

# Solution of the Train Platforming Problem

Alberto Caprara, Laura Galli, and Paolo Toth

DEIS, University of Bologna  
Viale Risorgimento 2, 40136 Bologna, Italy  
{alberto.caprara,l.galli,paolo.toth}@unibo.it

**Abstract.** In this paper we study a general formulation of the train platforming problem, which contains as special cases all the versions previously considered in the literature as well as a case study from the Italian Infrastructure manager that we addressed. In particular, motivated by our case study, we consider a general quadratic objective function, and propose a new way to linearize it by using a small number of new variables along with a set of constraints that can be separated efficiently by solving an appropriate linear program. The resulting integer linear programming formulation has a continuous relaxation that leads to strong bounds on the optimal value. For the instances in our case study, we show that a simple diving heuristic based on this relaxation produces solutions that are much better than those produced by a simple heuristic currently in use, and that often turn out to be (nearly-)optimal.

## 1 Introduction

The objective of train platforming, which is the routing problem that generally follows any timetabling phase, is to find an assignment of trains to platforms in a railway station. The practical relevance of the problem inspired the definition of a few different versions, which are relatively easy for small contexts, i.e., stations with very few platforms and alternative paths to route the trains, but become extremely difficult when applied to complex railway station topologies such as those associated with the main European stations, leading to instances with hundreds of trains and tens of platforms. Moreover, most versions are not concerned with the station topology and ignore the routing phase, whereas the main European stations frequently have complex topologies and the routing issue can be quite a complicated task.

A main station typically has several external lines (also called corridors, generally with two tracks) connecting it to other main stations; these lines are called *directions* in our context. Moreover, there are several points at which a train may stop within the station to download/upload passengers and/or goods; these points are called *platforms* in our context, and can be of different type and length, some being dead-end and some being through-platforms. The connection between directions and platforms is achieved by internal lines, called *paths* in our context, which define a route within the station linking a given direction to a given platform. *Arrival paths* can be used to go from an arrival direction to

a platform, *departure paths* can be used to go from a platform to a departure direction, and *two-way paths* can be used for both purposes.

Moreover, depending on the particular context, there may be other constraints or preferences due to the particular station layout, safety or signalling requirements, operating or marketing policy. The problem aims at defining for each train the platform where it will stop and the corresponding arrival and departure paths, while ensuring that all the constraints are satisfied and minimizing the deviation from some specified “desired” arrival/departure times and stopping platforms for each train.

In this paper, we propose a general formulation of the problem, along with an *Integer Linear Programming* (ILP) formulation whose *Linear Programming* (LP) relaxation is used to drive a heuristic that turns out to widely outperform a simple heuristic currently in use for the instances in our case study. Our main contribution is to consider a general quadratic objective function, given that the objective function is indeed quadratic in our case study, and to propose an efficient way to linearize it by using a small number of new variables along with a set of constraints that can be separated efficiently by solving an appropriate LP.

## 1.1 Literature Review

In the following, we try to give a very quick but comprehensive view of the existing work, referring to the survey by Caprara et al. [2] for a more detailed description. As it is often the case with this type of problems, every reference generally considers a different version, making it difficult to compare the proposed methods. The easiest version is the one considered by De Luca Cardillo and Mione [4] and Billionet [1], who address a simplified version in which, for each train, the scheduled arrival and departure times cannot be changed and the paths used to route the trains within the station are uniquely determined by the choice of the platform. A more general version of the problem, in which arrival and departure times and arrival and departure routes are not fixed a priori is addressed in Zwaneveld [7], Zwaneveld et al. [9], Zwaneveld et al. [8], Kroon et al. [6]. Finally, the version addressed in Carey and Carville [3] is an intermediate one, in that arrival and departure times can be changed, but the assignment of a train to a platform uniquely determines the routes that the train will follow on its way to and from the platform.

## 1.2 The General Problem Considered

In this paper, we deal with a fairly general version of the problem, referred to in the sequel as the *Train Platforming Problem* (TPP). The specific versions previously considered in the literature, as well as the version of our case study, are special cases of TPP.

The input to the problem is a timetable for a set of trains with complete service details, i.e. train number, arrival and departure times and directions.

In the following we will use the concept of *pattern* for a train  $t$  corresponding to a stopping platform, an arrival and a departure path connecting respectively the arrival and departure direction of train  $t$  to the given platform and a time interval of occupation of the platform, implicitly defined by the variation on the arrival and departure time specified in the timetable.

In this general version, we are given a set  $B$  of platforms, a set  $T$  of trains to be routed to a platform, and, for each train  $t \in T$ , a collection  $\mathcal{P}_t$  of possible *patterns*. For convenience, let  $T^2 := (T \times T) \setminus \{(t, t) : t \in T\}$  denote the set of pairs of distinct trains.

Operational constraints forbid the assignment of patterns to trains if this implies occupying the same platform at the same time, or also using routes that intersect at the same time or too close in time. In the general version, this is represented by defining a pattern-incompatibility graph with one node for each train-pattern pair  $(t, P)$ , with  $P \in \mathcal{P}_t$ , and an edge joining each pair  $(t_1, P_1), (t_2, P_2)$  of incompatible patterns.

TPP requires the assignment of a pattern  $P \in \mathcal{P}_t$  to each train  $t \in T$  so that no two incompatible patterns are assigned and the *quadratic* objective function defined by the following coefficients is minimized. There are a cost  $c_b$  for each platform  $b \in B$  that is used in the solution, a cost  $c_{t,P}$  associated with the assignment of pattern  $P \in \mathcal{P}_t$  to train  $t \in T$ , and a cost  $c_{t_1, P_1, t_2, P_2}$  associated with the assignment of pattern  $P_1 \in \mathcal{P}_{t_1}$  and the assignment of pattern  $P_2 \in \mathcal{P}_{t_2}$  to train  $t_2$  for  $(t_1, t_2) \in T^2$ .

A key issue of our approach is to avoid, in the model formulation, the canonical approaches to linearize the objective function, e.g., by introducing additional binary variables to represent the product of the original binary variables — the number of these variables would be very large and the resulting LP relaxation fairly weak. This will be illustrated in detail in the following.

For the applications we are aware of, including our case study, the overall number of patterns  $\sum_{t \in T} |\mathcal{P}_t|$  allows us to handle explicitly all of them. The model that we will present is valid even if this is not the case. As to the solution approach, we will illustrate it assuming the explicit list of patterns is given. If this is not the case, the applicability of the method strongly depends on the specific way in which patterns are described implicitly, indeed in the column generation phase we would need to solve a pricing problem whose nature is directly connected to the description of the patterns.

### 1.3 The Italian Case

The instances in our benchmark come from Rete Ferroviaria Italiana, the Italian Infrastructure Manager. The resulting problem is the special case of TPP with the following characteristics.

It is important to notice that time is discretized considering the minutes in a day, thus time instants are always integer values in the range  $[1, 1440]$ .

The set  $B$  of platforms includes *regular platforms*, corresponding to platforms that one foresees to use, and *dummy platforms*, corresponding to platforms that one would like not to use but that may be necessary to find a feasible solution.

Besides sets  $T$  and  $B$ , we also have a set  $D$  of *directions* for train arrivals and departures and a collection  $\mathcal{R}$  of routes, called *paths*, connecting directions to platforms. Some of these directions are associated with *shunting areas* for the trains that begin/end at the station. For each direction  $d \in D$ , we have a *travel time*  $g_d$  for all paths connecting  $d$  to any platform (independent of the specific path, platform, and train). Moreover, for each ordered pair  $(d_1, d_2) \in D \times D$  corresponding to arrival direction  $d_1$  and departure direction  $d_2$ , the input specifies a *preference list*  $L_{d_1, d_2} \subseteq B$  of preferred platforms for all trains that arrive from direction  $d_1$  and depart to direction  $d_2$ .

For each direction  $d \in D$  and platform  $b \in B$ , we have a (possibly empty) set  $\mathcal{R}_{d,b} \subseteq \mathcal{R}$  of paths linking direction  $d$  to platform  $b$ . Specifically, we have  $\mathcal{R}_{d,b} = \mathcal{R}_{d,b}^a \cup \mathcal{R}_{d,b}^d$ , where the paths in  $\mathcal{R}_{d,b}^a$  are *arrival paths* to get from  $d$  to  $b$  and  $\mathcal{R}_{d,b}^d$  are *departure paths* to get from  $b$  to  $d$ . Note that we may have *two-way paths* in case  $\mathcal{R}_{d,b}^a \cap \mathcal{R}_{d,b}^d \neq \emptyset$ . For each path  $R \in \mathcal{R}$ , we are given a list  $\mathcal{I}_R \subseteq \mathcal{R}$  of *incompatible* paths, these are paths crossing each other at one or more points. (In particular, a path  $R$  is always incompatible with itself.)

Each train  $t \in T$  has an associated *ideal arrival time*  $u_t^a$  at a platform, along with a maximum *arrival shift*  $s_t^a$ , and an associated *ideal departure time*  $u_t^d$  from the platform, along with a maximum *departure shift*  $s_t^d$ , meaning that the train must arrive to a platform in the interval  $[u_t^a - s_t^a, u_t^a + s_t^a]$  and depart in the interval  $[u_t^d - s_t^d, u_t^d + s_t^d]$ . Moreover, each  $t \in T$  has an associated arrival direction  $d_t^a \in D$ , a departure direction  $d_t^d \in D$  and a set  $C_t \subseteq B$  of candidate platforms where it may stop, corresponding to the platforms for which there exist at least two paths linking respectively the arrival and departure directions of  $t$  to the given platform. I.e.,  $C_t = \{b \in B : \mathcal{R}_{d_t^a, b}^a \neq \emptyset, \mathcal{R}_{d_t^d, b}^d \neq \emptyset\}$ .

A pattern  $P \in \mathcal{P}_t$  is defined by a platform  $b \in C_t$ , an arrival path  $R^a \in \mathcal{R}_{d_t^a, b}^a$ , a departure path  $R^d \in \mathcal{R}_{d_t^d, b}^d$ , and the corresponding *actual arrival time*  $v_t^a \in [u_t^a - s_t^a, u_t^a + s_t^a]$  and *actual departure time*  $v_t^d \in [u_t^d - s_t^d, u_t^d + s_t^d]$ . Conventionally, the pattern occupies platform  $b$  for the interval  $[v_t^a - h, v_t^d + h]$ , where  $h$  is a buffer time called *headway* introduced for safety reasons. Moreover, the pattern occupies arrival path  $R^a$  for the interval  $[v_t^a - g_{d_t^a}, v_t^a]$  and the departure path  $R^d$  for the interval  $[v_t^d, v_t^d + g_{d_t^d}]$ , recalling the travel times defined above.

As we have just pointed out the arrival and departure times are always expressed in (an integer number of) minutes, which strongly limits the total number of patterns. Moreover, the problem is periodic with period 1440 minutes (one day), and therefore all times should be considered modulo this period. Nevertheless, given that all occupation times are much smaller than 1440, it is easier for the reader to imagine a linear time window, for which everything is equivalent (except when it comes to the usual boring implementation details).

Two patterns  $P_1 \in \mathcal{P}_{t_1}$  and  $P_2 \in \mathcal{P}_{t_2}$  are incompatible if either their platform occupation intervals overlap for a time window of duration  $> 0$  or if they occupy incompatible paths for a time window of duration  $> \pi$ , where  $\pi$  is a so-called *threshold*. Note that there may be two disjoint time windows in which  $P_1$  and  $P_2$  occupy incompatible paths (e.g., one time window associated with incompatible

arrival paths and one associated with incompatible departure paths), and in this case  $P_1$  and  $P_2$  are incompatible if and only if the largest duration between the two time windows is  $> \pi$ .

For each dummy platform  $b$ , we have infinite two-way paths for each direction  $d \in D$ , all of which are compatible with each other, meaning that the only incompatibilities between trains stopping at  $b$  are related with the occupation of platform  $b$  itself (still associated with headway  $h$ ), as the trains can always use compatible arrival and departure paths.

The objective function is computed by using the following coefficients, for which we also report the numerical values to give an idea of their relative importance:  $\alpha_1 = 1000$ ,  $\alpha_2 = 100000$ ,  $\alpha_3 = 1$ ,  $\alpha_4 = 100$ ,  $\alpha_5 = 10000$ ,  $\alpha_6 = 5$ .

Each platform cost is given by  $c_b = \alpha_1$  if  $b$  is a regular platform, and  $c_b = \alpha_2$  if  $b$  is a dummy platform (in other words the cost for using a dummy platform is two orders of magnitude larger than the cost for using a regular platform).

Each coefficient  $c_{t,P}$  is given by  $\alpha_3 \cdot p_t \cdot s_P$ , where  $p_t$  is a *train priority* value given in input and  $s_P$  is the total shift of pattern  $P$  (counting both the arrival and departure shifts), plus  $\alpha_4$  if pattern  $P$  stops at a regular platform not in the preference list  $L_{d_t^a, d_t^d}$ , plus  $\alpha_5$  if, instead, the pattern stops at a dummy platform.

Finally, each coefficient  $c_{t_1, P_1, t_2, P_2}$  is given by  $\alpha_6 \cdot p_{t_1} \cdot p_{t_2} \cdot w_{P_1, P_2}$ , where  $p_t$  is again the train priority and  $w_{P_1, P_2}$  is the sum of the durations of the (up to two, see above) time windows in which  $P_1$  and  $P_2$  occupy incompatible paths.

## 2 An ILP Formulation

In this section we present an ILP model for the general version of TTP that we consider. The model is mostly standard, but the quadratic term in the objective function is modelled in a non-standard (although fairly simple) way that makes it possible to handle the large-size instances that we encountered in our case study.

The most straightforward 0-1 quadratic programming formulation of the problem, using a binary variable  $y_b$  for each  $b \in B$ , indicating whether platform  $b$  is used, and a binary variable  $x_{t,P}$  for each  $t \in T$  and  $P \in \mathcal{P}_t$ , indicating whether train  $t$  is assigned pattern  $P$ , is the following:

$$\min \sum_{b \in B} c_b y_b + \sum_{t \in T} \sum_{P \in \mathcal{P}_t} c_{t,P} x_{t,P} + \sum_{(t_1, t_2) \in T^2} \sum_{P_1 \in \mathcal{P}_{t_1}} \sum_{P_2 \in \mathcal{P}_{t_2}} c_{t_1, P_1, t_2, P_2} x_{t_1, P_1} x_{t_2, P_2} \quad (1)$$

subject to

$$\sum_{P \in \mathcal{P}_t} x_{t,P} = 1, \quad t \in T, \quad (2)$$

$$\sum_{(t,P) \in K} x_{t,P} \leq y_b, \quad K \in \mathcal{K}_b, \quad (3)$$

$$\sum_{(t,P) \in K} x_{t,P} \leq 1, \quad K \in \mathcal{K}, \quad (4)$$

$$y_b, x_{t,P} \in \{0, 1\}, \quad b \in B, t \in T, P \in \mathcal{P}_t, \quad (5)$$

where  $\mathcal{K}_b$  is the collection of cliques in the pattern-incompatibility graph associated with sets of patterns that use platform  $b$  at the same time, and  $\mathcal{K}$  is the whole collection of cliques in the pattern-incompatibility graph. Constraints (2) guarantee that each train is assigned a pattern, constraints (3) impose that at most one train at a time occupies a given platform  $b$ , and if this ever happens that variable  $y_b$  takes the value 1, and constraints (4) forbid the assignment of patterns that are pairwise incompatible.

## 2.1 A Convenient Version of the Clique Inequalities

We first discuss how to modify constraints (3) and (4), whose number is exponential in the number of patterns, so that they can be handled in practice. First of all, each clique in  $\mathcal{K}_b$  corresponds to a set of intervals (associated with the platform occupation) that intersect pairwise. It is well known from the basic theory of interval graphs that each maximal clique is defined by an interval starting at point  $j$  together with all the intervals  $[l, k]$  with  $l \leq j$  and  $k > j$ . Therefore, the number of maximal cliques cannot be larger than the number of intervals. In our case, letting  $J_b$  denote the set of instants associated with the beginning of the occupation of platform  $b$  by a pattern, and  $K(b, j) \subseteq \mathcal{K}$  the set of patterns that occupy platform  $b$  for an interval  $[l, k]$  with  $l \leq j$  and  $k > j$ , we have the following alternative version of constraints (3):

$$\sum_{(t,P) \in K(b,j)} x_{t,P} \leq y_b, \quad b \in B, j \in J_b, \quad (6)$$

whose number is  $\sum_{b \in B} |J_b|$  and thus can be easily enumerated.

As to constraints (4), they are in general hard to separate. However, if we restrict attention to cliques in  $\mathcal{K}$  containing patterns of two trains only, we get a family of relaxed constraints that are still strong enough to be useful in practice (besides sufficing to define a model) and can be separated efficiently (provided the explicit list of all patterns is known), as explained in the next section. Given two trains  $t_1$  and  $t_2$ , we let  $\mathcal{K}(t_1, t_2) \subseteq \mathcal{K}$  denote the collection of cliques containing only incompatible patterns in  $\mathcal{P}_{t_1} \cup \mathcal{P}_{t_2}$  and define the following alternative version of constraints (4):

$$\sum_{(t_1, P_1) \in K} x_{t_1, P_1} + \sum_{(t_2, P_2) \in K} x_{t_2, P_2} \leq 1, \quad (t_1, t_2) \in T^2, K \in \mathcal{K}(t_1, t_2). \quad (7)$$

## 2.2 Linearizing the Objective Function

We finally illustrate how we linearize the quadratic term in the objective function (1). The textbook approach to linearization amounts to introducing additional binary variables  $z_{t_1, P_1, t_2, P_2}$  that are forced, by linear constraints, to be one if  $x_{t_1, P_1} = x_{t_2, P_2} = 1$ . The number of  $z$  variables is in this case very large and

the resulting LP relaxation fairly weak. On the other hand, the following linearization method requires a much smaller number of variables and leads to provably stronger linear programming relaxations. We introduce the  $\binom{|T|}{2}$  additional continuous variables  $w_{t_1, t_2}$  for  $(t_1, t_2) \in T^2$ , each representing the term  $\sum_{P_1 \in \mathcal{P}_{t_1}} \sum_{P_2 \in \mathcal{P}_{t_2}} c_{t_1, P_1, t_2, P_2} x_{t_1, P_1} x_{t_2, P_2}$ . This leads to the linear objective function:

$$\min \sum_{b \in B} c_b y_b + \sum_{t \in T} \sum_{P \in \mathcal{P}_t} c_{t, P} x_{t, P} + \sum_{(t_1, t_2) \in T^2} w_{t_1, t_2}. \quad (8)$$

We now show how to link the new  $w$  variables with the old ones, by first discussing how to do it in general and then illustrating it through an example, to which the reader may refer while reading the general description.

An elementary link between the  $x$  and the  $w$  variables could be expressed by the linear constraints:

$$w_{t_1, t_2} \geq c_{t_1, P_1, t_2, P_2} (x_{t_1, P_1} + x_{t_2, P_2} - 1), \quad (t_1, t_2) \in T^2, P_1 \in \mathcal{P}_{t_1}, P_2 \in \mathcal{P}_{t_2}, \quad (9)$$

which would however lead to a model equivalent to the textbook one with the  $z$  variables mentioned above. Instead, we can define the following stronger inequalities to bound the  $w$  variables from below. Taking into account the assignment constraints (2) and observing that there are up to  $|\mathcal{P}_{t_1}| |\mathcal{P}_{t_2}|$  possible values for  $w_{t_1, t_2}$ , we can consider the simple polyhedron in  $\mathbb{R}^{|\mathcal{P}_{t_1}| + |\mathcal{P}_{t_2}| + 1}$  corresponding to the convex hull of the  $|\mathcal{P}_{t_1}| |\mathcal{P}_{t_2}|$  possible values taken at the same time by vectors  $(x_{t_1, P_1})_{P_1 \in \mathcal{P}_{t_1}}, (x_{t_2, P_2})_{P_2 \in \mathcal{P}_{t_2}}$  and by variable  $w_{t_1, t_2}$  in a solution:

$$Q_{t_1, t_2} := \text{conv}\{(e_{P_1}, e_{P_2}, c_{t_1, P_1, t_2, P_2}) : P_1 \in \mathcal{P}_{t_1}, P_2 \in \mathcal{P}_{t_2}\}, \quad (10)$$

where, with a slight abuse of notation, for  $i = 1, 2$ , we let  $e_{P_i}$  denote the binary vector in  $\mathbb{R}^{|\mathcal{P}_{t_i}|}$  with the  $P_i$ -th component equal to 1 and all other components equal to 0.

Among the valid inequalities for  $Q_{t_1, t_2}$ , we are interested in those of the form

$$w_{t_1, t_2} \geq \sum_{P_1 \in \mathcal{P}_{t_1}} \alpha_{P_1} x_{t_1, P_1} + \sum_{P_2 \in \mathcal{P}_{t_2}} \beta_{P_2} x_{t_2, P_2} - \gamma. \quad (11)$$

We let  $\mathcal{F}_{t_1, t_2} \subseteq \mathbb{R}^{|\mathcal{P}_{t_1}| + |\mathcal{P}_{t_2}| + 1}$  be the collection of vectors  $(\alpha, \beta, \gamma)$  such that inequality (11) is valid for  $Q_{t_1, t_2}$  and not dominated by other valid inequalities.

*Example 1.* Consider the very simple case in which  $\mathcal{P}_{t_1} = \{P_1, P_3\}$ ,  $\mathcal{P}_{t_2} = \{P_2, P_4\}$ ,  $c_{t_1, P_1, t_2, P_2} = 5$ ,  $c_{t_1, P_1, t_2, P_4} = 3$ ,  $c_{t_1, P_3, t_2, P_2} = 2$ ,  $c_{t_1, P_3, t_2, P_4} = 6$ . In this case, the “weak” inequalities (9) have the form:

$$\begin{aligned} w_{t_1, t_2} &\geq 5x_{t_1, P_1} + 5x_{t_2, P_2} - 5, \\ w_{t_1, t_2} &\geq 3x_{t_1, P_1} + 3x_{t_2, P_4} - 3, \\ w_{t_1, t_2} &\geq 2x_{t_1, P_3} + 2x_{t_2, P_2} - 2, \\ w_{t_1, t_2} &\geq 6x_{t_1, P_3} + 6x_{t_2, P_4} - 6. \end{aligned}$$

We have

$$Q_{t_1, t_2} = \text{conv}\{(1, 0, 1, 0, 5), (1, 0, 0, 1, 3), (0, 1, 1, 0, 2), (0, 1, 0, 1, 6)\}$$

and the “strong” non-dominated inequalities (11), found by enumerating the facets of  $Q_{t_1, t_2}$ , read:

$$\begin{aligned} w_{t_1, t_2} &\geq 5x_{t_1, P_1} + 2x_{t_1, P_3} + 5x_{t_2, P_2} + 3x_{t_2, P_4} - 5, \\ w_{t_1, t_2} &\geq 3x_{t_1, P_1} + 3x_{t_1, P_3} + 2x_{t_2, P_2} + 3x_{t_2, P_4} - 3, \\ w_{t_1, t_2} &\geq 3x_{t_1, P_1} + 2x_{t_1, P_3} + 3x_{t_2, P_2} + 3x_{t_2, P_4} - 3, \\ w_{t_1, t_2} &\geq 2x_{t_1, P_1} + 2x_{t_1, P_3} + 2x_{t_2, P_2} + 2x_{t_2, P_4} - 2, \\ w_{t_1, t_2} &\geq 3x_{t_1, P_1} + 6x_{t_1, P_3} + 2x_{t_2, P_2} + 6x_{t_2, P_4} - 6, \end{aligned}$$

meaning  $\mathcal{F}_{t_1, t_2} = \{(5, 2, 5, 3, 5), (3, 3, 2, 3, 3), (3, 2, 3, 3, 3), (2, 2, 2, 2, 2), (3, 6, 2, 6, 6)\}$ .

### 2.3 The Final ILP Model

To summarize, the ILP formulation that we use has objective function (8) and constraints (2), (5), (6), (7), and:

$$w_{t_1, t_2} \geq \sum_{P_1 \in \mathcal{P}_{t_1}} \alpha_{P_1} x_{t_1, P_1} + \sum_{P_2 \in \mathcal{P}_{t_2}} \beta_{P_2} x_{t_2, P_2} - \gamma, \quad (t_1, t_2) \in T^2, (\alpha, \beta, \gamma) \in \mathcal{F}_{t_1, t_2}. \quad (12)$$

## 3 Solution of the LP Relaxation

As is often the case for the ILP formulations whose LP relaxations yield strong bounds on the optimal integer value, the ILP formulation of the previous section has a large number of variables and constraints. We adopt a canonical approach in which we work with a reduced current LP with all the  $y$  and  $w$  variables and a subset of the  $x$  variables, and all constraints (2) and (6) and only a subset of constraints (7) and (12). Variables and constraints are added dynamically as follows, taking into account the fact that in our case study (as well as in the other TPP case studies we are aware of) all patterns can be listed explicitly,

### 3.1 Variable Pricing

We check if there are negative-reduced-cost  $x$  variables to be added to the current LP by explicitly computing all the reduced costs. This is conceptually easy but not entirely trivial since the constraints that are present in the current LP are defined only with respect to the  $x$  variables that are present. Consequently, computation of the reduced cost of a variable  $x_{t, P}$  requires determining the coefficients of this variable for the constraints in the current LP. This is immediate for constraints (2), the coefficient being 1 for the constraint associated with train  $t$ , and (6), the coefficient being 1 for all constraints associated with the platform

$b$  at which pattern  $P$  stops and with instants  $j \in J_b \cap [l, k]$ , where  $[l, k]$  is the platform occupation interval of pattern  $P$ .

As to constraints (7) and (12), there are several (in general, exponentially many) ways to extend them to include also the  $x$  variables that are not in the current LP. For the purpose of pricing, it is easy to check that one can consider, for each variable  $x_{t,P}$  and for each of these constraints, the maximum possible coefficient for the variable in an extension of the constraint.

Specifically, for each constraint (7), the maximum possible coefficient of variable  $x_{t,P}$  is 1 if and only if  $t_1 = t$  and  $(t, P)$  is incompatible with all  $(t_2, P_2) \in K$  or  $t_2 = t$  and  $(t, P)$  is incompatible with all  $(t_1, P_1) \in K$ . Otherwise, the coefficient is necessarily 0.

Moreover, for each constraint (12), the coefficient of variable  $x_{t,P}$  can clearly be positive only if  $t_1 = t$  or  $t_2 = t$ . Assuming  $t = t_1$ , and letting  $\mathcal{P}'_{t_2}$  be the set of patterns associated with variables  $x_{t_2, P_2}$  in the current LP, the maximum possible coefficient for  $x_{t,P}$  in the constraint is given by

$$\min_{P_2 \in \mathcal{P}'_{t_2}} c_{t,P,t_2,P_2} + \gamma - \beta_{P_2}.$$

### 3.2 Separation of Constraints (7)

Given that all patterns associated with the same train are pairwise incompatible due to constraints (2), the pattern-incompatibility graph with nodes corresponding to the patterns in  $\mathcal{P}_{t_1} \cup \mathcal{P}_{t_2}$  turns out to be the complement of a *bipartite* graph, with the two sides of the bipartition (of the complement) corresponding to the patterns in  $\mathcal{P}_{t_1}$  and those in  $\mathcal{P}_{t_2}$ , respectively.

Therefore, separation of constraints (7) calls for the separation of clique inequalities on the complement of a bipartite graph, or, equivalently, to the separation of stable set inequalities on a bipartite graph. This in turn corresponds to the determination of a maximum-weight stable set in a bipartite graph (with weight  $x_{t_i,P}^*$  for each node  $(t_i, P)$ ,  $i = 1, 2$ , where  $y^*, x^*, w^*$  is the current LP solution), which is well-known to be a minimum  $s, t$ -cut problem on a directed network with source  $s$ , terminal  $t$ , and the other nodes corresponding to the nodes in the bipartite graph.

### 3.3 Separation of Constraints (12)

The separation of constraints (12) is done by a sort of “polyhedral brute force”, given that, for each pair of trains  $t_1, t_2$ , the number of vertices in  $Q_{t_1, t_2}$  is “small”. Specifically,  $Q_{t_1, t_2}$  has  $|\mathcal{P}_{t_1}| |\mathcal{P}_{t_2}|$  vertices and lies in  $\mathbb{R}^{|\mathcal{P}_{t_1}| + |\mathcal{P}_{t_2}| + 1}$ , we can separate over it by solving the following LP with  $|\mathcal{P}_{t_1}| |\mathcal{P}_{t_2}|$  variables and  $|\mathcal{P}_{t_1}| + |\mathcal{P}_{t_2}| + 1$  constraints.

Recall the form of the vertices of  $Q_{t_1, t_2}$  given in its definition (10). Let  $y^*, x^*, w^*$  be the current LP solution. We have that the vector  $((x_{t_1, P_1}^*)_{P_1 \in \mathcal{P}_{t_1}}, (x_{t_2, P_2}^*)_{P_2 \in \mathcal{P}_{t_2}}, w_{t_1, t_2})$  belongs to  $Q_{t_1, t_2}$  if and only if it can be expressed as a convex combination of its vertices, i.e., letting  $\lambda_{P_1, P_2}$  be the multiplier associated

with vertex  $(e_{P_1}, e_{P_2}, c_{t_1, P_1, t_2, P_2})$ , there exists a solution to the linear system:

$$x_{t_1, P_1}^* = \sum_{P_2 \in \mathcal{P}_{t_2}} \lambda_{P_1, P_2}, \quad P_1 \in \mathcal{P}_{t_1}, \quad (13)$$

$$x_{t_2, P_2}^* = \sum_{P_1 \in \mathcal{P}_{t_1}} \lambda_{P_1, P_2}, \quad P_2 \in \mathcal{P}_{t_2}, \quad (14)$$

$$1 = \sum_{P_1 \in \mathcal{P}_{t_1}} \sum_{P_2 \in \mathcal{P}_{t_2}} \lambda_{P_1, P_2}, \quad (15)$$

$$w_{t_1, t_2}^* = \sum_{P_1 \in \mathcal{P}_{t_1}} \sum_{P_2 \in \mathcal{P}_{t_2}} c_{t_1, P_1, t_2, P_2} \lambda_{P_1, P_2}, \quad (16)$$

$$\lambda_{P_1, P_2} \geq 0, \quad P_1 \in \mathcal{P}_{t_1}, \quad P_2 \in \mathcal{P}_{t_2}. \quad (17)$$

Applying Farkas' Lemma, and letting  $\alpha_{P_1}$ ,  $\beta_{P_2}$ ,  $\gamma'$  and  $\varepsilon$  be the dual variables associated with constraints (13), (14), (15) and (16), respectively, we have that the linear system (13)–(17) has a solution if and only if the optimal value of the following LP is zero:

$$\max \sum_{P_1 \in \mathcal{P}_{t_1}} \alpha_{P_1} x_{t_1, P_1}^* + \sum_{P_2 \in \mathcal{P}_{t_2}} \beta_{P_2} x_{t_2, P_2}^* + \gamma' + \varepsilon w_{t_1, t_2}^* \quad (18)$$

subject to

$$\alpha_{P_1} + \beta_{P_2} + \gamma' + \varepsilon c_{t_1, P_1, t_2, P_2} \leq 0, \quad P_1 \in \mathcal{P}_{t_1}, \quad P_2 \in \mathcal{P}_{t_2}. \quad (19)$$

In other words, the vector does not belong to  $Q_{t_1, t_2}$  if and only if the optimal value of LP (18)–(19) is positive (in fact, infinity). Given that we are interested in separating constraints of the form (12), it is easy to check that we can replace “=” by “ $\geq$ ” in constraints (15) and (16), leading to  $\gamma', \varepsilon \leq 0$ , and then add the normalization condition  $\varepsilon = -1$  and replace  $\gamma'$  by  $\gamma := -\gamma'$  — in this way the objective function (18) calls exactly for the determination of the constraint (12) that is violated by the largest amount. Then, for each  $(t_1, t_2) \in T^2$ , we separate constraints (12) by solving LP (18)–(19) after the small changes above.

## 4 Overall Method and Experimental Results

In this section we describe our solution approach TPP, whose main component is the solution of the LP relaxation of the ILP model of Sect. 2 by the method in Sect. 3. Moreover, we illustrate the results obtained for our case study.

#### 4.1 A Branch-and-Bound Method

Our overall method is a branch-and-bound method in which branching is aimed at quickly finding a “good” heuristic solution. This makes it essentially a canonical *diving heuristic* that, rather than terminating at the end of the “dive”, continues as a regular branch-and-bound method until optimality is proved (or the time limit is reached).

Specifically, given the optimal LP solution  $y^*, x^*, w^*$ , if  $x^*$  is integer this is also the optimal ILP solution of the current branch-and-bound problem (defined as the original ILP with the addition of the branching constraints, see below). Otherwise, we select the variable  $x_{t,P}$  which is not fixed by branching constraints and whose value  $x_{t,P}^*$  is closest to 1 (*possibly* it is 1). We generate two problems by imposing, respectively, the *branching constraints*  $x_{t,P} = 1$  and  $x_{t,P} = 0$ , and explore the first problem generated before the second, in a *depth-first* fashion. (Note that, if  $x_{t,P}^* = 1$ , there is no need to solve again the LP relaxation of the first problem.) The first backtracking occurs when we have an integer solution for a problem for which the branching constraints have fixed  $x_{t,P} = 1$  for the  $x$  components with largest LP value encountered. Until this backtracking, the method is a basic textbook diving heuristic.

The solution of the LP relaxation in the problems after the original *root* one is still carried out by pricing and separation, which makes the method a branch-and-cut-and-price one.

#### 4.2 Implementation Details

Our method was implemented in ANSI C and tested on a PC Pentium 4, 3.2 GHz, with a 2 GB RAM.

For the root problem, we initialize the current LP with the  $x$  variables corresponding to the  $|T|$  patterns selected by an elementary greedy heuristic, which considers the trains by decreasing values of the train priority (defined for our case study, see Sect. 1.3) and, for each train, chooses the pattern that is compatible with the patterns already chosen and leads to the smallest increase in the objective function.

The solution of the current LPs is done by using ILOG CPLEX 9.0. Given the solution of each current LP, we perform pricing by finding, for each train, the pattern with most negative reduced cost. If any patterns are found, we add them to the current LP and solve it by primal simplex. Otherwise, i.e., if there is no pattern with negative reduced cost, we separate constraints (7) by solving the minimum  $s, t$ -cut problem by an implementation of the method of [5], for each pair  $(t_1, t_2) \in T^2$ . If any violated constraints (7) are found, we add them to the current LP and solve it by dual simplex. Otherwise, we separate constraints (12) by solving the LP defined in Sect. 3.3, again for each pair  $(t_1, t_2) \in T^2$  and by using ILOG CPLEX 9.0. If any violated constraints (12) are found, we add them to the current LP and solve it by dual simplex. Otherwise, the LP for the current branch-and-bound problem is solved.

### 4.3 Experimental Results for the Case Study

**Table 1.** Instance characteristics

instance	station name	$ T $	$ B $	$ D $	$ \mathcal{R} $	# inc.	$g_d^{\max}$
PA C.LE.	Palermo Centrale	204	11	4	64	1182	3
GE P.PR.	Genova Piazza Principe	127	10	4	174	7154	4
BA C.LE.	Bari Centrale	237	14	5	89	1996	4

Table 1 summarizes the characteristics of the instances used in our case study, reporting the instance name, the full name of the corresponding station, the numbers of trains ( $|T|$ ), platforms ( $|B|$ ), directions ( $|D|$ ), and paths ( $|\mathcal{R}|$ ), the number of pairs of incompatible paths (# inc.), and the maximum travel time ( $g_d^{\max} := \max_{d \in D} g_d$ ).

**Table 2.** Results

instance	$\pi$	HEUR	LP	BEST	time
PA C.LE.	0	749012	334038	449044	200
PA C.LE.	1	410139	10159	120155	230
PA C.LE.	2	380182	10159	10172	339
GE P.PR.	0	745000	306020	306020*	115
GE P.PR.	1	705005	147069	147079	281
GE P.PR.	2	458065	8116	8116*	4617
GE P.PR.	3	336340	8116	8116*	13647
BA C.LE.	0	1576300	653264	808255	350
BA C.LE.	1	1398330	373486	438685	262
BA C.LE.	2	1197485	128896	148867	359
BA C.LE.	3	838235	8885	8924	270

In Table 2 we compare the solution obtained by a (computationally very fast) greedy randomized heuristic algorithm currently used by Rete Ferroviaria Italiana with the the best integer solution produced by our approach with a time limit of 24 hours. For the instances considered, we tested various values of the dynamic threshold  $\pi$ , whose meaning is illustrated in Sect. 1.3. In the table, we report the value of  $\pi$ , the solution value found by the heuristic currently used (HEUR), the optimal value of the LP relaxation at the root problem (LP), the best heuristic solution value found by our method (BEST) — a “\*” means that the solution is optimal, and the computing time in seconds at which this solution was found (time).

The table shows that in all cases our approach was able to improve significantly over the heuristic solution, in most cases finding the best solution after

a fairly small running time (some minutes). In 3 out of 11 cases the solution is provably optimal, in other 3 cases the relative gap between the solution value found and the LP lower bound is less than 1%, whereas in the remaining 5 cases the gap is not negligible, ranging from about 15% to the huge gap for PA C.LE. with  $\pi = 1$ , for which we do not know if the dummy platform that is used by the best solution found is really necessary.

The main practical impact of our approach, if applied in place of the simple heuristic currently in use, is to extend the current “capacity” of the stations considered, using a smaller number of platforms for the current trains and then allowing new trains to stop at the station (if the capacity along the lines associated with the directions allows this.)

Future experiments will be devoted to testing our method on the largest stations of the Italian railway network, such as Milano Centrale.

### Acknowledgments

This work was partially supported by the EU Project ARRIVAL.

### References

- [1] Billionnet A.: Using Integer Programming to Solve the Train Platforming Problem. *Transportation Science* **37** (2003) 213-222
- [2] Caprara A., Kroon L., Monaci M., Peeters M., Toth P.: Passenger Railway Optimization. in Barnhart C., Laporte G. (eds.): *Transportation, Handbooks in Operations Research and Management Science* **14** Elsevier (2007) 129-187
- [3] Carey M., Carville S. : Sceduling and Platforming Trains at Busy Complex Stations. *Transportation Research* **37** (2003) 195-224
- [4] De Luca Cardillo D., Mione N. : k L-List T Colouring of Graphs. *European Journal of Operational Research* **106** (1999) 160-164
- [5] Goldberg A.V., Tarjan R.E.: A New Approach to the Maximum Flow Problem. *Proceedings of the 18th ACM Symposium on the Theory of Computing* (1986)
- [6] Kroon L.G., Romeijn H.E., Zwaneveld P.J.: Routing Trains Through Railway Stations: Complexity Issues. *European Journal of Operations Research* **98** (1997) 485-498.
- [7] Zwaneveld P.J.: Railway Planning and Allocation of Passenge Lines. Ph.D. Thesis, Rotterdam School of Management (1997).
- [8] Zwaneveld P.J., Kroon L.G., van Hoesel C.P.M.: Routing Trains through a Railway Station based on a Node Packing Model. *European Journal of Operations Research* **128** (2001) 14-33.
- [9] Zwaneveld P.J., Kroon L.G., Romeijn H.E., Salomon M., Dauzere-Peres S., van Hoesel C.P.M., Ambergen H.W.: Routing Trains Through Railway Stations: Model Formulation and Algorithm. *Transportation Science* **30** (1996) 181-194.