

## Research Article

# A Branch and Bound Algorithm and Iterative Reordering Strategies for Inserting Additional Trains in Real Time: A Case Study in Germany

Yuyan Tan<sup>1</sup> and Zhibin Jiang<sup>2</sup>

<sup>1</sup>*Institute of Railway Systems Engineering and Traffic Safety, Technical University of Braunschweig, Pockelsstrasse 3, 38106 Braunschweig, Germany*

<sup>2</sup>*School of Transportation Engineering, Tongji University, 4800 Caoan Road, Shanghai 201804, China*

Correspondence should be addressed to Zhibin Jiang; [jzb@tongji.edu.cn](mailto:jzb@tongji.edu.cn)

Received 6 June 2014; Revised 1 September 2014; Accepted 2 September 2014

Academic Editor: Huimin Niu

Copyright © 2015 Y. Tan and Z. Jiang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the aim of supporting the process of adapting railway infrastructure and future traffic needs, we have developed a method to insert additional trains efficiently to an existing timetable without introducing large consecutive delays to scheduled trains. In this work, the problem is characterized as a job-shop scheduling problem. In order to meet the limited time requirement and minimize deviations to the existing timetable, the modification that consists of retiming or reordering trains is implemented if and only if it potentially leads to a better solution. With these issues in mind, the problem of adding train paths is decomposed into two subproblems. One is finding the optimal insertion for a fixed order timetable and the other is reordering trains. The two subproblems are solved iteratively until no improvement is possible within a time limit of computation. An innovative branch and bound algorithm and iterative reordering strategy are proposed to solve this problem in real time. Unoccupied capacities are utilized as primary resources for additional trains and the transfer connections for passengers can be guaranteed in the new timetable. From numerical investigations, the proposed framework and associated techniques are tested and shown to be effective.

## 1. Introduction and Literature Review

This paper will give an account of how to reconstruct an existing train schedule by inserting additional train services. This timetable-based extra train paths inserting (TETPI) problem is an integration of railway dispatching and scheduling. Train dispatcher both modifies the given timetable to manage delay in the running operation and establishes schedules for extra trains.

*1.1. Background.* The primary motivation of this research occurs as a result of the following application areas.

(1) *The Demands of Extra Trains for Train Operating Companies in European Railways.* Pachl [1] describes the open access networks in European railways. In the train paths management, the train operating companies order train paths from the infrastructure operator, and then the infrastructure

operator allocates these train paths on its lines in accordance with the orders of the train operating companies. From the view of scheduling, there is a distinction between regular trains and extra trains. Regular trains are all trains that have a schedule in the yearly timetable. Train paths for regular trains have to be ordered several months before the yearly timetable comes into effect. Extra trains are trains that have no schedule in the yearly timetable. Train paths for extra trains can be ordered at any time. Thus, when establishing the yearly timetable, a lot of traffic is not yet known. It contains only a part of the real traffic, mainly passenger trains and freight trains that run on a regular basis. Nowadays, in order to meet the needs of the shippers, an increasing share of freight trains runs as extra trains. Train paths for many freight trains, which do often not even appear in the scheduling systems, are ordered in a very short time in advance, sometimes just a few hours [1]. For this reason, the train operating companies have to rise to the challenge that establishing conflict-free

train paths for these trains in a short time and keeping the disruptions to regular trains minimised or similarly within acceptable levels.

(2) *A key Concept for Noncyclic Timetable in China Railway System.* The cyclic timetable has been widely adopted on high-speed railways because of its obvious advantages in transport marketing and train operation. However, it is not wise to apply the complete cyclic timetable model on Chinese railway due to its own features. Recently, researchers have shown an increased interest in an incomplete cyclic timetable model, an integration model of cyclic and noncyclic timetables, on China high-speed railways, such as [2, 3]. When constructing incomplete cyclic timetables, the trains of short (or medium long) distances and high frequencies are scheduled as cyclic trains, and then trains of long distances or low frequencies need to be inserted into the planned cyclic timetable as extra acyclic trains. Nowadays, the cyclic timetable models have been well developed, and the time-based train insertion technique without breaking the original cyclic structure is still a significant demand for research.

In this paper, the TETPI problem via a job shop scheduling approach is considered. More specifically the purpose of the TETPI problem in this paper is to establish a new conflict-free timetable for additional trains in a short time, guarantee the necessary transfer connections between passenger trains, and aim to minimize the delays incurred by additional trains, where delay is defined as deviation of actual service schedules and the initial schedules.

In order to meet the limited time requirement and minimize deviations to the existing timetable, in practice it is not necessary to take all of the scheduled trains into consideration. The modification which usually consists of retiming or reordering trains is implemented if and only if it potentially leads to a better solution. With these issues in mind the following process is possible for solving the TETPI problem.

*Phase 1* (FX strategy). Fix all previously scheduled services.

*Phase 2* (RM strategy). The scheduled services can be retimed, but the relative order between trains is kept.

*Phase 3* (RO strategy). The scheduled services can be reordered.

Note that the TETPI process does not necessarily have to be applied in the order shown above since not every phase is required. For instance when track infrastructure utilisation is light then Phase 1 would be applied. When track capacity is already heavily utilised, additional trains can only be inserted by disrupting existing services. Phase 1 will be inappropriate and Phase 2 should be applied. In both Phase 1 and Phase 2, additional train services may be added using a constructive algorithm. In Phase 3, a decision is taken in this phase to restate a priority order between trains, competing for the same single track section, in order to get a better solution.

Based on the above analysis, the contributions of this paper are as follows.

- (1) An iterative procedure for inserting extra trains to an existing timetable is defined in this paper. The TETPI

problem is decomposed into two subproblems; one is finding the optimal insertion for a fixed order timetable (Phase 1 and Phase 2) and the other is reordering trains (Phase 3). The two subproblems are solved iteratively until no improvement is possible within a time limit of computation.

- (2) An innovative branch and bound search algorithm is introduced to solve the first subproblem. In this step, the additional trains mainly utilize the unoccupied capacities to be inserted in order to minimise the deviations to existing trains.
- (3) The order between trains is restated based on dispatching rules to get a better solution. Note that only the change between train services which occur in the *critical path* is propitious to decrease the makespan (i.e., objective in the model) of the alternative graph.

## 1.2. Review of the Related Literature

*1.2.1. Literature Review on Scheduling and Rescheduling Problem.* The TETPI problem is related to a variety of topics in the literature. The first and foremost is rail timetable optimization. Recently, train scheduling and rescheduling problems have a great deal of attention. Corman et al. [4], for example, present a list of foremost papers published on this area between 1999 and 2007. There are varied models for formulating train operation problem. Since then we have observed the following papers based on *graph theory* to solve these problems efficiently: D'Ariano et al. [5] propose a fixed speed model and variable speed model to find a conflict-free timetable in real time after train operations are perturbed. D'Ariano et al. [6] consider the problem of managing disturbance in real time. In this paper, a real-time traffic management system called Railway traffic Optimization by means of alternative graph (ROMA) is introduced. They model the railway traffic optimization based no-store alternative graph, including constraints of rolling stock and passenger connections. This problem is decomposed into two subproblems, one is reordering which is solved by branch and bound algorithm and the other is rerouting which is solved by a local search algorithm. The two subproblems are then solved iteratively. Cacchiani et al. [7] deal with the problem of timetabling non-cyclic trains. A mixed-integer programming model (MIP) is proposed to look for the maximum-weight path in a compatibility graph. Schachtebeck [8] considers the delay management in public transportation. Based on graph theory he uses integer programming model (IP) formulation and suggests various heuristic solution approaches to solve large-scale real-world instances to optimality. In addition, papers such as [9–13] also present innovative models which are of certain reference value in the field of railway operation optimization.

Furthermore, there are indications that some of the previous models and techniques in job shop scheduling could be modified and adapted to solve the TETPI problem. For instance, Kis and Hertz [14] give for the classical job shop a polyhedral description of the feasible job insertions and use it to derive a lower bound for the minimum makespan job insertion problem. Gröflin et al. investigate [15, 16] insertion

TABLE 1: Characteristics of adding paths problem and solution approaches.

Publication	Background	Model	Objective	Solution	Real time	Connection	Problem size evaluated for
[17]	(F)	ILP	Maximizes the number of additional trains and minimize the violations to the ideal insertion	(HA)	No	No	679/202/24-48-96, 520/202/24-48-96, 0/202/188-338-554-64
[18]	(P)	MIP	Minimizes the total weighted time window violations and the makespan	(CA), (HA)	No	No	6/3/1-5, 6/5/1-5, 10/10/1-5, 24/10/1-5, 15/5/1-5, 54/30/1-5, 20/20/1-5, 20/12/1-5, 20/24/1-5
[19]	(F/P)	CSM	Minimizes the average traversal time of new train	(DP), (PR)	No	No	81/65/20
[20]	(F/P)	LPM, CSM	Support railway planners by computing a set of Pareto optimal solutions with respect to travel time and expected delay to additional trains	(SP)	No	No	—/—/1
Our paper	(F/P)	MIP	Minimizes consecutive delay to existing timetable	(BB), (AG), (SP), (PR)	Yes	Yes	36/60/1-15

(i) Background: passenger trains insertion (P); freight trains insertion (F).

(ii) Model: mixed integer programming (MIP); computer simulation model (CSM); integer linear programming (ILP); linear regression model (LPM).

(iii) Solution: constructive algorithm (CA); alternative graphs (AG); shortest path algorithm (SP); branch-and-bound (BB); heuristics algorithm (HA); dynamic programming (DP); local search (LS); practical rules (PR).

(iv) Problem size evaluated for number of initial trains/number of stations or block sections/number of additional trains. —: between double / means missing the information.

problems in a general disjunctive scheduling framework capturing a variety of job shop scheduling problems and inserting types. They propose a short cycle property for job insertion problem. A polyhedral description of all the feasible job insertions is derived to find the lower and upper bound for the minimum makespan.

### 1.2.2. Literature Review on Problem of Adding Train Paths.

Although the adding extra train paths technology is very important, there has been few direct related discussion about adding paths problem. The only papers to our knowledge are presented in Table 1 which summarizes the studies, like ours, dealing with inserting passenger or freight trains into an exiting timetable.

References [17, 19] solve the problem of inserting freights trains with assumption that all of the initial trains can not be changed. In [17], the additional trains are inserted with predefined ideal departure/arrival time and minimum stopping time at each station that must visit; meanwhile, alternative routes are taken into account. In [19], additional trains are inserted at a randomly fixed time belonging to the time window at each iteration and priority rule is predefined for each overtaking and meeting. Reference [20] computes a set of Pareto optimal train schedules with respect to risk and travel time. Their method aims to find robust train paths in the sense that the additional train has a low risk of delay upon arrival at its final station and supporting railway planners by computing a set of recommended train paths for a given train request.

Our study is different from the previous ones in two things. Firstly, we guarantee the transfer connections during insertion. Secondly, we solve the problem of adding train paths in real time. The difficulty in this paper is that the timetable scheduling and rescheduling are considered as two complex and time-consuming problems and usually researched independently. However, as introduced in Section 1.1, the problem of adding train paths, an integration of railway scheduling and rescheduling problem, is often required to be solved within a short time in Germany.

The format of the paper in the following sections is as follows. The model and algorithms for the TETPI problem are described in Section 2. The details of a novel bound search algorithm and an iterative reordering structure are presented to get optimal decision. Numerical investigations and case studies then demonstrate in Section 3 the quality of solutions that are obtained from the proposed approaches. In Section 4, the outcomes and the significance of the paper are summarized and the future research directions are given.

## 2. Model and Algorithms for the TETPI Problem

The job shop scheduling models with blocking and no-store constraints have been used by many authors to formulate the problem of scheduling train movements in railway networks. With this model the sequence of block sections traversed by a train is viewed as a set of machines in a job shop problem, where trains correspond to jobs and the usage of a block section by a train is called an operation.

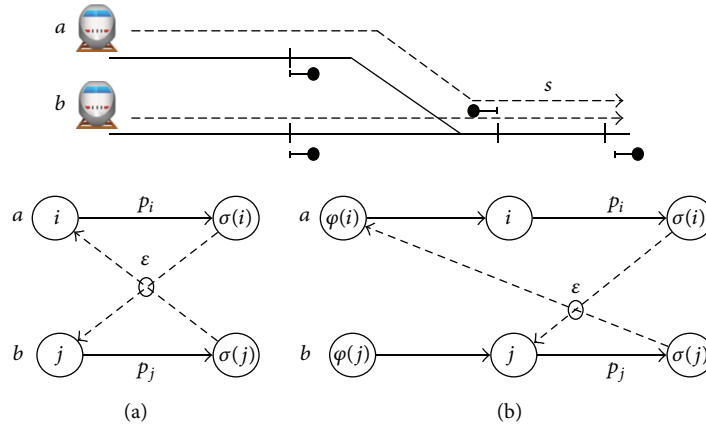


FIGURE 1: Two train operations to be processed on the same section.

**2.1. Basic Railway Scheduling Model.** For a basic railway scheduling problem, its corresponding alternative graph model is usually defined as a triple  $G = (V, A, E)$ , where nodes set  $V$  refers to a finite set of nodes, which correspond to operations.  $A$  is a set of direct conjunctive arcs modeling a precedence relation between two consecutive operations.  $E$  is a set of pairs of *alternative* arcs representing all the potential permutations of trains in common sections.

In Figure 1(a), operations  $i$  and  $j$  represent trains  $T_a$  and  $T_b$  entering the same section  $s$ .  $\sigma(i)$  and  $\sigma(j)$  are referred to the subsequent operations of  $i$  and  $j$ , respectively. Conjunctive arcs are depicted with solid arrows, which are weighted with running times  $p_i$  and  $p_j$  of the associated operations  $i$  and  $j$ , respectively. A conjunctive arc restricts the starting time for an operation with respect to the starting of its predecessor operation; that is,

$$t_{\sigma(i)} \geq t_i + p_i, \quad t_{\sigma(j)} \geq t_j + p_j. \quad (1)$$

In a fixed block safety system, each block section can host at most one train at a time; otherwise, some potential conflicts occur whenever two or more trains require the same block section. Therefore, after processing, every train has to wait on the current section and blocks it until the subsequent section becomes available, thus preventing other trains from entering the same section at the same time. This constraint is formulated by a pair of alternative arcs which defines different processing order for every two trains in the common section. A pair of alternative arcs  $\langle (\sigma(j), i), (\sigma(i), j) \rangle$  are represented with dashed lines in Figure 1(a). If  $(\sigma(j), i)$  is selected,  $T_b$  precedes  $T_a$ . Conversely, if  $(\sigma(i), j)$  is selected,  $T_a$  precedes  $T_b$ . The conflict between  $T_a$  and  $T_b$  must be solved choosing exactly one of the two arcs. In practice, an additional safety margin  $\mathcal{E}$ , referred to in this paper as *setup time*, is introduced to every alternative arc. It is a permutation based constant. Then the alternative arcs can be formulated as follows:

$$t_{\sigma(i)} \geq t_j + p_j + \mathcal{E}_{ji} \quad \text{or} \quad t_{\sigma(j)} \geq t_i + p_i + \mathcal{E}_{ij}. \quad (2)$$

However, for fast trains, the model is slightly different. Note that a fast train in a fixed safety system is only allowed to enter a block section if the next two block

sections are not occupied by other trains. As shown in Figure 1(b), the corresponding pair of alternative arcs lead to  $\langle (\sigma(i), j), (\sigma(j), \varphi(i)) \rangle$ , where  $\varphi(i)$  and  $\varphi(j)$  are the predecessor operations of faster operation  $i$  and slower operation  $j$ , respectively [21]. This can be seen as a multiprocessor job scheduling problem, where operations have to be processed simultaneously on a set of dedicated machines instead of a single machine.

**2.2. Model for the TETPI Problem.** In this section, the TETPI problem is formulated by means of an alternative graph model. Combining the description of the basic scheduling model and the features of inserting problem, the conjunctive arcs  $A$  and alternative arcs  $E$  for the TETPI problem are defined as follows:

- (i) *Conjunctive* arcs set  $A = A_O \cup A_N$  is composed of original conjunctive arcs set  $A_O$ , which belongs to existing trains, and new conjunctive arcs set  $A_N$ , which belongs to additional trains.  $A$  is fixed when train routes are defined.
- (ii) *Alternative* arcs set  $E = E_O \cup E_N$ , where  $E_O$  represents permutation of original trains and predefined sequence between additional trains and  $E_N$  represents all of the possible insertions for new trains set  $N$ . In the TETPI problem, even though  $E_O$  is settled, it is still a set of variables when some trains need to be reordered, and then selection of  $E_N$  depends on the choice of  $E_O$ .

For example, an alternative graph for a part of railway lines is presented in Figure 2. It contains 10 block sections numbered from  $m_1$  to  $m_{10}$ . Two existing trains labeled  $T_a$  and  $T_b$  are using this part of tracks. The routes of trains  $T_a$  and  $T_b$ , respectively, are as follows:

$$T_a: \{m_4 \rightarrow m_5 \rightarrow m_3 \rightarrow m_6 \rightarrow m_7 \rightarrow m_{10}\} \quad (3)$$

$$T_b: \{m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_6 \rightarrow m_7 \rightarrow m_{10}\}.$$

One additional train denoted  $T_n$  with the route

$$T_n: \{m_4 \rightarrow m_5 \rightarrow m_8 \rightarrow m_9 \rightarrow m_7 \rightarrow m_{10}\} \quad (4)$$

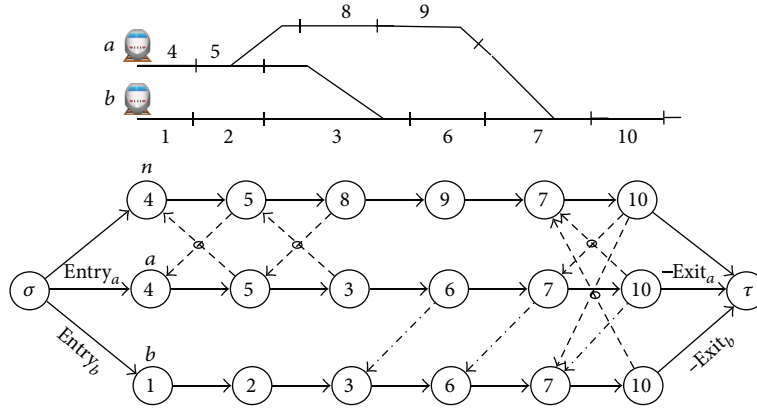


FIGURE 2: An alternative graph for the TETPI problem.

needs to be inserted. Let  $\sigma$  and  $\tau$  be dummy start and finish operations, having to be performed before and after all operations, respectively. For the sake of clarity, the weights of arcs which represent minimum running time and dwell time of the associated operations are not showed.

In this alternative graph,  $A$  is represented with solid arrows, while  $E_O$  and  $E_N$  are depicted with dash-dotted arrows and dashed arrows, respectively. To differ in that a precedence constraint in  $A$  always must hold, while an arc in  $E$  must be selected if its corresponding alternative arc does not hold. A *selection* in graph  $G$  is an arbitrary subset of alternative arcs; that is,  $S \subseteq E$ . If for every alternative pair at least one arc is selected, we called that  $S$  is *complete* in  $G$ .  $S$  is an *acyclic* selection if the graph  $G(V, A \cup S)$ , denoted by  $G(S)$ , contains no positive length cycles. Finally,  $S$  is a feasible selection if it is complete and acyclic. In fact, a positive length cycle causes a deadlock in a graph, which is infeasible. So the TETPI problem is equivalent to the following problem: given a set of trains  $T$  and a selection  $S_O$  that is feasible for all trains of  $T - \{N\}$ , where  $N$  is a set of additional trains, then find a selection  $S_N$  of alternative arcs incident to  $N$  such that among all feasible selections selection  $S = S_O \cup S_N$  minimizes the length of the longest path  $\pi$  from  $\sigma$  to  $\tau$  in graph  $G(S)$ .

Furthermore, D'Ariano et al. [6] use an innovative way to represent the maximum consecutive delay with the makespan of the associated alternative graph by introducing the weights of entry time and negative exit time on the start and end operations, respectively. In this way, the objective for the TETPI model is changed into minimizing consecutive delay.

In practice, extra freight trains may be set flexibly when they will enter the planned system, but at most of time passenger trains have to be set at predefined time or meet transfer requirements with existing trains, and moreover some trains may already be partly within the system at the start of the schedule. In such cases, strategies like retiming and reordering existing trains are unavoidable which may cause scheduled connections to be dismissed. In order to guarantee some scheduled connections or orders, the corresponding alternative arcs in  $E_O$  should not be changed. Meanwhile, for the purpose of new connections between additional trains and scheduled trains, the corresponding

alternative arcs in  $E_N$  should be preselected. A set of these preselected and preforbidden alternative arcs are referred to  $Q$  and  $F$ , respectively, in this paper.

Then the basic constrained TETPI problem based on alternative graph can be formulated as the model (M1):

(M1):

$$\min t_\tau - t_\sigma$$

$$\begin{cases} t_j - t_i \geq p_i & \forall i, j \in V, (i, j) \in A \\ t_j - t_i \geq p_i + \mathcal{E}_{ij} \text{ or} & \\ t_k - t_h \geq p_h + \mathcal{E}_{hk} & \forall i, j, k, h \in V, \\ & \langle (i, j), (h, k) \rangle \in E \\ t_j - t_i \geq p_i + \mathcal{E}_{ij} & \forall i, j \in V, (i, j) \in Q \\ t_j - t_i \leq p_i + \mathcal{E}_{ij} & \forall i, j \in V, (i, j) \in F. \end{cases} \quad (5)$$

Besides the above constraints, the alternative graph allows formulation of situations more general than the ones we have mentioned. For example, Strotmann [21] introduces additional constraints modeled in terms of the alternative graph model, including connection constraints between trains, out-of-service intervals for certain block sections and so on.

**2.3. Algorithms for the TETPI Problem.** In our definition, the TETPI problem is asked to develop a new conflict-free plan compatible with the strict time limit for some real-time traffic situations and modify the scheduled requests as little as possible, including reordering as few as possible. In order to meet these minimum deviation and limited time requirements, it is not necessary to take all the scheduled arcs  $E_O$  into consideration, because only the change of alternative arcs occurring in the *critical path* is propitious to decrease the makespan of alternative graph. In this section, the TETPI problem is decomposed into two subproblems:

- (1) find the optimal insertion for a fixed order timetable: in this phase, the additional services mainly utilize unoccupied capacities to process. By this means, it can not only minimize the deviation of scheduled timetable, but also keep the existing transfer plans to the fullest extent;

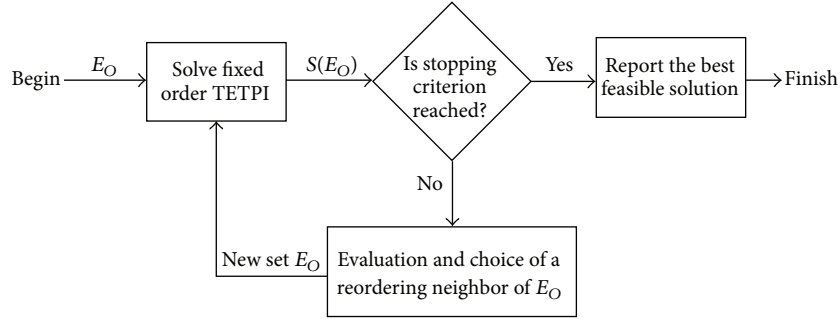


FIGURE 3: Architecture of insertion.

- (2) give a solution to subproblem (1) and define new sequence of  $E_O$  potentially leading to better schedules. In this phase, unavoidable permutation changes are introduced in order to set more additional restricted services without leading to large deviation or consecutive delay to the existing timetable.

These two subproblems are then iteratively solved until no improvement is possible within a time limit of computation. The architecture of this inserting procedure is represented in Figure 3.

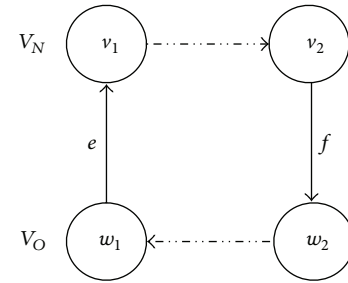
**2.3.1. Branch and Bound Search Algorithm with Fixed Permutation.** Let the length  $\pi^S(\sigma, \tau)$  be the critical path of the graph  $G(S)$ . This value can be reduced either by modifying the location of the new trains  $S_N$  or reordering the original train sequencing  $S_O$ . In this subsection, we start from  $G(V, A \cup S_O)$  with fixed permutation of original trains  $S_O$  and then derive a polyhedral description of all feasible insertions  $S_N$  for the TETPI problem.

In order to formulate this problem, we will use the following definitions and notions. For any directed arc  $e = (v, w)$ ,  $t(e) = v$  is the tail node of arc  $e$  and  $h(e) = w$  is the head node. Given an existing timetable  $G(V, A \cup S_O)$ , insert all operations of train set  $N$  into this schedule simultaneously. A family of alternative arc pairs  $E_N$  which are incident with  $N$  are composed of two distinct alternative edge sets  $E_N^-$  and  $E_N^+$ , respectively, entering and leaving  $N$ ; that is,  $E_N = E_N^- \cup E_N^+$ , where

$$\begin{aligned} E_N^- &= \{e \in E_N, t(e) \notin N, h(e) \in N\} \\ E_N^+ &= \{f \in E_N, t(f) \in N, h(f) \notin N\}. \end{aligned} \quad (6)$$

Insertion problem concerning how to reallocate the resources then can be produced on determining which arc should be selected from each pair of alternative arcs. Each arc in the pair represents mutually exclusive permutation between two operations. A core concept in the reduction of the computational effort of the algorithm introduced in this paper is the characterization of *incompatible arcs pairs*.

**Proposition 1.** *Given a TETPI problem, for any selection  $S \in E$ , if  $G(S)$  contains a cycle, then the selection  $S$  must contain at least one arc  $e \in E_N^-$  entering to and one arc  $f \in E_N^+$*

FIGURE 4: Arbitrary operation path with a cycle, where  $v_1, v_2 \in Op_N$  and  $w_1, w_2 \in Op_O$ .

leaving from the additional operations. This arcs pair  $\langle e, f \rangle$  is incompatible in a feasible inserting selection [15].

*Proof.* Given an operation path in graph  $G$ , let  $V_N$  and  $V_O$  denote the additional and exiting train operation sets, respectively. Since existing timetable and additional timetable are feasible separately, there is no cycle in  $V_N$  or  $V_O$ . If a cycle arises in the given path (see Figure 4 as a simple example), it must contain at least one arc  $e$  entering  $v_1$  from  $w_1$  and simultaneously at least one arc  $f$  leaving  $v_2$  to  $w_2$ , where  $v_1, v_2 \in V_N$  and  $w_1, w_2 \in V_O$ . For this reason, the corresponding alternative arc pair  $\langle e, f \rangle$  which composes a cycle is incompatible in a feasible selection  $S_N$ .  $\square$

Based on Proposition 1, let  $U$  be the collection of all such positive cyclic selections that visit additional trains  $N$  exactly once. For example, in Figure 2,

$$E_N^- = \{(a_5, n_4), (a_3, n_5), (a_{10}, n_7), (b_{10}, n_7)\} \quad (7)$$

indicates that existing trains precede additional trains in common sections, while

$$E_N^+ = \{(n_5, a_4), (n_8, a_5), (n_{10}, a_7), (n_{10}, b_7)\} \quad (8)$$

represents that an additional train goes first and then an existing train follows. Then the family of alternative arcs pair is  $\xi = \{\langle e, \bar{e} \rangle : e \in E_N^-, \bar{e} \in E_N^+\}$ , where

$$\begin{aligned} \xi &= \{ \langle (a_5, n_4), (n_5, a_4) \rangle \cup \langle (a_3, n_5), (n_8, a_5) \rangle \\ &\quad \cup \langle (a_{10}, n_7), (n_{10}, a_7) \rangle \cup \langle (b_{10}, n_7), (n_{10}, b_7) \rangle \}. \end{aligned} \quad (9)$$

According to the definition of  $U$ , then

$$U = \xi \cup \{ \langle (a_3, n_5), (n_5, a_4) \rangle \cup \langle (a_5, n_4), (n_8, a_5) \rangle \cup \langle (b_{10}, n_7), (n_{10}, a_7) \rangle \}. \quad (10)$$

Clearly,  $U$  is incompatible arc pairs set. Exactly only one arc in each pair of  $\xi$  must be chosen in a feasible selection and at most one arc (may be none of arc) could be selected in each pair of  $U/\xi$ . Let  $x_e$  be a binary variable to identify whether arc  $e$  is chosen or not; that is,

$$x_e = \begin{cases} 1 & \text{if arc } e \text{ is chosen,} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Then the TETPI problem can be described as model (M2):

$$(M2): \quad \min C \quad \begin{cases} x_e + x_f \leq 1 & \forall e \in E_N^-, f \in E_N^+, \langle e, f \rangle \in U \\ x_e + x_{\bar{e}} = 1 & \forall e \in E_N \\ x_e = 1 & \forall e \in Q \\ x_e = 0 & \forall e \in F \\ x_e = \{0, 1\} & \forall e \in E_N - Q - F \\ \pi(x) \leq C. \end{cases} \quad (12)$$

Since the critical path  $\pi(x)$  of the new timetable is nonlinear in  $x$ , huge computational effort will be taken to solve (M2). However, it can be observed that there are three different circumstances for  $\pi(x)$  in an arbitrary feasible insertion [16].

- (1) If  $\pi(x)$  contains no operations of additional trains  $N$ , the corresponding makespan for the new graph holds  $\pi_0 = \pi_{\sigma\tau}$ , where  $\pi_{\sigma\tau}$  is the makespan for existing timetable.
- (2) If  $\pi(x)$  contains exactly one arc in  $E_N$ , the corresponding makespan for the new graph satisfies  $\pi_1 = \pi_{\sigma,t(e)} + C_e + \pi_{h(e),\tau}$ , where  $\pi_{ij}$  is the longest path from node  $i$  to node  $j$  and  $C_i$  is the weight of arc  $i$ .
- (3) If  $\pi(x)$  contains exactly  $e \in E_N^-$  and  $f \in E_N^+$ , the corresponding makespan for the new graph is  $\pi_2 = \pi_{\sigma,t(e)} + C_e + \pi_{h(e),t(f)} + C_f + \pi_{h(f),\tau}$ .

Let  $\pi'(x)$  denote the maximum length path for these three circumstances. Clearly,  $\pi'(x) \leq \pi(x) \leq C$ . Then, the (M2) can be released into

(M3):

$$\min C \quad \begin{cases} x_e + x_f \leq 1 & \forall e \in E_N^-, f \in E_N^+, \langle e, f \rangle \in U \\ x_e + x_{\bar{e}} = 1 & \forall e \in E_N \\ x_e = 1 & \forall e \in Q \\ x_e = 0 & \forall e \in F \\ x_e = \{0, 1\} & \forall e \in E_N - Q - F \\ \pi_0 \leq C \\ \pi_1 x_e \leq C & \forall e \in E_N^- \cup E_N^+ \\ \pi_2 (x_e + x_f - 1) \leq C & \forall e \in E_N^-, f \in E_N^+. \end{cases} \quad (13)$$

**Proposition 2.** Given a TETPI problem with the makespan  $C$  for the optimal insertion, if  $\langle e, \bar{e} \rangle$  is a pair of alternative arcs and  $\pi_1 = \pi_{\sigma,t(e)} + C_e + \pi_{h(e),\tau} > C$ , then  $e$  is forbidden and  $\bar{e}$  is implied by the optimal inserting selection. In particular,  $x_e = 0$  and  $x_{\bar{e}} = 1$ , when  $\pi_1 > C$  [5].

**Proposition 3.** Given a TETPI problem with the makespan  $C$  for the optimal insertion, if there exists a path  $\pi_2$  that contains arcs  $e$  and  $f$  and  $\pi_2 = \pi_{\sigma,t(e)} + C_e + \pi_{h(e),t(f)} + C_f + \pi_{h(f),\tau} > C$ , then arcs pair  $\langle e, f \rangle$  is incompatible in the optimal inserting selection. In particular,  $x_e + x_f \leq 1$ , when  $\pi_2 > C$ .

*Proof.* Suppose that  $e$  and  $f$  are selected simultaneously in the optimal selection  $S_N$  and there exists a path  $\pi_2 = \pi_{\sigma,t(e)} + C_e + \pi_{h(e),t(f)} + C_f + \pi_{h(f),\tau}$ . The makespan  $C$ , the longest path in this inserting alternative graph, should satisfy  $C \geq \pi_2$ , but then it is a contradiction to  $\pi_2 > C$ .  $\square$

Based on Propositions 2 and 3, let

$$U_c = \{e : e \in E_N, \pi_1 > C\} \cup \{(e, f) : e \in E^-(N), f \in E^+(N), \pi_2 > C\}. \quad (14)$$

Then (M3) can be changed into

$$(M4): \quad \min C \quad \begin{cases} x_e + x_f \leq 1 & \forall e \in E_N^-, f \in E_N^+, \langle e, f \rangle \in U \cup U_c \\ x_e + x_{\bar{e}} = 1 & \forall e \in E_N \\ x_e = 1 & \forall e \in Q \\ x_e = 0 & \forall e \in F \cup U_c \\ x_e = \{0, 1\} & \forall e \in E_N - Q - F. \end{cases} \quad (15)$$

Then the lower bound LB for the TETPI problem can be relaxed into finding a minimum  $C$  to make (M4) have existence of solution; that is,  $C_{\min} = LB$ . It is evident that LB is tight if and only if there is a critical path that contains exactly one arc leaving  $N$  and one arc entering  $N$ . The procedure of inserting algorithm with fixed train permutation is showed in Procedure 1.

According to the above three propositions, let  $Q^*$  denote arcs implied by the optimal insertion based on Propositions 2

**Input:** existing timetable  $G(V_O, A_O, E_O)$   
**Output:** lower bound and critical path

- (1) **Begin**
- (2) calculate reachable matrix  $R_O$  and  $R_N$  for original and additional operation respectively, where 1/0 represents that it is (not) reachable from node  $i$  to node,
- (3) **while**  $se \neq Last\_section$  **do**
- (4)  $O_O = Get\_operation(T_O, se), O_N = Get\_operation(T_N, se)$
- (5) **for**  $O_O$  and  $O_N$  **do**
- (6)  $\sigma(O_O) = Get\_operation(O_O), \sigma(O_N) = Get\_operation(O_N)$
- (7)  $e = e \cup (\sigma(O_O), O_N), f = f \cup (\sigma(O_N), O_O)$
- (8) **end for**
- (9)  $se = se + 1$
- (10) **end while**
- (11) **for all**  $e$  and  $f$  **do**
- (12) **if**  $R_O(h(f), t(e)) = 1 \ \& \ R_N(h(e), t(f)) = 1$  **then**
- (13)  $U(e, f) = 1$ , i.e.  $e$  and  $f$  are incompatible arcs
- (14) **end if**
- (15) **end for**
- (16) **for all**  $U \neq 1$  **do**
- (17)  $\pi_1(e) = \pi_{\sigma, t(e)} + P_e + \pi_{h(e), \tau}, \pi_1(f) = \pi_{\sigma, t(f)} + P_f + \pi_{h(f), \tau}$
- (18) **if**  $R_N(h(e), t(f)) = 1$  **then**
- (19)  $\pi_2(e, f) = \pi_{\sigma, t(e)} + P_e + \pi_{h(e), t(f)} + P_f + \pi_{h(f), \tau}$
- (20) **else if**  $R_O(h(f), t(e)) = 1$
- (21)  $\pi_2(f, e) = \pi_{\sigma, t(f)} + P_f + \pi_{h(f), t(e)} + P_e + \pi_{h(f), \tau}$
- (22) **end if**
- (23) **end for**
- (24)  $C_1 = \pi_{\sigma, \tau}, C_2 = \max(\pi_1, \pi_2), C = C_1, U_c = U$
- (25) **while**  $C_1 \neq C_2$  **do**
- (26) **if**  $\pi_2(e, f) \leq C$  **then**
- (27)  $U_c(e, f) = 1$ , that is  $e$  and  $f$  are incompatible arcs
- (28) **end if**
- (29) **if**  $\pi_1(e) > C \ \& \ U_c(\bar{e}, f) = 1$  **then**
- (30)  $\bar{e}, f \in Select\_f\_Nr$
- (31) **end if**
- (32) **if**  $\pi_1(f) > C \ \& \ U_c(e, \bar{f}) = 1$  **then**
- (33)  $e, \bar{f} \in Select\_f\_Nr$
- (34) **end if**
- (35) **if**  $Select\_e\_Nr \cap Select\_f\_Nr = \emptyset$  **then**
- (36)  $C_{LB} = C, C_2 = C, C = (C_1 + C_2)/2$
- (37) **end if**
- (38) **if**  $C_1 = C$  **then**
- (39)  $C = (C_1 + C_2)/2$
- (40) **end if**
- (41) **end while**
- (42) **Return**  $C_{LB}$  and  $\pi$

PROCEDURE 1: Lower bound and critical path.

and 3 and let  $H$  be a set of alternative arc pairs which are still not determined. Then insertion  $Q^* \cup H^-$  and  $Q^* \cup H^+$  are two feasible inserting selections; for example, if  $(n_5, a_4)$  and  $(n_8, a_5)$  are implied in Figure 2, selections

$$Q^* \cup H^- = \{(n_5, a_4), (n_8, a_5), (b_{10}, n_7), (a_{10}, n_7)\}, \quad (16)$$

$$Q^* \cup H^+ = \{(n_5, a_4), (n_8, a_5), (n_{10}, n_7), (n_{10}, a_7)\},$$

are two feasible insert solutions. The better solution of the two insertions yields an upper bound UB for this TETPI problem, denoted by  $S_{UB}$ .

**2.3.2. Iteration of Reordering Optimization.** The evaluation and choice of a reordering neighbor of  $S_O$  are solved by a local search approach. The sequence of operations which compose critical path is modified if the new solution improves the objective function.

It should be noticed that, in the circumstance of inserting trains one by one, there are two types of orders to be exchanged, which occur between two original trains or between one additional train and one original train. When inserting trains simultaneously, rough permutation of additional trains should be drafted based on predefined time and transfer requirements. Then, besides orders between two original



**Input:**  $S_{LB}$ ,  $C_{LB}$  and critical path  $\pi$   
**Output:** best solution found

- (1) **while** (time limit not reached) & (consecutive delay > 0) **do**
- (2) Reorder the permutation of operations on critical path according to the dispatching rules (ARI) in Section 2.3.2, get  $S'_O$
- (3) **Do** Procedure 1 Lowerbound & Criticalpath, get  $S'_{LB}$  and  $C'_{LB}$
- (4) **if**  $C'_{LB} < C_{LB}$  **then**
- (5)  $S_O = S'_O$ ,  $C_{LB} = C'_{LB}$ ,  $S_{LB} = S'_{LB}$
- (6) **end if**
- (7) **end while**
- (8) **calculate**  $S_N$  according to the current  $S_{LB}$
- (9) **Return**  $S = S_O \cup S_N$

PROCEDURE 2: Local search and iterative step.

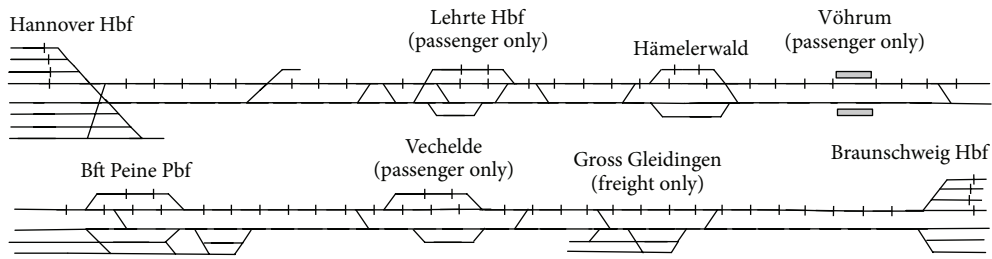


FIGURE 5: Hannover-Braunschweig railway area.

trains, the order of additional trains also should be taken into consideration.

Normally, the guarantee of original orders and connections plays an important factor in minimum consecutive delay to existing timetable and in convenience for passengers. It means that if there is no improvement found by exchanging orders of additional trains, then switch between two original trains is executed. Automatische Rijweg Instelling (ARI) is used as a dispatching rule in this paper. This rule is used in The Netherlands and is briefly introduced in Corman et al. [4]. Procedure 2 represents the iterative optimization procedure.

### 3. Computational Tests

We report on our computational experiments on a real-life scenario. The experiments are based on the dispatching area of Hannover-Braunschweig. The primary aim of the numerical investigations is to identify whether good solutions can be obtained using the methodology and techniques proposed in this paper. The bound search and permutation refinement algorithms are implemented in MATLAB R2010b and executed on a laptop equipped with a processor Intel Core 2 Duo (2.4 GHZ), 2 GB Ram. Computational time and delays are always expressed in seconds.

*3.1. Description of Instances.* The Hannover-Braunschweig railway area consists of double tracks divided into one long corridor for each traffic direction, one dedicated stop for freight trains, and seven passenger stations, see Figure 5.

We evaluate a four-hour (8:00 am–12:00 pm) timetable of the year 2003 in the direction of Braunschweig to Hannover. In this experimental study, 9 freight trains and 27 passenger trains, including 17 intercity trains and 10 regional trains, are scheduled in 60 blocking sections in the existing timetable. The total number of exiting operations is 2064. A setup time  $\mathcal{E}_{ij}$  is set to a constant of 120 seconds for simplification.

This insertion problem increases in the number of additional trains which contain a mixture of 120, 160, and 180 km/h trains in order to make the problem more challenging. In particular, the instances of simultaneity inserting 1 to 15 additional trains with various speed are tested in this section.

For each instance, unoccupied capacity is used as main resource for additional trains. If and only if consecutive delays are inevitable, then iterative reordering optimizations are executed within a computational time limit in order to minimize consecutive delays to the existing timetable.

In this section, computational tests impose a strict time limit of 90 seconds, which will be attained limiting the execution of the iterative reordering optimizations and the number of reordering possibilities.

*3.2. Computational Results.* The results for inserting additional trains to the existing timetable are summarized in Table 2. In this phase, the relative order of exiting trains is not allowed to change; that is,  $E_O$  are fixed, as described in subproblem (1) in Section 2.3. Each row of Table 2 reports the results for the insertion of one to fifteen additional trains, respectively. It describes the performance of the innovative search method in terms of lower bound, upper bound for

TABLE 2: Results of inserting additional trains to a fixed order timetable (subproblem 1).

Number of additional trains	Lower bound (LB)	Upper bound (UB)	Percentage of $Q^*$ in $E_N$	Time
1	0	0	100	0.02
2	0	0	100	0.26
3	0	0	100	0.27
4	0	0	100	0.29
5	0	0	100	1.26
6	0	0	100	1.56
7	0	0	100	1.93
8	0	0	100	2.90
9	0	0	100	3.38
10	1	143	87.22	3.67
11	1	143	87.29	4.14
12	1	143	87.37	4.62
13	1	423	93.10	16.20
14	2	423	92.91	17.82
15	2	423	93.00	24.17

TABLE 3: Results for the scenarios from Table 1 with Reordering strategy (sub-problem 2).

Number of additional trains	Total number of changed orders	Lower bound (LB)	Upper bound (UB)	Objective value <sup>1</sup>	Time <sup>2</sup>
10	1	0	0	0	10.00
11	2	0	0	0	12.31
12	2	0	0	0	18.65
13	3	0	75	(75)	(90)
14	3	0	75	(75)	(90)
15	3	0	75	(75)	(90)

<sup>1</sup>The numbers in the parentheses refer to the current best feasible solution ( $S_{UB}$ ) found when the time limit of 90 seconds was exceeded without an optimal solution being verified.

<sup>2</sup>The numbers in the parentheses refer to exceed the time limit of 90 seconds.

consecutive delay of exiting trains, percentage of essential selected arcs, and computation time.

In particular, the percentage of essential selected arcs implies the decrease of search space for an optimal solution. It evaluates the proportion of the edges  $Q^*$  in the initial edges  $E_N$  which need to be decided for an optimal insertion.  $Q^*$  is the set of edges that must be selected and calculated with the methodology proposed in this paper. It is calculated by

$$\text{Percentage of } Q^* \text{ in } E_N = \frac{\text{the number of } Q^*}{\text{the number of } E_N} \times 100. \quad (17)$$

Clearly, it is equal to 100% when all of the arcs  $E_N$  are determined in this phase. In such a case, there is no difference between lower and upper bounds, which represents that the bounds are tight and the exactly optimal insertion is  $S_{UB}$ .

In the instances of inserting 1–9 trains, the search technique proposed in this paper allows these additional trains to utilize the unoccupied capacities to be inserted without introducing any consecutive delay to the existing timetable. The optimal insertion does not introduce any disturbance to the exiting timetable and it can be got in 0.02 seconds for inserting a train and within 3.5 seconds even when 9 additional trains are added simultaneously.

For the scenarios of inserting 10–15 trains, iterative permutation optimizations are executed for decreasing consecutive delays to existing timetable and for narrowing the search range for optimal insertion, as described in subproblem (2) in Section 2.3. In each reordering iteration, a pair of trains occurring on the critical path is exchanged and a new graph is computed to find new feasible insertions.

Table 3 presents the results after reordering within the time limits of 90 seconds. It is apparent that, in the instance of 10 additional trains, the algorithm allows us to find the best solution within 10 seconds and only a pair of additional trains is exchanged to avoid delay for scheduled timetable. For inserting 11 and 12 trains, the best insertions without any consecutive delay also can be calculated efficiently within 20 seconds and two pairs of additional trains are exchanged. The instances for adding 13 to 15 trains are significantly harder. This result may be explained by two factors. Firstly, the computational time required for bound search in each iteration increased with the number of trains to be inserted, as indicated in Table 2. Secondly, as previously analyzed in Section 2, the lower bound is tight if and only if the longest path consists of at most one alternative arc which entering the additional trains  $N$  and at most one alternative arc which

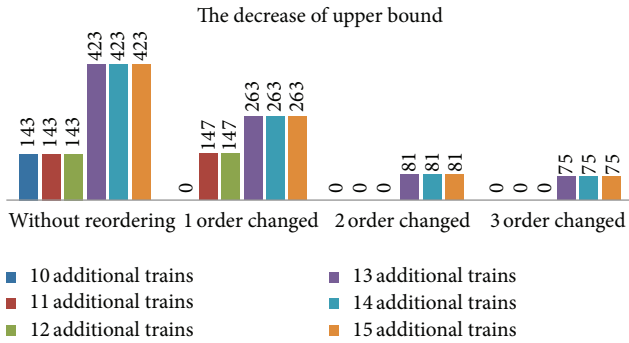


FIGURE 6: The decrease of upper bound.

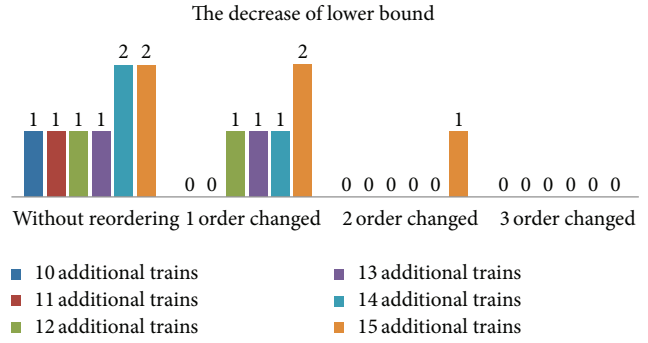


FIGURE 7: The decrease of lower bound.

leaving the additional trains  $N$ . With the increase of  $N$ , namely, the difference between the number of additional and existing operations becoming smaller, critical path for the new graph  $G$  which consists of existing and additional operations has more complicated status. Then the time for the iteration of reorder also increased with the number of trains which are inserted simultaneously. However, we can get a good feasible solution  $S_{UB}$  in a short time, which is very important in real-time inserting. On the other hand, since more than 87% of arcs are already determined by the lower bound, the optimal insertions for instances 13–15 can be found with minimal additional effort.

Figures 6 and 7 give a detailed view on the performance of iteration procedure for inserting 10 to 15 trains. The upper and lower bounds decrease with the iteration of reordering.

#### 4. Conclusions

The problem of real-time scheduling additional trains without leading to large deviation is considered in this paper. An innovative branch and bound search method and an iterative reordering structure have been developed for this aim. In the fixed permutations phase, the additional trains are inserted using unoccupied capacities to minimize deviation and keep connections of scheduled trains. In the reordering phase, certain permutations are exchanged to insert more trains with acceptable consecutive delays to existing trains.

From numerical investigations on a real-life study, it is observed that the lower bounds and upper bounds are very tight for inserting a certain amount of trains simultaneously relative to the number of existing trains. For relative large number of additional trains, moderate amounts of reordering actions are sufficient to handle the disturbances in the reference test cases, and the technique reduces the search space allowing us to solve real and complex problems efficiently.

Further research is required on a number of issues. From the computational point of view, the development of bound search and iterative reordering algorithms, which enable finding the optimal solutions for large instances within acceptable computation time, is worthwhile. For the theoretical and practical points of view, it would also be interesting to be combined with rerouting strategies. This can be modeled based on flexible job-shop problem which

involves many additional alternative machines and requires more computational effort.

#### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

#### Acknowledgments

This work was supported by the Railway Automation Graduate School (RA:GS!) of Siemens AG. The study has benefited from the comments from Dipl.-Ing. Tobias Lindner and Dipl.-Ing. Niels Hohn and the encouragement and support by Professor Dr.-Ing. Jörn Pachl. The authors appreciate this support.

#### References

- [1] J. Pachl, "Control of railway operation on open access networks," in *Proceedings of the 7th International Congress on Advances in Civil Engineering*, Istanbul, Turkey, October 2006.
- [2] M. Xie and L. Nie, "Model of cyclic train timetable," *Journal of the China Railway Society*, vol. 4, pp. 7–13, 2009.
- [3] D. Yang, L. Nie, Y. Tan, Z. He, and Y. Zhang, "Working out an incomplete cyclic train timetable for high-speed railways by computer," *WIT Transactions on the Built Environment*, vol. 114, pp. 899–999, 2010.
- [4] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Centralized versus distributed systems to reschedule trains in two dispatching areas," *Public Transport*, vol. 2, no. 3, pp. 219–247, 2010.
- [5] A. D'Ariano, M. Pranzo, and I. A. Hansen, "Conflict resolution and train speed coordination for solving real-time timetable perturbations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 208–222, 2007.
- [6] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo, "Reordering and local rerouting strategies to manage train traffic in real time," *Transportation Science*, vol. 42, no. 4, pp. 405–419, 2008.
- [7] V. Cacchiani, A. Caprara, and P. Toth, "Non-cyclic train timetabling and comparability graphs," *Operations Research Letters*, vol. 38, no. 3, pp. 179–184, 2010.

- [8] M. Schachtebeck, *Delay management in public transportation: capacities, robustness, and integration [Ph.D. thesis]*, Georg-August-Universität Göttingen, 2010.
- [9] L. Wang, Y. Qin, J. Xu, and L. Jia, "A fuzzy optimization model for high-speed railway timetable rescheduling," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 827073, 22 pages, 2012.
- [10] L. Wang, W. Mo, Y. Qin, F. Dou, and L. Jia, "Optimization based high-speed railway train rescheduling with speed restriction," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 934369, 14 pages, 2014.
- [11] H. Fu, L. Nie, B. R. Sperry, and Z. He, "Train stop scheduling in a high-speed rail network by utilizing a two-stage approach," *Mathematical Problems in Engineering*, vol. 2012, Article ID 579130, 11 pages, 2012.
- [12] H. Niu and M. Zhang, "An optimization to schedule train operations with phase-regular framework for intercity rail lines," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 549374, 13 pages, 2012.
- [13] H. Niu and X. Zhou, "Optimizing urban rail timetable under time-dependent demand and oversaturated conditions," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 212–230, 2013.
- [14] T. Kis and A. Hertz, "A lower bound for the job insertion problem," *Discrete Applied Mathematics*, vol. 128, no. 2-3, pp. 395–419, 2003.
- [15] H. Gröflin and A. Klinkert, "Feasible insertions in job shop scheduling, short cycles and stable sets," *European Journal of Operational Research*, vol. 177, no. 2, pp. 763–785, 2007.
- [16] H. Gröflin, A. Klinkert, and N. P. Dinh, "Feasible job insertions in the multi-processor-task job shop," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1308–1318, 2008.
- [17] V. Cacchiani, A. Caprara, and P. Toth, "Scheduling extra freight trains on railway networks," *Transportation Research Part B: Methodological*, vol. 44, no. 2, pp. 215–231, 2010.
- [18] R. L. Burdett and E. Kozan, "Techniques for inserting additional trains into existing timetables," *Transportation Research Part B: Methodological*, vol. 43, no. 8-9, pp. 821–836, 2009.
- [19] L. Ingolotti, F. Barber, P. Tormos, A. Lova, M. Salido, and M. Abril, "An efficient method to schedule new trains on a heavily loaded railway network," in *Advances in Artificial Intelligence—IBERAMIA 2004*, vol. 3315 of *Lecture Notes in Computer Science*, pp. 164–173, 2004.
- [20] H. Flier, T. Graffagnino, and M. Nunkesser, "Scheduling additional trains on dense corridors," in *Experimental Algorithms*, vol. 5526 of *Lecture Notes in Computer Science*, pp. 149–160, Springer, 2009.
- [21] C. Strotmann, *Railway scheduling problems and their decomposition [Ph.D. dissertation]*, Universität Osnabrück, 2007.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

