and the turnback time for each train and set it as a weight of the corresponding arc.
- We relax the condition of formula (1) as follows:

$$\text{weight} \leq (\text{actual time of the end node of the arc} - \text{actual time of the start node}) + \text{offset.} \quad (2)$$

In the following, we denote an offset for running time as  $\alpha$ , an offset for headway and cross over time as  $\beta$  and an offset for turnback time as  $\gamma$ .

- For dwell times, we do not consider formula (2). Instead, we set a threshold  $y$  and if a dwell time is smaller than the planned dwell time +  $y$ , we regard the dwell arc is on a critical path.

Concepts behind these ideas are as follows:

- Running times: A weight of a running arc should be the smallest one among actual running times in a certain period of time. But in order to avoid an irregular value, we use  $x$  percentile. This is the same for turn back times, headways and cross over times.
- $\alpha$ ,  $\beta$  and  $\gamma$  are offsets to compensate difference between actual values and train traffic record data.
- $\beta$  has another meaning at the same time. We set  $\beta$  as a value considering an increase of train interval when the train was compelled to stop before it arrives at a station. Hence,  $\beta$  is usually larger than  $\alpha$  and  $\gamma$ .
- If a dwell time is too large we consider there must be some trouble at the station to increase the dwell time. We make the decision by using the threshold  $y$ .



## 8 Numerical experiments

We have conducted numerical experiments using real world train traffic record data of a railway line in Tokyo area. The track length is about 30 km and about a half of it is underground and the other half is on the ground. Several types of trains are operated; one regular train and several types of express trains which pass some stations. Tracks are double and the number of stations is 23. During the busiest hours, 27 trains are operated per hour per direction. The number of passengers is about 1,300 thousands a day in average. We used the data of 2008 because at that time small delays very often happened.

We show a result for one day in Figure 8. Bold lines are critical paths. We set  $x$  as 10 (percentile),  $y$  as 60 (seconds) and  $\alpha$ ,  $\beta$ ,  $\gamma$  as 15, 15 and 30 (seconds) respectively. We applied our algorithm to train traffic record data of half a year and learned that the algorithm can output primary delays which very often happen and give significant influence to many other trains. Some of them were already notorious among the timetable planners but our algorithm showed the seriousness of their influence quantitatively and also discovered primary delays which give wide influence but not so well recognized by them.

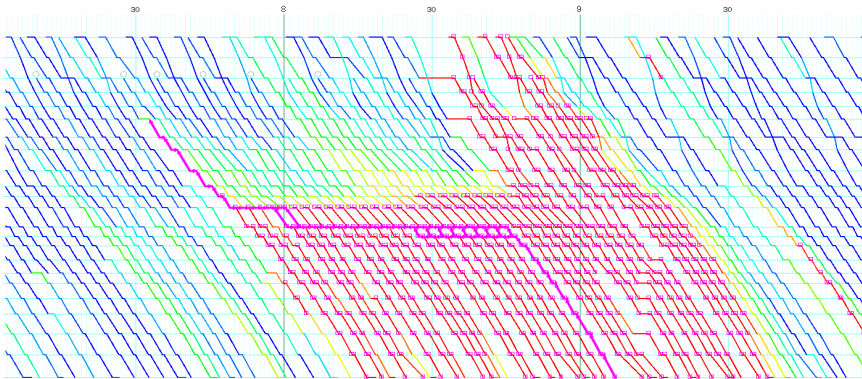


Figure 8: An example of numerical experiment for one day.

## 9 Conclusion

We have introduced an algorithm to seek for primary delays which very often happen and are causes of a lot of secondary delays using train traffic record data. The algorithm is based on an idea to trace back a critical path on a Train Operation Record Network, which is a kind of PERT network constructed from daily train traffic record data.

The results of our experiments are very encouraging. We have succeeded to find out the primary delays of trains which were already notorious among timetable planners. In addition, our algorithm nominated primary delays which had not been recognized by them.

Our future work should be to validate the results of the algorithm based on statistical approach.

## Acknowledgement

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C) 24510199.

## References

- [1] Report by Ministry of Land, Transport and Infrastructure: <http://www.mlit.go.jp/common/000047882.pdf>
- [2] Ushida, K., Makino, S., Tomii N.: Increasing Robustness of Dense Timetables by Visualization of Train Traffic Record Data and Monte Carlo Simulation, *WCRR2011-World Congress on Railway Research*, Lille, France, 2011.
- [3] Annual report of urban transportation 2010, Institution for Transport Policy Studies, Tokyo, 2012.
- [4] Tomii, N.: *How to make timetables of urban railways* (in Japanese), Ohmsha Publishing, Tokyo, 2012.
- [5] Conte C., Shöbel, A.: Identifying dependencies among delays, *Second International Seminar on Railway Operations Modeling and Analysis: RailHanover*, 2007.
- [6] Flier, H., Gelashvili, R., Graffagnino, T., Nunkesser, M.: Mining Railway Delay Dependencies in Large-Scale Real-World Delay Data, *ROBUST AND ONLINE LARGE-SCALE OPTIMIZATION*, Lecture Notes in Computer Science, 2009, Volume 5868/2009, 354-368, 2009.
- [7] van der Meer, D., Goverde, R., Hansen, I.: Prediction of Train Running Times and Conflicts using Track Occupation Data, *12th WCTR-World Congress of Transportation Research*, Lisbon, Portugal, 2010.
- [8] Cule, B., Goethals, B., Tassenoy, S. Verboven, S.: Mining Train Delays, *Proceedings of the 10th international conference on Advances in intelligent data analysis X*, 113-124, 2011.
- [9] Abe, K. and Araya, S.: Train Traffic Simulation Algorithm using Longest Path Method (in Japanese), *Transactions of Information Processing Society of Japan*, Vol.27, No.1, 1986.
- [10] Tomii, N. and Ikeda, H.: A Train Traffic Rescheduling Simulator Combining PERT and Knowledge-Based Approach, *ESS'95, European Simulation Symposium*, Elrangen, Germany, Nov. 1995.
- [11] Tomii, N., Tashiro, Y., Tanabe, N., Hirai, C., Muraki, K.: Train Rescheduling Algorithm which minimizes Passengers' Dissatisfaction, in Moonis. Ali *et al.* eds., *Innovations in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence 3533*, Springer, 2005.

