Non-cyclic Train Timetabling and Comparability Graphs

Valentina Cacchiani   Alberto Caprara   Paolo Toth

Department of Electronics, Computer Science and Systems, University of Bologna
Viale Risorgimento 2, I-40136 Bologna, Italy

Abstract

We consider the customary formulation of non-cyclic train timetabling, calling for a maximum-profit collection of compatible paths in a suitable acyclic network. The associated ILP models look for a maximum-weight clique in a(n exponentially-large) compatibility graph. By taking a close look at the structure of this graph, we analyze the existing ILP models and propose some new stronger ones, all having the essential property that both separation and column generation can be carried out efficiently. Computational results show that the LP relaxations of the new formulations can lead to much stronger bounds on highly-congested instances.

Keywords*: Train Timetabling, Comparability Graph, Constraint Separation*

1 Introduction

This paper concerns a general version of the Non-cyclic Train Timetabling Problem (NTTP), which calls for a maximum-profit set of timetables for a set T of trains traveling on a railway network. Non-cyclic simply indicates that time instants in the given time horizon can be ordered linearly, i.e., it does not exclude the fact that the timetables are repeated with a given period (e.g., every day): if this is the case, it is enough to have a sufficiently "wide" period during which no train is running. The train timetabling problem has been widely studied in the literature: we refer to [6] and [1] (Deliverable D3.1) for a comprehensive survey on the problem, in the cyclic and non-cyclic versions. Other works in this topic are [8], [3], and [7]. In the most recent years, many papers (see e.g. technical reports in [1]) have been devoted to the study of dealing with delays both in the planning phase (for the design of a robust timetable plan) and in the operational phase (i.e., considering online and rescheduling techniques). This work focuses on a more theoretical aspect, analyzing the existing ILP models and proposing some new stronger ones, all having the essential property that both separation and column generation can be carried out efficiently. In addition, computational results show that the LP relaxations of the new formulations can lead to much stronger bounds on highly-congested instances. A customary formulation of the problem (see, e.g., [6]) considers a discretization of the time horizon along with the definition of a directed graph $G = (V, A)$ in which nodes correspond to events, namely to arrival or departures of trains in stations (along specified tracks) at given time instants. In this way, there is a correspondence between the feasible timetables for a train $t \in T$ and the collection $\mathcal{P}^t$ of the paths from an artificial source to an artificial sink in a suitable arc-induced acyclic subgraph G^t of G.

We consider the (typical) case in which the profit associated with a timetable can be expressed as a linear function of the arcs in the corresponding path. In this case, the best timetable for a single train $t \in T$ is given by a maximum-profit path on the acyclic graph G^t, and can be computed in linear time (in the size of G^t) by dynamic programming. The difficulty of the problem comes from the fact that paths for distinct trains can conflict, due to the track capacity constraints illustrated next.

Track capacity constraints are specified by listing a set L of line segments, corresponding to tracks joining two stations in the network without intermediate stations in between. Each line segment $\ell \in L$ is associated with a subset $A_\ell \subseteq A$ of the arcs in G. Moreover, for each train $t \in T$ and line segment $\ell \in L$, every path in $\mathcal{P}^t$ can contain at most one arc in A_ℓ (meaning that the train path can traverse each line segment at most once). Consider two trains t_1, t_2 along with two paths $P_1 \in \mathcal{P}^{t_1}, P_2 \in \mathcal{P}^{t_2}$ containing two arcs $a_1 \in P_1 \cap A_\ell, a_2 \in P_2 \cap A_\ell$. Arc a_1 represents the departure of t_1 from the first station at time d_1 and its arrival at the second station at time r_1. Similarly, arc a_2 represents the departure of t_2 from the first station at time d_2 and its arrival at the second station at time r_2. Assuming without loss of generality that $d_1 \leq d_2$, we have that P_1, P_2 respect the track capacity constraints on line segment ℓ if

$$d_2 \geq d_1 + \alpha_\ell \quad \text{and} \quad r_2 \geq r_1 + \beta_\ell, \tag{1}$$

where α_ℓ and β_ℓ are given parameters, possibly depending on the line segment. In words, there is a minimum time distance between departures and arrivals along each line segment, and trains cannot overtake each other along this line segment.

In this paper, we will take a close look at old and new ILP formulations for NTTP, all involving constraints that can be associated with stable sets in suitable graphs. In order to have a general view of these formulations and be able to compare them, it will be fundamental to point out the underlying graphs; for this reason we conclude the introduction with some basic graph-theoretic notions and notations that we will use extensively.

Given a graph G, we let $\mathcal{S}(G)$ denote the collection of all maximal stable sets of G, i.e., of all maximal node subsets of G for which there exists no edge of G with both endpoints in the subset. Given two graphs G_1, G_2 on the same node set, we let $G_1 \cap G_2$ denote the edge intersection of G_1 and G_2, i.e., the graph on the same node set whose edges are those that are present in both G_1 and G_2.

A comparability graph is an undirected graph whose edges can be oriented so as to get an acyclic directed graph $D = (N, A)$ which is transitive, i.e., such that $(i, j), (j, k) \in A$ implies $(i, k) \in A$. The comparability graph is associated with the partial order $\prec$ on node set N given by $i \prec j$ if and only if $(i, j) \in A$.

2 ILP Formulations, Graphs, and Separation

For each $t \in T$, recalling that $\mathcal{P}^t$ denotes the collection of possible paths for train t, let π_P be the profit of path $P \in \mathcal{P}^t$. Moreover, let $\mathcal{P} := \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^t$ be the overall (multi-)collection of paths. Two paths $P_1, P_2 \in \mathcal{P}$ are *compatible*, i.e., they can be both selected in the solution, if the following hold:

- the two paths are associated with distinct trains;
- for each line segment $\ell \in L$ traversed by both P_1, P_2, the two paths respect the track capacity constraints on ℓ.

The objective is the maximization of the profits of the paths selected with the constraint that all paths selected are compatible. The compatibility relation is naturally represented by an auxiliary graph $F = (\mathcal{P}, E)$ with one node for each path and an edge joining each pair of compatible paths. Then, NTTP calls for a maximum-weight clique in F. Note that F is the edge intersection of the following $|L| + 1$ graphs on node set $\mathcal{P}$:

- F_T, in which two nodes are joined by an edge if and only if the corresponding paths are associated with distinct trains;
- F_ℓ, $\ell \in L$, in which two nodes are joined by an edge if and only if the corresponding paths either do not both traverse segment ℓ, or they traverse it by respecting the track capacity constraints.

The structure of F_T and F_ℓ

The structure of F_T is elementary, namely it is a collection of the $|T|$ stable sets $\mathcal{P}_t$, $t \in T$, with edges joining each pair of nodes belonging to distinct stable sets. In other words:

Observation 1 *F_T is a complete $|T|$-partite graph;*

The structure of F_ℓ is way more interesting:

Observation 2 *For $\ell \in L$, F_ℓ is a comparability graph.*

A general, impractical ILP formulation

By associating a binary variable x_P with each path $P \in \mathcal{P}$, the most natural ILP formulation of NTTP would be to associate a constraint with each stable set of F. The formulation reads:

$$\max \sum_{P \in \mathcal{P}} \pi_P x_P, \tag{2}$$

$$\sum_{P \in S} x_P \leq 1, \qquad S \in \mathcal{S}(F), \tag{3}$$

$$x_P \in \{0,1\}, \qquad P \in \mathcal{P}. \tag{4}$$

Although the corresponding LP relaxation is fairly weak for maximum-weight clique in general, in the cases in which the objects represented by the nodes have a special structure (e.g., paths or cycles in a graph) the resulting upper bound often turns out to be strong. On the other hand, even putting aside the fact that $|\mathcal{P}|$ may be exponential, the solution of this LP relaxation turns out to be hard, recalling the well-known equivalence between separation and optimization [9].

Proposition 1 *The separation of constraints* (3) *is strongly NP-complete even when $|\mathcal{P}| = |T|$ (one feasible path per train), namely the problem of finding a maximum-weight stable set on a generic graph with n nodes can be reduced to it, setting $|T| := |\mathcal{P}| := n$.*

Jointly with the fact that $|\mathcal{P}|$ is in general exponentially large, and therefore that the separation of (3) would also complicate the associated column generation problem, there appears to be no chance in practice to solve the LP relaxation of the above ILP formulation for NTTP instances of interest.

Practical ILP formulations

Forgetting about the whole set of (3), a natural alternative is to concentrate on alternative constraints with the following structure.

Definition 1 *A set of constraints of the form*

$$\sum_{P \in S} x_P \leq 1, \qquad S \in \mathcal{S}', \tag{5}$$

is said to be practical *for NTTP if:*

(i) together with the binary condition (4)*, it defines a valid ILP formulation for NTTP;*

(ii) can be separated in polynomial time;

(iii) the column generation problem for the variables associated with each train $t \in T$ can be carried out by computing an optimal path on the graph G^t with appropriate arc costs.

In other words, (iii) is the natural requirement that the column generation problem has the same structure as the problem of finding the best path for a given train for the original profits.

Requirement (i) is easy to deal with, namely:

Observation 3 *Consider a collection of graphs $F_1, \ldots, F_m$ whose edge intersection yields F. The set of constraints*

$$\sum_{P \in S} x_P \leq 1, \qquad S \in \mathcal{S}(F_1) \cup \cdots \cup \mathcal{S}(F_m),$$

satisfies (i).

(Note that also the converse holds, namely, given any set of constraints (5) that satisfies (i), we can define a collection of graphs whose edge intersection yields F and such that these constraints correspond to the maximal stable sets of the graphs in the collection.)

Requirement (ii) needs to be addressed separately case by case (as it is generally the case with complexity issues). As to (iii), it is satisfied if a technical condition discussed in the full paper holds.

The ILP formulation in [4]

In [4], we considered the following ILP model. For a train pair $\{t_1, t_2\} \subseteq T$, let $F_T(\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2})$ and $F_\ell(\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2})$ be the subgraphs of F_T and F_ℓ induced by node set $\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2} \subseteq \mathcal{P}$. We considered the constraints associated with the stable sets of the edge intersection $F_T(\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2}) \cap F_\ell(\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2})$, called *overtaking constraints* in [4]:

$$\sum_{P \in S} x_P \leq 1, \qquad \{t_1, t_2\} \subseteq T,\ \ell \in L,\ S \in \mathcal{S}(F_T(\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2}) \cap F_\ell(\mathcal{P}^{t_1} \cup \mathcal{P}^{t_2})). \tag{6}$$

Observation 4 *Constraints* (6) *satisfy requirements (i)-(iii) in Definition 1.*

Although constraints (6) are sufficient to define an ILP formulation, they tend to be fairly weak in practice and, in any case, rather slow to separate. For this reason, the following stronger constraints are used in [4]. First of all, we use the obvious constraints associated with the maximal stable sets of F_T, which, due to Observation 1, read:

$$\sum_{P \in \mathcal{P}^t} x_P \leq 1, \qquad t \in T, \tag{7}$$

(i.e., for each train we select at most one path in the solution) and do not need to be separated as they are only $|T|$. Moreover, we considered the edge-induced subgraph F_ℓ^d of F_ℓ associated with the relaxation of (1):

$$d_2 \geq d_1 + \alpha_\ell,$$

and the edge-induced subgraph F_ℓ^r of F_ℓ associated with the relaxation of (1):

$$r_2 \geq r_1 + \beta_\ell.$$

It is easy to check that both F_ℓ^d and F_ℓ^r are not only comparability graphs, but also the complement of an interval graph. The corresponding constraints, called *departure* and *arrival constraints*, respectively, in [4], read:

$$\sum_{P \in S} x_P \leq 1, \qquad \ell \in L,\ S \in \mathcal{S}(F_\ell^d), \tag{8}$$

$$\sum_{P \in S} x_P \leq 1, \qquad \ell \in L,\ S \in \mathcal{S}(F_\ell^r), \tag{9}$$

and can be separated in linear time (in the size of G and the number of nonzero variables of the current LP solution). Only if these constraints are satisfied, we proceed with the separation of (6).

A simpler, natural ILP formulation

A natural formulation which is in fact simpler than the one in [4] is obtained by combining Observation 3 and the fact that F is the edge intersection of F_T and F_ℓ for $\ell \in L$, leading to the following constraints. For F_T, we have constraints (7) already mentioned above. As to F_ℓ, we have the constraints:

$$\sum_{P \in S} x_P \leq 1, \qquad \ell \in L,\ S \in \mathcal{S}(F_\ell), \tag{10}$$

which are clearly stronger than (8) and (9), and whose separation calls for the determination of a maximum-weight stable set in a comparability graph, which can be found efficiently by maximum flow techniques.

Proposition 2 *Constraints* (7) *and* (10) *satisfy requirements (i)-(iii) in Definition 1.*

A third natural, and stronger, ILP formulation

A third alternative to constraints (3) can be obtained by merging the main ideas in the previous two ILP formulations, and the resulting formulation is stronger than both. Specifically, we consider, for $\ell \in L$ the edge intersection of $F_T \cap F_\ell$, noting that F itself is the edge intersection of these $|L|$ graphs. The constraints that replace (3) in this third formulation are:

$$\sum_{P \in S} x_P \leq 1, \qquad \ell \in L,\ S \in \mathcal{S}(F_T \cap F_\ell). \tag{11}$$

These constraints are clearly stronger than (7) and (10) (and also stronger than (6)), and are easily checked to satisfy requirements (i) and (iii) in Definition 1 (see below). On the other hand the complexity of their separation is unclear in general. Nevertheless, for the instances in our case study, in which the travel time along each line segment depends only on the train (type), we are able to devise a polynomial time algorithm. Formally, we say that travel times are fixed if, for each train $t \in T$ and line segment $\ell \in L$, there exists a value $\theta(t, \ell)$ such that all arcs in A_ℓ in some path $P \in \mathcal{P}^t$ have an arrival time equal to the departure time plus $\theta(t, \ell)$.

Proposition 3 *If travel times are fixed, a maximum-weight stable set in $F_T \cap F_\ell$ can be found in polynomial time by dynamic programming.*

Proposition 4 *If travel times are fixed, constraints* (11) *satisfy requirements (i)-(iii) in Definition 1.*

The dynamic programming procedure in Proposition 3 has a fairly high time and space complexity, which make it slow in practice. Accordingly, we developed two different methods to separate these constraints heuristically.

The first heuristic separation procedure is a simple (randomized) greedy heuristic for maximum-weight stable set that, starting from the empty solution, at each iteration selects a node with a probability proportional to the ratio between the weight of the node and the sum of the weights of its neighbors. The node selected is added to the stable set and it is removed from the graph together with all its neighbors.

The second heuristic procedure uses the fact that, as already mentioned in Section 2, a maximum-weight stable set in a comparability graph can be found efficiently. Specifically, we consider graph $F_T \cap F_\ell$, along with the comparability graph F_ℓ and the associated transitive directed graph D. We orient all edges in $F_T \cap F_\ell$ as the corresponding arcs in D: the resulting graph D' needs not be transitive as it has only a subset of the arcs of D. Then, we compute the transitive closure D'' of D', and finally find a maximum-weight stable set in the comparability graph obtained by ignoring the edge orientations in D''. In the sequel we will refer to this heuristic separation method as the transitivization procedure.

	LP in [4]		LP (7),(10)		LP (11)		LP (11) - Greedy		LP (11) - Trans.	
Instance	Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
Bologna-Milano	1210.7	8	1187.9	2266	1110.5	11750	1135.7	202	1113.1	1967
Bologna-Roma	1184.0	28	1120.1	948	996.9	41050	1044.9	171	1007.5	1519
Brennero-Bologna	1169.7	15	1147.9	1933	1056.5	47691	1067.7	430	1058.5	1388
Milano-Roma	1105.1	177	1029.6	694	947.1	31010	972.5	350	951.1	1098
Modane-Milano	1136.5	49	1079.1	569	993.9	29479	1015.5	150	999.9	770

Table 1: Upper bound values and solution times for small, highly-congested instances.

		Best	LP in [4]			LP (11) - Trans.		
Instance	$\|T\|$	Value	Value	Time	Gap%	Value	Time	Gap%
Bologna-Milano	30	2441	2734.4	99	10.7%	2618.4	TL	6.8%
Bologna-Roma	45	4444	4499.9	4681	1.2%	4444.0	17963	0.0%
Brennero-Bologna	45	3260	3737.5	405	12.8%	3616.9	TL	9.9%
Milano-Roma	65	5290	5483.5	2558	3.5%	5302.1	TL	0.2%
Modane-Milano	40	3135	3448.9	626	9.1%	3271.3	TL	4.2%

Table 2: Upper bound values, solution times, and optimality gap for realistic congested instances.

3 Computational Results

We first considered a set of highly-congested instances containing 12 trains running on distinct main corridors of the Italian railway network. In Table 1 we report the upper bounds associated with the LP relaxations of the ILP formulations in the previous section and the associated solution time in seconds. Versions *Greedy* and *Trans.* of constraints (11) refer to the heuristic separation of these constraints by the greedy procedure and the transitivization procedure of Section 2, respectively. Not counting these two versions, the left part of the table shows that the quality of the upper bound improves slightly from the LP in [4] to the LP with constraints (7),(10), and significantly from the latter to the LP with constraints (11). On the other hand, the solution time increases by more than order of magnitude going from one formulation to the other. Still, if we resort to heuristic separation of constraints (11), with the greedy procedure we get a lower bound which is still much better than those of the two other LPs within a relatively small running time, whereas with the transitivization procedure we get a lower bound which is basically the same as the one obtained by exact separation by dynamic programming, within a running time that is one order of magnitude smaller.

The instances in Table 1 are in fact over-congested, and were considered only to carry over the comparison of the upper bounds and computing times. Realistic congested instances are instead considered in Table 2. These instances contain a larger number of trains $|T|$, and for them the use of the dynamic programming procedure for the separation of (11) is out of reach. For these instances, we report the value of the best solution found by the method of [5] (*Best*), along with the upper bound, the computing times and the final percentage gap over the best solution value for the LP in [4] and the LP with constraints (11) that are separated by the transitivization procedure, with a time limit of 10 hours. The table shows that, although the latter reaches the time limit (indicated by *TL*) in all cases except one, the final gap between the heuristic and upper bound value is much smaller with the new constraints, and in one case optimality of the best solution is proved (for the only instance that ends before the time limit).

References

[1] ARRIVAL Research Project, 6^{th} Framework Program (http://arrival.cti.gr/).

[2] U. Brännlund, P.O. Lindberg, A. Nöu and J.E. Nilsson, *Allocation of Scarce Track Capacity Using Lagrangian Relaxation*, Transportation Science **32** (1998) pp. 358–369.

[3] R. L. Burdett and E. Kozan, *Techniques for absolute capacity determination in railways*, Transportation Research Part B **40** (2006) pp. 616–632.

[4] V. Cacchiani, A. Caprara and P. Toth, *A Column Generation Approach to Train Timetabling on a Corridor*, 4OR **6** (2008) pp. 125–142.

[5] A. Caprara, M. Fischetti and P. Toth, *Modeling and Solving the Train Timetabling Problem*, Operations Research **50** (2002) pp. 851–861.

[6] A. Caprara, L. Kroon, M. Monaci, M. Peeters and P. Toth, *Passenger Railway Optimization*, in *Handbook in Operations Research and Management Science 14*, C. Barnhart, G. Laporte, eds., Elsevier, Amsterdam, 2007, pp. 129–187.

[7] M. Carey and I. Crawford, *Scheduling trains on a network of busy complex stations*, Transportation Research Part B **41** (2007) pp. 159–178.

[8] M. Carey and D. Lockwood, *A model, algorithms and strategy for train pathing*, Journal of the Operational Research Society **46** (1995) pp. 988-1005.

[9] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, (1988).