4OR (2008) 6:125–142
DOI 10.1007/s10288-007-0037-5

REGULAR PAPER

A column generation approach to train timetabling on a corridor

Valentina Cacchiani · Alberto Caprara · Paolo Toth

Received: 15 January 2006 / Revised: 15 October 2006 / Published online: 29 March 2007

Abstract We propose heuristic and exact algorithms for the (periodic and non-periodic) train timetabling problem on a corridor that are based on the solution of the LP relaxation of an ILP formulation in which each variable corresponds to a full timetable for a train. This is in contrast with previous approaches to the same problem, which were based on ILP formulations in which each variable is associated with a departure and/or arrival of a train at a specific station in a specific time instant, whose LP relaxation is too expensive to be solved exactly. Experimental results on real-world instances of the problem show that the proposed approach is capable of producing heuristic solutions of better quality than those obtained by these previous approaches, and of solving some small-size instances to proven optimality.

Keywords Train timetabling · ILP-formulation · Column generation · Separation · Constructive heuristics · Experimental results

MSC classification 90B06 · 90C10 · 90C57

V. Cacchiani (✉) · A. Caprara · P. Toth
D.E.I.S., University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy
e-mail: vcacchiani@deis.unibo.it

A. Caprara
e-mail: acaprara@deis.unibo.it

P. Toth
e-mail: ptoth@deis.unibo.it

1 Introduction

The train timetabling problem (TTP) is a fundamental problem in railway optimization, and has received considerable attention in the optimization literature in recent years, mainly due to the ongoing reorganization and privatization of European railways, which is a strong incentive towards efficient use of resources. The general aim of TTP is to provide a timetable for a number of trains on a certain part of the railway network. According to the current situation, an Infrastructure Manager is handling the railway network and receiving requests from Train Operators, concerning trains to be operated for a given time horizon. Each of these requests specifies a path for a train along with the arrival and departure times for all stations along the path. Given that these requests are mutually incompatible, the Infrastructure Manager has to solve a TTP, modifying the arrival and/or departure times of some trains (and possibly canceling some other trains), in order to come up with a proposed feasible scenario for the Train Operators, who may either accept it (given that they will pay less for the requests that were modified), or come up with new proposals. The process is iterated until the scenario proposed by the Infrastructure Manager is accepted by all Train Operators. We refer to Borndörfer et al. (2005) for a more detailed description of this process in the German case.

Given the practical relevance of the problem and the different organization rules of the various Infrastructure Managers, several variants of the problem were formulated and solved. Although it is out of the scope of this paper to present in detail all the methods proposed in the literature to tackle these variants (we refer the interested reader to the survey by Caprara et al. 2006), they can be roughly classified according to the following criteria: (a) the method focuses on a single line or on a general network, and (b) the method takes advantage or not of the fact that the solution to be constructed is periodic, if this is the case. As to (a), the reason for focusing on a single (main) line, often called corridor, is that, in many cases, once the timetable for the trains on the corridor has been determined, it is relatively easy to find a convenient timetable for the trains on the other lines of the network. On the other hand, there are other situations in which there are a few congested lines, and solving the problem on each line separately would be a too rough approach. As to (b), in some real cases (nearly) all trains are repeated every period (typically 1 h). When this holds, not only the size of the problem is widely reduced by considering only one period, but also the mathematical formulation can be stronger since the "big M" needed to write disjunctive constraints can actually be as small as the period itself (see again, e.g., Caprara et al. 2006 for details). As a result, there are methods that take advantage of the periodicity and are not suited for cases in which the problem is non-periodic. At present, given their effectiveness, these appear to be the majority of the methods that have been tested on networks as opposed to single lines. Among the methods that consider a single line and can be used also in the non-periodic case, Szpigel (1973), Jovanovic and Harker (1991), Cai and Goh (1994), Carey and Lockwood (1995), and Higgings et al. (1997) present mixed ILP formulations in which the arrival and

departure times are represented by continuous variables and there are binary variables expressing the order of the train departures from each station, and heuristic algorithms based on these formulations. Moreover Brännlund et al. (1998) and Caprara et al. (2002), Caprara et al. (2006) illustrate pure ILP formulations obtained by discretizing time and representing the problem on a suitable graph. Due to the large sizes of the instances considered, these approaches avoid the explicit solution of the associated LP relaxation, resorting to computationally faster approaches based on Lagrangian relaxation combined with subgradient optimization.

The references dealing with approaches tested on networks and that assume periodicity build on the seminal paper on the periodic event scheduling problem by Serafini and Ukovich (1989). Schrijver and Steenbeek (1994) and Odijk (1996) consider a simple but rather weak ILP formulation in which, again, there are continuous (bounded) variables representing arrival and departure times modulo the period, and there are binary variables expressing the order of the train arrival/departures at stations. The formulation is solved by branch-and-bound. Lindner (2000), Peeters (2003), Peeters and Kroon (2001), Kroon and Peeters (2003), and Lindner and Zimmermann (2005) deal with analogous ILP formulations in which the binary variables are removed at the cost of introducing a larger and more complex set of constraints related with the notion of cycle bases. The resulting LP relaxations are much stronger, leading to much smaller solution times as discussed in Liebchen et al. (2004), again by using branch-and-bound.

In this paper, we illustrate a solution method for TTP on a single line, that can be used also when the problem is non-periodic, although the case on which we focus is periodic with period 1 day. The method uses an ILP formulation that is a natural variant of the ones presented in Caprara et al. (2002), Caprara et al. (2006), and is based on the explicit solution of the LP relaxation of this ILP formulation. The LP relaxation is used to drive both heuristic and exact branch-and-bound algorithms. The paper is organized as follows: The version of TTP we consider is defined formally in Sect. 2, and its canonical representation on a space-time graph, common to many of the approaches mentioned above, is described in Sect. 3. The new ILP formulation, with variables associated with paths (rather than nodes/arcs) in this graph, is given in Sect. 4, and the solution of its LP relaxation by separation and column generation techniques is discussed in Sect. 5. Sections 6 and 7 discuss heuristic and exact algorithms based on this LP relaxation, respectively. Finally, in Sect. 8 we report computational results on real-world instances of the problem, showing that our heuristics provide improved solutions with respect to previous approaches, whereas our exact method is capable of solving to proven optimality for the first time some of the instances in the test bed.

2 The train timetabling problem

We face the same problem as in Caprara et al. (2002), Caprara et al. (2006), therefore its description is a synthesis of the one given in those references.

We consider a single, one-way track corridor linking two major stations, with a number of intermediate stations in between together with a set of trains that are candidate to be run every day of a given time horizon along the corridor. Let $S = \{1, \ldots, s\}$ represent the set of stations, numbered according to the order in which they appear along the corridor for the running direction considered, and $T = \{1, \ldots, t\}$ denote the set of candidate trains. 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 ($j \in T$).

The *track capacity constraints* impose that overtaking between trains occurs only within a station. 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. 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 $f_j + 1, \ldots, l_j - 1$. 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, with respect to the ideal timetable, one is allowed to modify (anticipate or delay) the departure time of each train from its first station, and to increase (but not decrease) the stopping time interval at the (intermediate) stations. Moreover, one can also *cancel* the train. The timetable for train j in the final solution will be referred to as the *actual timetable*.

Observe that, differently from Caprara et al. (2002), we consider the version of the problem in which the travel time between consecutive stations must be the same in the ideal and actual timetables; see Caprara et al. (2006) for a comparison with the version in which this time can be increased as well. In this new version one can write the overtaking constraints in the ILP in a form that is much stronger for the LP relaxation than in the old version.

The *objective* is to maximize the sum of the profits of the scheduled trains, defined as follows. The *profit* achieved for each train $j \in T$ is given by $\pi_j - \alpha_j \nu_j - \gamma_j \mu_j$, where π_j is the *ideal profit*, that is the profit that is achieved if the train travels according to its ideal timetable, ν_j is the *shift*, that is the absolute difference between the departure times from station f_j in the ideal and actual timetables, μ_j is the *stretch*, that is the (nonnegative) difference between the travel time from f_j to l_j in the actual and ideal timetables (equal to the sum of the stopping time increases over all intermediate stations), and α_j, γ_j are given nonnegative parameters.

For convenience, we will use the acronym TTP to denote the specific problem defined above. It was shown in Caprara et al. (2002) that TTP is NP-hard in the strong sense, being a generalization of the well-known maximum stable set problem.

Note that it would be easy to impose in the model additional constraints on the train timetables, such as the requirement that a train does not arrive at a station later than a certain time instant, to guarantee a connection with a train on a different line whose timetable is fixed. However, we did not consider these constraints in our case study.

3 A graph representation

We next outline the representation of the problem on a graph described in Caprara et al. (2002), Caprara et al. (2006). Times are here discretized and expressed as integers from 1 to $q := 1,440$ (the number of minutes in a day), though a finer discretization would also be possible (e.g., 1/2, 1/4 min) without changing the model, although the time and space complexity of the associated algorithms could increase considerably.

Let $G = (V, A)$ be the (directed, acyclic) space-time multigraph in which nodes represent arrivals/departures at a station in given time instants, and paths represent feasible timetables for trains, defined as follows (an illustrative picture taken from Caprara et al. (2002) is given in Fig. 1). 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, $i \in S \setminus \{1\}$, and W^i, $i \in S \setminus \{s\}$, 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 $\theta(v) \geq \theta(u)$, $\Delta(u, v) := \theta(v) - \theta(u) + q$ otherwise.

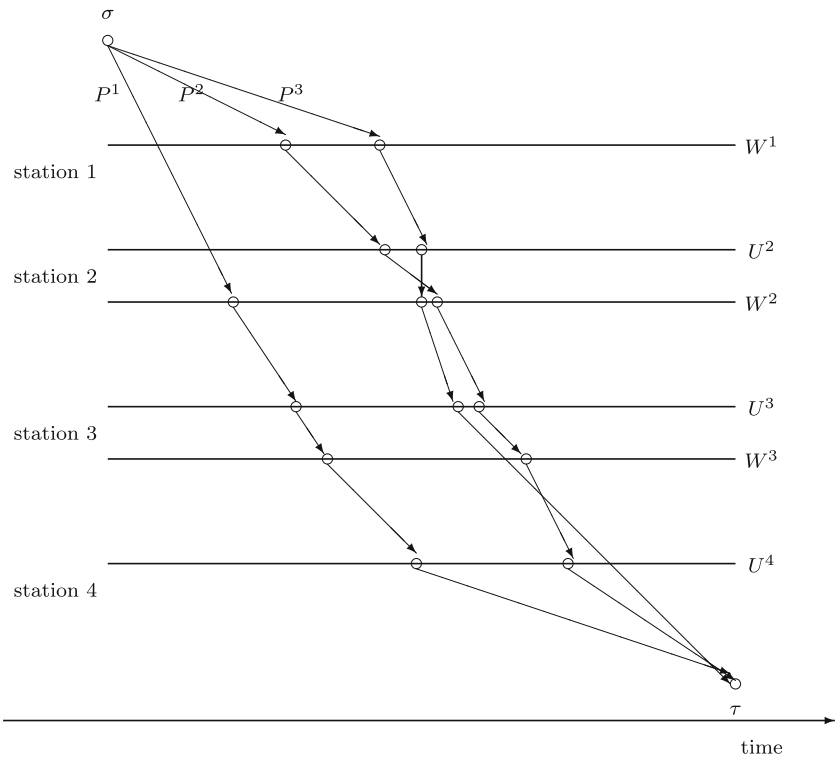

Fig. 1 An example of train paths in graph G (with $s = 4, t = 3, f_1 = 2, l_1 = 4, f_2 = 1, l_2 = 4, f_3 = 1, l_3 = 3$)

Because of the "cyclic" nature of the time horizon, we say that node u *precedes* node v (i.e., $u \preceq v$) if $\Delta(v,u) \geq \Delta(u,v)$ (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 $i \in S^j$. Accordingly, 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 set of nodes associated with time instants corresponding to possible arrivals/departures of train j in a positive-profit timetable (see Caprara et al. 2006 for a detailed definition of V^j).

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 $p_{(\sigma,v)} := \pi_j - \alpha_j \nu(v)$, with $\nu(v) := \min\{\Delta(v^*,v), \Delta(v,v^*)\}$, where v^* is the node associated with the departure of train j from station i in the ideal timetable;
- 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 $p_{(u,v)} := -\gamma_j \mu(u,v)$, with $\mu(u,v) := \Delta(u,v) - \Delta(u^*,v^*)$, where u^* and v^* are the nodes associated, respectively, with the arrival and departure of train j at station i in the ideal timetable;
- 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 equal to the travel time of train j from station i to station $i+1$, whose profit is $p_{(v,u)} := 0$;
- a set of *ending* arcs (u,τ), for each $u \in U^{l_j} \cap V^j$, whose profit is $p_{(u,\tau)} := 0$.

By construction, for each train $j \in T$, any 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.

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. This is discussed in detail in the next section.

4 An ILP model with path variables

The solution approaches in the literature that are based on the graph representation of the previous section define ILP formulations in which variables are associated with nodes and/or arcs of the graphs (see Brännlund et al. 1998, Caprara et al. 2002). As the associated LP relaxations turn out to be extremely expensive to solve exactly, even by the state-of-the-art LP solvers, a heuristic solution of their dual is obtained by combining Lagrangian relaxation of the track capacity constraints with subgradient optimization.

In this section we illustrate an alternative solution approach, based on an alternative (and equally natural) ILP formulation in which variables are associated with paths in G.

In the new ILP model there is a binary variable for each possible path for a train, indicating if the path is chosen or not in the solution. Considering that for each train $j \in T$ the corresponding path can be shifted and/or stretched with respect to the *ideal path*, i.e., the path associated with the ideal timetable

of train j, there is an exponentially large number of possible paths for a train, as will be discussed next. Therefore, our solution approach is based on column generation techniques, that are easy to apply as the column generation problem calls for an optimal path in the (acyclic) graph considered. Moreover, we will have a very large number of constraints, as was the case for the previous ILP formulation. These constraints will be handled by separation algorithms, which in this case have to deal with fractional solutions, differently from the case in which Lagrangian relaxation is used.

Recall the definitions of Sect. 3. Moreover, given two consecutive stations i and $i+1$ along with two trains j and k such that $i, i+1 \in S^j \cap S^k$, let

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

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 r_j and r_k are the travel times of j and k from i to $i+1$, respectively. Note that if j and k depart from i within a time interval smaller 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 $i+1$.

For each $j \in T$, let $\mathcal{P}^j$ be the collection of possible paths for train j, each associated with a path from σ to τ in G containing only arcs in A^j (paths are seen as arc subsets) and having positive profit. Furthermore, let $\mathcal{P} := \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^t$ be the overall (multi-)collection of paths, denoting by $p_P := \sum_{a \in P} p_a$ the actual profit for path $P \in \mathcal{P}$. Finally, let $\mathcal{P}_w^j \subseteq \mathcal{P}^j$ be the (possibly empty) subcollection of paths for train j that visit node $w \in V^j$, and $\mathcal{P}_w := \mathcal{P}_w^1 \cup \cdots \cup \mathcal{P}_w^t$ be the subcollection of paths that visit node $w \in V$.

Letting x_P, $P \in \mathcal{P}$, be a binary variable that is equal to 1 if and only if the path $P \in \mathcal{P}$ is selected in an optimal solution, the ILP model is the following:

$$\max \sum_{P \in \mathcal{P}} p_P x_P \tag{1}$$

subject to

$$\sum_{P \in \mathcal{P}^j} x_P \le 1, \qquad j \in T, \tag{2}$$

$$\sum_{w \in U^i : w \succeq u, \Delta(u,w) < a_i} \; \sum_{P \in \mathcal{P}_w} x_P \le 1, \quad i \in S \setminus \{1\}, u \in U^i, \tag{3}$$

$$\sum_{w \in W^i : w \succeq v, \Delta(v,w) < d_i} \; \sum_{P \in \mathcal{P}_w} x_P \le 1, \quad i \in S \setminus \{s\}, v \in W^i, \tag{4}$$

$$\sum_{w \in W^i \cap V^j : v_1 \preceq w \prec v_2} \; \sum_{P \in \mathcal{P}_w^j} x_P + \sum_{w \in W^i \cap V^j : w \succeq v_2, \Delta(v_2,w) < b_i^{kj}} \; \sum_{P \in \mathcal{P}_w^j} x_P + \sum_{w \in W^i \cap V^k : w \succeq v_2, \Delta(v_1,w) < b_i^{jk}} \; \sum_{P \in \mathcal{P}_w^k} x_P \le 1,$$

$$i \in S \setminus \{s\}, j, k \in T \text{ (with } j \neq k; i, i+1 \in S^j \cap S^k),$$
$$v_1, v_2 \in W^i \text{ (with } v_1 \preceq v_2, \Delta(v_1, v_2) < b_i^{jk}), \tag{5}$$
$$x_P \geq 0, \quad P \in \mathcal{P}, \tag{6}$$
$$x_P \text{ integer}, \quad P \in \mathcal{P}. \tag{7}$$

The interpretation of the objective function (1) and of constraints (2), that impose that at most one path for each train is selected, poses no problem. The *arrival time* constraints (3) and the *departure time* constraints (4), which have analogous counterparts in Caprara et al. (2002), Caprara et al. (2006), prevent two consecutive arrivals and departures at the same station i to be too close in time. The *overtaking* constraints (5), which are the immediate counterpart of constraints (13) in Caprara et al. (2006), are formally more complex. These constraints are defined by specifying a station $i \in S \setminus \{s\}$, two trains $j, k \in T$ such that $i, i+1 \in S^j \cap S^k$, and two nodes $v_1, v_2 \in W^i$ such that $v_1 \preceq v_2$ and $\Delta(v_1, v_2) < b_i^{jk}$. By definition of b_i^{jk}, we must have $x_{P_1} + x_{P_2} \leq 1$ for all $P_1 \in \mathcal{P}_{v_1}^j$ and $P_2 \in \mathcal{P}_{v_2}^k$, which would be the weakest possible form of overtaking constraints. A very simple strengthening would be $\sum_{P \in \mathcal{P}_{v_1}^j} x_P + \sum_{P \in \mathcal{P}_{v_2}^k} x_P \leq 1$. By proceeding further, taking also into account the arrival and departure constraints, one gets to (5), which are maximal in the sense that no other variable x_P for $P \in \mathcal{P}^j \cup \mathcal{P}^k$ could be added to the left-hand side maintaining validity. Specifically, nodes v_1 and v_2 represent the earliest departure times from i for trains j and k, respectively, involved in (5). Stated in words, constraints (5) 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. Finally, note that, although the ILP model would be valid also without constraints (3) and (4), as arrival/departures too close in time are forbidden also by (5), the former lead to a stronger LP relaxation and are also much faster to separate in practice, so it is much better to keep them in the formulation.

Note that the ILP model above is of *set packing type*, and is associated with the stable set problem defined on the graph with one node for each path $P \in \mathcal{P}$, with associated profit p_P, and one edge joining each pair of nodes corresponding to paths that cannot be both selected in the solution, i.e., that have both coefficient 1 in at least one of the inequalities.

5 Solution of the LP relaxation

As already mentioned, the number of variables in the new ILP formulation can be very large. Rough bounds on the number of paths associated with each train $j \in T$ are the following, letting $\nu_j^{\max} := \max\{\nu : \pi_j - \alpha_j \nu > 0\}$ 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. It is not difficult to see that, once the shift for a path of train j is fixed, the number of paths that accumulate a total stretch not larger than $\mu_j^{\max}$ in the $m_j := l_j - f_j - 1$ intermediate stations for train j is equal to the number of vectors with m_j integer nonnegative components whose sum is at most $\mu_j^{\max}$, which is equal to $\binom{\mu_j^{\max}+m_j}{m_j}$. Accordingly, we have that $|\mathcal{P}^j|$ is at least equal to the number of paths with zero shift and stretch not larger than $\mu_j^{\max}$, i.e., $|\mathcal{P}^j| \geq \binom{\mu_j^{\max}+m_j}{m_j}$, and at most equal to the number of paths with shift not larger than $\nu_j^{\max}$ and stretch not larger than $\mu_j^{\max}$, i.e., $|\mathcal{P}^j| \leq (2\nu_j^{\max}+1)\binom{\mu_j^{\max}+m_j}{m_j}$. In any case, $|\mathcal{P}^j|$ may be exponential in m_j, and for the instances considered in Sect. 8 this makes it absolutely impractical to consider explicitly all paths in $\mathcal{P}^j$.

The natural approach to tackle such a large number of variables is to use column generation techniques for the solution of the LP relaxation, working with an LP with only a subset of the variables (at the beginning, e.g., only the variables associated with the ideal path for each train), adding variables with positive reduced profit to the LP at each iteration. This is illustrated in detail in Sect. 5.1.

Although not exponential, also the number of constraints in the new ILP formulation is fairly large. Specifically, we have $O(|V|)$ arrival and departure constraints (3) and (4) and $O(t^2|V|b^{\max})$ overtaking constraints (5), where $b^{\max}$ is the maximum value of b_i^{jk} over all stations i and train pairs j, k. Accordingly, rather than imposing all these constraints since the beginning, it is much better to handle them by using separation techniques, as illustrated in Sect. 5.2.

Note that the column generation problem simply amounts to the computation of an optimal path in an acyclic graph, and the effect of adding new constraints with nonnegative dual variables simply corresponds to changing the "penalties" of some nodes in this path computation. Accordingly, the combination of separation and column generation poses no serious problem in our case (i.e., the addition of new constraints does not destroy the structure of the column generation problem, as opposed to what is frequently the case, see, e.g., Barnhart et al. 1998).

The general structure of our method to solve the LP relaxation is the following:

1. We initialize a reduced LP with only the t variables associated with the ideal paths for each train, and constraints (2) and (6);
2. We solve the reduced LP, obtaining the primal solution x^* and the dual solution y^*;
3. We apply column generation: if variables with positive reduced profit with respect to y^* are found, we add them to the reduced LP and go to Step 2.;
4. We apply separation for constraints (3) and (4): if constraints violated by x^* are found, we add them to the reduced LP and go to Step 2.;
5. We apply separation for constraints (5): if constraints violated by x^* are found, we add them to the reduced LP and go to Step 2.;

6. We terminate since x^*, y^* is an optimal primal-dual pair for the whole LP (1)–(6).

The reason for separating constraints (3) and (4) before constraints (5) is that the former are stronger as well as faster to check for violation. On the other hand, the choice to perform column generation before separation was motivated by experimental evidence, showing that convergence is faster if we start separation only when the current primal solution is optimal with respect to the constraints in the reduced LP. Of course, in the first iteration there are no variables with positive reduced profit, and separation amounts to checking feasibility of the solution defined by the ideal timetables.

5.1 Column generation

The column generation problem is solved by considering in turn each train $j \in T$ and checking if there exists a path $P \in \mathcal{P}^j$ with positive reduced profit.

Let the dual variables associated with constraints (2), (3), (4), and (5) be γ_j $(j \in T)$, α_u $(i \in S \setminus \{1\}, u \in U^i)$, β_v $(i \in S \setminus \{s\}, v \in V^i)$, and δ_{j,k,v_1,v_2} $(i \in S \setminus \{s\}, j, k \in T, v_1, v_2 \in W^i$ satisfying the requirements in (5)). Moreover, for $i \in S \setminus \{1\}$ and $w \in U^i$, let $\zeta_w := \sum_{u \in U^i : u \preceq w, \Delta(u,w) < a_i} \alpha_u$, and, similarly, for $i \in S \setminus \{s\}$ and $w \in W^i$, let $\eta_w := \sum_{v \in W^i : v \preceq w, \Delta(v,w) < d_i} \beta_v$. Finally, for $\ell \in T$, $i \in S \setminus \{l_\ell, \ldots, s\}$ and $w \in W^i$, let $\xi_{\ell w}$ be the sum of variables δ_{j,k,v_1,v_2} over all constraints for which ℓ is either j or k and node w is one of the nodes in the summations for train ℓ in (5) (i.e., the first two summations if $\ell = j$ and the third one if $\ell = k$).

For a path $P \in \mathcal{P}^j$, let U_P and W_P be, respectively, the set of arrival and departure nodes visited by path P. The reduced profit of path P is given by

$$p_P - \gamma_j - \sum_{u \in U_P} \zeta_u - \sum_{v \in W_P} (\eta_v + \xi_{jv}).$$

Accordingly, one may find the path $P \in \mathcal{P}^j$ with maximum reduced profit by finding, in $O(|A^j|)$ time, the path of maximum profit from σ to τ in G that uses only arcs in A^j, proceeding in a completely analogous way as in the solution of the Lagrangian relaxation in Caprara et al. (2002), where the profit of a path $P \in \mathcal{P}^j$ is now given by

$$\sum_{a \in P} p_a - \sum_{u \in U_P} \zeta_u - \sum_{v \in W_P} (\eta_v + \xi_{jv}).$$

(Note that both in Caprara et al. (2002) and here profits are associated both with nodes and arcs of G, and they can be handled in a way completely analogous to the case of arc profits only.)

If the maximum profit is not larger than γ_j, then all variables x_P, $P \in \mathcal{P}^j$, have nonpositive reduced profit. Otherwise, the path found is associated with the variable with the most positive reduced profit.

5.2 Separation

Separation of constraints (3), (4) and (5) is done by trivial enumeration, as it is the case within the relax-and-cut procedure associated with Lagrangian relaxation, with the difference that we have to deal with fractional variable values in this case.

Specifically, for constraints (3) and (4), for each node $v \in V$ we initialize a value y^*_w to 0. Then, we consider in turn each positive variable x^*_P in the reduced LP solution and increase the value y^*_w of all nodes $w \in U_P \cup W_P$ by x^*_P. Finally, for each station $i \in S \setminus \{1\}$ and node $u \in U^i$, we test if $\sum_{w \in U^i : w \succeq u, \Delta(u,w) < a_i} y^*_w > 1$, in which case constraint (3) associated with node u is violated. The computation of the node values can be carried out in $O(sn)$ time, where n is the number of positive variables in the reduced LP solution, and the subsequent check is easily done in $O(|V|)$ time by using accumulators to store, for each node $u \in U^i$, the sum of the values of consecutive nodes (i.e., of nodes associated with consecutive time instants) in U^i, starting from an (arbitrarily chosen) initial node and ending in u. The check for constraints (4) is perfectly analogous.

For constraints (5), for each train $j \in T$ and for each node $v \in V^j$ we first initialize a value z^*_{jv} to 0, and consider in turn each positive variable x^*_P in the reduced LP solution for $P \in \mathcal{P}^j$, increasing the value of all nodes in W_P by x^*_P. Then, we enumerate all pairs of trains $j, k \in T$ whose paths may be in conflict (by determining, for each train $j \in T$ and once for all, the set of trains whose paths may be in conflict with a path of j), and all stations $i \in \{f_j, \ldots, l_j - 1\} \cap \{f_k, \ldots, l_k - 1\}$. For each pair of nodes $v_1, v_2 \in W^i$, $v_1 \preceq v_2, \Delta(v_1, v_2) < b_i^{jk}$ we have that constraint (5) corresponding to j, k, v_1, v_2 is violated if and only if

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

The time complexity of the above procedure to separate constraints (5) is again $O(sn)$ for the computation of the values, and then $O(t^2|V|b^{\max})$ for the subsequent check, since, by using accumulators similar to those used for the separation of constraints (3) and (4), each constraint can be checked in constant time.

6 Heuristic algorithms based on the LP relaxation

In this section we present some constructive and local search heuristics, based on the optimal solution of the LP relaxation of the ILP model (1)–(7). We present some variants all of which have been implemented and tested.

6.1 Constructive heuristics

The construction of heuristic solutions from the LP relaxation is done following two main principles: the first one is to fix to 1 or 0 some variables x_P associated with train paths, the second one is to construct the path of each train step-by-step, as it is done in the enumerative algorithm of the previous section.

In all cases, we fix some part of the LP solution and then solve again the LP relaxation taking into account what was fixed (using dual simplex).

After extensive testing of various possibilities on our test-bed instances, we decided to apply only the following two path-fixing policies:

- fix to 1 all the variables x_P whose value x^*_P is 1, and additionally the variable x_P with the highest fractional value x^*_P;
- fix to 1 the variable x_P with the highest fractional value x^*_P.

6.2 Local search

We try to improve the solutions constructed by the heuristics of the previous section by using the following local search procedures.

The first procedure, which is analogous to the one used in the Lagrangian heuristic proposed in Caprara et al. (2002), considers, in increasing order of train index, each train j whose associated path has positive shift and/or stretch. For each j, the procedure removes the associated path from the solution and finds the optimal path for j with the paths of the other trains fixed (as usual, by computing an optimal path in the acyclic graph G). Once all trains have been considered, the trains that are currently canceled are considered, again in increasing order of train index, and the optimal path (if any) with the other train paths fixed is found. If, after having computed the optimal path for all these trains, the solution has been improved, the procedure is re-applied.

The second procedure is similar, with the difference that, each time a train j with shifted and/or stretched path is considered, we define the ILP model (1)–(7) for the problem in which all the paths of the other trains in the solution are fixed, and all the variables associated with paths for train j *as well as* with those of all the canceled trains are present. This ILP is solved heuristically by the constructive methods described in Sect. 6.1, and the best solution is taken.

A straightforward generalization of the second procedure is to consider, rather than a single train with shifted/stretched path, k trains, all of which are scheduled in the solution and at least one of which is shifted/stretched, and heuristically solve the ILP in which the paths for all the other scheduled trains are fixed. In order to limit the computational effort, we consider only the cases $k = 2$ and $k = 3$ ($k = 1$ being covered by the procedure above). Moreover, we consider only sets of k trains such that, by removing each of the k associated paths, at least one of the paths for the remaining trains can be improved. For each of the k paths, this is checked by removing the path from the solution and, for each remaining train j, recomputing the optimal path if all the other paths in the solution are fixed.

7 An exact branch-and-cut-and-price algorithm

In this section we illustrate an exact enumerative algorithm based on the ILP formulation proposed in Sect. 4, capable of finding the provably optimal solution for some (small-size) real-world instances. We use a canonical branch-and-bound approach in which upper bounds are computed by solving the LP relaxation (amended by the branching constraints) by means of column generation and separation. Accordingly, our method falls within the category of the so-called branch-and-cut-and-price algorithms.

Having specified how we solve the LP relaxation in Sect. 5, the other main issue of the method is the way in which branching is performed.

Our branching method amounts to specifying, for some station, which departure node must be visited by the path of a train—as the travel times between consecutive stations are fixed, as explained in Sect. 2, the specification of all departure nodes for a train uniquely identifies its path in the solution (if any, i.e., if the train is not canceled).

In order to decide the train on which to branch, starting from the optimal LP solution x^*, we proceed as follows: for each train $j \in T$, let $\rho_j := \sum_{P \in \mathcal{P}^j} x^*_P$ and $\omega_j := \max_{P \in \mathcal{P}^j} x^*_P$ be, respectively, the sum and the maximum of the values taken by the variables associated with the train in the LP solution. We branch on the train j for which ρ_j/ω_j is maximum. The reason for this choice is that a train with a large value of the ratio is a train for which the sum of the associated variables is large but, at the same time, many of these variables are fractional. Then, branching is aimed at trying to limit the number of paths whose variables can have positive value for the train, hopefully improving the upper bound.

After having decided the train j, we choose a station $i \in S^j$ and then branch by imposing that the path for train j visits a specific departure node $v \in W^i \cap V^j$, by setting $x_P = 0$ for all paths P that do not visit that node. Moreover, we explore, in this order, the subproblems associated with node $u \in W^i$ such that (1) $\Delta(u, v) = 0$ (i.e., $u = v$), (2) $\Delta(u, v) = 1$, (3) $\Delta(v, u) = 1$, (4) $\Delta(u, v) = 2$, (5) $\Delta(v, u) = 2$, and so on, excluding of course the cases in which $u \notin V^j$ as well as those in which forcing the path for train j to visit node u would lead to a violation of the track capacity constraints along with the constraints already imposed by branching. When we perform the first branching concerning train j, we also consider the possibility to have no path for the train, i.e., to cancel it, exploring this subproblem as the last one among those generated by the branching.

The choice of i and v works as follows. If there is no node already fixed for train j, we let $i := f_j$ and v be such that $\sum_{P \in \mathcal{P}^j_v} x^*_P$ is maximum, recalling that P^j_v is the set of paths for train j that visit node v. Otherwise, i.e., if there are nodes already fixed for train j, we consider the path $P \in P^j$ such that x^*_P is maximum. We let v be the node in V^j such that $(\eta_v + \xi_{jv})$ is maximum and i be the station associated with v, recalling that $(\eta_v + \xi_{jv})$ is the "penalty" associated with node v, as defined in Sect. 5.1. The reason for this choice is to try to identify a "good"

path for train j, and at the same time a node with a high "penalty" in order to have some impact on the LP solution.

The decision tree associated with the subproblems generated by branching is explored according to a depth-first policy, in order to limit the storage requirement of the method. Before branching at the root subproblem, we apply the heuristic algorithms of the previous section in order to have a valid (and hopefully tight) lower bound on the optimum to use in the bounding part.

Note that, differently from most algorithms in which column generation has to be performed for each subproblem, in our case the branching conditions are easily dealt within the column generation procedure. Indeed, in the associated optimal path computation for train j, the visit of departure nodes that have been already fixed is forced by skipping the exploration of nodes that are incompatible with these departure nodes.

8 Experimental results

The algorithms proposed in Sects. 5 to 6 were implemented in C and tested on a set of real-world instances from Rete Ferroviaria Italiana (the Italian railway infrastructure management company, RFI for short), a company of Ferrovie dello Stato (FS holding group).

8.1 The real-world instances

For each station i, the lower bounds a_i and d_i on the time intervals between two consecutive arrivals and departures of trains are set to 4 and 2 min, respectively, according to the current operational rules. 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 j is scheduled according to its ideal timetable is $\pi_j = 200$, and there is a penalty $\alpha_j = 7$ for each minute of shift as well as a penalty $\gamma_j = 10$ for each minute of stretch. Note that the shift is penalized less than the stretch for all train types.

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-1	Piacenza	Bologna	17	40 (6,0,10,0,0,12,2,10)	4,800
MU-VR	Munich	Verona	48	54 (0,0,0,7,0,0,47,0)	5,470
BZ-VR	Bolzano	Verona	27	128 (34,0,0,10,1,11,38,34)	16,300
PC-BO-2	Piacenza	Bologna	17	93 (12,3,13,10,5,20,6,24)	11,010
PC-BO-3	Piacenza	Bologna	17	60 (12,1,10,0,4,12,7,14)	7,450
PC-BO-4	Piacenza	Bologna	17	221 (28,3,17,35,35,28,14,61)	25,740
CH-RM	Chiasso	Rome	102	41 (17,0,11,0,0,0,7,6)	6,020
BN-BO	Brennero	Bologna	48	68 (1,0,5,13,0,11,38,0)	7,130
CH-MI	Chiasso	Milan	16	194 (20,1,20,8,9,19,66,51)	21,930
MO-MI-1	Modane	Milan	54	16 (1,0,3,0,0,4,5,3)	1,760
MO-MI-2	Modane	Milan	54	100 (19,0,26,0,0,13,22,20)	12,420

The main characteristics of the instances are outlined in Table 2. Column # *Stations* gives the total number of stations along the considered corridor. 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 Caprara et al. (2002). In particular, the original train type "Intercity" has been split into train types "Intercity" and "Combined", with different profit coefficients. In addition, few intermediate stations in which no overtaking can occur are now explicitly considered.

Most of these instances contain a small portion of the trains that are daily run along the associated lines, considering a limited time window within the day, and in some cases only some of the various train types.

8.2 The results

In this section we present a comparison of the upper bound and heuristic solution values obtained by the method of Caprara et al. (2006) and by the methods discussed in this paper (Sects. 5 and 6) as well as the results of the exact algorithm (Sect. 7) on the instances that it was able to solve. All times reported are expressed in CPU seconds on a PC Pentium IV 2.4 GHz with 512 MB RAM. We solved the LPs using CPLEX 9.0.

The Lagrangian method proposed in Caprara et al. (2006) is an improvement over the one in Caprara et al. (2002). The results reported in Caprara et al. (2006) were obtained by running the code on a different architecture (with a different floating point arithmetic). Accordingly, rather than reporting those result with a rough time conversion, we decided to run the code from Caprara et al. (2006) on the machine mentioned above. This explains why the results given here are slightly different.

As far as the heuristic solution values are concerned, among the many possible combinations of the options discussed in Sect. 6, we report the results for the following two heuristics. Heuristic H1 first constructs a solution by using the first fixing rule described in Sect. 6.1, and then applies the three local search procedures described in Sect. 6.2, in the order in which they were described. Heuristic H2 first constructs a solution by using the second fixing rule described in Sect. 6.1, and then applies only the first two local search procedures described in Sect. 6.2, again in that order, these two procedures being much faster than the third one.

Table 3 reports the results on the instances illustrated above. The columns in the table have the following meaning:

- *Lagr. UB* is the Lagrangian upper bound obtained by the method in Caprara et al. (2006).
- *Lagr. heur* is the value of the best heuristic solution found by the method in Caprara et al. (2006).
- *Gap%* is the percentage gap between the best upper bound and the value of the best heuristic solution found by the method in Caprara et al. (2006).
- *LP UB* is the optimal value of the LP relaxation discussed in this paper.
- *LP H1* and *LP H2* are the solutions found by heuristics H1 and H2, respectively, illustrated above.
- *Gap%* is the percentage gap between the best upper bound and the value of the best solution found by H1 and H2.

Table 3 shows that, on the one hand, the LP upper bound, which dominates the Lagrangian upper bound, is often significantly stronger and can be computed within much smaller computing times. The solutions found by heuristic H2 are of quality comparable to those found by the Lagrangian heuristic and require, on average, a similar computing time, although the time taken by the former has a much higher variance. On the other hand, in almost all cases, the solutions found by heuristic H1 are better than those found by the Lagrangian heuristic, but the associated computing time is much higher.

The exact algorithm described in Sect. 7, which initially applies heuristic H1, could solve 3 of the 11 instances within a time limit of 100,000 s (around 28 h). It does not appear that, by increasing the time limit by, say, a factor 2 or 3, the method could solve to proven optimality other instances among those considered. The associated optimal value, computing time, and number of subproblems explored are reported in Table 4. This is the first time that some instances in this test bed are solved to proven optimality.

9 Conclusions

In previous (unpublished) experiments we observed that the solution of the LP relaxation of the ILP models in Caprara et al. (2002), Caprara et al. (2006) is extremely cumbersome in practice. This paper shows that an equivalent LP relaxation in which the number of variables is exponentially large can be solved within a computing time that is not only reasonable (as one might have expected) but

Table 3 Comparison of the results obtained with the Lagrangian heuristic of Caprara et al. (2006) and the LP-based heuristic of the present paper

Name	Lagr. UB		Lagr. heur		Gap%	LP UB		LP H1		LP H2		Gap%
	Value	Time (s)	Value	Time (s)		Value	Time (s)	Value	Time (s)	Value	Time (s)	
PC-BO-1	4,314	65	3,629	54	15.9	4,091	14	3,776	5,836	3,725	137	7.7
MU-VR	5,032	128	4,211	97	16.3	4,894	19	4,253	5,210	3,968	42	13.1
BZ-VR	16,152	259	15,994	260	0.98	16,102	11	15,977	2,297	15,997	12	0.65
PC-BO-2	10,953	75	10,861	78	0.84	10,914	6	10,882	504	10,727	3	0.29
PC-BO-3	7,235	73	7,135	73	1.38	7,200	4	7,153	644	7,138	6	0.65
PC-BO-4	24,243	616	21,425	523	11.6	23,894	761	22,041	43,200	21,753	6,029	7.7
CH-RM	5,850	392	5,567	383	4.8	5,823	26	5,574	5,530	5,407	73	4.3
BN-BO	6,909	126	6,774	123	1.95	6,852	6	6,716	1,811	6,649	5	1.98
CH-MI	21,259	232	20,816	232	2.08	21,131	15	21,022	13,653	20,919	31	0.5
MO-MI-1	1,727	17	1,684	15	2.49	1,708	2	1,684	36	1,634	3	1.4
MO-MI-2	11,336	415	9,318	354	17.8	11,185	288	9,453	48,051	9,498	1,239	15.08

Table 4 Results with the branch-and-cut-and-price algorithm

Name	Optimal value	Time (s)	# Subproblems
PC-BO-2	10,882	683	6,611
PC-BO-3	7,161	77,562	699,687
MO-MI-1	1,684	44	680

also significantly smaller than the time needed to compute good Lagrangian bounds by the approaches proposed in Caprara et al. (2002), Caprara et al. (2006), yielding notably better bounds. On the other hand, although the solution values returned by the heuristic method based on the new LP relaxation tend to be better than those found by the Lagrangian heuristics of Caprara et al. (2002), Caprara et al. (2006), the corresponding solution times are much larger, which seems to indicate that the Lagrangian approach is preferable for practical purposes.

Future work will be concerned with facing real-world instances that contain (most of) the trains that should be run along the considered corridor on a day. Moreover, we will test our approach on instances associated with a network rather than a corridor, comparing it with the other approaches that are suited for this case.

Acknowledgment We are grateful to the anonymous referees for their helpful comments.

References

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: column generation for solving huge integer programs. Oper Res 46:316–329

Borndörfer R, Grötschel M, Lukac S, Mitusch K, Schlechte T, Schultz S, Tanner A (2005) An auctioning approach to railway slot allocation. ZIB Technical Report ZR-05-45

Brännlund U, Lindberg PO, Nöu A, Nilsson JE (1998) Allocation of scarce track capacity using Lagrangian relaxation. Transp Sci 32:358–369
Cai X, Goh CJ (1994) A fast heuristic for the train scheduling problem. Comput Oper Res 21:499–510
Caprara A, Fischetti M, Toth P (2002) Modeling and solving the train timetabling problem. Oper Res 50:851–861
Caprara A, Kroon L, Monaci M, Peeters M, Toth P (2006) Passenger railway optimization. In: Barnhart C, Laporte G (eds) Handbooks in operations research and management science, vol 14. Elsevier, Amsterdam, pp 129–187
Caprara A, Monaci M, Toth P, Guida PL (2006) A Lagrangian heuristic approach to real-world train timetabling problems. Disc Appl Math 154:738–753
Carey M, Lockwood D (1995) A model, algorithms and strategy for train pathing. J Oper Res Soc 46:988–1005
Higgings A, Kozan E, Ferreira L (1997) Heuristic techniques for single line train scheduling. J Heuristics 3:43–62
Jovanovic D, Harker PT (1991) Tactical scheduling of rail operations: the SCAN I system. Transp Sci 25:46–64
Kroon LG, Peeters LWP (2003) A variable trip time model for cyclic railway timetabling. Transp Sci 37:198–212
Liebchen C, Proksch M, Wagner FH (2004) Performance of algorithms for periodic timetable optimization. TU Preprint 021/2004, TU Berlin Computer-aided transit scheduling. Lecture notes in economics and mathematical systems. Springer, Berlin (to appear)
Lindner T (2000) Train schedule optimization in public rail transport. Ph.D. Thesis, University of Technology, Braunschweig
Lindner T, Zimmermann UT (2005) Cost optimal periodic train scheduling. Math Methods Oper Res 62:281–295
Odijk M (1996) A constraint generation algorithm for the construction of periodic railway timetables. Transp Res 30:455–464
Peeters LWP (2003) Cyclic railway timetable optimization. Ph.D. Thesis, Erasmus Research Institute of Management, Erasmus University, Rotterdam
Peeters LWP, Kroon LG (2001) A cycle based optimization model for the cyclic railway timetabling problem. In: Daduna J, Voss S (eds) Computer-aided transit scheduling, lecture notes in economics and mathematical systems, vol 505. Springer, Berlin, pp 275–296
Serafini P, Ukovich W (1989) A mathematical model for periodic event scheduling problems. SIAM J Disc Math 2:550–581
Schrijver A, Steenbeek A (1994) Timetable construction for railned. Technical Report, CWI, Amsterdam (in Dutch)
Szpigel B (1973) Optimal train scheduling on a single track railway. In: Ross M (ed) OR'72. North-Holland, Amsterdam, pp 343–351