The mathematical approach to the train-set routing problem in Metropolitan Subway

K.M. Kim, S.H Hong, D.S Moon
Korea Railroad Research Institute, Uijeongbu, Korea

Abstract

This paper considers subway routing problem. Given a schedule of train to be routed by a railway stock, the routing problem determines a sequence of trains while satisfying turnaround time and maintenance restrictions. Generally, the solution of routing problem is generated from set partition formulation solved by column generation method, a typical integer programming approach for train-set. However, we find the characteristics of metropolitan subway which has a simple rail network, a few end stations and 13 departure-arrival patterns. We reflect a turn-around constraint due to spatial limitations has no existence in conventional railroad. Our objective is to minimize the number of daily train-sets.

We develop two basic techniques that solve the subway routing problem in a reasonable time. In first stage, we formulate the routing problem as a Min-cost-flow problem. Then, in the second stage, we attempt to normalize the distance covered to each routes and reduce the travel distance using our heuristic approach. Applied to the current daily timetable, we could find the subway routings, which is an approximately 14% improvement on the number of train-sets reducing 15% of maximum traveling distance deviation.

1. Introduction

Given a schedule of train to be routed by a railway stock, the train-set routing problem determines a sequence of trains while satisfying turnaround time and maintenance restrictions (Fig. 1). The train-set routing problem, with additional constraints, is typically NP-hard. However, the line of metropolitan subway has a simple rail network, a few end stations and 13 departure-arrival patterns. Generally, the subway train schedule has many alternative routings. This means the routing problem fits well to a deterministic optimization model. In this sense, a mathematical model produces an effective train-set routing. Unlike general train-set routing problems are defined by a weekly time horizon, subway train-set routing problems contains the trains for a one-day time horizon because as a policy the number of routings starting at a station should be equal to the number of routings ending at that station. In addition, one of the vehicles which have been put in operation could be replaced by variety maintenance constraints, so that reason we relax the maintenance conditions. We do not allow that vehicle, returned to maintenance center, go back to the route covering trains.

![Routing diagram of a railway network](image)

Fig. 1: Routing diagram of a railway network
2. Literature review and Problem Classification

Generally, the solution of routing problem is generated from set partition formulation solved by column generation method, a typical integer programming approach for fleet. Kabbani et al.[4] make a model, aircraft maintenance routing problem with every 3 days maintenance restrictions as a set-partitioning problem for determining lines of flying. LOFs(lines of flying) means the origin at the start of a day and the destination at the end of a day for a plane. If the subproblem does not find an appropriate solution, they try to generate solutions swapping either flights at station for maintenance routing. Gopalan et al.[3] study the maintenance routing problem for US Air. Each aircraft needs to have an A-check every 3 days. First, they fix the lines-of-flight during a day and then, find the required routing. Clarke et al.[1] study a aircraft rotation problem which maximizes the total through values subject to acceptable maintenance. Here, through value is measured by desirability of the two immediate connecting flights. They formulate as a traveling salesman problem with side constraints and solve using Lagrangian relaxation and heuristics. Zhao et al.[7] model train set scheduling problem as a TSP and makes algorithm which is based on probabilistic search method. First, they formulate the problem as the network model and choice ordered inspection arcs by random choice. Second, construct the path between inspection arcs using shortest path method. After repeating these operations, find a tour. Schrijver[5] and Forbes et al.[2] present single/multi-commodity-flow formulation to the train routing problem. But they take no thought of the maintenance restrictions.

We classify the train-set routing problem into three problems, based on a characteristic of coupling. In the train-set path problem, a route corresponds to a path. This path has no concern how to connect the trains from the start to the end of schedule period. This problem has been formulated into the set partitioning problem with the column generation. The second class of train-set routing problem is the train-set repetition problem. In this problem, the number of routings starting at a station should be equal to the number of routings starting at that station. The last class is the train-set rotation problem which is called the aircraft rotation problem in aircraft industry. In this problem, all train-sets follow only one route. This problem can be formulated into the traveling salesman problem. Even if the maintenance condition is ignored, it is intractable.

3. An approach to the Subway routing problems

To achieve an improvement over the current implementation, we need to fully exploit key characteristics of the metropolitan railways. We present two stage approaches the train-set routing problem in a reasonable time. In first stage, we formulate the train-set routing problem as a Min-cost-flow problem. Then, in the second stage, we attempt to normalize the distance covered to each routes and reduce the travel distance using our heuristic approach. Our objective is to minimize the number of daily train-sets.

3.1 Network Model

![Fig. 2. Example of Network Model](image)
The train-set repetition problem can be modeled as the minimum cost flow model (See Fig. 2). To describe the network structure of the MCF, we need to split a train into two nodes (departure node, arrival node) and to make following three arcs \((C_{ij}, l_{ij}, u_{ij})\).

- **\(E_1\): Train arc set** represent trains \((0, 1, 1)\)

- **\(E_2\): Turn around arc set** represent the connection of two trains eligible to satisfy turn-around condition \((0, 0, 1)\)

- **\(E_3\): Backward ground arc set** represent the connection of two trains eligible to connect repeatedly \((1, 0, 1)\)

### 3.2 Modified Network Model

Figure 3 shows the feasible routings obtained from the 4 train schedules. In this case, solution B is better than solution A from the viewpoint of normalization operation time and practical turn-around capacity. Therefore, we modify the network model to minimize not only the total number of train-sets but also the operation time of each route if possible to following FIFT (First in First Turnaround) rule.

For example, i train has possible \(n\) candidate trains \((j_1, j_2, ... , j_n)\) to satisfy the turn-around conditions. (Fig. 4) In this case, we modify the cost of turn-around arcs and backward ground arcs.

- **Modified \(E_2\):** To increase the cost of turn-around arcs in turn-around order. Ex) \(c_{ij1}=1, c_{ij2}=2, ... c_{ijn}=n\). The modified \(E_2\) be able to reflect minimizing operation time on each route.

- **Modified \(E_3\):** To change the cost of backward ground arcs into Big-M (Big-M > n) so that modified \(E_2\) have not influence on objective value.
3.3 Mathematical Formulation

The mathematical model based on the minimum cost flow satisfies the Totally Unimodularity properties. So, we can be obtained the Integer solution (Wolsey[6]) with linear programming model and found the optimal solution in polynomial time.

Minimizing the number of train-sets problem formulation is the following.

\[ \text{Min} \quad \sum_{(i,j) \in E_2 \cup E_3} c_{ij} x_{ij} \quad (1) \]

s.t \[ \sum_{(i,j) \in E} x_{ij} - \sum_{(j,i) \in E} x_{ji} = 0, \quad i \in N \quad (2) \]

\[ 0 \leq x_{ij} \leq 1, \quad (i,j) \in E_2 \cup E_3 \quad (3) \]

\[ 1 \leq x_{ij} \leq 1, \quad (i,j) \in E_1 \quad (4) \]

Where, \( E_1 \): Train arc set
\( E_2 \): Turn-around arc set
\( E_3 \): Backward ground arc set

Constraint (2) ensures that flow conservation equation to each nodes. Constraint (3), (4) specifies that the arc capacity should be keep up the proper upper and lower bound.

3.4 Normalization Algorithm

As mentioned previously at section 3.3, the mathematical model based on modified min cost flow network find the minimal level of train-set with reducing turn-around time. There are numerous alternative solutions. To make more effective solution, we suggest the operation distance normalization algorithm. This motivates us to consider a routing heuristic based on generations of arc exchange of the minimum cost routes.

Let \( R_{ex1} \) and \( R_{ex2} \) be two disjoint routes. If \( R_{ex1} \) and \( R_{ex2} \), respectively, contain two consecutive trains, \( (i_{ex1} \rightarrow j_{ex1}) \) and \( (i_{ex2} \rightarrow j_{ex2}) \) satisfying following conditions, then we can generate arc exchange \( (i_{ex1} \rightarrow j_{ex2}) \) and \( (i_{ex2} \rightarrow j_{ex1}) \)

- Condition 1: \( i_{ex1} \) and \( i_{ex2} \) have the same destination;
- Condition 2: \( \text{arrival time of } i_{ex1} + \text{turn-around time} \leq \text{departure time of } j_{ex2} \) and \( \text{arrival time of } i_{ex2} + \text{turn-around time} \leq \text{departure time of } j_{ex1} \)

<Normalization Algorithm>

Step 1 (Initializing) : Apply a min-cost flow model to generate a daily routing

Step 2 (Sorting) : To sort route in an ascending order, \( R_1 \geq R_2 \geq \cdots \geq R_m \). \( \text{ex}_1 = 0, \text{ex}_2 = n+1 \)

Step 3 (Selecting \text{ex}_1) : If \text{ex}_1 be equal to \( n \), then go to Step 6,
Else \text{ex}_1 = \text{ex}_1 + 1 and select the route \( R_{ex1} \)

Step 4 (Selecting \text{ex}_2) : If \text{ex}_2 be equal to \text{ex}_1, then \text{ex}_2 = n + 1 and go to Step 3,
Else \text{ex}_2 = \text{ex}_2 - 1 and select the route \( R_{ex2} \)

Step 5 (Arc Exchange) : If the standard deviation of operation distance be decreased through arc exchange, then go to Step 2. Else go to Step 4

Step 6 (Terminating) : Terminate algorithm
4. Experimental Results

We have conducted computational experiments to determine the effectiveness and efficiency of our algorithm. The instance problem has 460 trains (up line 183, down line 183, deadhead 94), 11 end stations and 13 departure-arrival patterns. (See Fig 4, Table 1) We do not allow that vehicle, returned to maintenance center, go back to the route covering trains. Our computational experiments have been conducted with CPLEX 9.1/Visual C++ 6.0 and have been run on a Pentium(R) 4 CPU 2.80GHz

In Table 2, we look at the experiment result. We can see that column 3, 4 or 5 summarizes the performance of our solutions with respect to those provided by the current operation, column 2. Our approach finds the daily routes of reducing the number of train-set from 61 to 52, approximately 14%. In addition to, maximum distance of route, standard deviation of distance was decreased to 11% and 5%. Compared to the min-cost flow model, the method based on the modified min-cost flow network model can decrease the average operation time by almost 20 min and reduce the maximum operation time by 38 min. Our approach has a polynomial time complex. So, the computing time for is within a 2 min.

<table>
<thead>
<tr>
<th>Station</th>
<th>Turn Time</th>
<th>Num. Train</th>
<th>Num. Dead Head</th>
<th>Num. Train</th>
<th>Num. Dead Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center 1</td>
<td>∞</td>
<td>45</td>
<td>43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>159</td>
<td>40</td>
<td>180</td>
<td>21</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>21</td>
<td>2</td>
<td>-</td>
<td>22</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>5</td>
<td>123</td>
<td>-</td>
<td>123</td>
<td>-</td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>H</td>
<td>5</td>
<td>18</td>
<td>-</td>
<td>18</td>
<td>-</td>
</tr>
<tr>
<td>I</td>
<td>5</td>
<td>38</td>
<td>3</td>
<td>38</td>
<td>3</td>
</tr>
<tr>
<td>Center 2</td>
<td>∞</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>Sum</td>
<td>-</td>
<td>366</td>
<td>94</td>
<td>366</td>
<td>94</td>
</tr>
</tbody>
</table>

Table 2. Experiment Result
5. Conclusions

This paper describes a pragmatic algorithm for the train-set routing problems where the trains are frequent and train patterns are relatively simple. We suggest the two-phased method for the routing problem of minimizing the number of train-sets. In first stage, we formulate the train-set routing problem as a Min-cost-flow problem. Then, in the second stage, we attempt to normalize the travel distance covered to each routes. Our algorithm may be able to yield significant improvements in the efficiency of railway operations. To minimize the number of train-sets could save operating cost and increase flexibility of re-scheduling.

References


