Available online at www.sciencedirect.com
SCIENCE DIRECT
DISCRETE APPLIED MATHEMATICS
Discrete Applied Mathematics 154 (2006) 738–753
www.elsevier.com/locate/dam

A Lagrangian heuristic algorithm for a real-world train timetabling problem

Alberto Caprara[a], Michele Monaci[b], Paolo Toth[a], Pier Luigi Guida[c]

[a] *DEIS, University of Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy*
[b] *DEI, University of Padova Via Gradenigo 6/A, I-35131 Padova, Italy*
[c] *Rete Ferroviaria Italiana SpA, Piazza della Croce Rossa 1, I-00195 Roma, Italy*

Received 8 November 2003; received in revised form 25 July 2004; accepted 27 May 2005
Available online 14 November 2005

Abstract

The *train timetabling problem* (TTP) aims at determining an optimal timetable for a set of trains which does not violate track capacities and satisfies some operational constraints.

In this paper, we describe the design of a train timetabling system that takes into account several additional constraints that arise in real-world applications. In particular, we address the following issues:

- Manual block signaling for managing a train on a track segment between two consecutive stations.
- Station capacities, i.e., maximum number of trains that can be present in a station at the same time.
- Prescribed timetable for a subset of the trains, which is imposed when some of the trains are already scheduled on the railway line and additional trains are to be inserted.
- Maintenance operations that keep a track segment occupied for a given period.

We show how to incorporate these additional constraints into a mathematical model for a basic version of the problem, and into the resulting Lagrangian heuristic. Computational results on real-world instances from Rete Ferroviaria Italiana (RFI), the Italian railway infrastructure management company, are presented.

Keywords: Train scheduling; Path allocation; Lagrangian relaxation; Lagrangian heuristic; Computational results

1. Introduction

Management of main railway lines is increasingly becoming an important issue of European transport systems. Several reasons motivate better usage and planning of the rail infrastructure, particularly on the so-called European corridors, where track resource is limited due to greater traffic densities, and competitive pressure among the train operators is expected to increase in the near future. The availability of effective, computer-aided tools to improve the planning ability of railways over traditional methods is consequent to the new market scenario. Firstly, the re-organization of the European rail system, following the EU policy directives, has separated the activities of the Infrastructure Manager (who is responsible for train planning and real-time control) from the train operators (who provide rolling stock and

E-mail addresses: acaprara@deis.unibo.it (A. Caprara), monaci@dei.unipd.it (M. Monaci), ptoth@deis.unibo.it (P. Toth), pierluigi.guida@guidafs.inet.it (P.L. Guida).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2005.05.026

transport services). This introduces the so-called access-to-infrastructure problem, in which several operators request capacity on a common railway line. Secondly, European railways are being transformed into more liberalized and privatized companies, which are expected to compete on a more profit-oriented basis. Thirdly, the rail transport system is subject to increasing pressure by governments and social interest groups to improve its overall efficiency and quality of service. Finally, the strategic character of the sector is highlighted in view of ecological impacts and national policies aiming at spilling freight traffic shares from roads to rails.

In this context, the ability to undertake infrastructure planning in a very timely, smooth and efficient way is becoming one of the most important tasks of the infrastructure manager, who at the same time has to optimize the use of the infrastructure and to provide track allocation, through rational and transparent procedures, as required in normative directives, introduced in the European Union since the early 90s.

The general aim was to implement scheduling algorithms which can provide a timetable plan on heavy-traffic, long-distance corridors. Through this model the infrastructure manager can allocate "optimally" the paths requested by all transport operators and proceed with the overall timetable design process, possibly with final local refinements and minor adjustments, as in the tradition of railway planners. The algorithm we present here is called traffic capacity management (TCM), and is part of a more general telematic architecture developed within the EU Project PARTNER. In brief, this allows each train operator to submit requests for paths on the given railway line, and allows the infrastructure manager to collect all the requests, run the optimization algorithm to allocate (if possible) all of them at maximum profit, and eventually respond to train operators with the proposed plan of track allocation and relative "access fees".

The essential characteristics of the process can be summarized as follows:

- train paths are given a value (i.e., a priority), an ideal timetable, with ideal departure and arrival times, and tolerances within which they can be "moved";
- the optimal allocation is found by maximizing the difference between the values of the trains scheduled and a cost penalty function, which takes into account the deviations from the ideal timetables;
- a subset of paths can be fixed and shall not be moved, being either prioritary (as is the case for long-term or clock-phased services) or already allocated (i.e., "sold" to some train operating company);
- the scheduling algorithm uses typical parameters as in the current timetable planning (e.g., minimum headway between trains, available tracks at each station);
- maintenance operations, also called "possessions", can impose constraints or forbid regular operations of the planned timetable during some intervals;
- signaling failures can cause some degraded operational mode, switching from an automatic to a manual block system.

In addition, under the assumption of competitive market, the process can be iterated if some operator does not accept the solution and asks for a re-evaluation by the infrastructure manager, e.g., by using modified path values, since the paths allocated take into account the value or access fee that the train operator is willing to pay.

The train timetabling problem has received considerable attention in the literature.

Many references consider mixed integer linear programming formulations in which the arrival and departure times are represented by continuous variables and there are logical (binary) variables expressing the order of the train departures from each station. Szpigel [14] considers a variant of these models in which the order of the train departures from a station is not represented by binary variables but by disjunctive constraints. The problem is then solved by branch-and-bound for small size instances by computing bounds through the relaxation of these disjunctive constraints. Jovanovic and Harker [8] solve a version of these models, that calls for a feasible schedule rather than for the optimization of a suitable objective function, by branch-and-bound techniques. Cai and Goh [2] illustrate a constructive greedy heuristic driven by one of these models. Carey and Lockwood [4] define a heuristic that considers the trains one at a time (in appropriate order), and for each train solves a mixed integer linear program analogous to those mentioned above in order to schedule the train optimally, keeping the path of the previously scheduled trains partially fixed. More precisely, the relative order of the train departures for these trains is kept fixed, whereas their arrival and departure times may be changed. Higgins et al. [7] define local search, tabu search, genetic and hybrid heuristics, finding a feasible solution by using a model in the family above.

Brännlund et al. [1] discretize the time into 1-min *time slots* and subdivide the track line into *blocks*. Operational constraints impose that two trains cannot be in the same block in the same time slot. There is a binary variable x_{sbj}

each time the timetable constraints allow train j to be in block b in time slot s. This model is not suited for large size instances as those arising for the main European corridors.

Oliveira and Smith [10] model the problem as a special case of the Job-Shop Scheduling Problem, considering trains as jobs to be scheduled on tracks regarded as resources, and present a hybrid algorithm devised under the Constraint Programming paradigm, showing how to adapt this framework in some special real-life applications.

Schrijver and Steenbeek [13], Lindner and Zimmermann [9], and Peeters and Kroon [12] consider the case in which the timetable is identical with a period of one hour (rather than one day as is the case of the problem considered in the other references), and address the general case of a railway network instead of a single (main) line. The problem is solved through a mixed integer linear programming formulation in which the times are again represented by continuous variables and integer variables are used to impose that the differences between pairs of time variables belong to a certain interval modulo one hour. Further references on this version of the problem can be found in Peeters [11].

In the next section, we give a formal description of the *basic train timetabling problem* (TTP) considered by Caprara et al. [3], which addresses the main characteristics of the problem, together with the mathematical formulation and the associated Lagrangian heuristic proposed in [3]. In Section 3, we give a stronger mathematical formulation of the problem under an assumption which is widely satisfied in practice. Finally, in Section 4, we describe a number of additional real-world constraints with respect to the basic problem, discussing how the approach illustrated in the previous sections can be modified to take them into account, and presenting computational results that show the impact of the new constraints on the quality of the solutions found. Some concluding remarks are given in Section 5.

2. The basic problem and its solution

In this section, we illustrate the basic version of TTP considered by Caprara et al. [3] and the corresponding solution approach. This research was performed within the 4th EU-RDT framework, under the TRIS (Teleconferencing Railway Information System) project, in order to develop innovative algorithms to support the access-to-infrastructure on the European main lines.

2.1. The basic train timetabling problem

The basic problem illustrated below is a brief summary of the problem considered by the traditional rail timetable planners, whose final product is conventionally exhibited as timetable "diagrams".

We consider a single, one-way track corridor linking two major stations, with a number of intermediate stations in between. The trains have to be run every day of a given time horizon (e.g., 6–12 months). Times are here discretized and expressed as integers from 1 to $q := 1440$ (the number of minutes in a day), though a finer discretization would also be possible (e.g., $\frac{1}{2}$, $\frac{1}{4}$ minute) without changing the model. In the following, *time instants* indicate a particular instant within the day, while *time intervals* indicate a continuous interval (not longer than q). Let $S = \{1, \ldots, s\}$ represent the set of stations, numbered according to the order in which they appear along the track for the running direction considered, and $T = \{1, \ldots, t\}$ denote the set of trains which are candidate to be run every day of the given time horizon. For each train $j \in T$, a *first* (departure) station f_j and a *last* (destination) station l_j ($l_j > f_j$) are given. Let $S^j := \{f_j, \ldots, l_j\} \subseteq S$ be the ordered set of stations visited by train j.

The *track capacity constraints* impose that overtaking between trains occurs only within a station. To this end, a train is allowed to stop in any intermediate station to give the possibility to some other train to overtake it. Furthermore, for each station $i \in S$, there are lower bounds a_i and d_i on the time interval between two consecutive arrivals and two consecutive departures, respectively. This guarantees that the trains can be scheduled regularly, that is having a sufficient headway between each other for safe and regular train operations, besides contingent delays or schedule disruptions.

The line equipment which takes care of safely controlling the train sequencing on the track is commonly known as the "safety block system", that divides the line track between two consecutive stations into a number of block sections, allowing each section to be occupied by at most one train in each time instant. Actually, the system uses *protection signals* at each block entry, and, for two consecutive blocks b and $b + 1$, if a train is currently traveling along block $b + 1$, there is a *red light* at the beginning of block $b + 1$ and a *yellow light* at the beginning of block b, whereas in the other cases there is a *green light*. This protection system goes under the name of "three aspects block", and corresponds to the so-called *automatic block signaling* between two consecutive stations. Accordingly, the regular running situation is (approximately) modeled by imposing the minimum time distance between consecutive arrivals and departures of trains at stations, as illustrated above.

A *timetable* defines, for each train $j \in T$, the departure time from f_j, the arrival time at l_j, and the arrival and departure times for the intermediate stations $f_j+1, \ldots, l_j-1$. Note that, for some intermediate station, the arrival and departure time can coincide. The timetable of each train is *periodic*, i.e, it is kept unchanged every day. The *running time* of train *j* in the timetable is the time interval between the departure from f_j and the arrival at l_j.

Each train is assigned by the train operator an *ideal timetable*, representing the most desirable timetable for the train, that may be modified in order to satisfy the track capacity constraints. In particular, one is allowed to slow down (but not speed up) each train with respect to its ideal timetable, and/or to increase (but not decrease) the stopping time interval at the (intermediate) stations. Moreover, one can modify (anticipate or delay) the departure time of each train from its first station, or even *cancel*, i.e., not schedule, the train. The timetable for train *j* in the final solution for TTP will be referred to as the *actual timetable*.

Note that track capacity constraints, along with the fact that the actual timetable has to be repeated every day, may force some trains to be canceled to obtain a feasible solution.

The *objective* is to maximize the overall profit for the infrastructure manager, i.e., the sum of the profits of the scheduled trains. According to the charging rules generally adopted by the infrastructure managers, the *profit* achieved for each train $j \in T$ is given by $\pi_j - \phi_j(v_j) - \gamma_j \mu_j$, where π_j is the *train priority*, that is the profit that would be achieved if the train traveled according to its ideal timetable, v_j is the *shift*, that is the absolute difference between the departure times from station f_j in the ideal and actual timetables, and μ_j is the *stretch*, that is the (nonnegative) difference between the running times in the actual and ideal timetables. Moreover, $\phi_j(v_j)$ is a user-defined nondecreasing function penalizing the train shift (with $\phi_j(0)=0$), and γ_j is a given nonnegative parameter (i.e., the function penalizing the train stretch is assumed to be linear). Of course, if the profit of train *j* turns out to be nonpositive, it is better to cancel train *j*.

The objective function is aimed at optimizing the profit value of the railway line capacity from the infrastructure manager's perspective, reaching an economic equilibrium with the overall willingness of the train operators to pay. On the other hand, the trains are scheduled with the smallest possible "deviation" from the ideal commercial timetable initially requested.

It was shown in [3] that the problem is NP-hard in the strong sense, being a generalization of the well-known maximum independent set problem.

2.2. A graph representation

Let $G=(V,A)$ be the directed acyclic multigraph defined as follows. The node set *V* has the form $\{\sigma,\tau\} \cup (U^2 \cup \cdots \cup U^s) \cup (W^1 \cup \cdots \cup W^{s-1})$, where σ and τ are an *artificial source node* and an *artificial sink node*, respectively, whereas sets U^i, $i \in S \backslash \{1\}$, and W^i, $i \in S \backslash \{s\}$, represent the set of time instants in which some train can arrive at and depart from station *i*, respectively. We call the nodes in $U^2 \cup \cdots \cup U^s$ and $W^1 \cup \cdots \cup W^{s-1}$ *arrival* and *departure* nodes, respectively. Let $\theta(v)$ be the time instant associated with a given node $v \in V$. Moreover, let $\Delta(u,v) := \theta(v) - \theta(u)$ if $\theta(v) \geqslant \theta(u)$, $\Delta(u,v) := \theta(v) - \theta(u) + q$ otherwise. Because of the "cyclic" nature of the time horizon, we say that node *u precedes* node *v* (i.e., $u \preceq v$) if $\Delta(v,u) \geqslant \Delta(u,v)$ (i.e., if the cyclic time interval between $\theta(v)$ and $\theta(u)$ is not smaller than the cyclic time interval between $\theta(u)$ and $\theta(v)$).

Note that not all time instants correspond to possible arrivals/departures of train *j* at a station $i \in S^j$. More precisely, let $v_j^{\max} := \max\{v : \pi_j - \phi_j(v) > 0\}$ and $\mu_j^{\max} := \max\{\mu : \pi_j - \gamma_j \mu > 0\}$ be the maximum shift and stretch, respectively, in a timetable for train *j* that has positive profit. Considering a station $i \in \{f_j+1, \ldots, l_j\}$ and the node $u \in U^i$ corresponding to the ideal arrival time of train *j* at station *i*, all nodes $w \in U^i$ such that either $w \prec u$ and $\Delta(w,u) > v_j^{\max}$ or $w \succ u$ and $\Delta(u,w) > \max\{v_j^{\max}, \mu_j^{\max}\}$ do not correspond to arrivals of train *j* in any positive-profit timetable. The same holds for departure nodes $w \in W^i$ for $i \in \{f_j, \ldots, l_j-1\}$, noting that for $i=f_j$ condition $\Delta(u,w) > \max\{v_j^{\max}, \mu_j^{\max}\}$ can be replaced by the stronger $\Delta(u,w) > v_j^{\max}$, *u* being now the node corresponding to the ideal departure time of train *j* from f_j. Accordingly, we let $V^j \subseteq \{\sigma,\tau\} \cup (U^{f_j+1} \cup \cdots \cup U^{l_j}) \cup (W^{f_j} \cup \cdots \cup W^{l_j-1})$ denote the nodes associated with possible arrivals/departures of train *j* in a positive-profit timetable.

The arc set A is partitioned into sets $A^1, \ldots, A^t$, one for each train $j \in T$. In particular, for every train $j \in T$, A^j contains

- a set of *starting* arcs (σ, v), for each $v \in W^{f_j} \cap V^j$, whose profit is $p_{(\sigma,v)} := \pi_j - \phi_j(v(v))$, where $v(v) := |\theta(v) - [\text{ideal departure time of train } j \text{ from station } f_j]|$ is the *shift* associated with node v;
- a set of *station* arcs (u, v), for each $i \in S^j \setminus \{f_j, l_j\}$, $u \in U^i \cap V^j$ and $v \in W^i \cap V^j$ such that $\Delta(u, v)$ is at least equal to the minimum stop time of train j in station i, whose profit is $p_{(u,v)} := -\gamma_j \mu(u, v)$, where $\mu(u, v) := \Delta(u, v) - [\text{ideal stop time for train } j \text{ in station } i]$ is the stretch associated with arc (u, v);
- a set of *segment* arcs (v, u), for each $i \in S^j \setminus \{l_j\}$, $v \in W^i \cap V^j$ and $u \in U^{i+1} \cap V^j$ such that $\Delta(v, u)$ is at least equal to the minimum travel time of train j from station i to station $i + 1$, whose profit is $p_{(v,u)} := -\gamma_j \mu(u, v)$, where $\mu(v, u) := \Delta(v, u) - [\text{ideal travel time for train } j \text{ from station } i \text{ to station } i + 1]$ is the stretch associated with arc (v, u);
- a set of *ending* arcs (u, τ), for each $u \in U^{l_j} \cap V^j$, whose profit is $p_{(u,\tau)} := 0$.

Note that, for each train $j \in T$, the number of arcs in A^j is proportional to the number of stations visited by j, to the maximum number of nodes in V^j in each station, and to the maximum number of arcs starting from each node in V^j, i.e.,

$$|A^j| = \mathrm{O}((l_j - s_j) \max\{v_j^{\max}, \mu_j^{\max}\} \mu_j^{\max}). \tag{1}$$

By construction, G is *acyclic* and each path from σ to τ in G which uses only arcs in A^j and has profit p corresponds to a feasible timetable for train j having profit p; in other words all the feasibility constraints pertaining to train j, as well as the objective function, are implicitly enclosed in graph G.

To satisfy the track capacity constraints, one should impose that certain pairs of arcs, associated with different trains, cannot be selected in the overall solution. In particular, it is sufficient to state that, for each pair of trains j, k and for each station $i \in (S^j \setminus \{l_j\}) \cap (S^k \setminus \{l_k\})$, the two segment arcs $(v_1, u_1) \in A^j$ and $(v_2, u_2) \in A^k$, $v_1 \in W^i \cap V^j$, $v_2 \in W^i \cap V^k$, $u_1 \in U^{i+1} \cap V^j$, $u_2 \in U^{i+1} \cap V^k$ cannot be both selected if either $v_1 \preceq v_2$ and $\Delta(v_1, v_2) < d_i$, or $u_1 \preceq u_2$ and $\Delta(u_1, u_2) < a_{i+1}$, or $v_1 \preceq v_2$ and $u_2 \preceq u_1$ (the two arcs *cross* each other).

2.3. An integer linear programming formulation

An *integer linear programming* (ILP) formulation of TTP is the following. For each $j \in T$ and each arc $a \in A^j$, we introduce a binary variable x_a equal to 1 if and only if arc a is selected in an optimal solution, i.e., the path in the solution associated with train j contains arc a. For notational convenience, for each node $v \in V$ and $j \in T$, let $\delta_j^+(v)$ and $\delta_j^-(v)$ denote the sets of arcs in A^j leaving and entering node v, respectively.

It is convenient to introduce additional variables, associated with the nodes of G, in order to model the track capacity constraints.

For each node $v \in V$, let y_v be a binary variable equal to 1 if and only if there exists a train j whose associated path visits node v. Moreover, for each train $j \in T$ and for each node $v \in V^j$ let z_{jv} be a binary variable equal to 1 if and only if the path of train j visits node v.

The overall model for the problem then reads

$$\max \sum_{j \in T} \sum_{a \in A^j} p_a x_a \tag{2}$$

$$\text{s.t.} \sum_{a \in \delta_j^+(\sigma)} x_a \leqslant 1, \quad j \in T, \tag{3}$$

$$\sum_{a \in \delta_j^-(v)} x_a = \sum_{a \in \delta_j^+(v)} x_a, \quad j \in T, \quad v \in V^j \setminus \{\sigma, \tau\}, \tag{4}$$

$$z_{jv} = \sum_{a \in \delta_j^-(v)} x_a, \quad j \in T, \; v \in V^j, \tag{5}$$

$$y_v = \sum_{j \in T : v \in V^j} z_{jv}, \quad v \in V, \tag{6}$$

$$\sum_{w \in U^i : w \succeq u, \Delta(u,w) < a_i} y_w \leqslant 1, \quad i \in S \backslash \{1\}, \; u \in U^i, \tag{7}$$

$$\sum_{w \in W^i : w \succeq v, \Delta(v,w) < d_i} y_w \leqslant 1, \quad i \in S \backslash \{s\}, \; v \in W^i, \tag{8}$$

$$\sum_{w \in W^i \cap V^j : w \preceq v} z_{jw} + \sum_{w \in U^{i+1} \cap V^j : w \succeq u} z_{jw} + \sum_{w \in W^i \cap V^k : w \succeq v} z_{kw} + \sum_{w \in U^{i+1} \cap V^k : w \preceq u} z_{kw} \leqslant 3,$$

$$i \in S \backslash \{s\}, \quad j,k \in T, \quad j \neq k, i, i+1 \in S^j \cup S^k, \quad v \in W^i, \quad u \in U^{i+1}, \tag{9}$$

$$x_a \geqslant 0, \quad a \in A, \tag{10}$$

$$x_a \text{ integer}, \quad a \in A. \tag{11}$$

The objective function (2) is defined as the sum of the profits of the arcs associated with each path in the solution. Constraints (3) impose that at most one arc associated with a train is selected among those leaving the starting node σ, while constraints (4) impose equality on the number of selected arcs associated with a train entering and leaving each arrival or departure node. Consequently, the set of selected arcs associated with a train can either be empty, or define a path from the source σ to the sink τ. More precisely, (3) and (4), along with the nonnegativity constraints on the x variables, define the *convex hull* of the incidence vectors of the set of paths from σ to τ for train j, including the empty path. Constraints (5) and (6) express the link among the x, z, and y variables, noting that $\sum_{j \in T} z_{jv} \leqslant 1$ for $v \in V^j$ since two trains cannot depart/arrive at the same station at the same time. The *arrival time* constraints (7) and the *departure time* constraints (8) prevent two consecutive arrivals and departures at the same station i to be too close in time. (Recall that a_i and d_i represent the minimum time intervals between consecutive arrivals at and departures from i, respectively.) Both the arrival and the departure time constraints are *clique* inequalities. Note that these constraints are not sufficient since two trains traveling at different speeds may still overtake each other between stations i and $i+1$, even if their departures from i and arrivals at $i+1$ are sufficiently distant in time. Accordingly, the *overtaking* constraints (9) forbid overtakings between consecutive stations i and $i+1$. Specifically, for a station i, considering two trains j, k and nodes $v \in W^i$ and $u \in U^{i+1}$, the constraints forbid thesimultaneous selection of arcs $(v_1, u_1) \in A^j$, $(v_2, u_2) \in A^k$ such that $v_1, v_2 \in W^i$, $v_1 \preceq v \preceq v_2$ and $u_1, u_2 \in U^{i+1}$, $u_2 \preceq u \preceq u_1$ (i.e., the two arcs cross).

Given the size of the above ILP (2)–(11) for the real-world instances that we consider (see Section 2.5), for which the order of magnitude of both the number of variables and the number of constraints is 10^5–10^6, its exact solution cannot be obtained in reasonable computing time. Indeed, even the solution of the corresponding continuous relaxation (2)–(10) is very time consuming. Hence one has to resort to heuristic approaches.

2.4. Lagrangian relaxation and heuristic

By relaxing the inequalities (7)–(9) in a Lagrangian way, the resulting Lagrangian relaxed problem reads

$$\max \sum_{j \in T} \sum_{a \in A^j} p_a x_a - \sum_{v \in V} \bar{q}_v y_v - \sum_{j \in T} \sum_{v \in V^j} \bar{r}_{jv} z_{jv} + \sum_h b_h \lambda_h \tag{12}$$

subject to (3)–(6), (10), (11), where $\bar{q}_v$, $v \in V$, and $\bar{r}_{jv}$, $j \in T$, $v \in V^j$, are *Lagrangian penalties* associated with the node variables and λ_h denotes the Lagrangian multiplier associated with the hth relaxed constraint, with $b_h = 1$ if this constraint belongs to (7) or (8) and $b_h = 3$ if it belongs to (9). Since there is no constraint involving variables associated with different trains, the Lagrangian relaxed problem can be optimally solved by computing, for each train $j \in T$, the path of maximum Lagrangian profit from σ to τ in G that uses only arcs in A^j. Given a path for train j, with nodes

$Q^j \subseteq V^j$ and arcs $P^j \subseteq A^j$, the corresponding Lagrangian profit is

$$\sum_{a \in P^j} p_a + \sum_{v \in Q^j} (\bar{q}_v + \bar{r}_{jv}).$$

A maximum Lagrangian profit path for train j can easily be computed in $O(|A^j|)$ time (recalling bound (1) on $|A^j|$). The corresponding value of the objective function (12) gives a valid upper bound for the original problem.

Near optimal Lagrangian multipliers are determined by an iterative subgradient procedure. The very large number of track capacity constraints that are relaxed in a Lagrangian way is handled according to a so-called *relax-and-cut* framework, see Fisher [6] and Escudero et al. [5], explicitly considering each constraint only when it turns out to be violated by the relaxed solution at some iteration of the subgradient procedure, in which case the constraint is stored in a *pool* structure and assigned a nonnegative Lagrangian multiplier (the multipliers for the constraints not considered explicitly are obviously set to 0).

At each iteration of the subgradient optimization procedure, besides solving the corresponding Lagrangian relaxed problem, the approach of [3] also computes a heuristic solution by ranking the trains by decreasing values of the Lagrangian profit of the associated path in the relaxed solution, and then by scheduling the trains one by one. The schedule of the previous trains being fixed, finding the (locally optimal) schedule of the next train j in the sequence simply amounts to the solution of a maximum profit path problem in the subgraph of G resulting from the removal of the arcs of A^j which are not compatible with those corresponding to the trains already scheduled. This maximum profit path is computed with respect to *Lagrangian* profits. Afterwards, once a complete solution S is found, one tries to improve it by rescheduling the trains according to the same sequence, but using the *actual* profits (i.e., not considering the Lagrangian penalties), possibly scheduling also trains that were canceled in solution S. The use of Lagrangian profits to determine S, beside leading to better solutions than those obtained by using the actual profits, yields substantially different heuristic solutions for different Lagrangian multipliers, increasing the likelihood of improving the incumbent solution.

After a certain number of subgradient iterations, a suitable subset of trains is removed or fixed, and the subgradient optimization procedure is restarted, taking into account the removed and fixed trains also in the computation of the Lagrangian solution. Of course, the value of this solution is no more a valid upper bound for the original problem. In particular, if some train is canceled in the incumbent solution, the r canceled trains having the smallest Lagrangian profits are *removed*. Otherwise, the paths of the r trains with largest Lagrangian profits are *fixed* in the solution. When the Lagrangian upper bound certifies that, with the trains currently removed and fixed, the incumbent solution cannot be improved, the r removed trains having the largest Lagrangian profits are *unremoved* and the r fixed trains having the smallest Lagrangian profits are *unfixed*. The overall process is stopped after a prefixed number of subgradient iterations.

For a detailed description of the above Lagrangian heuristic, the reader is referred to [3].

2.5. Real-world instances

The algorithm was implemented in C and tested on a set of instances from Rete Ferroviaria Italiana (the Italian railway infrastructure management company, RFI for short), a company of Ferrovie dello Stato (FS holding group).

For each station i, the lower bounds a_i and d_i between consecutive arrivals and departures of trains are set to 4 and 2 minutes, respectively, according to the current operational rules. The function penalizing the train shift of each train j is defined as $\phi_j(v_j) := \alpha_j v_j$, i.e., the penalty is linear in the shift. The coefficients used in the objective function are illustrated in Table 1. In particular, the profit coefficients are identical for the trains of the same type. For instance, the profit achieved if a Eurostar train is scheduled according to its ideal timetable is 200, and there is a penalty of 7 for each minute of shift as well as a penalty of 10 for each minute of stretch. Note that the shift is penalized less than the stretch for all train types.

The main characteristics of the instances are outlined in Table 2. Column # trains gives the total number of trains on input followed by an array with the number of Eurostar, Euronight, Intercity, Express, Combined, Direct, Local and Freight Trains, respectively. Furthermore, Column Ideal profit gives the sum of the profits achievable by scheduling each train according to its ideal timetable. We remark that the instances used in this paper are slightly different with respect to those considered in [3]. In particular, the original train type "Intercity" has been split into train types "Intercity" and

Table 1
Train profit coefficients depending on the train type

Train type	π_j	α_j	γ_j
Eurostar	200	7	10
Euronight	150	7	10
Intercity	120	6	9
Express	110	5	8
Combined	100	6	9
Direct	100	5	8
Local	100	5	6
Freight	100	2	3

Table 2
Characteristics of the instances considered

Instance	First station	Last station	# Stations	# Trains	Ideal profit
PC-BO-a	Piacenza	Bologna	17	221 (28,3,17,35,35,28,14,61)	25 740
PC-BO-b	Piacenza	Bologna	17	60 (12,1,10,0,4,12,7,14)	7450
PC-BO-c	Piacenza	Bologna	17	40 (6,0,10,0,0,12,2,10)	4800
BN-BO-a	Brennero	Bologna	48	68 (1,0,5,13,0,11,38,0)	7130
MU-VR-a	Munich	Verona	49	54 (0,0,0,7,0,0,47,0)	5470

Table 3
Results with the original code

Name	Best UB	Greedy sol.	Best sol.	% Gap	# Sched.	Avg. ν	Avg. μ	Time
PC-BO-a	24 372 (5.3%)	19 757	21 200 (7.3%)	13.01%	188 (85.0%)	0.8	0.6	6225
PC-BO-b	7231 (2.9%)	6676	7117 (6.6%)	1.58%	60 (100.0%)	0.7	0.7	132
PC-BO-c	4269 (11.1%)	3181	3624 (13.9%)	15.11%	35 (87.5%)	3.9	0.9	422
BN-BO-a	6897 (3.3%)	6746	6771 (0.4%)	1.83%	68 (100.0%)	0.6	0.3	78
MU-VR-a	5068 (7.3%)	3332	4233 (27.0%)	16.48%	48 (88.8%)	1.3	1.1	166

"Combined", with different profit coefficients. In addition, few intermediate stations in which no overtaking can occur are now explicitly considered.

Table 3 reports the results obtained by the algorithm proposed in [3] on the instances illustrated above. We ran the algorithm on a single processor Digital Ultimate Workstation 533 MHz (running Digital Unix V4.OD, and having 16.1 SPECint95 value), with a limit of 1000 subgradient iterations, and the fixing/removing parameter r set to 1. For each instance, we performed a separate run without the train fixing/removing phase to compute a tighter upper bound. The columns in the table have the following meaning:

- Best UB is the best upper bound found by the subgradient optimization procedure, with (in brackets) the percentage improvement (i.e., decrease) with respect to the ideal profit (which is a trivial upper bound on the optimal solution value).
- Greedy sol. is the solution value found by scheduling the trains by decreasing values of π_j (breaking ties arbitrarily) and assigning each train the timetable corresponding to the maximum profit path compatible with the previous trains. The manual methods proceed in a similar way, therefore the quality of the solution provided by the practitioners is close to the value given in this entry.
- Best sol. is the value of the best solution found by the Lagrangian heuristic method, with (in brackets) the percentage improvement with respect to the greedy solution.
- % Gap is the percentage gap between Best UB and Best sol.
- # Sched. is the number of trains scheduled in the best solution, with (in brackets) the percentage with respect to the total number of trains.

- Avg. v is the average shift (in minutes) for the trains scheduled in the best solution.
- Avg. μ is the average stretch (in minutes) for the trains scheduled in the best solution.
- Time is the overall running time needed to find the best solution (expressed in seconds).

Note that, due to the relatively large values of α_j and γ_j with respect to π_j (for each $j \in T$), the average shift and stretch have small values. Moreover, the method often finds alternative solutions that schedule a larger number of trains with respect to the best solution, but with a smaller global profit.

Finally, we observe that the structure of the solution is not significantly affected by the specific values assigned to the train profit coefficients. In order to verify this, we ran the algorithm on the instances above leaving π_j unchanged and setting, starting from the parameter values of Table 1, (i) $\alpha_j := \gamma_j$; (ii) $\gamma_j := \alpha_j$; (iii) $\alpha_j := 2/3\alpha_j$ and $\gamma_j := 2/3\gamma_j$; (iv) $\alpha_j := 3/2\alpha_j$ and $\gamma_j := 3/2\gamma_j$ (for each train $j \in T$). In all cases, the solution found with the original coefficients, when evaluated with the new coefficients, turned out to have a value within 2% of that of the best solution found by the algorithm with the new coefficients.

3. A stronger version of the overtaking constraints

With respect to the model proposed in [3] and illustrated in the previous section, the new method is based on the assumption that the travel time of each train along each track segment joining two stations is *fixed* and coincides with that of the ideal timetable, i.e., it is not possible to slow down the train along the track. In other words, the departure time from station i uniquely determines the arrival time at station $i+1$. Even if, of course, slowing down is something that has to be done at the operational level, within our model, at the planning level, this assumption is motivated by the fact that in practice slowing down a train between stations i and $i+1$ is in almost all cases equivalent to forcing the train to stop in i for a longer time (and then to travel at its regular speed from i to $i+1$). The latter statement is not true in general, as shown by some pathological examples, but it holds for realistic cases. In particular, experimental results show that the best solution value found by the heuristic procedure described in Section 2.4 is marginally affected by this additional constraint, whereas the corresponding running time per subgradient iteration is widely reduced, since the graph G turns out to be much smaller (for every train, the number of segment arcs turns out to be equal to the number of departure nodes). Furthermore, the above assumption simplifies the mathematical representation of the problem, yielding simpler and stronger overtaking constraints, and makes it also easier to model some of the operational constraints described in the next section.

In the following, we describe a stronger formulation of the overtaking constraints derived by taking into account the assumption above. Given two consecutive stations i and $i+1$ along with two trains j, k such that $i, i+1 \in S^j \cap S^k$, let

$$b_i^{jk} := \max\{d_i, a_{i+1} + t_j - t_k\}$$

denote the minimum time interval between a departure of j from i and a departure of k from i (in this order) in a feasible solution, where t_j and t_k are the travel times of j and k from i to $i+1$, respectively. Note that if j and k depart from i with distance less than b_i^{jk}, either their departures are too close in time, or their arrivals are too close in time, or k overtakes j between i and $i+1$.

The new overtaking constraints are defined by specifying a station $i \in S \backslash \{s\}$, two trains $j, k \in T$ such that $i, i+1 \in S^j \cap S^k$, and two nodes $v_1 \in W^i$ and $v_2 \in W^i$ such that $v_1 \preceq v_2$ and $\Delta(v_1, v_2) < b_i^{jk}$. Nodes v_1 and v_2 represent the earliest departure from i for trains j and k, respectively, involved in the new constraints, that read

$$\sum_{w \in W^i \cap V^j : v_1 \preceq w \prec v_2} z_{jw} + \sum_{w \in W^i \cap V^j : w \succeq v_2, \Delta(v_2, w) < b_i^{kj}} z_{jw} + \sum_{w \in W^i \cap V^k : w \succeq v_2, \Delta(v_1, w) < b_i^{jk}} z_{kw} \leqslant 1,$$

$$i \in S \backslash \{s\}, \quad j, k \in T, \quad j \neq k, \; i, i+1 \in S^j \cap S^k, \quad v_1, v_2 \in W^i, \quad v_1 \preceq v_2, \quad \Delta(v_1, v_2) < b_i^{jk}. \tag{13}$$

Stated in words, the constraints forbid the simultaneous departure from station i of train j at a time instant between $\theta(v_1)$ and $\theta(v_2) + b_i^{kj} - 1$ and of train k at a time instant between $\theta(v_2)$ and $\theta(v_1) + b_i^{jk} - 1$. Since the departure of j at time $\theta(v_2) + b_i^{kj}$ is compatible with the departure of k at time $\theta(v_2)$, and the departure of k at time $\theta(v_1) + b_i^{jk}$ is compatible with the departure of j at time $\theta(v_1)$, the time windows in the constraint cannot be enlarged.

Proposition 1. *Overtaking constraints* (13) *dominate overtaking constraints* (9).

Proof. We show that (9) are linearly implied by (13) and the remaining linear constraints in ILP (2)–(11).

For convenience and with an abuse of notation, in this proof, given a time interval t and a node u, we will write $u-t$ to denote a node w such that $\theta(w)=\theta(u)-t$. Consider a constraint (9). Assuming the travel time of train j from station i to station $i+1$ to be constant and equal to t_j, we have

$$\sum_{w\in U^{i+1}\cap V^j:w\succeq u} z_{jw}=\sum_{w\in W^i\cap V^j:w\succeq u-t_j} z_{jw}.$$

Moreover,

$$\sum_{w\in W^i\cap V^j}\ \sum_{a\in\delta_j^-(w)} x_a\leqslant 1$$

is valid for any path for train j, and therefore implied by (3), (4) and (10), that define the convex hull of the incidence vectors of paths for each train. Jointly with (5) this yields

$$\sum_{w\in W^i\cap V^j} z_{jw}\leqslant 1.$$

Using the above inequalities, we get

$$\begin{aligned}\sum_{w\in W^i\cap V^j:w\preceq v} z_{jw}+\sum_{w\in U^{i+1}\cap V^j:w\succeq u} z_{jw}&=\sum_{w\in W^i\cap V^j:w\preceq v} z_{jw}+\sum_{w\in W^i\cap V^j:w\succeq u-t_j} z_{jw}\leqslant\sum_{w\in W^i\cap V^j} z_{jw}\\&\quad+\sum_{w\in W^i\cap V^j:u-t_j\preceq w\preceq v} z_{jw}\leqslant 1+\sum_{w\in W^i\cap V^j:u-t_j\preceq w\preceq v} z_{jw}.\end{aligned}$$

Reasoning in the same way for train k, we get

$$\sum_{w\in W^i\cap V^k:w\succeq v} z_{kw}+\sum_{w\in W^i\cap V^k:w\preceq u} z_{kw}\leqslant 1+\sum_{w\in W^i\cap V^k:v\preceq w\preceq u-t_k} z_{kw},$$

where t_k is the travel time of train k from i to $i+1$. This implies that the left-hand-side of (9), due to the other linear constraints in (2)–(11), is at most

$$\begin{aligned}&2+\sum_{w\in W^i\cap V^j:u-t_j\preceq w\preceq v} z_{jw}+\sum_{w\in W^i\cap V^k:v\preceq w\preceq u-t_k} z_{kw}\\&\leqslant 2+\sum_{w\in W^i\cap V^j:u-t_j\preceq w\prec v} z_{jw}+\sum_{w\in W^i\cap V^j:w\succeq v,\varDelta(v,w)<b_i^{kj}} z_{jw}+\sum_{w\in W^i\cap V^k:w\succeq v,\varDelta(u-t_j,w)<b_i^{jk}} z_{kw}\end{aligned}\tag{14}$$

noting that $\theta(u)-t_j+b_i^{jk}\geqslant\theta(u)-t_j+(a_{i+1}+t_j-t_k)>\theta(u)-t_k$. Now, defining $v_1:=u-t_j$ and $v_2:=v$, we have that (13) implies that (14) is at most 3, completing the proof. □

The handling of constraints (13) within the subgradient optimization procedure is analogous to the case of constraints (9) in [3]. Namely, suppose in the relaxed solution we have $z_{jw_1}=z_{kw_2}=1$ for two trains j,k and nodes $w_1,w_2\in W^i$ such that $w_1\preceq w_2$, $\varDelta(w_1,w_2)<b_i^{jk}$, $\varDelta(w_1,w_2)\geqslant d_i$ and $\varDelta(u_2,u_1)\geqslant a_{i+1}$, where $u_1,u_2\in U^{i+1}$ are the arrival nodes for j and k in $i+1$ corresponding to the departure at w_1 and w_2, respectively. In other words, the departure of the two trains from i violates a constraint (13) but no constraint (7) or (8). In this case, we check if the constraint pool contains already a constraint (13) in which the coefficient of both z_{jw_1} and z_{kw_2} is 1 (in which case we increase the Lagrangian multiplier of the constraint). If this is not the case, we add to the pool the constraint (13) for which the time window associated with variables z_{jw} with coefficient 1 is centered around w_1 and the time window associated with variables z_{kw} with coefficient 1 is centered around w_2. Table 4 gives the upper bounds provided by the old model with constraints

Table 4
Comparison of the upper bounds provided by the old and the new model

Name	Best UB with constr. (9)	Best UB with constr. (13)
PC-BO-a	24 174 (6.1%)	24 226 (5.8%)
PC-BO-b	7225 (3.0%)	7236 (2.8%)
PC-BO-c	4223 (12.0%)	4302(10.3%)
BN-BO-a	6903 (3.2%)	6909 (3.0%)
MU-VR-a	5028 (8.1%)	5032 (8.0%)

Table 5
Results with the new code

Name	Best UB	Greedy sol.	Best sol.	% Gap	# Sched.	Avg. v	Avg. μ	Time
PC-BO-a	24 226 (5.8%)	19 757	21 250 (7.5%)	11.34%	192 (86.8%)	1.1	0.5	2398
PC-BO-b	7236 (2.8%)	6676	7112 (6.5%)	1.08%	60 (100.0%)	0.6	0.7	190
PC-BO-c	4302(10.3%)	3181	3593 (13.0%)	14.78%	34 (85.0%)	3.2	1.0	217
BN-BO-a	6909 (3.0%)	6746	6771 (0.4%)	1.74%	68 (100.0%)	0.6	0.3	227
MU-VR-a	5032 (8.0%)	3332	4222 (26.7%)	16.01%	48 (88.8%)	1.2	1.2	296

(9) (by imposing fixed travel times between stations) and by the new model with constraints (13). Observe that the upper bound values for the old model reported in Table 4 are slightly smaller than those reported in Table 3 since we imposed fixed travel times between stations. The only exception is instance BN-BO-a, due to the heuristic nature of the subgradient optimization procedure.

Finally, in Table 5 we report the results of the new version of the algorithm on the real-world instances. On average, this new version produces solutions whose quality is comparable with that of the original approach, within smaller computing times.

4. Additional constraints of the real-world problem

In this section, we illustrate how to extend the model illustrated in the previous section to take into account more realistic situations, in the attempt to make the model able to handle additional scenarios arising in railway operations and increase its overall flexibility in real-life case studies. The section is organized into small subsections, each dealing with a different case and presenting computational results illustrating the impact of the variations on the solution of real-world problems. Computational experience showed that, for these more constrained problems, better results are obtained by setting the fixing/removing parameter r to $\max\{1, \lfloor t'/20 \rfloor\}$, where t' denotes the current number of trains to be scheduled.

4.1. Manual block signaling

In the basic model, we consider the automatic block signaling between stations, illustrated in Section 2. Although the modern railway networks use automatic block signaling only, there are some lines (or parts of lines) in which the block signaling is still manual. Moreover, *failures* in the automatic block signaling system between two or more stations may force downgrading to manual block signaling.

Essentially, manual signaling is like having a unique block between two consecutive stations. Formally, considering the line segment between stations i and $i+1$, the track capacity constraints impose that there cannot be two or more trains in the segment at the same time, i.e., a train cannot depart from station i if there is already a train traveling from i to $i+1$. More precisely, given two trains traveling from i to $i+1$ one after the other, a minimum *block reset* time equal to c_i must elapse between the arrival of the first train in $i+1$ and the departure of the second train from i.

Table 6
Results with manual block signaling

Name	Best UB	Greedy sol.	Best sol.	% Gap	# Sched.	Avg. v	Avg. μ	Time
PC-BO-a	22385 (13.0%)	13809	15 186 (10.0%)	32.16%	133 (60.1%)	2.0	0.9	3250
PC-BO-b	6998 (6.1%)	6007	6533 (8.8%)	6.64%	56 (93.3%)	1.2	1.4	490
PC-BO-c	4136 (13.2%)	2541	2957 (16.4%)	28.51%	28 (70.0%)	3.8	1.9	148
BN-BO-a	6418 (10.0%)	6081	6259 (2.9%)	2.48%	65 (95.9%)	1.1	0.4	814
MU-VR-a	3505 (35.9%)	2486	3075 (23.7%)	12.27%	36 (66.6%)	3.2	0.0	644

In order to model these constraints, we can first set the minimum intervals d_i and a_{i+1} between consecutive departures from i and arrivals in $i+1$, respectively, to the maximum between the original values and

$$c_i + [\text{minimum travel time from } i \text{ to } i+1],$$

where the minimum is taken over all trains in the instance.

As is the case for the original problem (with automatic signaling), the resulting constraints (7) and (8) are not sufficient, and we have to introduce the following additional constraints, that are intended to replace (13) for all stations $i \in S\backslash\{s\}$ such that the block signaling between i and $i+1$ is manual. Actually, these constraints are given by (13) where the minimum time intervals are now defined by

$$b_i^{jk} := t_j + c_i.$$

In order to take into account the manual signaling constraints in our heuristic algorithm, after a train j has been scheduled, we consider all the line segments on which j travels and forbid the associated nodes as follows. Suppose train j departs from station i at time t_1 and arrives at station $i+1$ at time t_2. We forbid all the departure nodes from i associated with time instants between t_1 and $t_2 + c_i - 1$. Moreover, for each unscheduled train k, we forbid the segment arcs corresponding to traveling from station i to station $i+1$ between t_1 and t_2.

Clearly, with the above modification, the algorithm can handle the case in which signaling is automatic in some parts of the line and manual in others.

For the instances considered, we simulated manual block signaling between all stations along the line (assuming a failure in the automatic signaling system). The results, given in Table 6, show that the effect of manual block signaling on the line strongly depends on the congestion of the instance. Indeed, for the most congested instances, i.e., PC-BO-a, MU-VR-a and PC-BO-c, the decrease of both the number of scheduled trains and the value of the best solution is about 30%, 25% and 18%, respectively, while for the remaining instances, the decrease is about 5%.

These results can also be interpreted as the effect of block installation breakdown if the line is highly congested.

4.2. Station capacities

The basic model assumes that the capacity of each station is infinite, i.e., any number of trains could be standing at a station at a given time instant. Although this situation is clearly unrealistic, in practice, for some of the instances considered, the number of trains that are simultaneously present in a station does not exceed two for any solution that satisfies the other constraints, due to the minimum time that must elapse between consecutive departures and arrivals and to the high penalty incurred if a train stops in a station (for a time interval not less than $a_i + d_i$) to be overtaken by another train.

Nevertheless, in some other situations, especially for highly congested instances, one has to explicitly impose the capacity constraint for each station to ensure feasibility. Formally, station capacity constraints impose that, for each time instant t, there can be at most k_i trains present in each station i, taking into account also the trains that are passing through station i at time instant t without stopping. In order to model these constraints, let $T(i,t) \subseteq T$ be the set of trains that may be in station i at time t, and, for each $j \in T(i,t)$, let $A^j(i,t) \subseteq A^j$ be the set of arcs corresponding to the presence of train j in station i at time t, i.e., $(u,v) \in A^j(i,t)$ if $\theta(u) \leqslant t \leqslant \theta(v)$. The capacity constraint

Table 7
Results with station capacities

Name	# Major	Best UB	Greedy sol.	Best sol.	% Gap	# Sched.	Avg. ν	Avg. μ	Time
PC-BO-a	3	23 091 (10.3%)	16 086	17 506 (8.8%)	24.19%	155 (70.1%)	1.6	0.6	3190
PC-BO-b	3	6989 (6.2%)	5654	5751 (1.7%)	17.71%	48 (80.0%)	2.0	0.7	723
PC-BO-c	3	3876 (19.2%)	2784	3291 (18.2%)	15.09%	29 (72.5%)	2.0	0.6	161
BN-BO-a	6	6866 (3.7%)	6455	6528 (1.1%)	4.92%	67 (98.5%)	1.1	0.2	1091
MU-VR-a	8	5013 (8.3%)	3252	4055 (24.7%)	19.11%	43 (79.6%)	1.4	0.0	838

reads

$$\sum_{j \in T(i,t)} \sum_{a \in A^j(i,t)} x_a \leqslant k_i, \quad i \in S, \quad t = 1, \ldots, q. \tag{15}$$

Note that these are the first operational constraints that involve the arc variables. On the other hand, since their number is not too large, we can handle them in the solution of the Lagrangian relaxation as follows. First of all, these constraints are added to the pool of constraints with a nonnegative multiplier only when they are violated by the relaxed solution, as is the case for all other constraints. Then, when we have to find an optimal path for a train j, either within the solution of the Lagrangian relaxation or in the heuristic algorithm, the Lagrangian penalty associated with each arc $a=(u,v) \in A^j$ such that $u \in U^i$, $v \in W^i$ is given by

$$\alpha_a = \sum_{t=\theta(u)}^{\theta(v)} \lambda_{it},$$

where λ_{it} is the Lagrangian multiplier associated with (15). Accordingly, if we compute the quantities:

$$\beta_{i,t} := \sum_{p=1}^{t} \lambda_{ip}$$

(once for all after each Lagrangian multiplier updating) the penalty of arc a is obtained in constant time as $\alpha_a = \beta_{i,\theta(v)} - \beta_{i,\theta(u)-1}$ if $\theta(u) \leqslant \theta(v)$, $\alpha_a = \beta_{i,\theta(v)} + \beta_{i,q} - \beta_{i,\theta(u)-1}$ otherwise, where $\beta_{i,0} := 0$. The complexity of the optimal path computation for each train $j \in T$ remains then $\mathrm{O}(|A^j|)$ also after the addition of the station capacity constraints.

When we construct a heuristic solution, for each station i having capacity k_i, we keep a list of the time intervals in which the number of trains (scheduled so far) present in i is equal to k_i. This list contains pairwise nonoverlapping intervals of the form (t_1, t_2), and is stored by increasing values of t_1. Clearly, all the arrival and departure nodes whose time instants lie in these intervals are forbidden. Moreover, the arcs corresponding to a stop in the station which overlaps some of these intervals are forbidden as well (even if the arrival and the departure associated with the stops are feasible). To this end, in the max-profit path procedure, each time we explore an arrival node at station i, labeling all the connected departure nodes, we check whether the corresponding arc is infeasible by exploring, through binary search, the above list.

For the instances considered, we imposed a station capacity equal to 1 for the minor stations (those in which only the Local trains stop) and to 2 for the major stations along the line. The results are reported in Table 7. The number of stations having capacity equal to 2 is given in Column # Major. For the least congested instance, i.e., BN-BO-a, only one train is canceled and the best solution value decreases by less than 4%. For instance MU-VR-a, 10% more trains are canceled with respect to the case without capacities, while the value of the best solution decreases by about 4%. For the other instances, the reduction of the number of trains scheduled and of the value of the best solution is more than 15% and 9%, respectively.

These simulations can help the railway planners in identifying some infrastructural bottlenecks and considering focused investments on specific stations or parts of the line.

Table 8
Results by scheduling first the high priority trains and then the remaining trains

Name	# Trains	Ideal profit	Best UB	Greedy sol.	Best sol.	% Gap	# Sched.	Avg. ν	Avg. μ	Time
PC-BO-a	138	11656	9947 (14.7%)	8923	9237 (3.5%)	7.14%	106 (76.8%)	1.0	1.8	3547
PC-BO-b	37	3440	3318 (3.5%)	3135	3285 (4.8%)	0.99%	36 (97.3%)	1.1	1.3	113
PC-BO-c	24	2108	1831 (13.1%)	1148	1355 (18.0%)	26.00%	17 (70.8%)	1.6	2.7	76
BN-BO-a	49	4586	4543 (0.9%)	4516	4543 (0.6%)	0.00%	49 (100.0%)	0.8	0.4	7
MU-VR-a	47	3965	3645 (8.1%)	2645	3085 (16.6%)	15.36%	37 (78.7%)	1.4	1.4	612

4.3. Prescribed timetable for a subset of the trains

In some cases, the timetables of some trains are fixed and should not be changed (nor the trains be left unscheduled). For instance, one may have a published timetable for high priority trains and would like to run additional low priority trains for which only an ideal timetable (that may be changed) is specified.

This situation is handled by implementing a procedure that fixes the paths associated with the fixed timetables permanently, i.e., both in the solution of the Lagrangian relaxation and in the heuristic solution.

For the instances considered, we first solved TTP for the high priority trains only, i.e., trains of type Eurostar, Euronight, Intercity and Express (*Phase* 1). Then, for the scheduled trains, we kept fixed the paths according to the solution found and scheduled the remaining trains (*Phase* 2). The results are given in Table 8. The table gives the number of low priority trains considered in Phase 2 (# Trains), the corresponding ideal profit, and all the values reported in the previous tables, computed with respect to these trains. Note that instance BN-BO-a has been solved to proven optimality.

This kind of analysis can provide very useful information on assessing the capability of the line to accommodate more train paths, using spare capacity, especially if there is high flexibility for the timetables of the additional trains, i.e., if their shift/stretch penalty is small.

4.4. Maintenance operations

Maintenance operations along the track forbid the use of part of the track by the trains for specified periods. Typically, these operations take place overnight, according to the following schedule, illustrated by an example for clarity. Maintenance operations are called *possessions* to denote that a segment of the track must be possessed for a specified time interval by the maintenance gangs.

Let i_1, i_2, i_3, i_4 be stations that appear along the track in this order and are not necessarily consecutive, i.e., there may be other stations between i_1 and i_2 and so on. Typically, the distance between i_1 and i_2, i_2 and i_3, i_3 and i_4 may be about 25–30 km. A maintenance operation may keep the track from i_1 to i_2 occupied from 1 AM to 3 AM, the track from i_2 to i_3 from 1:30 AM to 3:30 AM, and the track from i_3 to i_4 from 2 AM to 4 AM. This means that no train can be traveling along the occupied parts of the track in the specified periods. For instance, the last train running before the maintenance period must depart from i_2 before 1 AM, whereas the first train running after the maintenance cannot depart from i_1 before 3 AM.

The maintenance is implemented by explicitly forbidding train paths from visiting nodes corresponding to an arrival at/departure from a station in a time instant at which maintenance is keeping the station occupied. Forbidding arcs in the graph in correspondence of maintenance operations is not necessary, since possessions keep each track segment occupied for a time which is much longer than the maximum feasible travel time for each train along the segment. So, it cannot happen that a train departs from a station before the maintenance period and arrives at the following station after. This model enables us to carry out studies for both scheduled and unscheduled (i.e., line breakdowns) maintenance requirements.

For each instance we denote by ψ the smallest departure time (after midnight) of a train from its first station, and forbid the trains from being between stations 1 and 3 from ψ to $\psi + 120$, between stations 3 and 6 from $\psi + 30$ to $\psi + 150$, and between stations 6 and 9 from $\psi + 60$ to $\psi + 180$. The results are given in Table 9, showing that the maintenance operations do not affect significantly the solution value and the number of scheduled trains for instances BN-BO-a and MU-VR-a. The contrary holds for instances PC-BO-b and, mainly, PC-BO-c, for which the decrease of

Table 9
Results with maintenance operations

Name	Best UB	Greedy sol.	Best sol.	% Gap	# Sched.	Avg. ν	Avg. μ	Time
PC-BO-a	22 338 (7.6%)	17751	19 631 (10.6%)	12.12%	175 (79.2%)	1.1	0.5	2411
PC-BO-b	5526 (4.5%)	4974	5417 (8.9%)	1.97%	48 (80.0%)	1.7	0.7	185
PC-BO-c	1472 (22.9%)	1247	1313 (5.3%)	10.80%	16 (40.0%)	9.8	1.2	10
BN-BO-a	6710 (3.0%)	6586	6615 (0.4%)	1.42%	67 (98.5%)	0.8	0.3	148
MU-VR-a	4818 (8.2%)	3509	4102 (16.9%)	14.86%	47 (87.0%)	1.2	1.2	231

both the solution value and the number of scheduled trains is more than 20% and 50%, respectively. This suggests that the trains canceled by the solution should not be run every day, but every day apart from the days in which maintenance operations take place (e.g., over week-ends).

We point out that maintenance requirements can impose severe constraints on the overall timetable process, and their effect on the line capacity is difficult to analyze, especially in the case of long distance lines where residual "local" capacity can be useless due to some other bottlenecks further along the line (e.g., in another country, in case of an international traffic corridor). Moreover, it is useful to analyze the effects of possessions due to contingency or extraordinary maintenance.

5. Conclusions

The algorithm presented in this paper compares to previous works in that:

- solutions can be provided within limited computing resources and running times even for long distance and heavily loaded lines (e.g. railway lines with hundreds of trains and more than 500 km long);
- both technical and economical factors which intervene in infrastructure management are considered;
- several characteristics which are taken into account in real-life planning at railway companies are included.

The model provides a track allocation plan which can be very close to the final timetable design and can be easily validated or refined by other techniques, like interactive graphical workstations and simulators, generally working at a more detailed level. However it is known that the latter are too "short sighted" to find global feasible solutions, and require a more intelligent control at a higher level of decision making.

Furthermore, the proposed algorithm allows one to easily find residual infrastructure capacities, i.e., spare paths (or slots) that can be allocated on short-notice basis, following short-term or contingency planning needs (e.g., special trains), which are particularly addressed in freight operations, where traffic demand is more variable and unpredictable, i.e., it has to follow the changing market requirements.

In addition, the model can be modified so as to support the real-time train management, providing controllers with some long range view, e.g., international corridors, about the traffic forecasting and conflict detection and resolution.

Finally, the approach is able to quantify in operational terms the definitions of the so-called "co-ordinated" and "capacity constraint" infrastructure, which are also introduced in newly proposed EU directives according to the congestion levels of the main railway lines. To sum up, the method described here can facilitate other traditional techniques of infrastructure planning, enhance the market attitude and the quality of service of the railway transportation sector, and improve international cooperation.

The described method, developed within the EU Project PARTNER, has been evaluated through extensive testing in real case studies from the Italian network, providing very encouraging results. The TCM model is now being implemented as part of a newly developed capacity allocation and timetable planning system of Rete Ferroviaria Italiana, the FS Infrastructure Management company.

The method behaves well in dealing with large-size real-world case studies as they occur in railways, and can be modified to tackle other situations than those described in the present paper. For instance, our research group plans to extend the analysis to situations involving the so-called *bidirectional block* with one track carrying traffic in both directions, whereas so far we considered the case of two tracks, one dedicated to each direction. This occurs in presence of line sections which are still single track, or when one out of two tracks cannot be used (e.g., must be subject to maintenance operations) leaving the other track working both ways. Moreover, in some cases, with the purpose of handling higher traffic flow in one direction, both tracks can be used in that direction. More complex infrastructure situations allow more than two tracks in particularly congested parts of the network, thus having more parallel tracks to manage traffic in both directions. These situations arise in particular in large railway junctions or so-called metropolitan nodes where two or more lines overlap.

References

[1] U. Brännlund, P.O. Lindberg, A. Nöu, J.E. Nilsson, Allocation of scarce track capacity using Lagrangian relaxation, Transportation Sci. 32 (1998) 358–369.
[2] X. Cai, C.J. Goh, A fast heuristic for the train scheduling problem, Comput. Oper. Res. 21 (1994) 499–510.
[3] A. Caprara, M. Fischetti, P. Toth, Modeling and solving the train timetabling problem, Oper. Res. 50 (2002) 851–861.
[4] M. Carey, D. Lockwood, A model, algorithms and strategy for train pathing, J. Oper. Res. Soc. 46 (1995) 988–1005.
[5] L.F. Escudero, M. Guignard, K. Malik, A Lagrangian relax and cut approach for the sequential ordering with precedence constraints, Ann. Oper. Res. 50 (1994) 219–237.
[6] M. Fisher, Optimal solution of vehicle routing problems using minimum *k*-trees, Oper. Res. 42 (1994) 626–642.
[7] A. Higgings, E. Kozan, L. Ferreira, Heuristic techniques for single line train scheduling, J. Heuristics 3 (1997) 43–62.
[8] D. Jovanovic, P.T. Harker, Tactical scheduling of rail operations: the SCAN I system, Transportation Sci. 25 (1991) 46–64.
[9] T. Lindner, U.T. Zimmermann, Cost-oriented train scheduling, in: Eighth International Conference on Computer Aided Scheduling of Public Transport (CASPT 2000), Berlin, June 2000.
[10] E. Oliveira, B.M. Smith, A job-shop scheduling model for the single-track railway scheduling problem, Technical Report 2000.21, School of Computing Research Report, University of Leeds, 2000.
[11] L.W.P. Peeters, Cyclic railway timetable optimization, Ph.D. Thesis, Erasmus Research Institute of Management, Erasmus University Rotterdam, 2003.
[12] L.W.P. Peeters, L.G. Kroon, A cycle based optimization model for the cyclic railway timetabling problem, in: J. Daduna, S. Voss (Eds.), Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems, vol. 505, Springer, Berlin, 2001, pp. 275–296.
[13] A. Schrijver, A. Steenbeek, Timetable construction for railned, Technical Report, CWI, Amsterdam, 1994 (in Dutch).
[14] B. Szpigel, Optimal train scheduling on a single track railway, in: M. Ross (Ed.), OR'72, North-Holland, Amsterdam, 1973, pp. 343–351.