An Algorithm for Freight Train Driver Rescheduling in Disruption Situations

Keisuke SATO  
Researcher,  
Planning Systems, Transport Information Technology Division

Naoto FUKUMURA  
Senior Researcher,  
Laboratory Head,  
Planning Systems, Transport Information Technology Division

Railway operators adjust timetables and reschedule rolling stock operation plans and crew duties accordingly when operations are disrupted. This paper discusses the problem of rescheduling driver assignment to freight trains after timetable adjustment has been completed. We model the problem as an integer programming problem with set-covering constraints, and solve it using the column generation technique. Numerical experiments using real data have revealed that the method can provide a quality driver-rescheduling plan within an acceptable computing time.

Keywords: crew scheduling, rescheduling, mathematical optimization, set-covering problem, column generation

1. Introduction

Planned schedules for railway operations are occasionally disrupted by external influences or internal failures. In such situations, dispatchers in charge working in the traffic control room first adjust timetables by delaying or even canceling trains, and accordingly reschedule rolling stock operation plans and crew duties (refer to [1]). In Japan, the whole rescheduling process has so far been entrusted to manual work by experienced dispatchers, but there is now a need to speed up the process time for rescheduling and reduce dispatchers' workloads.

This paper presents the results of a study on an algorithm for rescheduling freight driver duties in disrupted situations in Japan on the assumption that timetable adjustment has already been completed, i.e., information on the extent of delays and train cancellations is available. Several algorithms were proposed to cope with the problem, mainly targeting the drivers of passenger trains. Some of these resorted to meta-heuristic approaches by defining an exchange of trains for operation between two drivers or instruction for an extra driver to operate a train as a neighborhood of the search process (see [2 − 3]). In the case of freight trains, the start times of particular drivers' shifts vary day by day, and areas of driving also vary from driver to driver. These variables mean that it is difficult to obtain a feasible solution using such meta-heuristic frameworks; instead, we adopted a mathematical optimization model as applied to airline crew rescheduling [4 − 5] and recently to railway driver rescheduling [6 − 7]. Under this method, we enumerate, for each driver, paths by which the driver returns, from the current place that he/she is on to the crew base that he/she belongs to, by driving or deadheading trains. One path is chosen for each driver so that all trains are covered by one or more drivers' paths. Tens of thousands of paths will be counted if we enumerate them all, which will require extensive amounts of computation time. We therefore use the column generation technique, which requires only a limited number of paths, to obtain a good solution within a reason-
scheduling problem is, as described above, that all legs must be covered by one or more drivers. In addition, a driver’s duty must end at the crew base to which he/she belongs, and the types of locomotives that each driver can operate are limited. There are also several other constraints that make a driver’s duty valid. An assessment criterion with which to evaluate driver-rescheduling plans is the degree of change in duties. The exchange of duties between drivers belonging to the same crew base is also preferable, since communication among different crew bases is thereby avoided.

3. Model description

3.1 Network representation

Given an adjusted timetable and the current sites that the drivers are on as shown in Fig. 2, we enumerate candidate duties from a driver’s current site to his/her crew base and choose one for each driver so that any one of the legs will be included in at least one driver’s duty. To this end, the adjusted timetable and the driver duties are modeled as a network [8]. Each driver’s current site, legs and crew bases are defined as the nodes of the network. Directed arcs are drawn between two nodes if a driver can move from the preceding node to the following one. Figure 3 shows a network developed from Fig. 2. A candidate duty for a certain driver corresponds to a path that begins at his/her current site and ends at the crew base to which he/she belongs.

We next set a cost for each arc and each driver. The cost of an arc for a driver is zero if the connection between nodes is included in his/her planned duty; otherwise, a positive value is set. A different positive value is added to it when the connection causes overtime work for the driver. We then have the cost of a candidate duty, defined as the sum of the cost of the arcs included in the corresponding path. The cost of the planned duty without overtime work caused by a train delay is zero, and a potentially changed duty that is very different from the planned one has a greater cost. We hence adopt the cost of a candidate duty as an evaluation criterion.

3.2 Formulation

We formulate driver rescheduling as an optimization problem based on a set of enumerated candidate duties obtained from the network representation of the disrupted situation. Notations of the data that can be obtained from the network representation are given as follows:

- $I$: the set of legs
- $K$: the set of drivers
- $P^k$: the set of candidate duties of the driver $k \in K$
- $c_{ip}^k$: the cost of candidate duty $p \in P^k$ of the driver $k$

$$a_{ip}^k = \begin{cases} 
1 & \text{if leg } i \in I \text{ is included in the candidate duty } p \text{ of the driver } k \\
0 & \text{otherwise}
\end{cases}$$

We then introduce a decision variable $x$, which indicates whether a driver will take on a candidate duty or not. It is defined as follows:

$$x_p^k = \begin{cases} 
1 & \text{if the driver } k \text{ takes on the candidate duty } p \\
0 & \text{otherwise}
\end{cases}$$

Given these notations, we formulate driver rescheduling as an integer programming representation of a set-covering problem – a well-known approach in mathematical optimization theory [8]. A side constraint is added to it, and the full description of the formulation is given below:

Minimize $\sum_{i \in I} \sum_{p \in P^k} c_{ip}^k x_p^k$ \hspace{1cm} (1)

subject to $\sum_{p \in P^k} a_{ip}^k x_p^k \geq 1 \quad \forall i \in I$, \hspace{1cm} (2)

$\sum_{i \in I} x_p^k = 1 \quad \forall k \in K$, \hspace{1cm} (3)
\[ x_k^p \in \{0, 1\} \quad \forall k \in K \quad \forall p \in P^k. \quad (4) \]

Equation (1) is an evaluation criterion or an objective function that seeks to minimize the total cost of the selected duties. There are three types of constraints, as shown in (2) – (4). Equation (2) is a set-covering constraint indicating that all legs must be contained in one or more duties. It is specified by (3) that each driver has to take on exactly one duty among his/her candidate duties. The constraint applied by (4) restricts the values that the variables can take to either zero or one.

4. Crew-rescheduling algorithm

4.1 Overview

By solving the formulation presented in the previous section, an optimal solution that minimizes the value of the evaluation criterion can be obtained using general mathematical optimization software, if all the candidate duties \( P \) are available. However, the enumeration of all such duties is not reasonable, since they number in the tens of thousands and would take too long to enumerate fully. In fact, none of them are available before rescheduling, since the adjusted timetable and consequently the topology of the network cannot be presumed. We therefore adopt the column generation technique, which alternately enumerates duties and solves our optimization problem with integer constraints relaxed. For a comprehensive understanding of the column generation technique, refer to [9].

4.2 Detailed flow of algorithm

A flowchart of our algorithm is shown in Fig. 4. Section 3 has already outlined the network construction procedure (a), on the condition that planned driver duties and an adjusted timetable are given. In procedure (b), we obtain a set of initial candidate duties for each driver by enumerating paths from the driver node to his/her crew base within the network. The next procedure, labeled as (c), relaxes the constraint by equation (4) of our optimization formulation, and allows the variables to take continuous values so that the column generation technique can be applied.

Procedures (d) – (f) form the framework of the column generation approach. In procedure (d), we optimize our optimization problem on the condition that the candidate duties are not fully enumerated and that the variables are not restricted to zero or one. We can use general optimization software to solve this procedure. After optimization, we make a judgment on whether the evaluation criterion defined by equation (1) may or may not be improved by adding other candidate duties. The details of procedure (e) are described in the next subsection. If duties are found that would decrease the cost, they are added to the candidate duties and re-optimization is executed (procedure (f)).

Procedure (g) is executed after procedure (e) if the total cost of the selected duties cannot be further improved. We solve our optimization problem with the candidate duties enumerated thus far and variables restricted to binary values using optimization software, and the algorithm is then completed. A variable taking a value of one for a certain driver in the solution represents his/her modified duty. If no solution is obtained, the dispatcher in charge is urged to re-adjust the timetable.

4.3 Termination of column generation

The optimization in procedure (d) of Fig. 4 gives us a numerical value called a dual price, which corresponds to the respective constraints given by equations (2) and (3) in addition to the optimal solution itself. Here, we denote the dual price of equation (2) by \( \lambda \), and that of equation (3) by \( \mu \). From the duality theorem of mathematical optimization theory [8], the following search problem indicates the existence of a candidate duty that has not yet been enumerated and may decrease the value of the optimization criterion: for each \( k \in K \),

\[
\text{find} \quad p \in P^k \\
\text{such that} \quad c_p^k - \sum_{i=0}^{k} a_{pi}^k \lambda_i - \mu^k < 0.
\]

This search problem seeks a candidate duty for a driver for which the cost is less than the sum of the dual prices of the legs included in the duty plus the dual price imposed on the driver. If such a path is found, it is added to his/her set of candidate duties; otherwise, the column generation process is terminated since the evaluation criterion will not be improved further.

Recall that the cost of a candidate duty, or a path, is the sum of the cost of the arcs on that path, and that the dual prices correspond to the nodes of the network. Hence, we can reduce the search to a well-known shortest-path problem, and our task is to examine whether the length of the shortest path is less than zero. Moreover, the acyclic nature of the network enables us to solve the problem very effectively using Dijkstra’s algorithm with the nodes selected according to a topological order of the network, even where the arcs of the network have negative costs. The topological order is easily obtained by sorting the nodes by the leg departure times.

The search for candidate paths using Dijkstra’s algorithm can be applied to procedure (b) in Fig. 4. We set all the dual values to zero and enumerate the minimum cost.
duty and others obtained during the execution of the algorithm, even if they have a nonnegative cost. Note that parallel enumeration of different driver duties can be performed, as the shortest path problem for a particular driver is independent of those for other drivers.

4.4 Examples

Here we present a test disruption scenario and show the results of driver rescheduling obtained by the proposed algorithm. Figure 5 shows a situation in which one train is cancelled and the departures of three trains at Station U are delayed by 3.5 hours due to an accident. The planned duty of Driver C is also depicted in the figure, and shows that he/she will miss his/her next train at Station X because of the delay.

We apply the algorithm to this adjusted timetable and obtain the driver-rescheduling plan shown in Fig. 6. This figure indicates the planned duties (before rescheduling) and the modified versions (after rescheduling) of four drivers whose duties are changed as a result of the algorithm’s application. The leg that Driver C is due to drive at Station X is assigned to Driver A, and the two legs due to be operated by Driver A are assigned to Driver B. Driver D moves to Station X by deadheading because of the cancellation of the train that he/she is due to drive, but he/she drives trains as planned from then on.

![Fig. 5 Disruption scenario](image)

![Fig. 6 Driver-rescheduling plan according to the algorithm shown in Fig. 4](image)

5. Computational results

We carried out computational experiments based on real data, i.e., a freight train timetable, and planned driver duties for the area with Japan’s highest frequency of freight train operation, in which 200 such trains are operated per day. We tried rescheduling 102 duties with 413 legs influenced by actual disruptions using a personal computer with a 2.66-GHz CPU, 3.5 GB of RAM and a Windows XP OS. CPLEX 10.1 [10] commercial mathematical optimization software was also used with the algorithm. We set the cost of a duty change higher than that of overtime work to reduce the number of duties altered.

Table 1 shows the rescheduling results for three cases. For each one, the number of delayed trains, the average delay time and the number of cancelled trains are displayed. Also shown are the number of duties changed, the number of duties with overtime work and the sum of overtime work hours in order to give a summary of the rescheduling solution. The computation time is the total elapsed time, including the time taken for input and output of data from/to the hard disk. Only a small number of duties were modified with respect to the number of delayed trains. This is a result of our optimization algorithm’s performance, and is also in part due to the

<table>
<thead>
<tr>
<th>Case no.</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of delayed (time)/cancelled trains</td>
<td>11 (2.0 h)/0</td>
<td>25 (4.0 h)/7</td>
<td>29 (4.5 h)/1</td>
</tr>
<tr>
<td>Number of duties changed</td>
<td>8</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Number of duties with overtime work (total overtime hours)</td>
<td>13 (32.1 h)</td>
<td>10 (13.8 h)</td>
<td>7 (14.2 h)</td>
</tr>
<tr>
<td>Computation time</td>
<td>13.0 sec</td>
<td>15.7 sec</td>
<td>12.3 sec</td>
</tr>
</tbody>
</table>
fact that there are long rest or wait times for drivers in planned duties. The solutions were obtained in less than 20 seconds for all cases; we believe this represents an acceptable speed for dispatchers.

6. Conclusions

This paper presents a study on the problem of freight train driver rescheduling in a case where an adjusted timetable is given in a situation of disruption. We modeled the timetable and the planned driver duties as a network, and created a mathematical optimization formulation for the problem. This was then solved using the column generation technique, which alternately enumerates candidate duties and optimizes rescheduling. Numerical experiments using real data revealed that the method provides a driver-rescheduling plan of satisfactory quality within an acceptable computing time.

In future work, we plan overall optimization of freight train rescheduling for disrupted situations. We have already proposed a rescheduling method for locomotives in [11], and our future interests involve the integration or simultaneous optimization of locomotive and driver rescheduling.

Acknowledgment

The authors would like to thank Mr. Eiki Shigeta and Mr. Hiroshi Kawakami of the Japan Freight Railway Company for their guidance and teaching in the course of this research.

References