Constraints Based Effective railway Rescheduling System

Hiroyuki SHIMIZU, Hirohisa FUJIMOTO
JR East Japan Information Systems Company
JR East Japan Head Office Bldg 9F 2-2-2 Yoyogi, Shibuya-ku, Tokyo, JAPAN
e-mail : shimizuh@inet.jeis.co.jp

Jyun TAKAHASHI
New Media Research Institute Co., Ltd.

Naotsugu NOZUE
Math-model Research Inc.

Summary

Delays of trains occur owing to accidents, bad weather conditions, vehicle troubles, etc. Trains run according to the departure order of trains at each station (called "SOT") in the scheduled plan. Therefore, a delay of a train delays its succeeding train, which may result in a chain reaction of delays and confusion arises to the whole network finally. The rescheduling needs to adjust SOT and the order of using resources (called "ROT").

Rescheduling involves many factors intricately related. One of major factors is related to resources such as line sections, arrival-departure tracks, cars, drivers, etc. To use these resources, there are many constraints such as safety requirements, performance of cars, labor regulations, etc. Some constraints may break out dynamically among a group of trains which are interrelated by SOT.

A group of trains are also related each other by car rostering. In view of each car, a trajectory of its movement is folded over at each terminal station of trains operated by this car.

As about 5 to 15 trains are operated in an hour from each direction, these trajectories exhibit a chaotic nature.

As a result, a decision on a train at a station may greatly change the future situations of succeeding trains.

At present a train dispatcher carries out a reschedule on the basis of his expertise with the help of a traffic simulator. However, if a large scale disorder occurs, it is difficult for a train dispatcher to predict the train diagram of the future and to suitably reschedule them. For example, a one hour delay may influence over 6 hours in the Shinkansen Network of East Japan Railway Company (JR-East).

To support the decision making process of a train dispatcher in these rescheduling environments, we have been studying the architecture of this decision support system and adopted the constraints-based approach. Implementation of the prototype system CIBERS is based on constraint programming packages provided by ILOG. Case studies have been carried out with real data from the Shinkansen Network of JR-East.

Keyword

Constraint programming, Train traffic rescheduling system, Train operation, Car operation, Driver operation
1. Introduction

Delays of trains occur owing to accidents, bad weather conditions, vehicle troubles, etc. Trains run according to the departure order of trains at each station (called “SOT”) in the scheduled plan. Therefore, a delay of a train delays its succeeding train, which may result in a chain reaction of delays and confusion arises to the whole network finally. The rescheduling needs to adjust SOT and the order of using resources (called “ROT”).

Rescheduling involves many factors intricately related. One of major factors is related to resources such as line sections, arrivals-departure tracks, cars, drivers, etc. To use these resources, there are many constraints such as safety requirements, performance of cars, labor regulations, etc. Some constraints may break out dynamically among a group of trains which are interrelated by SOT.

A group of trains are also related each other by car rostering. In view of each car, a trajectory of its movement is folded over at each terminal station of trains operated by this car.

As about 5 to 15 trains are operated in an hour from each direction, these trajectories exhibit a chaotic nature. (Fig. 1)

As a result, a decision on a train at a station may greatly change the future situations of succeeding trains.

2. Constraint based approach

The algorithmic aspect of this problem belongs to a class of “resource-constrained project scheduling problems” and is known as a hard combinatorial problem. A general branch and bound algorithm for this class was proposed[1], but our problem is too hard to apply this naive algorithm.

To avoid difficulties, an expert system was developed for comparatively simple urban transportation within a one hour time span by trial and error [2]. However, this approach based on trial and error can’t settle a large disturbance in the nationwide Shinkansen Network where many types of trains are operated in a densely and intricately interrelated environment.

Recently, this type of problem has been successfully solved by Constraint Programming based on the constraint propagation mechanism [3]. The programming style of Constraint Programming is different from conventional procedure languages[4].

We have to consider following items.

* Declarative definition of constraints involved in target application
* Objective function
* Searching heuristics based on backtracking
* Reduction of search space by constraint propagation with partial solution

In our prototype system, we adopted ILOG Solver and Scheduler, and developed the following [4].

* The general constraint model of Shinkansen railway traffic in terms of specific constraints to railways.
* The objective function : minimization of the sum of weighted delays
* The search heuristics specialized in railway traffic rescheduling
* Constraints and constraint propagation method for railway traffic
* The GUI system for rescheduling.
* The case study for the Shinkansen of JR-East.
The result of our case study show that our prototype system is practical concerned with quality of rescheduling plan, processing time and GUI.

3. The railway traffic rescheduling system

Constraints are classified into two groups:

- *Hard constraints which must be observed by schedules.
- *Soft constraints which are desirable to be observed but can be changed with some penalty.

3.1 Hard constraints

1. Minimum running time
2. Minimum stopping time
3. Minimum turn back time
4. Minimum time spacing of consecutive departures (Fig. 2)
5. Minimum time spacing of consecutive arrivals (Fig. 2)
6. Minimum time spacing between departure and its successive arrival
7. Line capacity (A train can't use the same line section which has as many trains as its capacity allows at the same time.)
8. Arrival-departure track capacity (Two trains can't use the same arrival-departure track at the same time.)
9. Car rostering (The train can't depart earlier than the arrival time of its preceding train operated by the same car.)
10. Driver rostering (The train can't depart earlier than the arrival time of its preceding train operated by the same driver.)
11. Scheduled departure time (The train can't depart earlier than scheduled time.)

![Fig. 2 – Example of constraint](image)

3.2 Soft constraints

1. Allocation of arrival-departure tracks to original ones
2. No extra stopping besides the original stopping pattern
3. Overtaking at the first possible station
4. No influence on other areas

4. Architecture of CIBERS

4.1 Outline of the system

CIBERS consists of four layers as shown in Fig. 3.

1. 1st layer: ILOG Solver and Scheduler.
   In this layer, the basic constraints are expressed in terms of linear algebra. The constraints and their propagation mechanism are provided by the ILOG system. For example, if any of SOT changes, the earliest departure times for each station will be updated. Similarly, if the upper bound of the total delay time is reduced, the latest arrival time for each station will be updated.

2. 2nd layer: The general constraints model of Shinkansen railway traffic.
   To reduce the computation time, we devised several original constraints and their propagation methods extensively. In this layer, the constraints specialized in railways are added. Some are added dynamically. The constraint propagation is performed by the constraint propagation mechanism specialized in railway traffic. For example, if a train tries to enter into a line section which has as many trains as its capacity allows, the earliest departure time is set to the earliest arrival time among trains inside.

3. 3rd layer: Heuristics of search engine
   In this layer, the heuristics of rescheduling is defined. For example, considering time difference between schedule time and current earliest possible time, the heuristics assists to search optimal SOT, that is, optimal overtakings among trains.

4. 4th layer: Graphical User Interface
   This layer offers a graphical user interface such as a setup of conditions of a simulation and a display of rescheduled train diagram, and summary of simulation.
4.2 Model

A train diagram can be determined in terms of SOT and ROT. A train is represented as the sequence of activities such as "stopping" activity and "running" activity. (Fig. 4)

Each activity requires some resources. As for resource, there exists arrival-departure tracks at the station and line sections between stations. (Fig. 5)

* A "stopping" activity requires an arrival-departure track resource.
* A "running" activity requires a line section resource.

Car rostering is represented as a sequence of trains operated by the same car.
Fig. 5 - The resources

Fig. 6 shows the example of the constraints defined among activities.

Fig. 6 - Example of the constraints
A train consists of both "stopping" activities and "running" activities. To express train operations, we add constraints between two types of activities:

* Constraints are defined between the end event of "running" activity and the start event of "stopping" activity.
* Constraints are defined between the end event of "stopping" activity and the start event of "running" activity.

The constraints of SOT are defined between the start events of "running" activities related to the same line section. The constraints of the arrival-departure track are defined between the end event of the "stopping" activity of a train and the start event of the "stopping" activity of the succeeding train, if the two trains use the same track at the station.

If a train tries to enter into a line section which has as many trains as its capacity, the start event of a "stopping" activity is set to the end event of a "stopping" activity among trains inside.

4.3 Constraint propagation

All activities are connected as a network according to constraints. Therefore, a change to the time of the start event of an activity effects the time of the end event of an incident activity, and repeatedly to other succeeding activities in the network.

For example, as shown in Fig 7, if a train makes a turn-back at a destination station, the following constraints are defined.

1. \( T_{11} + a_1 \leq T_{12} \) (\( a_1 \): Minimum running time between Sta. 1 and Sta. 2)
2. \( T_{12} + b \leq T_{22} \) (\( b \): Minimum turn-back time at Sta. 2)
3. \( T_{22} + a_2 \leq T_{21} \) (\( a_2 \): Minimum running time between Sta. 2 and Sta. 1)

Therefore, a delay of \( T_{11} \) propagates to \( T_{12}, T_{22}, \) and \( T_{21} \).

Fig. 10 shows that train #3001B overtakes train #31B at Furukawa Sta. which is a new passing station by rescheduling.

The SOT at Sendai St. is also changed so that train #3001B departs before train #6735B which is for non-commercial service.

Fig. 11 shows the graph of total delay of the rescheduled SOT.

Changing 2 elements of SOT reduces total delay from 296 minutes to 206 minutes (Table 1).
Fig. 8 - Original SOT of Tohoku line (From Sendai to Morioka)

Fig. 9 - Original delay graph
Fig. 10 - Rescheduled SOT of Tohoku line (From Sendai to Morioka)

Fig. 11 - Rescheduled delay graph

<table>
<thead>
<tr>
<th></th>
<th>Change of “SOT” [sections]</th>
<th>Total Delay [minutes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Rescheduling</td>
<td>Down 0 Up 0</td>
<td>296</td>
</tr>
<tr>
<td>After Rescheduling</td>
<td>2 0</td>
<td>203</td>
</tr>
</tbody>
</table>
5.2 Example 2 : Heavy traffic and 60 min. suspension  
Suspended section : Between Nasu-shiobara Sta. and Shin-shirakawa Sta.
Suspended time : From 1:00 p.m. to 2:00 p.m.

Fig. 12, 13, and 14 show the diagram before rescheduling. Fig. 15 shows the graph of total delay of original SOT.  
Suspension of trains for one hour on the Tohoku line causes delays which amount to total of 4496 minutes and largely influences  
other areas – Jyoetu line and Hokuriku line if SOT is not changed (Table 2).  

Fig. 16, 17, and 18 shows the diagram after rescheduling. Fig. 19 shows the graph of total delay of rescheduled SOT.  
Changing 67 elements of SOT reduces the total delay from 4496 minutes to 2119 minutes and the influences on the other lines are  
minimized (Table 2). Time required to recover is also shortened from 18:00 to 16:20, that is, from 240 minutes to 140 minutes.  

This is an excellent result and also marvelous considering the computation time is within 1 minute.

We discussed the results of case studies with experts of traffic dispatchers and had great confidence in CIBERS.
Fig. 13 - Original SOT of Jo-etsu line

Fig. 14 - Original SOT of Hokuriku line
Fig. 15 - Original delay graph

Fig. 16 - Rescheduled SOT of Tohoku line
Fig. 17 - Rescheduled SOT of Jo-etsu line

Fig. 18 - Rescheduled SOT of Hokuriku line
6. Conclusions

We have made a study of the prototype system CIBERS of railway traffic rescheduling based on Constraint Programming. The results of the case studies with real data from JR-East Shinkansen Network found that we can get practical solutions within about 1 minute, and proved that our prototype system is practical concerning quality of rescheduling plan, processing time and Graphical User Interface.

In view of software development, our approach based on constraint programming was also effective as the development of our system took only 1.5 man years.

We are now planning the advanced version of this system.

Acknowledgements

This research project was carried out in cooperation with JR East Japan Information Systems Company, New Media Research Co. Ltd. and Math-Model Research Inc.. We would like to thank the Shinkansen operations control center of JR-East for making data available and for helpful discussions.

7. References