Transportation Research Part B 45 (2011) 409–429

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network

Yun-Hong Min [a,*], Myoung-Ju Park [a], Sung-Pil Hong [a], Soon-Heum Hong [b]

[a] Dept. of Industrial Eng., Seoul National University, San 56-1 Shilim-Dong, Kwanahk Gu, Seoul 151-742, Republic of Korea
[b] Dept. of Railway Transport & Logistics Research, Korea Railroad Research Institute, 360-1 Woulam-Dong, Uiwang-City, Kyonggi-Do 437-757, Republic of Korea

ARTICLE INFO

Article history:
Received 5 January 2009
Received in revised form 16 August 2010
Accepted 16 August 2010

Keywords:
Train-conflict resolution
NP-hardness
Column generation
Fix-and-regenerate

ABSTRACT

In practice, a train-conflict resolution is decentralized around dispatchers each of whom controls a few segments in a global railway network with her rule-of-thumb to operational data. Conceptually, the global sub-optimality or infeasibility of the decentralized system is resolved by a network controller who coordinates the dispatchers and train operators at the lower layers on a real-time basis. However, such notion of a multi-layer system cannot be effectual unless the top layer is able to provide a global solution soon enough for the dynamic lower layers to adapt in a seamless manner. Unfortunately, a train-conflict resolution problem is NP-hard as formally established in this paper and an effective solution method traded off between computation time and solution quality has been lacking in literature. Thus, we propose a column-generation-based algorithm that exploits the separability of the problem. A key ingredient of the algorithm is an efficient heuristic for the pricing subproblem for column generation. Tested on the real data from the Seoul metropolitan railway network, the algorithm provides near-optimal conflict-free timetables in a few seconds for most cases. The performance of the proposed algorithm is compared to the ones of the previous MIP-based heuristic by Törnquist and Persson (2007) and the priority-based heuristic by Sahin (1999).

1. Introduction

An important goal of daily traffic management is to restore the feasibility of a timetable disrupted by unforeseen events (technical or human failures) with minimal consequences. By nature, this *train-conflict resolution* is a real-time problem as it involves the control of trains in operation, and a comprehensive and centralized control of the real size railway network is far from a feasible idea when the lines are complex and the traffic is heavy. In practice, the railway network is decomposed into local areas each of which, normally spanning a few railway segments between stations, is controlled by a single dispatcher based on operational data in an autonomous way. A remedy of the probable sub-optimality or infeasibility of such decentralized system is the notion of multi-layer traffic management: a network controller coordinates the traffic management of the dispatchers and train operators at the lower layers. Therefore, it is a crucial task of the top layer to compute a global solution promptly enough for the dynamic lower layers to adapt in a seamless manner. The more computationally efficient the top layer gets, the more measures can be transferred from the lower layers and integrated into the top layer so that the global optimality of train-conflict resolution may be enhanced. Furthermore, a major conflict may be resolved only by the reconstruction of a global timetable.

* Corresponding author. Tel.: +82 2 880 7181.
E-mail address: myh@snu.ac.kr (Y.-H. Min).
URL: http://polytope.snu.ac.kr/sphong.html (S.-P. Hong).

0191-2615/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trb.2010.08.001

However, a train-conflict resolution problem is NP-hard. As shown in this paper, a very simple railway topology (three stations connected with two single-tracks) makes the problem NP-hard. In a real network which is a concatenation of such subnetworks, even the sub-optimality of the upstream is easily multiplied to the downstream via delayed trains. Indeed, an effective method for the problem has not been found in the literature. Perhaps, an exception is the heuristic by Törnquist and Persson (2007) which has been reported to produce the solutions of very reasonable quality in small computation times for a certain range of problem size. Unfortunately, when the problem size gets as large as a metropolitan area network, the computation times not only get extensive but also fluctuate significantly on the instances. Such fluctuation is typical result of commercial codes based on an enumerative method such as the branch-and-bound method.

Therefore, the centralized train-conflict resolution problem requires to guarantee a minimal and predictable computational effort. An observation is that without train dwelling constraints at stations, the problem is separated into independent train dispatching problems at segments between stations. A further observation is that when a train schedule is fixed for a segment, the reduced problem is again a train-conflict resolution problem. Also, the fixed schedule is easily extended to a feasible global timetable by fixing the schedules in a topological order of segments in the underlying railway network. From these observations, an economic variant of the column-generation method called the *fix-and-regenerate* method is particularly promising: at each iteration, after solving the LP-master problem, we fix the most promising column to reduce the size of the problem. Here, each column corresponds to a feasible train schedule of a segment. To maintain minimal computation, we use a heuristic instead of an exact method in obtaining a feasible train schedule of a segment.

The algorithm is tested on the real data from the Seoul metropolitan railway network. For most cases, it provides near-optimal or reasonable conflict-free timetables w.r.t. the total weighted deviation from the original timetable. Its performance is also compared to the ones of two previous algorithms, one by Törnquist and Persson (2007) and the other by Sahin (1999). Considering a trade-off between computation time and solution quality, the proposed algorithm outperforms the previous heuristics. It seems particularly suitable for the centralized approach in the sense that computation times are not only minimal but also predictable.

This paper is organized as follows. Section 2 reviews the previous studies on the train-conflict resolution problem. In Section 3, the train-conflict resolution problem is formally defined and formulated as a mixed integer program. The intractability of the train-conflict resolution problem is explored in Section 4. Section 5 is devoted to the discussion of the column-generation based heuristic. The heuristic is then tested on the real data from the Seoul metropolitan railway network in comparison with the previous heuristics in Section 6. Section 7 provides some concluding remarks and further studies.

2. Previous studies

The train-conflict resolution problem has attracted numerous studies that can be categorized according to a convenient trichotomy: whether they are focused on the top, middle, or bottom layer as in Table 1.

The studies of the first category aim to find the global optimality of a large railway network. They mainly deal with global timetables, track re-routing and possible dwell time changes for the restoration of train operation feasibility.

The studies of the second category model the micro levels of traffic management. They manage traffics in a few segments of a large railway network and focus on the decentralized conflict resolution of a dispatcher who controls the dispatching sequence and the routes of trains. Such studies can be classified further according to how the headway safety time is enforced. Some studies (Kraay et al., 1991; Cai and Goh, 1994; Cai et al., 1998; Sahin, 1999; Dessouky et al., 2006) enforce each train to observe the minimum and maximum values of its transit time over each segment. On the other hand, the others (Mascis and Pacciarelli, 2002; D'Ariano et al., 2007a; Mazzarello and Ottaviani, 2007; Mascis et al., 2008) explicitly model the blocks of segments and prevent two consecutive trains from being located in the same block in the same epoch. However, to reduce the computational complexity of the problem, they rely on the average speeds of trains.

The distinction between the studies on the middle and bottom layers does not appear as clear as between the top two layers. However, the studies on the bottom layer mainly deal with the real-time speed controls in complying with the controls from the middle layer. The bottom layer checks the feasibility of the average train speeds and dispatching sequences

Table 1
Literature summary.

Layer	Top	Middle	Bottom
Controller	Network controllers	Dispatchers	Dispatcher/Drivers
Subjects	Railway network	A few segments	Operational Trains
Decision variables	Global time table, track re-routing, dwell time change	Mainly sequencing, partially re-routing	Mainly speed control, partially sequencing
Reference	Carey (1994a,b), Carey and Lockwood (1995), Kraay and Harker (1995), Higgins et al. (1996), Caprara et al. (2002, 2006), Törnquist (2007), Törnquist and Persson (2007), and Cacchiani et al. (2008)	Kraay et al. (1991), Cai and Goh (1994, 1998), Sahin (1999), Mascis and Pacciarelli (2002), Dessouky et al. (2006), D'Ariano et al. (2007a), Mazzarello and Ottaviani (2007), Mascis et al. (2008), and Sahin (1999)	D'Ariano et al. (2007b), Mazzarello and Ottaviani (2007), and Mascis et al. (2008)

from the middle layer after taking account of deceleration and acceleration. If not, it informs the middle layer of the time epochs during which the system becomes infeasible and recommends the new speeds of trains. Accordingly, the middle layer generates a new dispatching sequence of the trains based on the recommended speeds. The procedure is repeated until a feasible set of the speeds and dispatching sequences of trains is obtained (D'Ariano et al., 2007b; Mazzarello and Ottaviani, 2007; Mascis et al., 2008).

The *timetabling problem* is closely related to the train-conflict resolution problem in the sense that it also computes a train schedule over a given time horizon for a global network, thereby considering the same kind of constraints as a train-conflict resolution problem at the top layer. However, the timetabling problem pursues a maximal profit from the satisfied demands. As pointed out by Brannlund et al. (1998), the principles of timetabling problem algorithms can be applied to the train-conflict resolution problem at the top layer. However, such methods (Carey, 1994a,b; Carey and Lockwood, 1995; Brannlund et al., 1998; Caprara et al., 2002, 2006; Cacchiani et al., 2008), without considering the restriction on computation times, are reported to require too excessive computational efforts for train-conflict resolution purposes.

From train-conflict resolution or timetabling models proposed in the literature, the one proposed by Törnquist and Persson (2007) seems the closest to ours. As compared in detail to our MIP-model in the next section, a difference in our model is the consideration of the platform assignment to trains within a station. Regarding the solution method, they solved, by a commercial code, an MIP-formulation augmented with the linear cuts reflecting a set of additional routing strategies that exclude some "impractical" solutions. These cuts are not valid in the ordinary sense that they may exclude some feasible solutions. But, as demonstrated in their paper (Törnquist and Persson, 2007) and also in our comparison test, for a certain range of problem size, such strategies seem very effective and produce near-optimal solutions in significantly reduced computation times in comparison to the same approach without the cuts.

From our experience, however, the frequency of trains, the multiplicity of tracks, and the time horizon of the problem are the three major factors in determining computational efforts required for a problem instance. Unfortunately, when these factors get as large as in a metropolitan railway network, the computation times of the heuristic (Törnquist and Persson, 2007) not only get extensive but also fluctuate significantly on the instances. This is typical result of a commercial code based on an enumerative method such as the branch-and-bound method.

This paper is motivated by the goal of finding an efficient method for the top layer train-conflict resolution for the Seoul metropolitan railway network, many of whose segments between stations are double-tracked with high train frequencies, e.g. 15 trains per hour in average.

3. The problem

3.1. Some terminologies

Definition 3.1 (*Railway networks*). By a *railway network*, we mean an undirected graph consisting of the nodes of stations and the edges of *segments*. A segment is the set of tracks connecting two adjacent stations.

This definition reflects that the model proposed in this paper does not consider the routing of trains within a station.

Definition 3.2 (*Train timetables*). A *train timetable* is the set of quadruples (`station`, `arrival time`, `depature time`, `track assignment`) for all trains and the stations which they visit over a predetermined time horizon.

Definition 3.3 (*Train-conflicts*). By a *train-conflict*, we mean a situation in which two or more trains claim resources in an infeasible manner, namely, in the way that violates a safety regulation.

To this broad definition, we add that in most practical cases, conflicts arise when a set of trains fail to respect the safe headway times due to delays.

Problem 3.4 (*Train-conflict resolution problem*).

Input A railway network, a timetable, and a set of train-conflicts.
Output A conflict-free timetable with the minimum total weighted deviation w.r.t. the original timetable which satisfies the constraints on the earliest departure times, the headway times, the transit times, and the dwelling times.

3.2. Assumptions

Our train-conflict resolution model relies on the following restrictions.

Assumption 3.5. Each station has enough number of platforms.

As mentioned earlier, this is motivated by the empirical data from KORAIL (Korea Railroad) which shows that the stations of Seoul metropolitan area have enough number of platforms for train-conflict resolution. Indeed, in the experiment of Sec-

tion 6, we will observe that the number of platforms required by our algorithm does not exceed the number of platforms in any station. If this is not the case, our model does not guarantee a feasible platform assignment.

Assumption 3.6. A railway network is a tree.

Assumption 3.7. Each track is unidirectional.

Assumptions 3.6 and 3.7 also reflects the Seoul metropolitan railway network. Based on these assumptions, we duplicate each segment between two adjacent stations. Then, two adjacent stations have two parallel segments to which we assign opposite directions. We can view a segment as a set of tracks with the same direction. See Fig. 1.

Now choose a station in the railway network as the *root* of a tree. A segment is called *inbound* if it is oriented towards the root, and *outbound* otherwise. Then every train is one of the following types: (1) an inbound train which only traverses inbound segments, (2) an outbound train which only traverses outbound segments, and (3) an in-outbound train which traverses inbound segments first and outbound segments afterwards. An in-outbound train can be thought of as two trains. One of them traverses inbound segments from the origin to a certain station from which the other traverses outbound segments to the destination. For example, in Fig. 1, an in-outbound train from station 2 to station 4 is the concatenation of an inbound train from station 2 to the root and an outbound train from the root to station 4. Then, we can obtain a feasible timetable of all trains by first computing the timetable of inbound trains and then that of outbound trains. This method does not guarantee an optimal solution. But, at the same time, it does not violate the optimality by much since there are few in-outbound trains in practice. For instance, there are no in-outbound trains in the Seoul metropolitan area. Thus the method guarantees the optimality. In addition, it is pragmatic since the problem size can be halved, corresponding to our purpose. Therefore, we will solve the conflict resolution problem by devoting a segment to two opposite directions.

The train-conflict resolution problem can be formulated as an MIP. Table 2 summarizes constraints considered in the literature of Table 1. The second column indicates the equation numbers of the corresponding features of the MIP-model (1)–(10) presented below. Note that intra-track headway times, dispatch sequencing, and dwelling time requirements at stations are the most common features of the MIP-models for the train-conflict resolution problem or timetabling problem.

Regarding the dwelling modes of trains, notice that even when a station is originally not a stopping station for a train, the total deviation of a new timetable can be smaller when the train is allowed to dwell at the station. If so, the train will have a longer transit time on the incident segments due to deceleration and acceleration. Such difference may be significant for a

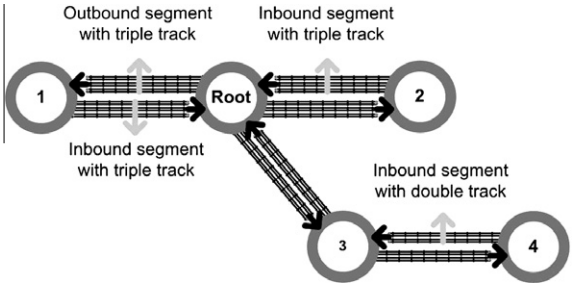

Fig. 1. A railway network.

Table 2
MIP-model comparison.

Constraints	Our model	Group A	Group B	Group C	Group D	Group E
Multiple tracks	(3)	✓				✓
Intra-track headway time	(6)	✓	✓	✓	✓	✓
Inter-track headway time	(7)					
Train-dispatch sequencing	(4)	✓	✓	✓	✓	✓
Dwelling modes at stations	Only in postprocessing				✓	
Dwelling time requirements	(10)	✓	✓	✓	✓	✓
Platform assignments		✓				
Bi-direction		✓		✓		✓

Group A	Carey (1994a), DÁriano et al. (2007a,b), Dessouky et al. (2006), Mascis and Pacciarelli (2002), Mascis et al. (2008), Mazzarello and Ottaviani (2007), Törnquist (2007), and Törnquist and Persson (2007)
Group B	Cacchiani et al. (2008), Caprara et al. (2002), Caprara et al. (2006), and Carey and Lockwood (1995)
Group C	Cai and Goh (1994), Cai et al. (1998), Carey (1994b), Higgins et al. (1996), Kraay and Harker (1995), and Sahin (1999)
Group D	Zhou and Zhong (2005)
Group E	Kraay et al. (1991)

metropolitan network where segments are relatively short and trains are more frequent. Zhou and Zhong (2005), to the best of the author's knowledge, were first to explicitly incorporate the dwelling modes of trains as decision variables into their timetabling model for a high speed train line where the difference is also significant. However, it would be too excessive for our goal of achieving minimal computation to design an algorithm searching over all possible combinations of the dwelling modes of trains over segments. Thus, in our algorithm, we initially fix the dwelling modes of trains and search a better combination in the postprocessing.

The feature of our model is the intra-track safe headway time: on the same segment, the departure or arrival times of any pair of consecutive trains should not be too close even if they use different tracks of a segment. This is to enforce the safe headway time at the points where tracks cross.

As mentioned earlier, the platform assignment is not considered in our model as in most studies with a few exceptions (Carey, 1994a; Törnquist and Persson, 2007). However, in the actual experiment, Carey (1994a) only considered the instances with a single platform in each direction.

3.3. An MIP-formulation

Throughout the paper, we will reserve Q, P, and I to denote the set of stations, segments, and trains, respectively. P and I here are either all inbound or all outbound. Also, $I_p \subset I$ denotes the set of trains traversing segment p.

The parameters and decision variables of the proposed MIP-formulation are as follows:

Parameters

- c_p^i = cost of the unit departure delay of train i at segment p.
- w_p^i = cost of the unit deviation (delay or early arrival) from original arrival time of train i at segment p.
- d_p^i = original departure time of train i at segment p.
- a_p^i = original arrival time of train i at segment p.
- h_q = safe headway time between two consecutive trains arriving or departing at station q via the same track.
- h_m = safe headway time between two consecutive trains arriving or departing at a station via different tracks.
- s_q^i = minimum dwelling time of train i at station q (defined as *0* if q is not a stopping station for train i).
- $\underline{t}_p^i(\circ\circ)$ = minimum transit time over segment p of train i when dwelling mode of i at the two end stations are $\circ\circ$; thus $\circ\circ$ = ss, sp, ps, or pp meaning, in their order, stop–stop, stop–pass, pass–stop, or pass–pass.
- $\overline{t}_p^i(\circ\circ)$ = maximum transit time over segment p of train i with dwelling mode $\circ\circ$.
- b_p^i = earliest possible (nonnegative) departure time of train i at segment p.
- n_p = number of tracks of segment p.
- M = a sufficiently large positive number.

Decision variables

- D_p^i = departure time of train i at end station q of segment $p = (q,r)$.
- A_p^i = arrival time of train i at end station r of segment $p = (q,r)$.
- E_p^i = difference between original arrival time, a_p^i and actual arrival time, A_p^i.
- $Z_{p,k}^i = 1$ if train i is assigned to the kth track of segment p and 0 otherwise.
- $X_p^{ij} = 1$ if train i departs earlier than train j at end station q of segment $p = (q,r)$ and 0 otherwise.
- $Y_p^{ij} = 1$ if train i arrives earlier than train j at end station r of segment $p = (q,r)$ and 0 otherwise.
- $\circ\circ_p^i = 1$ if train has a dwelling mode $\circ\circ$ at segment p and 0 otherwise, where $\circ\circ$ = ss, sp, ps, or pp meaning in their order, stop–stop, stop–pass, pass–stop, or pass–pass.

Objective function

Our objective is to minimize the total weighted deviation (namely, both earliness and tardiness) between original and rescheduled timetables of trains:

$$\min \sum_{p\in P} \sum_{i\in I_p} \left\{ c_p^i (D_p^i - d_p^i) + w_p^i E_p^i \right\}. \tag{1}$$

Constraints

Deviation from the original timetable. These constraints are two linear constraints that are equivalent to $E_p^i \geqslant \left| A_p^i - a_p^i \right| : \ \forall p \in P, \ \forall i \in I_p$,

$$A_p^i - a_p^i \leqslant E_p^i \quad \text{and} \quad -\left(A_p^i - a_p^i\right) \leqslant E_p^i. \tag{2}$$

Track assignment. If a segment p has multiple tracks, a train i traversing segment p has to choose exactly one track: $\forall p \in P$, $\forall i \in I_p$,

$$\sum_{k=1}^{n_p} Z_{p,k}^i = 1. \tag{3}$$

Train-dispatch sequencing. A departing sequence of trains should be equal to their arrival sequence over the same track: $\forall p \in P$ and $\forall i, j \in I_p$ with $i \neq j$, $k = 1, \ldots, n_p$,

$$\begin{aligned} &2 - Z_{p,k}^i - Z_{p,k}^j + \left(X_p^{ij} - Y_p^{ij}\right) \geqslant 0 \quad \text{and} \\ &2 - Z_{p,k}^i - Z_{p,k}^j - \left(X_p^{ij} - Y_p^{ij}\right) \geqslant 0. \end{aligned} \tag{4}$$

Earliest departure times of trains. Each train i should observe its earliest possible departure time: $\forall p \in P$, $\forall i \in I_p$,

$$D_p^i \geqslant b_p^i. \tag{5}$$

Intra-track headway safety. Two trains i and j traversing segment p via track k have to observe the minimum headway time: $\forall p = (q,r) \in P$ and $\forall i, j \in I_p$ with $i \neq j$, $k = 1, \ldots, n_p$,

$$M\left(1 - Z_{p,k}^i\right) + M\left(1 - Z_{p,k}^j\right) + \begin{cases} M\left(1 - X_p^{ij}\right) + D_p^j - D_p^i \geqslant h_q, \\ M\left(1 - X_p^{ij}\right) + A_p^j - A_p^i \geqslant h_r, \\ MX_p^{ij} + D_p^i - D_p^j \geqslant h_q, \quad \text{and} \\ MX_p^{ij} + A_p^i - A_p^j \geqslant h_r. \end{cases} \tag{6}$$

Inter-track headway safety. Two trains i and j traversing segment p have to observe the minimum headway time: $\forall p \in P$ and $\forall i, j \in I_p$ with $i \neq j$,

$$\begin{aligned} &M\left(1 - X_p^{ij}\right) + D_p^j - D_p^i \geqslant h_m, \\ &M\left(1 - Y_p^{ij}\right) + A_p^j - A_p^i \geqslant h_m, \\ &MX_p^{ij} + D_p^i - D_p^j \geqslant h_m, \quad \text{and} \\ &MY_p^{ij} + A_p^i - A_p^j \geqslant h_m. \end{aligned} \tag{7}$$

Dwelling modes and corresponding transit times at segments. For each segment $p = (q,r)$, each train chooses a dwelling mode. (However, if station q is originally a dwelling station, then a train has no other choice than to stop at q.) Accordingly, the transit time of each train is determined between its lower- and upper-bounds: $\forall p = (q,r) \in P$ and $\forall i \in I_p$,

$$\begin{aligned} &SS_p^i + SP_p^i + PS_p^i + PP_p^i = 1, \\ &\left(SS_p^i + SP_p^i = 1, \text{ added if } q \text{ is originally a dwelling station of } i,\right) \end{aligned} \tag{8}$$

$$\begin{aligned} &\underline{t}_p^i(ss)SS_p^i + \underline{t}_p^i(sp)SP_p^i + \underline{t}_p^i(ps)PS_p^i + \underline{t}_p^i(pp)PP_p^i \leqslant A_p^i - D_p^i, \quad \text{and} \\ &\overline{t}_p^i(ss)SS_p^i + \overline{t}_p^i(sp)SP_p^i + \overline{t}_p^i(ps)PS_p^i + \overline{t}_p^i(pp)PP_p^i \geqslant A_p^i - D_p^i. \end{aligned} \tag{9}$$

Dwelling time requirements. Each train has to dwell for a positive predetermined time at station q if it stops at station q. Otherwise the dwelling time should be zero: $\forall q \in Q$, for each pair of consecutive segments incident to q, $p' = (s,q)$ and $p = (q,r)$, and for each train traversing p' and p, namely $\forall i \in I_{p'} \cap I_p$, we have

$$\begin{aligned} &s_q^i\left(SS_p^i + SP_p^i\right) \leqslant D_p^i - A_{p'}^i \leqslant M\left(SS_p^i + SP_p^i\right) \quad \text{and} \\ &s_q^i\left(SS_{p'}^i + PS_{p'}^i\right) \leqslant D_p^i - A_{p'}^i \leqslant M\left(SS_{p'}^i + PS_{p'}^i\right). \end{aligned} \tag{10}$$

Notice that each pair of (10) also implies $SS_{p'}^i + PS_{p'}^i = SS_p^i + SP_p^i$ so that the dwelling modes over p' and p are consistent.

Notice that the constraints from (10) are the only coupling constraints that control the movement of a train over adjacent segments sharing the same station. Hence, if (10) is relaxed, the MIP-model is separable into the independent subproblems with respect to the segments $p \in P$. More specifically, each independent subproblem is a train-conflict resolution problem on a single segment. Conversely, a feasible solution of the original problem, namely a global timetable, is the combination of the timetables from the subproblems that satisfy the coupling constraints of (10). In Section 5, we will propose an algorithm for the MIP-model (1)–(10) relying on this separability.

4. Intractability of train-conflict resolution

As pointed out in Törnquist (2006), a formal proof of the intractability of train-conflict resolution is lacking in the literature. In this paper, we show that a very simple version of our problem consisting of two single-tracked segments (hence of three stations) is strongly NP-hard. To do so, we use a reduction from the strongly NP-hard two machine flow shop problem with transportation delays and unit processing times, denoted by $F2|p_{ij}=1,t_j,r_j|\sum C_j$ in the conventional notation of the field (Brucker et al., 2004).

Problem 4.1. Each job $j = 1,2,\ldots,n$, released by time r_j, has to be processed by two machines i in the order of $i = 1,2$. The p_{ij} represents the processing time of job j on machine i. We assume a unit processing time $p_{ij} = 1$ for all i and j. But, there is a variable transportation delay t_j of job j between its completion time on machine 1 and the moment it becomes available for machine 2. Let C_j denote the completion time of job j on machine 2. Then our objective is to find a combined processing sequence of jobs on two machines that minimizes the total completion time $\sum C_j$.

Theorem 4.2. *The train-conflict resolution problem is strongly NP-hard.*

Proof. Given an instance of Problem 4.1, construct an instance of the train-conflict resolution problem, referred to as SP, as follows: SP has two single-tracked segments $p = 1$ and 2 connecting three stations $q = 1$, 2, and 3. Every station has a unit safe headway time. We assume that the n jobs have been sorted in a nondecreasing order of transportation delays: $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_n$. Construct n trains $j = 1,2,\ldots,n$ whose transit times are all units on every segment. Train j is originally scheduled to depart from station 1 at $-n+j$ and arrive at station 2 at $-n+j+1$: $d_1^j = -n+j$ and $a_1^j = -n+j+1$. After dwelling for t_j at station 2, it departs from station 2 at $-n+j+t_j+1$ and arrives at station 3 at $-n+j+t_j+2 : s_2^j = t_j,\ d_2^j = -n+j+t_j+1$, and $a_2^j = -n+j+t_j+2$. Its earliest possible departure time b_1^j at station 1 is r_j. Finally, regarding the coefficients of the objective function, we only consider the deviation cost of arrival time at station 3 by a unit weight: $c_1^j = c_2^j = w_1^j = 0$, and $w_2^j = 1$.

First of all, to see SP is a legitimate instance of the train-conflict resolution problem, notice that the original train schedules constructed above constitute a conflict-free timetable as $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_n$. Since the departure of each train has been delayed by $b_1^j - d_1^j$, we need to construct a new conflict-free timetable. From our construction, a departure delay does not incur a cost and no early arrival may occur. Thus the objective of SP is to minimize the sum of arrival delays at station 3 and it is equivalent to minimization of the sum of arrival times.

Now we establish the equivalence of SP and Problem 4.1. To do so, it suffices to show that there exists a feasible schedule for the instance of Problem 4.1 with the total completion time $\leqslant C$ if and only if there exists a conflict-free timetable of SP whose total weighted deviation from the original timetable is $\leqslant C - \sum_j a_2^j$.

For the "only if" part, suppose we are given a feasible schedule of Problem 4.1 with the total completion time $\leqslant C$. Construct a conflict-free timetable of SP as follows: for each train, set the departure time at station 1 and the arrival time at station 2, respectively, to the starting and completion time of the corresponding job on machine 1. Since every job has a unit processing time, every train observes the safe headway time. Similarly, for each train, let the departure time at station 2 and the arrival time at station 3, respectively, be the starting and completion time of the corresponding job on machine 2. Since $s_2^j = t_j$, each train dwells at least for t_j units of time at s_2. Similarly, the train schedules on segment (2,3) also satisfy the safety requirement. Since the sum of train arrival times at station 3 is $\sum_j A_2^j = \sum_j C_j \leqslant C$, the objective value of the constructed timetable of SP is $\sum_j A_2^j - \sum_j a_2^j \leqslant C - \sum_j a_2^j$. The proof of "if" part is almost identical and hence omitted.

Finally, it is easy to see from the construction the largest absolute value of a number in SP is bounded by the order of the largest number from the instance of Problem 4.1. Thus the strong NP-hardness of Problem 4.1 implies the strong NP-hardness of the train-conflict resolution problem. □

5. The algorithm

This section is devoted to the development of an efficient algorithm relying on the separability of the problem discussed in Section 3.3. Typically, this separability is exploited by a problem decomposition technique such as the Lagrangian relaxation and the column-generation method.

Of these alternatives, the latter appears more suitable to our problem. In the Dantzig–Wolfe reformulation of the MIP-model (1)–(10), the columns correspond to the timetables of the subproblems of segments. Thus, generating columns for solving the (primal) LP-relaxation via the simplex method has a very natural interpretation of searching timetables of the subproblems of segments that are promising in terms of current prices. This enables us to easily implement various heuristic ideas to accelerate the simplex method. For instance, by single pricing, we may consider multiple promising timetables when entering variables. This is not possible in the (dual) Lagrangian relaxation, where only single timetable corresponds to the current dual solution. Furthermore, the subgradient optimization method to solve the Lagrangian relaxation involves numerical steps, such as the choice of step sizes, that affect the convergence of solutions. Hence, the simplex method steps seem much more transparent for implementation.

The column generation algorithm for solving the LP-relaxation of an IP-problem, combined with an implicit enumeration scheme, can be used to compute an optimal solution of IP-problem. Such an exact method, however, requires excessive computational efforts for our purpose of the train-conflict resolution problem. An observation is that if we fix the timetable over a segment, then the reduced problem is again a train-conflict resolution problem. Furthermore, the feasibility of the combined timetable is easily maintained if timetables are fixed in a topological order of segments with respect to the underlying railway network.

Due to the correspondence between the timetables of segments and the columns, the sequential timetable-fixing idea can be naturally combined with the column generation algorithm to compute a feasible solution without resorting to an enumeration step. A similar idea has been used in Caprara et al. (2002, 2006) and Cacchiani et al. (2008). They consider the routing subproblems of trains which are obtained by relaxing the headway safety constraints. Also, in this case, a feasible solution is easily obtained by sequentially fixing the routes of trains. However, their approach is not applicable to our model since it is based on the discretization of time axis. Furthermore, as the number of trains is an order of magnitude larger than the number of segments especially in a metropolitan network, exploiting the separability of the problem with respect to trains may deteriorate the quality of solution.

In Section 5.1, we provide an LP-relaxation of the MIP-model (1)–(10) based on the Dantzig–Wolfe decomposition considering (10) as coupling constraints. Section 5.2 is devoted to an efficient heuristic to solve the pricing subproblems. Finally, Section 5.3 combines these two into the main train-conflict resolution algorithm.

5.1. Dantzig–Wolfe decomposition

Consider the problem consisting of (1)–(9) from the MIP-model which is, as observed earlier, the set of independent problems each of which is to find a conflict-free timetable of minimum weighted deviation for each segment $p \in P$. Let $T_p = \{\tau_{p,1}, \ldots, \tau_{p,k_p}\}$ be the set of the conflict-free timetables for segment $p \in P$ (hence k_p means the total number of conflict-free timetables for segment $p \in P$).

As mentioned in Section 3.1, our algorithm initially fixes the dwelling modes so that every train stops at every station: $SS_p^i \leftarrow 1$, for all p and $i \in I_p$. Then, by expressing each integer feasible solution of a subproblem as $\tau_p = \sum_{t=1}^{k_p} \tau_{p,t}\lambda_{p,t}$ using binary variables $\lambda_{p,t} \in \{0,1\}$ with $\sum_{t=1}^{k_p} \lambda_{p,t} = 1$, the MIP-model (1)–(10) can be restated as follows:

$$\min \quad \sum_{p\in P}\sum_{i\in I_p}\left(\sum_{t=1}^{k_p}\left(c_p^i D_{p,t}^i + w_p^i E_{p,t}^i\right)\lambda_{p,t} - c_p^i d_p^i\right) \tag{11}$$

$$\text{s.t.} \quad \sum_{t=1}^{k_p} D_{p,t}^i \lambda_{p,t} - \sum_{t'=1}^{k_{p'}} A_{p',t'}^i \lambda_{p',t'} \geqslant s_q^i, \quad \forall q \in Q, \quad \forall p' = (s,q), \quad p = (q,r), \quad \forall i \in I_p \cap I_{p'} \tag{12}$$

$$\sum_{t=1}^{k_p} \lambda_{p,t} = 1, \quad \forall p \in P \tag{13}$$

$$\lambda_{p,t} \in \{0,1\}, \quad \forall p \in P, \quad t = 1, \ldots, k_p. \tag{14}$$

Here, $D_{p,t}^i$, $A_{p,t}^i$, and $E_{p,t}^i$, respectively, are the departure time, arrival time, and arrival deviation of train i over segment p according to the tth timetable, $\tau_{p,t}$.

Now we consider the LP-relaxation obtained from the binary IP, (11)–(14) by replacing (14) with $\lambda_{p,t} \geqslant 0$. The column generation algorithm applied to this primal LP-relaxation, called *LP-master*, or *LPM* consists of the following steps.

Initialization Construct a *restricted LPM* or RLPM which is obtained by restricting LPM to the columns that are properly chosen from each and every segment. Solve the RLPM to get the primal optimal solution λ as well as the dual optimal solution (π, μ), where $\pi = \left(\pi_q^i\right)$ and $\mu = (\mu_p)$ are the dual variables corresponding to constrains (12) and (13), respectively.

Pricing Generate a column whose reduced cost (or price) is negative. If there is no such column, the current solution λ is an optimal solution of LPM. Otherwise, enter such a column to get a new pair of solutions λ' and (π', μ') and repeat this step.

The pricing step amounts to finding a variable $\lambda_{p,t}$ of the tth timetable over segment $p = (q,r)$ whose reduced cost is negative. In terms of the dual variables (π, μ), the reduced cost is given by $\sum_{i\in I_p}\left(\left(c_p^i - \pi_q^i\right)D_p^i + \pi_r^i A_p^i + w_p^i E_p^i\right) - \mu_p$. Thus, generating a negative reduced cost column of the pricing step is equivalent to the following subproblem:

$$(\text{PSP})\zeta_p = \min\left\{\sum_{i\in I_p}\left(\left(c_p^i - \pi_q^i\right)D_p^i + \pi_r^i A_p^i + w_p^i E_p^i\right) - \mu_p : \tau_p \in T_p\right\}. \tag{15}$$

If $\zeta_p \geqslant 0$ for all segments p, the current solution is optimal. Otherwise, we can enter the corresponding column to RLPM to improve the solution. Thus, it is essential to solve the pricing subproblem (15) fast enough, which is the theme of the next section.

5.2. A heuristic for pricing subproblem

From (15), the pricing subproblem is a train-conflict resolution problem of finding the timetable of a segment optimal with respect to the weights modified by the current values of dual variables (π, μ). Although the computational complexity of this single-segment train-conflict resolution problem is an interesting open problem, it bears some similarity with the NP-hard scheduling problem of minimizing weighted earliness and tardiness in Garey et al. (1988). As in Section 4, regarding each train and track as a job and a machine, respectively, two problems are identical except that in (15) a single machine can process two jobs if neither the starting nor finishing times of two jobs are too close to each other. Furthermore, our problem has an additional complication involving release times. Thus we conjecture that the pricing subproblem is an intractable problem and propose a heuristic instead of an exact method.

A scheduling problem of minimizing weighted earliness and tardiness typically has two phases (Yano and Kim, 1991; Davis and Kanet, 1993). In the first phase, a job processing sequence is determined. Then, based on the sequence, the second phase finds the starting time of each job to minimize the objective value, i.e. the weighted earliness and tardiness. An alternative approach is a rule-based method which dispatches jobs based on the *lookahead function*, a measure of the effects of job delay to the overall objective (Valente and Alves, 2005).

Our heuristic is a hybrid of these two approaches: we enumerate some partial sequences of trains and choose the best one according to a lookahead function defined for the train-conflict resolution problem.

Algorithm 5.1. Pricing heuristic

Step 1: Sort the trains in an increasing order of the earliest possible departure times. Repeat Steps 2–4 until all schedules of trains are fixed.
Step 2: Of the trains whose schedules are not fixed yet, choose the first three earliest trains in the earliest possible departure time. Then we have 6(=3!) possible sequences of the three trains.
Step 3: For each sequence from Step 2, do
  for each of the three trains in their order in the sequence, do
    for each track, assign to the train the departure and arrival times that minimize the weighted deviation from its original schedule and are conflict-free with the trains already assigned to the track.
    Choose a track with the minimum weighted deviation.
  (See Example 5.2 below.)
Step 4: Among the schedules of the six sequences, choose one whose lookahead function (16) is minimum:

$$\sum_{i=1}^{3} \left\{ \left(c_p^i - \pi_q^i\right) D_p^i + w_p^i A_p^i \right\} \Big/ \left(c_p^i - \pi_q^i + w_p^i\right). \tag{16}$$

Fix the schedule of the first train.

Example 5.2. Fig. 2 illustrates Step 3. Consider a double-tracked segment $p = (q, r)$. The intra-track and inter-track safe headway times are 3 and 2, respectively, at both end-stations. Let the first three trains be A, B, and C whose original departure time, arrival time, and minimum transit time triples are (0,9,9), (7,11,4), and (10,14,4), respectively. The maximum transit time is set to be ∞ for every train. Also, assume the earliest possible departure times are 5 for A and the same as the original schedule for the other two. Finally, for simplicity, assume there is no train assigned to any of the tracks. For the sequence (A,B,C), for instance, Step 3 assigns the departure time, the arrival time, and the track to each of three trains as follows.

First, it is easy to see that train A can start at its earliest possible departure time on track 1 which is currently empty. To minimize the weighted deviation, it travels at its minimum transit time: the assigned departure and arrival time pair is (5,14).

Train B can observe the original schedule by travelling on track 2: the assigned departure and arrival time pair is (7,11).

Now we consider the third train C. Then, to assign train C with the minimum weighted deviation without conflict, as easily seen from Fig. 2, the departure and arrival time pair should be (10,17) on track 1, and (10,16) on track 2. If the weights of both departure and arrival deviations of train C are 1, the weighted deviations on track 1 and 2 are 3(=10 − 10 + 17 − 14) and 2(= 10 − 10 + 16 − 14), respectively. Hence train C is assigned to track 2.

A greedy approach for a scheduling problem is to compute a sequence of jobs incrementally based on a *lookahead* function, an estimation of the change in an objective value from an increment of jobs in the sequence. For instance, to solve a single machine scheduling problem with earliness and tardiness costs, Ow and Morton (1989) and Valente and Alves

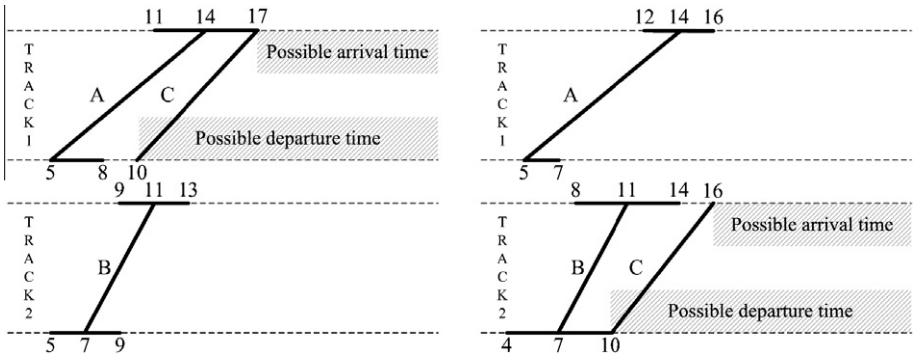

Fig. 2. An illustration of Step 3.

(2005) used a lookahead function that estimates, in each iteration, not only an increase in the objective function but also the tardiness of the remaining jobs due to a new job added to the current sequence. In our pricing problem as a scheduling problem, (16) serves as a lookahead function. This particular form was chosen from experiments. Notice that this function is a weighted average of the departure and arrival times. Thus, whenever a train is rescheduled, the earliness is preferred to the tardiness although they are penalized with the same weight in the objective function. This is probably because the earliness may leave more room to the following trains to optimize their schedules.

To analyze the time complexity of this heuristic, suppose segment p has m tracks and n trains. It takes $O(n\log n)$ time to sort the trains in Step 1. Obviously, the heuristic terminates in n iterations. Since we investigate a constant number of schedules to assign 3 trains over the m tracks, each iteration requires $O(m)$ operations. Hence the total complexity of the pricing subproblem heuristic is $O(n\log n + nm)$ time.

As will be seen in the experiments, our main algorithm provides almost the same quality in the solution when it employs this simple heuristic or an exact IP-based algorithm to solve the pricing subproblem.

5.3. Main algorithm

Our algorithm derives a feasible solution by rounding the fractional optimal solution λ^* of the LP-relaxation of (11)–(14). More specifically, we fix a promising schedule of a segment $p = (q,r)$ based on the fractional solution. (Recall that each variable of (11)–(14) corresponds to a schedule of the trains relevant to a segment.) Then the procedure is repeated for the reduced problem after *regenerating* it, namely after adjusting the earliest departure time of train i at station r so that the minimum dwelling time is met: $b^i_{p'} \leftarrow \max\left\{b^i_{p'}, a^i_p + s^i_r\right\}$, where p' is the segment the train i traverses right after p (hence both p and p' are incident to r). This fixing and regenerating steps are repeated for the segments in their topological order so that the feasibility of the timetable is readily maintained. Our main algorithm can be stated as follows.

Algorithm 5.3. Fix-and-regenerate algorithm (FRA)

Step 1: Let (11)–(14) be the current problem. In a topological order of segments p in the subnetwork downstream to the source segment of conflict, repeat Steps 2 and 3.
Step 2: Solve the LP-relaxation of the current problem by the column-generation method to obtain a fractional solution λ^* (as discussed in Section 5.1).
Step 3: Let $\lambda^*_{p,t'}$ be the largest among $\lambda^*_{p,t}$ from segment p. Then fix the corresponding timetable $\tau_{p,t'}$ for segment p. Let the regenerated problem be the current problem.
Step 4: (**Postprocessing**) As mentioned in Section 5.1, the dwelling mode of every train is initially set to "stop" at every station. If the arrival and departure times of a train are the same at a station, then modify the dwelling mode of the train as "pass" at the station. Accordingly, set the values of integer variables, namely, track assignments, dispatching sequences of trains, and dwelling modes in the MIP-model (1)–(10). Solve the resulted LP to recompute the continuous values, departure, arrival, and dwelling times.

We can illustrate the flow of algorithm as in Fig. 3.

In Step 2, we need to generate an initial feasible solution to construct RLPM in solving the LP-relaxation. As we perform a heuristic instead of exact pricing, we get a suboptimal solution of the LP-relaxation, in general, depending on the initial solution. To enhance the optimality, we may use multiple initial solutions for the same LP-relaxation. We refer to such scheme as a *partial branching*.

In the experiment, we used two initial solutions for partial branching. The first initial solution is obtained by fixing the timetable by the pricing heuristic setting $(\pi,\mu) = (0,0)$. This is repeated in a topological order of segments by regenerating the

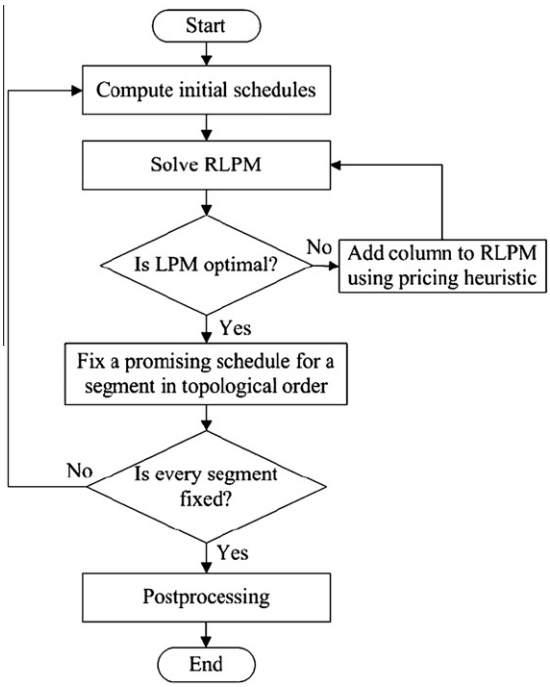

Fig. 3. Flow diagram of the fix-and-regenerate algorithm.

problem after fixing each timetable. The second initial solution is obtained from the first initial solution by exchanging the departing orders of the most delayed train and its successor or predecessor, whoever departure time is the closest to the departure time of the most delayed train from the first segment. The second solution is motivated by the observation that the delay in the first segment has the most impact on the schedules of trains of the subsequent segments.

6. Experimental results

In this section, the fix-and-regenerate algorithm is tested on the real-world instances from the Seoul metropolitan railway network. The performance is evaluated in comparison with two previous algorithms, by Törnquist and Persson (2007) and Sahin (1999). We also compare the performance of the two versions of the fix-and-regenerate algorithm: one employing the heuristic in Section 5.2 and the other one using an exact IP-based method for the pricing subproblem.

6.1. Instance description

The railway network consists of 23 stations and 22 segments. Among the segments, 5 are triple-tracked and 17 are double-tracked in each direction as illustrated in Fig. 4. This network spans 75.4 km in the Seoul metropolitan area and the average traffic is 66 trains per hour in each direction of a segment.

There are three types of trains: passenger trains, subway trains, and freight trains. From a consultation with KORAIL, they are assigned the costs of a unit deviation 1.5, 1.2, and 1.0, respectively. For all stations, the inter-track safe headway time is fixed to be 0.5 min. But the intra-track safe headway time varies from 2.5 min to 6.0 min. Each train has a lower bound on its transit time for each segment. But, the upper bound for transit time is not specified in the practice of KORAIL. Hence, it will be set to infinity for every segment in our experiment.

Additional parameters in designing instances are summarized in Table 3. First, as each track is unidirectional, we consider the instances consisting only of inbound trains to or outbound trains from Seoul. Inbound trains travel from Pyongtac or Incheon to Seoul, and outbound trains go in the opposite direction (Fig. 4). Second, we consider four times horizons: 30, 60, 90, and 120 min. The running times of all trains in the Seoul metropolitan area are within 120 min. Also, the length of the time horizon that the operator (in the control center) considers during conflict resolution is between 30 and 60 min. Thus KORAIL asked us to use 120 min for exploiting the advantage of the automated resolution procedure. Third, each time horizon starts at 8:00 AM, 9:00 AM, noon, 1:00 PM, 6:00 PM, or 7:00 PM according to a daily timetable. For example, an instance can consist of the inbound trains whose original arrival and departure times at stations fall in the 120 min interval starting at 8:00 AM. We generate the sources of train-conflicts by delaying the transit time of a train over a segment by 5, 10, 15, 20, 25, 30, 35, and 40 min. For the 30 and 60 min horizon instances, however, the delays longer than 30 min are not used.

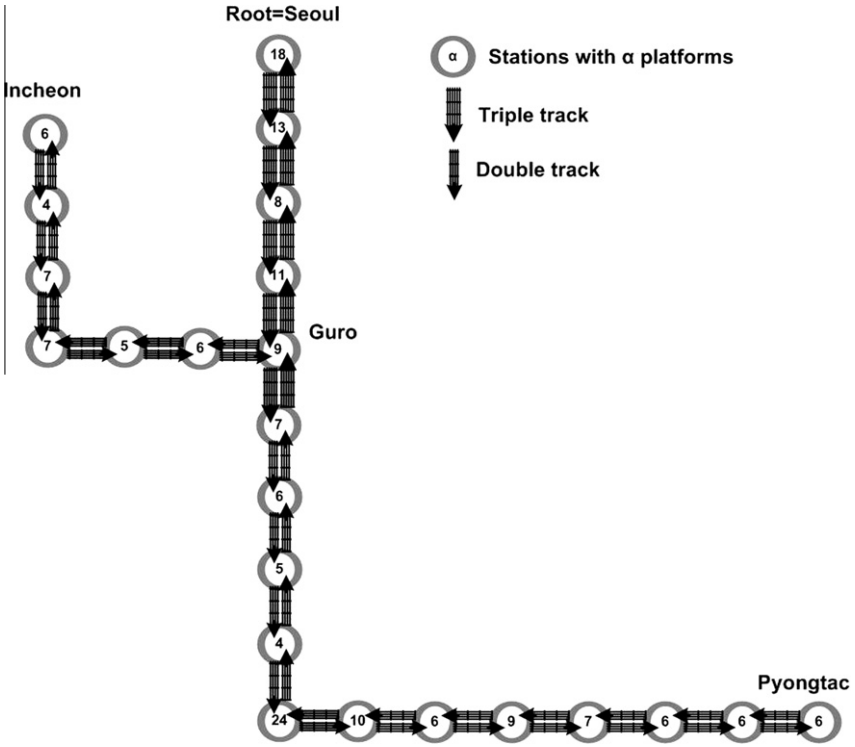

Fig. 4. Railway network of Seoul metropolitan.

Table 3
Instance parameters.

Directions	Lengths of time horizon (min)	Starting times of time horizon	Delays (min)
Inbound Outbound	30, 60, 90, 120	8 AM, 9 AM, noon, 1 PM, 6 PM, 7 PM	5, 10, 15, 20, 25, 30, 35, 40

6.2. Comparison with Törnquist and persson's algorithms

On the above instances, the performance of the fix-and-regenerate algorithm, or FRA, is compared to the ones of the previous methods proposed by Törnquist and Persson (2007). Their methods add the linear constraints restricting solutions to a set of routing strategies to an MIP-formulation. Then they solve the resulted MIP by a commercial code. The model of Törnquist and Persson (2007) does not exactly match with ours as their model can consider the bidirectional tracks and the platform assignments at stations but cannot consider the inter-track headway times and the dwelling mode of trains at stations. Their idea, however, can be applied to our model since the routing strategies used in Törnquist and Persson (2007) are independent of the particular differences of two models. Of the routing strategies from Törnquist and Persson (2007), we choose the following two which have been reported to yield near-optimal solutions in very reasonable computation times on their real instances.

6.2.1. Törnquist and Persson's strategy II (TP2)

Strategy II only considers the timetables in which the trains assigned to the same track should observe the original dispatching sequence. For example, suppose train i departs segment p earlier than train j in the original timetable. If train i and j use the same track of segment p in the new timetable, then train i must depart segment p earlier than train j. This strategy can be expressed as the following constraints:

$$2 - Z^i_{p,k} - Z^j_{p,k} + X^{ij}_p \geqslant 1, \quad \forall p \in P, \quad \forall i,j \in I_p, \quad d^i_p < d^j_p, \quad k = 1,\ldots,n_p.$$

6.2.2. Törnquist and Persson's strategy III (TP3)

Strategy III is identical to the previous strategy except that the conflict-source (hence the initially delayed) train does not have to observe the original dispatching sequence with directly subsequent θ trains. Let i^* denote the index of the

conflict-source train, and Ω_p be a set of θ trains that travel segment p immediately after i^*. Then strategy III can be expressed as the following constraints:

$$2 - Z^i_{p,k} - Z^j_{p,k} + X^{ij}_p \geqslant 1, \quad \forall p \in P, \quad i \neq i^*, \quad \forall j \in I_p, \quad d^i_p < d^j_p, \quad k = 1, \ldots, n_p,$$
$$2 - Z^i_{p,k} - Z^j_{p,k} + X^{ij}_p \geqslant 1, \quad \forall p \in P, \quad i = i^*, \quad \forall j \notin \Omega_p, \quad d^i_p < d^j_p, \quad k = 1, \ldots, n_p.$$

Törnquist (2007) developed an extended version of Strategy III by using the different set of Ω_p, a set of events for a conflict-source train and those that are immediately affected by the conflict-source train that travel segment p (see Definition 4.1 in Törnquist (2007)). However, in our instance, all tracks are unidirectional and all segments have multiple tracks. Therefore, when a conflict occurs, a subsequent train may use a different track. The resulting delay immediately affected is not significant. Hence we did not use the extended version. In the experiments, we set θ to be ∞ for a conservative evaluation and $\{2, \infty\}$ for a practical evaluation.

TP2 and TP3 coded in OPL studio 5.5 were executed on a desktop PC with an Intel Quad 2.50 GHz CPU and 4.0 GB RAM. FRA coded in C using CPLEX 9.0 solver for the LP subroutines was executed on the same machine.

6.2.3. A conservative evaluation

For a conservative performance evaluation of FRA, we first compared its solutions to those obtained by running TP2 and TP3 for 120 min. (FRA was run to its terminations.) We set θ to be ∞ because the quality of TP3 improves as θ increases. Table 4 reports the distribution of the relative errors $(z - \text{OPT})/\text{OPT}$ of the objective values z of the solutions provided by each of FRA, TP2, and TP3 to an optimal value OPT, obtained by the CPLEX 9.0 solver applied to the MIP-formulation (1)–(10) in 120 min. Each row represents a relative error interval and each column represents the percentage of instances whose solutions have relative errors falling into the interval. For example, the entry of the column "FRA" and the row "0–10%" of the table for the 30 min horizon shows that FRA produces for 73.6% of the instances a solution whose relative error to the optimal value is less than 10%. The row "Feasible Sol. N/A" shows the percentage of instances for which each method cannot find a feasible solution in the time limit. In case the CPLEX solver could not find an optimal solution in 120 min, we used, instead, the largest LP lower bound available. (Hence the relative gaps presented can be more conservative than the actual values for all methods, FRA, TP2, and TP3.)

For all time horizons, FRA performs better in solution quality than TP2. On the other hand, for small time horizons (30 and 60 min), TP3 produces better solutions than FRA. But, FRA provides better solutions than TP3 for larger horizons (90 and 120 min). Furthermore, while FRA always returns a feasible solution, TP2 and TP3 cannot find a feasible solution for more than 40% of the cases for larger time horizons. Therefore, in the conservative evaluation, FRA produces better solutions than TP2 in every time horizon, while it is outperformed by TP3 in the small time horizons such as 30 and 60 min.

6.2.4. A practical evaluation

In practice, the running time of a conflict resolution algorithm is restricted to a certain range. We set the upper bound to be a single minute after consulting with KORAIL.

As described in Section 5.3, FRA does not fully exploit the dwelling mode variables from (8)–(10). This strategy may result in a sub-optimality of solution while it may save the computation time significantly. We also adopt an analogy of this strategy for TP2 and TP3. In other words, we apply TP2 and TP3 to the MIP-formulation (1)–(10) in which the dwelling mode of

Table 4
Comparison of FRA and the algorithm by Törnquist and Persson (2007).

Gap	Time horizon of 30 min			Time horizon of 60 min		
	FRA	TP2	TP3	FRA	TP2	TP3
0–10%	73.6%	45.8%	80.6%	41.7%	30.6%	51.4%
10–20%	9.7%	16.7%	9.7%	11.1%	12.5%	6.9%
20–30%	9.7%	8.3%	6.9%	12.5%	1.4%	11.1%
30–40%	2.8%	11.1%	1.4%	11.1%	0.0%	6.9%
40–50%	0.0%	9.7%	0.0%	4.2%	1.4%	1.4%
Over 50%	4.2%	8.3%	1.4%	19.4%	52.8%	18.1%
Feasible Sol. N/A	0.0%	0.0%	0.0%	0.0%	1.4%	4.2%
Avg. time (s)	0.7	3451.1	1595.3	2.5	5929.1	6225.4
	Time horizon of 90 min			Time horizon of 120 min		
0–10%	37.5%	7.3%	20.8%	21.9%	3.1%	7.3%
10–20%	16.6%	5.2%	13.5%	29.2%	2.1%	4.2%
20–30%	9.4%	2.1%	0.0%	9.4%	3.1%	1.0%
30–40%	9.4%	4.2%	2.1%	10.4%	1.0%	1.0%
40–50%	7.3%	2.1%	4.2%	4.2%	0.0%	2.1%
Over 50%	19.8%	33.3%	17.7%	25.0%	12.5%	10.4%
Feasible Sol. N/A	0.0%	45.8%	41.7%	0.0%	78.1%	74.0%
Avg. time (s)	5.9	7043.4	6565.4	14.8	7200.0	6985.2

every train is fixed so that it stops at every station. Then, the returned solution is improved by the same postprocessing as in FRA. We call these two modifications, TP2′ and TP3′, respectively.

Thus, in this practical evaluation, we compare FRA with TP2′ and TP3′ with a single minute computation time. In a preliminary experiment, TP3′ outperformed TP2′ for most instances, and hence TP2′ was excluded in the experiment. The computation time of TP3′ decreases if a small value of θ is used. Thus, we also experimented the case of $\theta = 2$ along with $\theta = \infty$.

Table 5–8 show the experimental results on the instances consisting of outbound trains. Each ordered pair from the first column represents the starting time of the horizon and the duration of the delay causing a train conflict. The second column indicates the lower bound or optimum for each instance. Notice that the boldface objective value means it is at least as good as the objective values returned by the other methods.

Table 5 shows the experimental results on the instances with the time horizon of 30 min. FRA returns a better objective value for eight instances while TP3′ obtained a better objective value for 17 instances (12 for $\theta = \infty$ and 5 for $\theta = 2$). However, the average running time of FRA is 0.7 s while the average running time of TP3′ is about 40 s. Thus, overall, FRA and TP3′ show a trade-off between computation time and objective value on the instances with the 30 min time horizon.

Table 6 shows the experimental results on the instances with the time horizon of 60 min. TP3′ could not find a feasible solution for 23 instances with $\theta = \infty$ and only two instances with $\theta = 2$. However, FRA was able to find feasible solutions for all instances and produced better solutions than those of TP3′ except for four instances. The average running time of FRA is 2.3 s, whereas the average running time of TP3′ is 55 s. Thus for the 60 min horizon instances, FRA outperformed TP3′.

Table 5
Comparison of FRA and TP3′ on the time horizon of 30 min.[a]

Instance	LB or OPT	FRA (1 min)		TP3′ − θ (1 min)			
		Obj.	Time (s)	$\theta = \infty$		$\theta = 2$	
				Obj.	Time (s)	Obj.	Time (s)
(08,05)	53.8	64.1	0.8	60.8	60.0	**58.4**	40.0
(08,10)	105.6	109.5	0.7	**105.6**	30.7	110.8	47.9
(08,15)	162.6	166.5	0.7	**163.8**	60.0	172.6	60.0
(08,20)	216.4	222.7	0.7	**218.8**	60.9	232.4	60.0
(08,25)	267.6	271.5	0.7	**267.6**	19.1	282.8	44.4
(08,30)	321.6	325.5	0.8	**321.6**	24.2	336.8	54.4
(09,05)	53.4	111.0	0.8	231.0	60.0	**80.5**	60.0
(09,10)	105.4	**183.6**	0.8	185.9	60.0	239.9	60.0
(09,15)	159.5	184.2	0.7	**176.0**	60.0	219.4	60.0
(09,20)	213.8	**230.6**	0.7	259.1	60.0	274.4	60.0
(09,25)	266.4	**280.7**	0.8	285.9	60.0	377.0	60.0
(09,30)	320.4	**334.7**	0.8	336.0	60.0	399.7	60.0
(12,05)	55.0	**55.0**	0.6	**55.0**	5.1	**55.0**	19.4
(12,10)	109.0	**109.0**	0.5	**109.0**	4.2	**109.0**	7.6
(12,15)	162.1	**162.1**	0.5	**162.1**	4.3	**162.1**	6.6
(12,20)	217.8	**217.8**	0.6	**217.8**	4.2	**217.8**	12.2
(12,25)	270.0	**270.0**	0.6	**270.0**	3.2	**270.0**	0.6
(12,30)	324.0	**324.0**	0.6	**324.0**	4.2	**324.0**	0.7
(13,05)	42.0	43.2	0.6	**42.0**	6.1	**42.0**	2.7
(13,10)	85.4	102.9	0.6	91.0	60.0	**89.9**	17.2
(13,15)	127.4	134.2	0.6	**131.4**	60.0	134.9	19.7
(13,20)	168.0	178.7	0.6	**169.2**	60.0	175.0	15.9
(13,25)	211.2	212.7	0.6	**211.2**	60.0	217.0	17.7
(13,30)	252.0	254.7	0.6	**253.2**	18.9	259.0	16.7
(18,05)	42.0	**42.0**	0.9	**42.0**	27.2	**42.0**	2.2
(18,10)	87.9	110.8	0.8	**97.3**	60.0	101.7	60.0
(18,15)	126.0	148.7	0.9	**138.9**	60.0	156.3	60.0
(18,20)	168.0	**171.8**	0.8	173.9	60.0	199.3	60.0
(18,25)	213.7	**213.8**	0.7	215.9	60.0	240.4	60.0
(18,30)	252.0	**255.8**	0.8	257.9	60.0	282.4	60.0
(19,05)	49.3	49.6	0.7	-	60.0	**49.4**	29.4
(19,10)	104.0	117.7	0.8	117.1	60.0	**113.3**	60.0
(19,15)	156.8	**156.8**	0.8	159.6	60.0	171.9	60.0
(19,20)	211.8	**211.8**	0.8	**211.8**	20.1	224.7	60.0
(19,25)	264.8	**264.8**	0.7	**264.8**	12.0	276.5	60.0
(19,30)	318.8	**318.8**	0.8	**318.8**	13.9	330.5	60.0
Average			0.7		40.5		39.9

-: Feasible solution N/A.
[a] Each timetable consists of outbound trains: 36 instances in total.

Table 6
Comparison of FRA and TP3′ on the time horizon of 60 min.[a]

Instance	LB or OPT	FRA (1 min)		TP3′ − θ (1 min)			
		Obj.	Time (s)	$\theta = \infty$		$\theta = 2$	
				Obj.	Time (s)	Obj.	Time (s)
(08,05)	91.0	**100.8**	3.7	–	60.0	237.4	60.0
(08,10)	202.8	**206.9**	3.5	–	60.0	304.5	60.0
(08,15)	316.8	**332.1**	3.6	–	60.0	685.7	60.0
(08,20)	432.8	**460.6**	3.6	–	60.0	867.2	60.0
(08,25)	544.8	**568.0**	3.6	–	60.0	–	60.0
(08,30)	658.8	**662.9**	3.7	–	60.0	1126.8	60.0
(09,05)	76.2	**210.7**	3.8	–	60.0	418.0	60.0
(09,10)	179.8	**378.8**	3.7	–	60.0	732.8	60.0
(09,15)	280.0	**336.8**	3.7	–	60.0	805.6	60.0
(09,20)	382.4	**432.1**	3.6	–	60.0	967.6	60.0
(09,25)	483.6	**504.8**	3.5	–	60.0	912.5	60.0
(09,30)	588.6	**631.4**	3.5	–	60.0	1762.2	60.0
(12,05)	91.0	**91.4**	1.7	–	60.0	**91.4**	56.2
(12,10)	181.4	**181.4**	1.8	**181.4**	23.3	**181.4**	41.7
(12,15)	270.3	270.5	1.7	**270.3**	22.5	**270.3**	24.4
(12,20)	371.8	**371.8**	1.6	**371.8**	39.0	**371.8**	58.1
(12,25)	450.0	**450.0**	1.6	**450.0**	31.9	**450.0**	47.6
(12,30)	540.0	**540.0**	1.6	**540.0**	28.8	548.4	58.2
(13,05)	90.0	105.4	2.1	**94.0**	60.0	102.1	60.0
(13,10)	193.4	246.5	2.0	773.4	60.0	**244.5**	60.0
(13,15)	298.2	**335.8**	2.0	633.7	60.0	444.1	60.0
(13,20)	396.0	**426.7**	2.1	–	60.0	457.2	60.0
(13,25)	498.0	516.7	2.0	**508.3**	60.0	576.0	60.0
(13,30)	601.0	**617.3**	2.2	623.8	60.0	683.6	60.0
(18,05)	78.0	**111.2**	2.8	–	60.0	260.8	60.0
(18,10)	156.0	**225.1**	2.9	512.2	60.0	305.6	60.0
(18,15)	234.0	**350.7**	2.9	–	60.0	536.9	60.0
(18,20)	312.0	**425.9**	3.1	535.0	60.0	1461.5	60.0
(18,25)	391.6	**495.5**	2.9	675.5	60.0	656.1	60.0
(18,30)	468.0	**577.9**	3.0	–	60.0	1950.2	60.0
(19,05)	92.9	**107.1**	3.5	–	60.0	193.8	60.0
(19,10)	210.1	**289.1**	3.6	–	60.0	482.9	60.0
(19,15)	320.9	**330.2**	3.2	–	60.0	717.7	60.0
(19,20)	435.3	**466.8**	3.3	–	60.0	915.3	60.0
(19,25)	548.9	**624.4**	3.3	–	60.0	1270.0	60.0
(19,30)	662.9	**668.6**	3.2	–	60.0	–	60.0
Average			2.9		55.7		57.9

-: Feasible solution N/A.
[a] Each timetable consists of outbound trains: 36 instances in total.

Tables 7 and 8 show the experimental results for the 90 and 120 min time horizons, respectively, in which we could perform experiments for more variety of source delays due to their longer horizon. The number of instances is 48 for each time horizon. As we can see, the performance difference gets bigger for longer time horizons. With $\theta = \infty$, TP3′ had a difficulty in finding a feasible solution for these time horizons. With $\theta = 2$, it could obtain more feasible solutions than with $\theta = \infty$, but the solution quality was inferior to that of FRA.

Table 9 recaptures this single-minute practical evaluation for all instances in the format of Table 4. In addition, in the row "Best", we specified the ratio of the instances for which each algorithm returned a solution with a strictly lower objective value than others. Also, the standard deviations of the computation times are presented. The results are, as expected, compatible to the ones from the previous tables. Even for the instances with the time horizon of 30 min, the relative errors of the solutions provided by FRA and TP3′ do not show a big difference.

Regarding the computation time, FRA terminated in less than 1 min for all time horizons. Especially, the computation times not only had small variances but also did not increase too fast as the time horizon gets larger. This appears consistent to the $O(n\log n + mn)$ polynomial computation time of the pricing heuristic analyzed in Section 5.2.

Fig. 5 illustrates the difference between timetables obtained by TP3′($\theta = 2$) and FRA for the second instance in Table 6. (a) of Fig. 5 is the time-distance graph that corresponds to a part of the original timetable. A train represented by the dashed line is delayed by 10 min. This conflict is resolved by TP3′ and FRA, and the resulting time-distance graphs are shown in (b) and (c) of Fig. 5, respectively.

In sum, we may conclude that the fix-and-regenerate algorithm outperforms Törnquist and Persson's algorithms in both solution quality and computation time. Table 10 compares our instances with the time horizon of 60 min to the instances

Table 7
Comparison of FRA and TP3′ on the time horizon of 90 min.[a]

Instance	LB	FRA(1 min)		TP3′ − θ (1 min)			
		Obj.	Time (s)	$\theta = \infty$		$\theta = 2$	
				Obj.	Time (s)	Obj.	Time (s)
(08,05)	103.2	**115.6**	12.3	–	60.0	–	60.0
(08,10)	241.2	**246.2**	11.9	–	60.0	–	60.0
(08,15)	379.2	**448.8**	12.0	–	60.0	–	60.0
(08,20)	517.2	**613.7**	11.6	–	60.0	–	60.0
(08,25)	655.2	**723.0**	12.0	–	60.0	–	60.0
(08,30)	793.2	**797.7**	11.7	–	60.0	–	60.0
(08,35)	931.2	**935.3**	10.4	–	60.0	–	60.0
(08,40)	1057.8	**1151.8**	10.6	–	60.0	–	60.0
(09,05)	90.8	**276.5**	6.9	–	60.0	–	60.0
(09,10)	230.1	**541.8**	7.0	–	60.0	–	60.0
(09,15)	366.6	**460.2**	6.8	–	60.0	4788.7	60.0
(09,20)	504.6	**563.8**	6.8	–	60.0	–	60.0
(09,25)	642.6	**681.7**	6.8	–	60.0	–	60.0
(09,30)	783.6	**835.7**	6.8	–	60.0	–	60.0
(09,35)	918.6	**975.1**	6.8	–	60.0	–	60.0
(09,40)	1057.8	**1124.2**	7.1	–	60.0	–	60.0
(12,05)	115.0	**119.0**	3.6	–	60.0	151.3	60.0
(12,10)	229.4	**233.0**	3.6	329.7	60.0	274.3	60.0
(12,15)	342.0	**346.1**	3.5	–	60.0	406.1	60.0
(12,20)	471.6	**489.3**	3.5	630.0	60.0	526.8	60.0
(12,25)	570.0	**609.1**	3.5	–	60.0	1206.8	60.0
(12,30)	684.0	**687.6**	3.6	–	60.0	864.8	60.0
(12,35)	798.0	**812.3**	3.3	869.4	60.0	981.6	60.0
(12,40)	912.0	**915.6**	3.2	1047.6	60.0	1107.5	60.0
(13,05)	147.6	**207.9**	3.8	–	60.0	394.5	60.0
(13,10)	325.3	**443.6**	3.9	–	60.0	579.1	60.0
(13,15)	506.5	**547.9**	3.8	–	60.0	710.7	60.0
(13,20)	669.6	**732.2**	4.1	–	60.0	774.8	60.0
(13,25)	843.6	**904.5**	3.8	–	60.0	–	60.0
(13,30)	1017.6	**1110.6**	3.8	–	60.0	1285.4	60.0
(13,35)	1192.1	**1295.9**	3.8	–	60.0	–	60.0
(13,40)	1371.9	**1616.3**	3.6	–	60.0	2126.6	60.0
(18,05)	114.0	**151.3**	8.1	–	60.0	4600.4	60.0
(18,10)	228.0	**312.7**	8.2	–	60.0	–	60.0
(18,15)	342.0	**438.3**	8.5	–	60.0	–	60.0
(18,20)	457.6	**556.7**	8.4	–	60.0	–	60.0
(18,25)	570.0	**641.8**	8.8	–	60.0	–	60.0
(18,30)	684.0	**748.3**	8.7	–	60.0	–	60.0
(18,35)	798.0	**927.2**	7.6	–	60.0	–	60.0
(18,40)	912.0	**975.0**	7.2	–	60.0	–	60.0
(19,05)	112.5	**141.5**	7.3	–	60.0	2901.0	60.0
(19,10)	251.5	**380.5**	7.4	–	60.0	–	60.0
(19,15)	386.5	**431.0**	8.1	–	60.0	–	60.0
(19,20)	527.9	**775.0**	7.1	–	60.0	–	60.0
(19,25)	662.5	**1026.0**	8.3	–	60.0	–	60.0
(19,30)	800.5	**844.8**	7.0	–	60.0	–	60.0
(19,35)	938.5	**985.5**	6.0	–	60.0	–	60.0
(19,40)	1076.5	**1171.4**	6.2	–	60.0	–	60.0
Average			6.9		60.0		60.0

-: Feasible solution N/A.
[a] Each timetable consists of outbound trains: 48 instances in total.

with the same horizon on which TP3 was tested in Törnquist (2007). Each ordered pair from the second column represents the direction of segment(O: outbound, I: inbound) and the starting time of the horizon.

The instances in Törnquist (2007) have more segments but less trains than ours. The total number of events, namely the total number of trains over segments is larger in the instances in Törnquist (2007) than in ours. The encoding length of an instance is proportional to the total number of events. In this sense, the instances from Törnquist (2007) is larger than ours.

However, according to Törnquist (2007), the computation times of TP3 applied to the instances of Törnquist (2007) are all less than 10 s, while the practical version of TP3, namely TP3′, could not find even a feasible solution in 60 s for most of our instances with 90 and 120 min time horizons. From our experiments, the most critical factor in the intractability of an instance seems to be the number of binary variables in the MIP-formulation. (This is plausible in the sense that an IP commercial code is an enumerative method.) The number of binary variables is

Table 8
Comparison of FRA and TP3′ on the time horizon of 120 min.[a]

Instance	LB	FRA (1 min)		TP3′ − θ (1 min)			
		Obj.	Time (s)	$\theta = \infty$		$\theta = 2$	
				Obj.	Time (s)	Obj.	Time (s)
(08,05)	103.2	**120.1**	31.8	–	60.0	–	60.0
(08,10)	241.2	**252.8**	32.4	–	60.0	–	60.0
(08,15)	379.2	**498.0**	32.8	–	60.0	–	60.0
(08,20)	517.2	**638.5**	31.7	–	60.0	–	60.0
(08,25)	655.2	**757.3**	31.1	–	60.0	–	60.0
(08,30)	793.2	**802.2**	32.2	–	60.0	–	60.0
(08,35)	931.2	**940.0**	24.9	–	60.0	–	60.0
(08,40)	1069.2	**1196.0**	25.1	–	60.0	–	60.0
(09,05)	90.6	**286.6**	15.6	–	60.0	–	60.0
(09,10)	230.6	**544.6**	19.2	–	60.0	–	60.0
(09,15)	366.6	**463.0**	19.3	–	60.0	–	60.0
(09,20)	504.6	**566.6**	16.5	–	60.0	–	60.0
(09,25)	642.6	**684.5**	16.5	–	60.0	–	60.0
(09,30)	780.6	**840.9**	16.6	–	60.0	–	60.0
(09,35)	918.6	**996.9**	12.7	–	60.0	–	60.0
(09,40)	1056.6	**1221.8**	12.6	–	60.0	–	60.0
(12,05)	114.0	**117.9**	9.3	–	60.0	2629.8	60.0
(12,10)	228.0	**232.0**	10.0	–	60.0	1862.4	60.0
(12,15)	342.0	**345.0**	9.6	–	60.0	425.7	60.0
(12,20)	456.0	**488.3**	9.2	–	60.0	–	60.0
(12,25)	570.0	**608.1**	9.3	–	60.0	2883.3	60.0
(12,30)	684.0	**686.5**	9.4	–	60.0	2754.1	60.0
(12,35)	798.0	**814.7**	7.8	–	60.0	2896.1	60.0
(12,40)	912.0	**914.5**	7.6	–	60.0	–	60.0
(13,05)	156.6	**216.0**	8.7	–	60.0	653.9	60.0
(13,10)	345.1	**475.8**	9.3	–	60.0	2403.3	60.0
(13,15)	530.6	**584.0**	8.5	–	60.0	830.0	60.0
(13,20)	714.6	**785.3**	8.3	–	60.0	–	60.0
(13,25)	900.6	**1007.0**	8.3	–	60.0	–	60.0
(13,30)	1086.6	**1221.4**	8.2	–	60.0	3999.7	60.0
(13,35)	1272.6	**1424.9**	6.9	–	60.0	–	60.0
(13,40)	1458.6	**1844.2**	6.9	–	60.0	–	60.0
(18,05)	114.0	**154.7**	8.8	–	60.0	–	60.0
(18,10)	228.0	**340.2**	8.9	–	60.0	–	60.0
(18,15)	342.0	**436.7**	9.0	–	60.0	–	60.0
(18,20)	456.0	**571.8**	9.1	–	60.0	–	60.0
(18,25)	570.0	**652.8**	9.0	–	60.0	–	60.0
(18,30)	684.0	**762.1**	9.1	–	60.0	–	60.0
(18,35)	798.0	**1002.6**	8.9	–	60.0	–	60.0
(18,40)	912.0	**986.1**	9.0	–	60.0	–	60.0
(19,05)	110.5	**154.8**	18.1	–	60.0	–	60.0
(19,10)	248.5	**399.4**	18.1	–	60.0	–	60.0
(19,15)	386.5	**443.0**	17.5	–	60.0	–	60.0
(19,20)	524.5	**930.3**	19.3	–	60.0	–	60.0
(19,25)	662.5	**1152.2**	18.3	–	60.0	–	60.0
(19,30)	800.5	**911.9**	18.2	–	60.0	–	60.0
(19,35)	938.5	**1053.3**	15.3	–	60.0	–	60.0
(19,40)	1069.2	**1213.5**	15.1	–	60.0	–	60.0
Average			15.0		60.0		60.0

-: Feasible solution N/A.
[a] Each timetable consists of outbound trains: 48 instances in total.

$$\sum_{p\in P} |I_p| \times n_p + |I_p|(|I_p| - 1).$$

In the table, we reported the average number of binary variables of an instance from Törnquist (2007) and ours, that are 2154 and 6401, respectively. (From our instances, we excluded the number of binary variables representing the dwelling modes of trains at stations.)

6.3. Comparison with Sahin's algorithm

While FRA or the algorithms of Törnquist and Persson are based on an MIP-formulation, some heuristic methods aim at obtaining minimal computation by adopting a pragmatic algorithm design principle. Sahin's algorithm (Sahin, 1999) is such

Table 9
FRA and TP3′ with regard to ratios to optimal objective values.

Gap	Time horizon of 30 min			Time horizon of 60 min		
	FRA	TP3′ − θ (1 min)		FRA	TP3′ − θ (1 min)	
		$\theta = \infty$	$\theta = 2$		$\theta = \infty$	$\theta = 2$
0–10%	73.6%	72.2%	68.1%	41.7%	30.6%	19.4%
10–20%	9.7%	12.5%	11.1%	11.1%	4.2%	8.3%
20–30%	9.7%	8.3%	11.1%	12.5%	5.6%	2.8%
30–40%	2.8%	1.4%	4.2%	11.1%	1.4%	4.2%
40–50%	0.0%	0.0%	1.4%	4.2%	1.4%	4.2%
Over 50%	4.2%	4.2%	4.2%	19.4%	25.0%	58.3%
Feasible Sol. N/A	0.0%	**1.4%**	0.0%	0.0%	**31.9%**	**2.8%**
Best	58.3%	80.6%	47.2%	79.2%	26.4%	15.3%
Avg. time (s)	0.7	31.4	28.8	2.5	55.7	56.5
Std. dev. (s)	0.1	24.1	21.7	0.8	10.6	11.6
	Time horizon of 90 min			Time horizon of 120 min		
0–10%	37.5%	1.0%	1.0%	21.9%	0.0%	0.0%
10–20%	16.6%	1.0%	4.2%	29.2%	0.0%	0.0%
20–30%	9.4%	3.1%	5.2%	9.4%	0.0%	1.0%
30–40%	9.4%	4.2%	3.1%	10.4%	0.0%	0.0%
40–50%	7.3%	6.3%	5.2%	4.2%	0.0%	0.0%
Over 50%	19.8%	28.1%	29.2%	25.0%	0.0%	11.5%
Feasible Sol. N/A	0.0%	**56.3%**	**52.1%**	0.0%	**100.0%**	**87.5%**
Best	97.9%	1.0%	1.0%	100.0%	0.0%	0.0%
Avg. time (s)	5.9	60.0	59.7	13.6	60.0	60.0
Std. dev. (s)	2.4	0.0	3.1	6.6	0.0	0.0

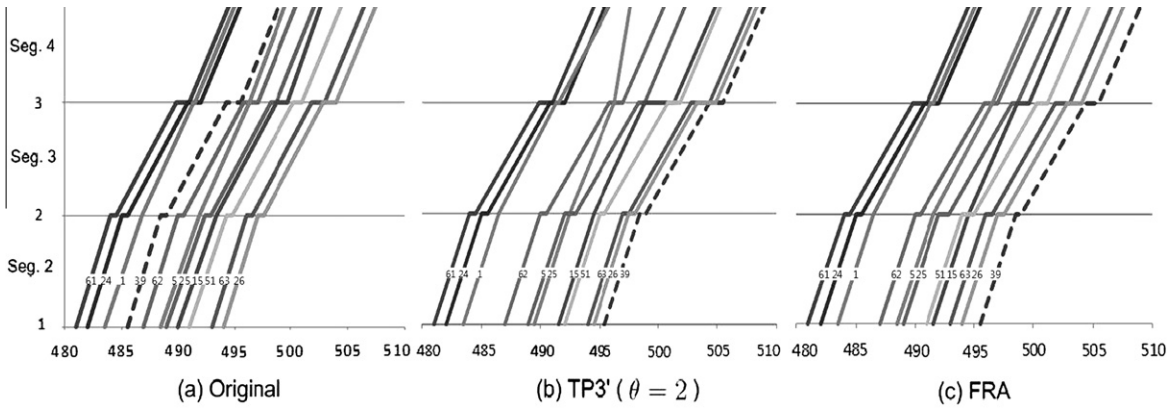

Fig. 5. A time–distance graph.

heuristic and uses a train-conflict resolution model that is the closest to ours. In this subsection, we perform a comparison test of FRA and Sahin's algorithm. The model in Sahin (1999) uses the same objective function as ours, but assumes only single-track segments between stations. Also, it allows a bidirectional track, but does not consider the platform assignment or the dwelling modes of a train at a station.

In the main step for sequencing trains on a segment, Sahin's algorithm uses a lookahead function for sequencing trains. More specifically, the algorithm constructs a conflict resolving timetable as follows. The input of the algorithm is the timetable obtained by updating the departure or arrival time of the train of the source delay in the original timetable. Then we look up the earliest conflict in the timetable and the two trains, say i and j, involved in the conflict. Suppose that the conflict occurs at segment p. Here, the algorithm considers only the option of shifting the schedule of train i or j (but, not both). Let Δ_i (Δ_j) be the minimum amount of time by which the schedule of train i (j, respectively) at segment p should be shifted to resolve the single conflict on the single segment p. The algorithm considers two timetables obtained by shifting the train i by Δ_i and train j by Δ_j, respectively, from the original timetable. Then, it computes, for each timetable, a lookahead function defined by a weighted sum of the following factors:

(1) the weight of train i,
(2) Δ_i,

Table 10
Instance description for the time horizon of 60 min in ours and Törnquist (2007).

No.	Instance	Trains	Segments	Events	Frequency[a]	Binary variables	Continuous variables
1	(O,08)	74	22	367	16.7	8387	1101
2	(O,09)	71	22	376	17.1	8515	1128
3	(O,12)	57	22	281	12.8	4502	843
4	(O,13)	60	22	310	14.1	5529	930
5	(O,18)	74	22	370	16.8	7843	1110
6	(O,19)	73	22	369	16.8	7820	1107
7	(I,08)	66	22	299	13.6	5667	897
8	(I,09)	65	22	315	14.3	5468	945
9	(I,12)	55	22	273	12.4	4275	819
10	(I,13)	55	22	298	13.5	5111	894
11	(I,18)	75	22	361	16.4	7719	1083
12	(I,19)	69	22	313	14.2	5977	939
Average in ours		66.2	22.0	327.7	14.9	6401.1	983.1
Average in Törnquist (2007)[b]		34.4	105.7	349.1	3.3	2153.7	1047.3

[a] Frequency = # of events/# of segments.
[b] Average of 40 instances for the time horizon of 60 min in Table 3 of Törnquist (2007).

(3) (Original arrival time of train i on the last segment − new departure time of train i on the segment p)/(the sum of the minimum transit times of train i from the segment p to the last segment), and
(4) the number of remaining conflicts in the new timetable.

Since Sahin's lookahead function is different from ours we chose a combination of weights that gives the best results from a preliminary experiment. Between two timetables, the algorithm chooses the one with the smaller value of the lookahead function. Then the procedure is repeated by looking up the earliest conflict in the new timetable until we get a conflict-free timetable.

In the adaption of the Sahin's algorithm for our multiple track instances, we assume that each train observes the track assignment of the original timetable. Also to guarantee a feasible timetable, we also considered the inter-track safe headway time. As in TP3′, the dwelling modes of trains are fixed to be "stop" at each station. After the application of Sahin's algorithm, the solution is improved by the postprocessing.

The comparison test of FRA and the Sahin's algorithm on 60 min time horizon instances are summarized in Table 11. The computational efforts of FRA are marginal to the ones of the Sahin's algorithm: 2.9 versus 1.9 s in average. As expected, however, FRA could provide better solution than the heuristic method. Their objective values are in different orders. Thus, FRA offers a better trade-off between solution quality and computation time.

6.4. Performance of pricing heuristic

To evaluate the pricing subproblem heuristic, over the 72 instances of the time horizon of 30 min, we tested two versions of FRA: the one solving the pricing subproblem exactly by the CPLEX solver and the other by the heuristic. Except for a single

Table 11
Comparison with Sahin's algorithm on 60 min time horizon instance.

Instance	FRA(1 min)		SAHIN		Instance	FRA(1 min)		SAHIN	
	Obj.	Time (s)	Obj.	Time (s)		Obj.	Time (s)	Obj.	Time (s)
(08,05)	**100.8**	3.7	2870.6	2.6	(13,05)	**105.4**	2.1	133.2	1.3
(08,10)	**206.9**	3.5	1789.9	2.4	(13,10)	**246.5**	2.0	1398.5	1.3
(08,15)	**332.1**	3.6	2770.8	2.2	(13,15)	**335.8**	2.0	2173.1	1.3
(08,20)	**460.6**	3.6	2834.8	2.2	(13,20)	**426.7**	2.1	2602.6	1.3
(08,25)	**568.0**	3.6	2738.1	2.2	(13,25)	**516.7**	2.0	2603.2	1.3
(08,30)	**662.9**	3.7	2888.7	2.2	(13,30)	**617.3**	2.2	2099.9	1.3
(09,05)	**210.7**	3.8	2968.2	2.6	(18,05)	**111.2**	2.8	969.0	2.6
(09,10)	**378.8**	3.7	3483.0	2.2	(18,10)	**225.1**	2.9	3158.9	2.1
(09,15)	**336.8**	3.7	3363.1	2.2	(18,15)	**350.7**	2.9	1290.7	2.1
(09,20)	**432.1**	3.6	4018.3	2.1	(18,20)	**425.9**	3.1	2864.9	2.1
(09,25)	**504.8**	3.5	3258.0	2.1	(18,25)	**495.5**	2.9	2889.1	2.1
(09,30)	**631.4**	3.5	3457.8	2.1	(18,30)	**577.9**	3.0	2644.0	2.1
(12,05)	**91.4**	1.7	586.8	1.7	(19,05)	**107.1**	3.5	793.2	2.5
(12,10)	**181.4**	1.8	661.6	1.1	(19,10)	**289.1**	3.6	1046.7	2.1
(12,15)	**270.5**	1.7	827.6	1.1	(19,15)	**330.2**	3.2	2096.2	2.1
(12,20)	**371.8**	1.6	1389.6	1.1	(19,20)	**466.8**	3.3	1698.1	2.1
(12,25)	**450.0**	1.6	1688.7	1.1	(19,25)	**624.4**	3.3	2883.6	2.1
(12,30)	**540.0**	1.6	1524.2	1.1	(19,30)	**668.6**	3.2	2435.2	2.1
					Average		2.9		1.9

instance for which the exact pricing resulted in a solution better by 0.1%, two versions returned the solutions of the same quality. However, the computation time of the exact pricing FRA was 100 times larger than that of the heuristic pricing FRA.

6.5. Platform capacity consideration

Finally, as our model does not explicitly consider the platform capacity, we examined the number of platforms required by the timetable provided by our algorithm. We could observe that at each station, the number of platforms required by the timetable of FRA does not exceed one plus the number of tracks on the incident segment. In this sense, FRA seems to provide a feasible solution when there are more platforms at a station than the number of outgoing tracks on the segment as in the case of the stations of the Seoul metropolitan railway network.

7. Conclusion and open question

In this paper, we have considered the train-conflict resolution problem for which the optimal conflict-free timetable should be provided in a few minute for the practical purposes. First of all, we have formulated this problem as an MIP. The feature of our model is the intra-track safe headway time: any pair of consecutive trains should not be too close even if they use different tracks of a segment. Second, we have established the intractability of the problem by showing that the instance with two single-tracked segments and three stations is NP-hard. Hence, it may be the best alternative to consider a method considering a trade-off between solution quality and computation time.

In this sense, we have proposed a method called FRA, exploiting the structure of the problem. From the observations on the Seoul metropolitan railway network case, we have made several assumptions that each station has enough number of platforms, the underlying railway network is tree, and each track is unidirectional. Theses assumptions imply that our problem is separable. Moreover, we can obtain a feasible (global) timetable by fixing a train schedule for a single segment iteratively in a topological order of segments.

Based on these observations, FRA is designed as an economic variant of the column-generation method. For this, we also have devised an efficient heuristic for the pricing subproblem. Tested on the real instances from a metropolitan railway network with a high train frequency, it could provide near-optimal conflict-free timetables within uniformly minimal computation times that appear practically feasible. In particular, it could provide a timetable within 1 minute, a requirement of the Seoul metropolitan railway network. Moreover, we have showed that our method is superior to other methods in the literature.

But our model does not consider the option of cancelling a train. Also, FRA relies on the above assumptions restricted to the cases similar to Seoul metropolitan railway. Thus, FRA cannot be used if those assumptions are violated.

We conjecture that the special case with one single-tracked segment and two stations is also NP-hard. Especially, the better understanding we get about this special case, the more efficient method may be possible for our pricing subproblem.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2009-0077035). The first and second authors were also partially supported by the second stage of Brain Korea 21 project in 2009.

References

Brannlund, U., Lindberg, P., Nou, A., Nilsson, J., 1998. Railway timetabling using lagrangian relaxation. Transportation Science 32 (4), 358–369.
Brucker, P., Knust, S., Cheng, T., Sharkhlevich, N., 2004. Complexity results for flow-shop and open-shop scheduling problems with transportation delays. Annals of Operations Research 129 (1–4), 81–106.
Cacchiani, V., Caprara, A., Toth, P., 2008. A column generation approach to train timetabling on a corridor. A Quarterly Journal of Operations Research 6 (2), 125–142.
Cai, X., Goh, C., 1994. A faster heuristic for the train scheduling problem. Computers & Operations Research 21 (5), 499–510.
Cai, X., Goh, C., Mees, A., 1998. Greedy heuristic for rapid scheduling of trains on a single track. IIE Transactions 30 (5), 481–493.
Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. Operations Research 50 (5), 851–861.
Caprara, A., Monaci, M., Toth, P., Guida, P., 2006. A lagrangian heuristic algorithm for a real-world train timetabling problem. Discrete Applied Mathematics 154 (5), 738–753.
Carey, M., 1994a. Extending a train pathing model from one-way to two-way track. Transportation Research Part B 28 (5), 395–400.
Carey, M., 1994b. A model and strategy for train pathing with choices of lines, platforms and routes. Transportation Research Part B 28 (5), 333–353.
Carey, M., Lockwood, D., 1995. A model, algorithms and strategy for train pathing. The Journal of the Operational Research Society 46 (8), 988–1005.
D'Ariano, A., Pacciarelli, D., Pranzo, M., 2007a. A branch and bound algorithm for scheduling trains in a railway network. European Journal of Operational Research 127 (2), 643–657.
D'Ariano, A., Pranzo, M., Hansen, I., 2007b. Conflict resolution and train speed coordination for solving real-time timetable perturbations. IEEE Transactions on Intelligent Transportation Systems 8 (2), 208–222.
Davis, J., Kanet, J., 1993. Single machine scheduling with early and tardy completion costs. Naval Research Logistics 40 (1), 85–101.
Dessouky, M., Lu, Q., Zhao, J., Leachman, R., 2006. An exact solution procedure to determine the optimal dispatching times for complex rail networks. IIE Transactions 38 (2), 141–152.
Garey, M., Tarjan, R., Wilfong, G., 1988. One-processor scheduling with symmetric earliness and tardiness penalties. Mathematics of Operations Research 13 (2), 330–348.
Higgins, A., Kozan, E., Ferreira, L., 1996. Optimal scheduling of trains on a single line track. Transportation Research Part B 30 (2), 147–161.

Kraay, D., Harker, P., 1995. Real-time scheduling of freight railroads. Transportation Research Part B 29 (3), 213–229.
Kraay, D., Harker, P., Chen, B., 1991. Optimal pacing of trains in freight railroads: model formulation and solution. Operations Research 39 (1), 82–99.
Mascis, A., Pacciarelli, D., 2002. Job-shop scheduling with blocking and no-wait constraints. European Journal of Operational Research 143 (3), 498–517.
Mascis, A., Pacciarelli, D., Pranzo, M., 2008. Scheduling models for short-term railway traffic optimisation. Lecture Notes in Economics and Mathematical Systems 600, 71–90.
Mazzarello, M., Ottaviani, E., 2007. A traffic management system for real-time traffic optimisation in railways. Transportation Research Part B 41 (2), 246–274.
Ow, P., Morton, E., 1989. The single machine early/tardy problem. Management Science 35 (2), 177–191.
Sahin, I., 1999. Railway traffic control and train scheduling based on inter-train conflict management. Transportation Research Part B 33 (24), 511–534.
Törnquist, J., 2006. Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms. In: Proceedings of ATMOS2005 (Algorithmic MeThods and Models for Optimization of RailwayS), Palma de Mallorca, Spain, October 2005, Dagstuhl Research Online Publication Server (DROPS).
Törnquist, J., 2007. Railway traffic disturbance management – an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. Transportation Research Part A 41 (3), 249–266.
Törnquist, J., Persson, J., 2007. N-tracked railway traffic re-scheduling during disturbances. Transportation Research Part B 41 (3), 342–362.
Valente, J., Alves, R., 2005. Improved heuristics for the early tardy scheduling problem with no idle time. Computers & Operations Research 32 (3), 557–569.
Yano, C., Kim, Y., 1991. Algorithms for a class of single machine weighted tardiness and earliness scheduling problems. European Journal of Operational Research 52 (2), 167–178.
Zhou, X., Zhong, M., 2005. Bicriteria train scheduling for high-speed passenger railroad planning applications. European Journal of Operational Research 167 (3), 752–771.