CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Exact Formulations and Algorithm for the Train Timetabling Problem with Dynamic Demand

Eva Barrena
David Canca Ortiz
Leandro C. Coelho
Gilbert Laporte

July 2013

CIRRELT-2013-41

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Exact Formulations and Algorithm for the Train Timetabling Problem with Dynamic Demand

Eva Barrena[1,2,*], David Canca Ortiz[3], Leandro C. Coelho[1,4,*], Gilbert Laporte[1,2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
[2] Department of Management Sciences, HEC Montréal, 3000, Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
[3] School of Engineering, University of Seville, Avenida de los Descubrimientos s/n, 41092,Seville, Spain
[4] Department of Operations and Decision Systems, Université Laval, 2325, de la Terrasse, Québec, Canada G1V 0A6

Abstract. In this paper we study the design and optimization of train timetabling adapted to a dynamic demand environment. This problem arises in rapid train services which are common in most important cities. We present four formulations for the problem, with the aim of minimizing passenger average waiting time. The first one consists of a mixed integer non-linear programming model in which binary variables represent train launching times and the objective function contains a quadratic term. The other three introduce flow variables, allowing for a linear representation of the objective function. We provide incremental improvements on these formulations, which allows us to evaluate and compare the benefits and disadvantages of each modification. We present a branch-and-cut algorithm applicable to all formulations. Through extensive computational experiments on several instances derived from real data provided by the Madrid Metropolitan Railway, we show the advantages of designing a timetable adapted to the demand pattern, as opposed to a regular timetable. We also perform an extensive computational comparison of all linear formulations in terms of size, solution quality and running time.

Keywords. Train timetabling, dynamic demand, regular timetable, exact algorithm, branch-and-cut.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 39682-10. This support is gratefully acknowledged. We also thank the Réseau québécois de calcul de haute performance (RQCHP) for providing computing facilities.

---

* Corresponding author: Eva.Barrena@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2013

1 Introduction

The railway planning process is a complex activity which is usually decomposed into a succession of stages, including network design, line design, scheduling, timetabling, rolling stock, and personnel planning [1, 12, 14, 15]. Timetabling design consists of determining launch and arrival times for each train service to and from each station along a railway line. A service is defined as a trip from an origin to a final destination station. In this paper, a train refers to the service it operates. We consider the case of a double direction rapid transit line with two tracks, in which case launch and arrival times can be designed without train conflicts on track segments, i.e., on the line portions between two consecutive stations.

Timetables are often constructed subject to a regularity or periodicity constraint, using a constant origin-destination peak-hour demand matrix [19, 20, 26]. Regular timetables are mainly used in rapid transit systems, where the frequency of the train services is high and their departures are equally spaced throughout the planning horizon, for example, every seven minutes. A periodic timetable repeats itself at every period of the planning horizon, for example, trains may be scheduled to depart at 3, 21 and 46 minutes every hour. Periodic timetables have proved their ability to deal with large-scale railway networks [19], they are easily memorized by passengers and, in the case of constant demand, they yield minimum waiting times [20]. Periodic solutions were initially proposed by Voorhoeve [26] who followed the formulation of Serafini and Ukovich [25].

We now describe some of the main scientific contributions available in this area. In Nachtigall and Voget [24], the authors present a genetic algorithm which is combined with a greedy heuristic and a local improvement procedure to obtain timetables while minimizing the average waiting time. Liebchen and Möhring [21] model the periodic event scheduling problem (PESP) as a digraph in which temporal restrictions on the arcs relate periodically recurring events. In Liebchen and Peeters [22], the authors introduce the concept of integral cycle bases for characterizing periodic tensions, following the work of Nachtigall [23]. Chierici et al. [11] study the quality of timetables and the corresponding demand captured by means of a logit model which computes the modal split between railway and an alternative transportation mode. Cordone and Redaelli [13] develop a branch-and-bound algorithm based on a piecewise-linear approximation of a non-convex objective function. These authors presented computational results concerning both random instances and a real-world regional network located in Northwestern Italy. Kroon et al. [17, 18] study the problem of improving periodic timetables in the Netherlands under a regular demand assumption. These authors developed a stochastic optimization model to allocate buffer times with the objective of minimizing random disturbances.

If demand cannot be assumed to be constant over time, the problem then becomes much more general. A periodic timetable applied to a general demand case leads to low occupancy levels of the trains and high average waiting times [6, 20]. Non-periodic timetabling is particularly appropriate in long corridors with high track densities.

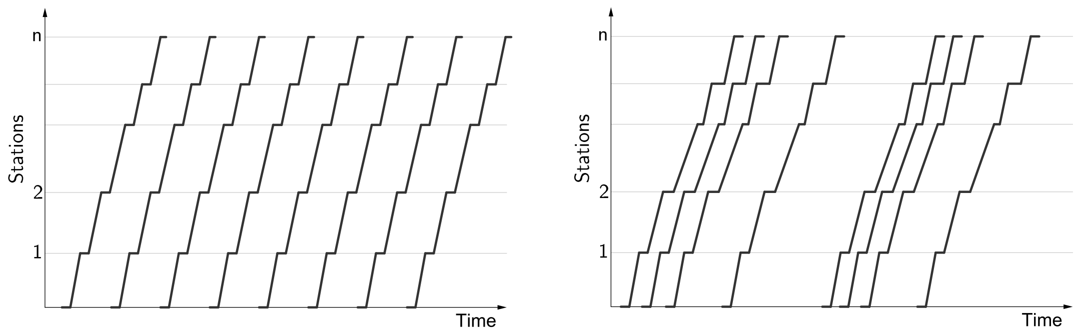
Caprara et al. [10] use an integer linear programming (ILP) model to determine trains timetable considering modifications over an ideal timetable provided by the train operator. Launch time of trains at the first station can be modified, trains can be cancelled, and speeds and dwell times can be reduced in order to satisfy track capacity constraints. The model incorporates manual block signalling for managing a train on a track segment and maintenance operations that can block a track segment for a given period. Cacchiani et al. [4] extend this ILP model and apply a Lagrangian heuristic algorithm to deal with additional real-world constraints. Cacchiani et al. [3] consider a similar problem in which solutions are forced to satisfy the track capacity constraints while minimizing deviations of launch and arrival train times with respect to an ideal known timetable. They use an ILP formulation, which is obtained from a so-called compatibility graph. Cacchiani et al. [2] consider an alternative ILP model in which each variable corresponds to a full train timetable. They propose heuristic and exact algorithms based on the solution of the LP relaxation model. Ingolotti et al. [16] implement a metaheuristic considering a set of realistic safety and operative constraints. The authors do not consider passenger demand. Their objective is to minimize the deviation between the train delays with respect to the minimum total running time.

In this paper, we focus on constructing timetables adapted to a dynamic demand pattern [7]. Our study is motivated by a collaboration with the Madrid Metropolitan Railway, which provided real demand data for their C5 line. We introduce four different formulations to model this problem. One of the main features of these formulations is that they do not assume any shape for the demand function; they can deal with non-monotonic and even non-convex demand functions. The objective of the problem is to minimize passenger waiting times at stations. We propose exact algorithms to optimize the models. The solutions are train timetables adjusted to a dynamic demand pattern over a finite planning horizon and are not necessarily regular, nor periodic. We believe this is the first exact algorithm ever proposed for this problem. We note that the train timetabling problem is NP-hard [5, 8], which justifies looking for tight formulations and efficient algorithms.

The remainder of this paper is organized as follows. In Section 2 we formally describe the problem and introduce some notation common to all models. In Section 3 we propose four mathematical formulations for the problem, followed by the description of a branch-and-cut algorithm applicable to all models in Section 4. We present the results of extensive computational experiments in Section 5, followed by conclusions in Section 6.

2 Problem description

Train timetables are normally represented in form of time-space diagrams as shown in Figure 1. The x-axis represents the planning horizon, and the y-axis the stations of the considered line, more concretely, the distance from each station to the first one. Figure 1(a) illustrates a regular timetable, i.e., the headway between consecutive trains is constant, and Figure 1(b) a non-regular timetable, i.e., headways are not necessarily constant and train frequency is normally higher around peak hours.



(a) Regular timetable (b) Non-regular timetable

Figure 1: Time-space diagrams of train timetables for a one line corridor

We now formally describe the train timetabling design problem for a two-track railway line, one in each direction. The determination of the timetable can then be decomposed into two independent problems. Let $\mathcal{S} = \{1, \ldots, n\}$ be the ordered set of stations defining a two-track railway line. The planning horizon is discretized into time intervals of length δ. Thus, time instant $t \in \mathcal{T} = \{0, 1, \ldots, p\}$ corresponds to δt time units elapsed since the beginning of the planning horizon. The discretization constant δ represents the length of the smallest time interval considered in the problem and so, from now on we will consider it as the time unit which can be as small as desired. Let d_{ij}^t be the passenger demand between stations $i, j \in \mathcal{S}, j > i$ during the interval $[t-1, t]$. We assume that passenger arrival data are available for each time interval. This demand description is very common in modern transit systems where data acquisition devices are installed at the entrance of stations and these data are used to compute the origin-destination matrices. Let l_{ij} be the length of the segment between stations i and j, h_{min} be the minimum headway, i.e., the minimum amount of time required between the launch of two consecutive trains at each station, w_{min} and w_{max} be the minimum and maximum allowed dwell time at stations, and s_{min} and s_{max} be the inverse of the minimum and maximum traveling speed of a train. Note that we work with the inverse of the speeds to avoid non-linear terms in the constraints of the problem.

The aim of the problem is to determine train launch times at stations and train speeds on segments such that the average waiting time of passengers on the stations is minimized.

3 Mathematical formulations

We propose four formulations for the railway timetabling problem with dynamic demand. In what follows we assume that a set $\mathcal{M} = \{1, \ldots, m\}$ of possible trains is available. Note that this is not a strong assumption and that the models can be easily solved with an unlimited number of trains. We use variables s_i^k to represent the inverse of the speed of train k when leaving station i, and variables w_i^k representing the dwell time of train k at station i.

3.1 Non-Linear Formulation

The first formulation uses binary variables $y_k, k \in \mathcal{M}$, equal to one if and only if train k is used in the solution, and binary variables z_{ik}^t equal to one if and only if train k leaves station i at time t. For all trains used, let the launch time of train k at station i be represented by an integer variable x_k^i, i.e., $x_k^i = \sum_{t \in \mathcal{T}} t z_{ik}^t$. Let D_i be the cumulative function of the outgoing demand at station i, i.e., $D_i(t) = \sum_{t'=0}^{t} \sum_{j \in \mathcal{S}, j>i} d_{ij}^{t'}$. Let $\hat{x}_k^i = \sum_{t \in \mathcal{T}} D_i(t) z_{ik}^t$ be the cumulative demand when train k leaves station i. These variables give rise to a non-linear objective function expressed in terms of launch times of each pair of consecutive trains. In order to write the following model, we consider two dummy trains, one at the beginning and another at the end of the planning horizon. The problem can then be formulated as follows:

$$\text{(NLF)} \quad \text{minimize} \quad \frac{1}{2} \sum_{i \in \mathcal{S}} \sum_{k \in \mathcal{M} \cup \{0, m+1\}} \left(\hat{x}_k^i - \hat{x}_{k-1}^i\right) \cdot \left(x_k^i - x_{k-1}^i\right) \tag{1}$$

subject to

$$x_0^i = 0 \quad i \in \mathcal{S} \tag{2}$$

$$x_{m+1}^i = p \quad i \in \mathcal{S} \tag{3}$$

$$p(1 - y_k) \leq x_k^i \leq p \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \tag{4}$$

$$x_k^i \leq x_{k+1}^i \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \setminus \{m\} \tag{5}$$

$$x_k^{i+1} = x_k^i + s_i^k l_{i,i+1} + w_{i+1}^k \quad i \in \mathcal{S} \setminus \{n\}, \quad k \in \mathcal{M} \tag{6}$$

$$y_k w_{min} \leq w_i^k \leq y_k w_{max} \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \tag{7}$$

$$y_k s_{min} \leq s_i^k \leq y_k s_{max} \quad i \in \mathcal{S} \setminus \{n\}, \quad k \in \mathcal{M} \tag{8}$$

$$x_{k+1}^i \geq x_k^i + h_{min} y_{k+1} + w_i^k \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \setminus \{m\} \tag{9}$$

$$\sum_{t \in \mathcal{T}} z_{ik}^t \leq 1 \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \tag{10}$$

$$x_k^i = \sum_{t \in \mathcal{T}} t z_{ik}^t \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \tag{11}$$

$$x_k^i \in \mathbb{N}^+ \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \tag{12}$$

$$y_k \in \{0, 1\} \quad k \in \mathcal{M} \tag{13}$$

$$z_{ik}^t \in \{0, 1\} \quad i \in \mathcal{S}, \quad k \in \mathcal{M}, \quad t \in \mathcal{T}. \tag{14}$$

In this formulation, the objective function (1) minimizes the total waiting time of the passengers. Constraints (2) and (3) define the dummy train launch times. Their role is to ensure that the objective function considers all waiting times and demands into account. Constraints (4) link variables x_k^i and y_k. Specifically, if train k is not used, i.e., $y_k = 0$, then it is launched at the end of the planning horizon, i.e., $x_k^i = p$. On the other hand, if train k is used, i.e., $y_k = 1$, then x_k^i can take any value smaller than p, i.e., the train must be launched before the end of the planning horizon. Constraints (5) order the trains in terms of launch times, i.e., together with constraints (4), they ensure that trains with lower index values are launched first. This is a necessary condition for this formulation, but it also breaks symmetric solutions. Constraints (6) ensure that if train k is launched from station i at time x_k^i, then this train has to be launched from the next station $i+1$ at time x_k^i, plus the time required to go from i to $i+1$, plus the dwell time at station $i+1$. Constraints (7) set the bounds for dwell times of launched trains and consider dwell times equal to zero for those that are not launched. Constraints (8) are similar to (7). They bound the inverse of the speed s_i^k of train k leaving station i for launched trains, and set it to zero for unlaunched trains. Note that constraints (6)$-$(8) ensure that if train k is not launched, then $x_k^i = p$ for all stations i. Constraints (9) play a stronger role than (5) and (6) by ensuring that no two trains are launched from the same station at the same time, and by also considering the minimum headway h_{min}. Constraints (10) ensure that each train is launched at most once from each station. Constraints (11) link binary variables z with their corresponding launch time x. Integrality and binary conditions on the variables are enforced through constraints (12)$-$(14).

3.2 Linear Formulation 1

This formulation introduces passenger flow variables in order to linearize the objective function. It uses binary variables x_{ki}^t equal to one if and only if train k leaves station i at time t, integer variables u_i^t representing the number of passengers boarding the train leaving station i at time t, and variables f_i^t representing the number of passengers waiting at station i at the end of interval $[t-1, t]$, i.e., if no train departs from station i at time t. We realistically assume that $f_i^0 = 0$ since no initial demand is considered. These variables give rise to a linear representation of the objective function with respect to a dynamic demand function. In order to compute the average waiting time (AWT) per passenger, we consider that a passenger arriving in the interval $[t, t+1]$ waits half of this time interval, i.e., $\delta/2$, plus the full δ for each of the remaining time intervals until boarding the next train. So, the total waiting time is $\frac{\delta}{2} \sum_{t \in \mathcal{T}} \sum_{i,j \in \mathcal{S}, j>i} d_{ij}^t + \delta \sum_{i \in \mathcal{S} \setminus \{n\}} \sum_{t \in \mathcal{T} \setminus \{p\}} f_i^t$. Since the first term is a constant, we do not consider it in the objective function. The problem can then be formulated as follows:

$$\text{(LF1)} \quad \text{minimize} \quad \delta \sum_{i \in \mathcal{S} \setminus \{n\}} \sum_{t \in \mathcal{T} \setminus \{p\}} f_i^t \tag{15}$$

subject to

$$f_i^t = f_i^{t-1} + \sum_{j \in \mathcal{S}, j > i} d_{ij}^t - u_i^t \quad i \in \mathcal{S} \setminus \{n\}, \quad t \in \mathcal{T} \tag{16}$$

$$u_i^t \leq \sum_{k \in \mathcal{M}} x_{ki}^t \sum_{t'=0}^{t} \sum_{j \in \mathcal{S}, j > i} d_{ij}^{t'} \quad i \in \mathcal{S} \setminus \{n\}, \quad t \in \mathcal{T} \tag{17}$$

$$x_{ki}^t \leq \sum_{t'=0}^{t} x_{k-1,i}^{t'} \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \setminus \{1\}, \quad t \in \mathcal{T} \tag{18}$$

$$\sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{M}} x_{k1}^t \leq m \tag{19}$$

$$\sum_{t \in \mathcal{T}} x_{ki}^t \leq 1 \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \tag{20}$$

$$\sum_{t \in \mathcal{T}} t x_{k,i+1}^t = \sum_{t \in \mathcal{T}} t x_{ki}^t + s_i^k l_{i,i+1} + w_{i+1}^k \quad i \in \mathcal{S} \setminus \{n\}, \quad k \in \mathcal{M} \tag{21}$$

$$w_{min} \sum_{t \in \mathcal{T}} x_{ki}^t \leq w_i^k \leq w_{max} \sum_{t \in \mathcal{T}} x_{ki}^t \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \setminus \{m\} \tag{22}$$

$$s_{min} \sum_{t \in \mathcal{T}} x_{ki}^t \leq s_i^k \leq s_{max} \sum_{t \in \mathcal{T}} x_{ki}^t \quad i \in \mathcal{S} \setminus \{n\}, \quad k \in \mathcal{M} \tag{23}$$

$$\sum_{t \in \mathcal{T}} t x_{k+1,i}^t \geq \sum_{t \in \mathcal{T}} t x_{ki}^t + h_{min} \sum_{t \in \mathcal{T}} x_{ki}^t \quad i \in \mathcal{S}, \quad k \in \mathcal{M} \setminus \{m\} \tag{24}$$

$$x_{ki}^t \in \{0, 1\} \quad i \in \mathcal{S}, \quad k \in \mathcal{M}, \quad t \in \mathcal{T} \tag{25}$$

$$f_i^t, u_i^t \geq 0 \quad i \in \mathcal{S}, \quad t \in \mathcal{T} \tag{26}$$

$$s_i^k, w_i^k \geq 0 \quad i \in \mathcal{S}, \quad k \in \mathcal{M}. \tag{27}$$

In this formulation, the objective function (15) minimizes the total waiting time of passengers. Constraints (16) define the flow conservation of passengers waiting at station i at time t. It is defined as the sum of passengers who were already at the station from previous periods, plus all passengers who arrived at station i at time t destined for other stations, minus the number of passengers who boarded the train leaving station i at time t. Constraints (17) define upper bounds on the number of passengers boarding a train. In particular, if no train is launched at time t, the flow of passengers boarding the train is zero. If a train is launched, the right-hand side of constraints (17) constitutes an upper bound on the number of passengers boarding the train. Note that capacities can easily be introduced in these constraints. From constraints (17) and (26), u_i^t is zero if no train is assigned to leave station i at time t. Constraints (18) order the trains by their indices. Again, this helps break the symmetry of this formulation. Constraints (19) limit the number of available trains. Constraints (20) ensure that a train is launched at most once. Constraints (21) ensure that if a train is launched from station i at time t, then this train has to be launched from station $i+1$ at time t, plus the travel time from i to $i+1$, plus the dwell time at station $i+1$. Constraints (22) set the thresholds for dwell times of launched trains. Similarly, constraints (23) set the thresholds for the inverse of speed of launched trains. Constraints (24) establish the relationship between launch times of two consecutive trains at station i, ensuring that the minimum headway h_{min} is respected. Finally, constraints (25)$-$(27) enforce integrality and non-negativity conditions on the variables.

In order to reduce the computational effort needed to solve this model exactly, we propose three sets of valid inequalities which tighten the formulation. The first set reduces the number of variables of the problem by effectively setting to zero all binary variables x_{ki}^t for which the launch time t from station i would not allow any train k to arrive to the last station n before the end of the planning horizon:

$$x_{ki}^t = 0 \quad i \in \mathcal{S}\backslash\{n\} \quad k \in \mathcal{M} \quad t \in \mathcal{T} \cap [p - (s_{min} l_{in} + (n-i-1)w_{min}), p]. \qquad (28)$$

The second set ensures that if a train k is launched from station i, it is also launched from next station $i+1$:

$$\sum_{t \in \mathcal{T}} x_{ki}^t = \sum_{t \in \mathcal{T}} x_{k,i+1}^t \quad i \in \mathcal{S}\backslash\{n\} \quad k \in \mathcal{M}. \qquad (29)$$

Finally, the third set limits the launch time of any train k from station j to the launch time of the same train k from any previous station $i < j$, considering the minimum and maximum times required to travel from station i to station j:

$$\sum_{t\in\mathcal{T}} tx_{kj}^t \leq \sum_{t\in\mathcal{T}} tx_{ki}^t + s_{max}l_{ij} + (j-i)w_{max} \quad i,j \in \mathcal{S}\backslash\{n\} \quad i<j \quad k\in\mathcal{M} \tag{30}$$

$$\sum_{t\in\mathcal{T}} tx_{ki}^t + s_{min}l_{ij} + (j-i)w_{min} \leq \sum_{t\in\mathcal{T}} tx_{kj}^t \quad i,j \in \mathcal{S}\backslash\{n\} \quad i<j \quad k\in\mathcal{M}. \tag{31}$$

3.3 Linear Formulation 2

This formulation is similar to LF1 in that it also uses flow variables and binary x_{ki}^t variables representing launch times. However, in LF2 we disregard speed and waiting times variables. We have observed that the previous formulation contains a number of symmetries since the same launch time at a station $i+1$ can be given by the launch time at previous station i and for different combinations of arrival and waiting times at station $i+1$. This leads to symmetric solutions which can be avoided by not considering the speeds s_i^k and waiting times w_i^k of each train k, but introducing instead integer variables v_i^k representing the amount of time elapsed between the launch times of train k from stations i and $i+1$. The formulation is as follows:

$$\text{(LF2)} \quad \text{minimize} \quad \delta \sum_{i\in\mathcal{S}\backslash\{n\}} \sum_{t\in\mathcal{T}\backslash\{p\}} f_i^t \tag{32}$$

subject to (16)−(20), (24)−(26), and to

$$\sum_{t\in\mathcal{T}} tx_{k,i+1}^t = \sum_{t\in\mathcal{T}} tx_{ki}^t + v_i^k \quad i\in\mathcal{S}\backslash\{n\} \quad k\in\mathcal{M} \tag{33}$$

$$v_i^k \leq (s_{max}l_{i,i+1} + w_{max}) \sum_{t\in\mathcal{T}} x_{ki}^t \quad i\in\mathcal{S}\setminus\{n\}, \quad k\in\mathcal{M} \tag{34}$$

$$(s_{min}l_{i,i+1} + w_{min}) \sum_{t\in\mathcal{T}} x_{ki}^t \leq v_i^k \quad i\in\mathcal{S}\setminus\{n\}, \quad k\in\mathcal{M}. \tag{35}$$

In this formulation, the objective function (32) is the same as (15). Constraints (33) ensure that if a train is launched from station i at time t, then this train has to be launched from station $i+1$ at time t, plus the time v_i^k elapsed between the launch times of train k from stations i and $i+1$. Finally, constraints (34) and (35) enforce the bounds on the v_i^k variables according to minimum and maximum speeds and waiting times.

3.4 Linear Formulation 3

The third linear formulation is similar to LF2, but is even more streamlined with respect to the variables linking the launch times from two consecutive stations. Here, the launch times at stations are constrained to lie within the interval in which a service may be launched in order to yield a feasible solution. The formulation is as follows:

$$\text{(LF3)} \quad \text{minimize} \quad \delta \sum_{i \in \mathcal{S} \backslash \{n\}} \sum_{t \in \mathcal{T} \backslash \{p\}} f_i^t \tag{36}$$

subject to (16)−(20), (24)−(26), and to

$$\sum_{t \in \mathcal{T}} t x_{k,i+1}^t \leq \sum_{t \in \mathcal{T}} t x_{ki}^t + s_{max} l_{i,i+1} + w_{max} \quad i \in \mathcal{S} \backslash \{n\} \quad k \in \mathcal{M} \tag{37}$$

$$\sum_{t \in \mathcal{T}} t x_{ki}^t + s_{min} l_{i,i+1} + w_{min} \leq \sum_{t \in \mathcal{T}} t x_{k,i+1}^t \quad i \in \mathcal{S} \backslash \{n\} \quad k \in \mathcal{M}. \tag{38}$$

In this linear formulation, the objective function (36) is again the same as (15). Constraints (37) and (38) enforce the bounds on the launch time from station $i+1$ according to the launch time from station i, and the minimum and maximum speeds and waiting times.

4 Branch-and-cut algorithm

We have implemented a branch-and-cut algorithm capable of solving all formulations. All variables and constraints of these formulations are explicitly handled by the algorithm. Given the large number of valid inequalities, these are not explicitly included in the initial subproblem, but are rather dynamically generated as cuts. These formulations can then be solved by branch-and-cut as follows. At a generic node of the search tree, a linear program with relaxed integrality constraints is solved, a search for violated constraints is performed, and violated valid inequalities are added to the current program which is then reoptimized. This process is reiterated until a feasible or dominated solution has been reached, or until no more cuts can be added. At this point branching on a fractional variable occurs. Let z^* represent the cost of best known solution. In Algorithm 1, we provide a sketch of the branch-and-cut scheme for all formulations.

As in other papers comparing a timetable with a benchmark solution, such as the ideal plans used in Caprara et al. [9] and Cacchiani et al. [3], we focus on obtaining improvements with respect to a regular timetable with the same number of trains, which can be instantaneously generated. We therefore provide the algorithm with a regular timetable as an initial solution. However, this feature is not essential to our algorithm.

5 Computational experiments

We now detail the computational experiments performed to evaluate our models and algorithm. All computations were carried out on a grid of Intel Xeon$^{\text{TM}}$ processors running at 2.66 GHz with up to 24 GB of RAM installed per node, with the Scientific Linux 6.1 operating system. A single thread was used. The algorithm just described were coded in C++ and we use IBM Concert Technology and CPLEX 12.5 as the MIP solver.

We describe in Section 5.1 the real data we received from Madrid Metropolitan Railway and how we have designed a set of instances to evaluate our models based on these data. In Section 5.2 we describe the results of our experiments on artificial instances and we compare the performance of each formulation between themselves and with respect to a regular timetable. In Section 5.3 we analyze the impact of

Algorithm 1 Branch-and-cut algorithm

At the root node of the search tree, generate and insert all variables and constraints (16)−(27) into the program.
Input a feasible incumbent solution of cost z^*.
Termination check:
if there are no more nodes to evaluate **then**
  Stop with the incumbent and optimal solution of cost z^*.
else
  Select one node from the branch-and-bound tree.
end if
Subproblem solution: solve the LP relaxation of the node and let z be its cost.
if the current solution is feasible **then**
  if $z \geq z^*$ **then**
    Go to termination check.
  else
    $z^* \leftarrow z$.
    Update the incumbent solution.
    Prune nodes with lower bound larger than or equal to z^*.
    Go to termination check.
  end if
end if
Cut generation:
if the solution of the current LP relaxation violates any cuts **then**
  Add violated cuts as new constraints.
  Go to subproblem solution.
end if
Branching: branch on one of the fractional variables.
Go to the termination check.

using optimized solutions compared to a regular timetable, on the real instances obtained from our partner.

5.1 Real data and instances generation

We have received real data from Line C5 of Madrid Metropolitan Railway. Although demand patterns significantly differ from one instance to another, an important common feature observed in real-world settings is the existence of demand peaks at different times. Based on this observation and on the shape of the demand functions obtained from our partner, we have generated a set of benchmark instances with a variable demand scenario. In particular, the demand functions are generated for each pair of stations, and their cumulative representation is composed of the sum of one or more S-shaped functions, or sigmoid curves.

The 36 instances whose demand data are provided by Madrid Metropolitan Railway contain six stations and between 200 and 1200 minutes.

In addition, we have created a set of 90 artificial instances which vary in terms of the number of stations, time horizon, discretization constant and number of trains. These instances were generated according to the following parameters:

- number of stations n: 3, 6, 10;
- horizon p: 200, 400, 600, 800, 1000 minutes;
- discretization constant δ: 1, 2, 4 minutes;
- maximum number of trains m: 5, 10;
- maximum inverse speed of the trains s_{min}: 0.0015 min/m (speed = 40 km/h);
- minimum inverse speed of the trains s_{max}: 0.00075 min/m (speed = 80 km/h);
- minimum headway h_{min}: 12 minutes;
- minimum stopping time at the stations w_{min}: 4 minutes;

- maximum stopping time at the stations w_{max}: 12 minutes.

These instances will be referred to as TT-n-p-δ-m, e.g., TT-3-800-2-10, corresponding to a train timetabling instance with three stations, a planning horizon of 800 minutes, discretization constant of two minutes, and a maximum of 10 trains. For the instances involving six and 10 stations, we do not consider a planning horizon 200 minutes since the regular case was infeasible for certain number of trains and discretization constants. Instead, instances of 1200 minutes are included. For all cases, a maximum running time of three hours was imposed. The set of instances as well as their solutions are available on `http://www.leandro-coelho.com`.

5.2 Computational results

In our preliminary computational experiments, we have observed that the non-linear formulation performed much worse than any of the linear ones. For this reason, in what follows we focus only on the three linear formulations we have presented.

Not all instances could be solved within the time limit. For instances that could not be solved optimally, we recorded the best AWT generated by the algorithm. We present in Table 1 a summary of the AWT per passenger over all instances. The columns represent the different cases of three, six and 10 stations, as well as the real case of the line C5 of Madrid Metropolitan Railway, and the rows present the regular timetabling and all linear formulations LF1, LF2 and LF3. It can be observed that for the three linear formulations presented, the AWT considerably improves with respect to a regular timetable. As expected, LF3 yields the best results, followed by LF2 and LF1. The only case where this order is not observed is when the are three stations, where LF2 yields a better AWT improvement than LF3. However, as we show in the following experiments, LF2 requires longer computational times that LF3. For these reasons, we consider LF3 to be the most effective formulation in terms of obtaining better solutions in shorter computational times.

For the case of three stations, we provide in Tables 2−4 the upper and lower bounds

Table 1: Summary of the average waiting times (AWT)

	3 stations	6 stations	10 stations	C5	Average	% AWT improvement over regular
Regular	45.19	60.07	60.56	40.07	51.47	-
LF1	10.60	41.80	56.08	37.60	36.52	29.05
LF2	10.28	38.34	51.08	37.12	34.21	33.53
LF3	**10.54**	**32.09**	**49.18**	**36.25**	**32.02**	**37.79**

on the objective function, the percentage optimality gap, the running time in seconds, and the AWT per passenger for each formulation. For the instances where the optimal solution was reached, the AWT is obviously the same in all formulations, but even for the cases where it is not reached, all formulations improve the AWT significantly with respect to the regular case.

It can be observed that all the instances parameters have an effect on the resolution difficulty. The number of binary variables x^t_{ki} increases with the number of trains, the number of stations and the length of the planning horizon, the latter being the most crucial parameter. The tables show that when the planning horizon is longer, fewer instances are solved optimally and the gap for the unsolved instances is larger. This effect tends to diminish when δ increases. Thus, a less fine discretization (a larger value of δ) enables instances with a larger planning horizon to be solved optimally. Tables 2$-$4 show that for a fixed number of three stations and five trains, optimality is reached for an instance with 200 minutes when $\delta = 1$, whereas it is obtained for 400 minutes when $\delta = 2$ and for 1000 minutes when $\delta = 4$. The drawback of increasing δ is that the AWT also goes up, which was expected since the number of possible launch times for trains is smaller and thus, the feasibility space is reduced.

As stated above, the number of stations is one of the parameters affecting the difficulty of resolution. For instances with six and 10 stations the optimum is not reached but still, the improvement with respect to the regular timetable is important. Figures 2 and 3 show the percentage improvement yielded by formulations LF1, LF2 and LF3 with respect to the regular case for each of the instances considered. Again, it can be observed that LF3 yields the best improvements, not only on average as was shown in Table 1, but also for most of the instances. The second best formulation is LF2 and the worst is LF1.

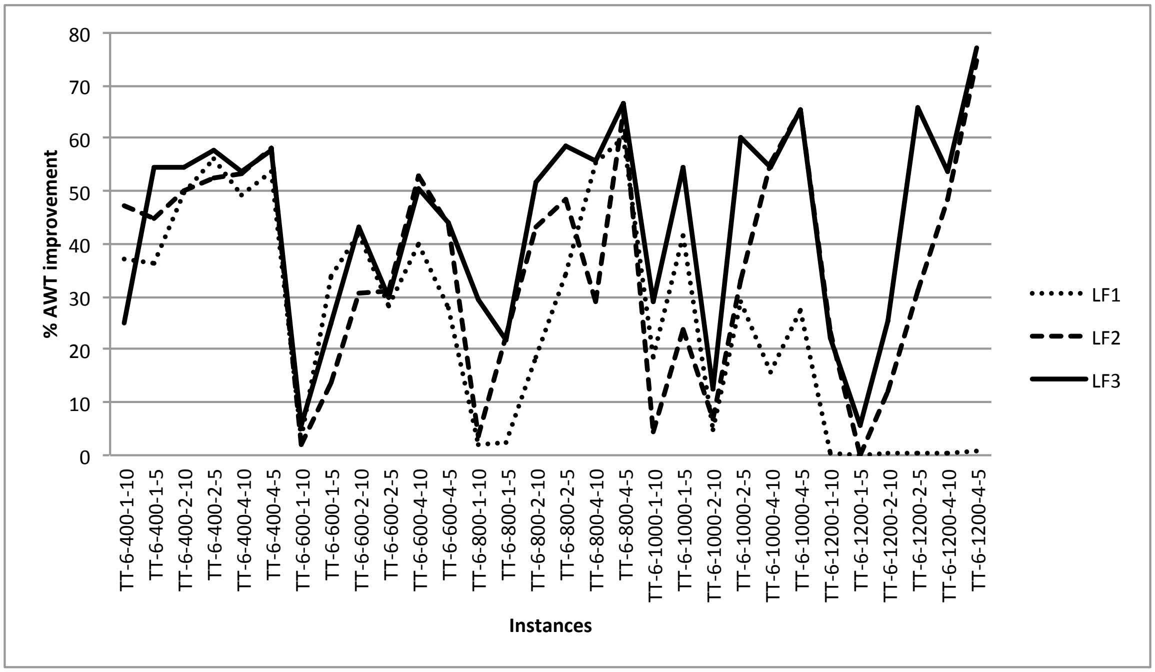


Figure 2: Percentage improvement of the AWT on instances with six stations with respect to a regular timetable

5.3 Computational results for the case of Line C5 of Madrid Metropolitan Railway

We now provide results for the instances obtained from the line C5 of Madrid Metropolitan Railway. We have run all the three formulations on the 36 real instances. We present in Table 5 the AWT per passenger when applying a regular timetable, as well as the corresponding values obtained with our formulations. Once again, it is clear that all three formulations yield better results than a regular timetable, and that LF3 is the best among them, with an average improvement of 7.64% over the regular case, of 3.37% over LF1 and of 2.28% over LF2. On some

Table 2: Summary of computational results of LF1 on instances with three stations

Instance	Regular AWT	UB	LB	Gap (%)	Time(s)	AWT
TT-3-200-1-5	19.39	37368	37368	0.00	831	6.27
TT-3-200-2-5	18.51	35210	35210	0.00	28	6.90
TT-3-200-4-5	15.81	29428	29428	0.00	3	6.93
TT-3-400-1-5	44.05	102418	55524	45.79	10800	9.08
TT-3-400-2-5	43.50	98412	98412	0.00	1974	9.72
TT-3-400-4-5	39.75	87260	87260	0.00	67	9.73
TT-3-600-1-5	44.99	167175	21111	87.37	10801	12.29
TT-3-600-2-5	43.33	162548	100065	38.44	10800	12.94
TT-3-600-4-5	39.44	150748	150748	0.00	1276	13.07
TT-3-800-1-5	87.36	290418	23753	91.82	10801	17.04
TT-3-800-2-5	89.26	284762	48520	82.96	10801	17.70
TT-3-800-4-5	97.84	270468	270468	0.00	3473	17.85
TT-3-1000-1-5	112.70	453416	18664	95.88	10802	26.60
TT-3-1000-2-5	111.55	294992	66854	77.34	10800	18.30
TT-3-1000-4-5	108.02	283516	283516	0.00	2532	18.62
TT-3-200-1-10	6.48	28961	28961	0.00	392	4.86
TT-3-200-2-10	8.17	26366	26366	0.00	36	5.42
TT-3-200-4-10	8.17	20744	20744	0.00	6	5.48
TT-3-400-1-10	22.48	60011	27737	53.78	10801	5.32
TT-3-400-2-10	19.24	55714	54168	2.77	10800	5.94
TT-3-400-4-10	19.24	45592	45592	0.00	516	6.04
TT-3-600-1-10	24.47	94637	15816	83.29	10801	6.95
TT-3-600-2-10	21.75	83130	41669	49.88	10801	7.11
TT-3-600-4-10	21.75	70584	55794	20.95	10801	7.18
TT-3-800-1-10	40.43	142836	18144	87.30	10803	8.38
TT-3-800-2-10	43.65	145422	25949	82.16	10801	9.53
TT-3-800-4-10	43.66	124388	55067	55.73	10801	9.29
TT-3-1000-1-10	55.08	213994	19106	91.07	10804	12.55
TT-3-1000-2-10	52.76	175016	24773	85.85	10802	11.26
TT-3-1000-4-10	52.76	131468	52491	60.07	10801	9.71
Average	45.19	138900	61309	39.75	6852	**10.60**

Table 3: Summary of computational results of LF2 on instances with three stations

Instance	Regular AWT	UB	LB	Gap (%)	Time(s)	AWT
TT-3-200-1-5	19.39	37368	37368	0.00	817	6.27
TT-3-200-2-5	18.51	35210	35210	0.00	37	6.90
TT-3-200-4-5	15.81	29428	29428	0.00	4	6.93
TT-3-400-1-5	44.05	102492	55905	45.45	10801	9.09
TT-3-400-2-5	43.50	98412	98412	0.00	809	9.72
TT-3-400-4-5	39.75	87260	87260	0.00	58	9.73
TT-3-600-1-5	44.99	168335	33204	80.28	10800	12.37
TT-3-600-2-5	43.33	162008	62634	61.34	10801	12.90
TT-3-600-4-5	39.44	150748	150748	0.00	1011	13.07
TT-3-800-1-5	87.36	298220	24552	91.77	10801	17.49
TT-3-800-2-5	89.26	282996	37555	86.73	10801	17.60
TT-3-800-4-5	97.84	270468	270468	0.00	3106	17.85
TT-3-1000-1-5	112.70	331849	22256	93.29	10802	19.47
TT-3-1000-2-5	111.55	294992	51674	82.48	10802	18.30
TT-3-1000-4-5	108.02	283516	283516	0.00	3478	18.62
TT-3-200-1-10	6.48	28961	28961	0.00	639	4.86
TT-3-200-2-10	8.17	26366	26366	0.00	48	5.42
TT-3-200-4-10	8.17	20744	20744	0.00	4	5.48
TT-3-400-1-10	22.48	60295	27894	53.74	10801	5.35
TT-3-400-2-10	19.24	55714	53835	3.37	10801	5.94
TT-3-400-4-10	19.24	45592	45592	0.00	211	6.04
TT-3-600-1-10	24.47	92664	17812	80.78	10802	6.81
TT-3-600-2-10	21.75	85330	36284	57.48	10801	7.27
TT-3-600-4-10	21.75	70460	70460	0.00	6859	7.17
TT-3-800-1-10	40.43	144628	17111	88.17	10803	8.48
TT-3-800-2-10	43.65	134594	34329	74.49	10801	8.89
TT-3-800-4-10	43.66	123356	59354	51.88	10801	9.23
TT-3-1000-1-10	55.08	162942	14851	90.89	10805	9.56
TT-3-1000-2-10	52.76	186518	23879	87.20	10801	11.94
TT-3-1000-4-10	52.76	131524	51637	60.74	10800	9.71
Average	45.19	133433	60310	39.67	6690	**10.28**

Table 4: Summary of computational results of LF3 on instances with three stations

Instance	Regular AWT	UB	LB	Gap (%)	Time(s)	AWT
TT-3-200-1-5	19.39	37368	37368	0.00	366	6.27
TT-3-200-2-5	18.51	35210	35210	0.00	28	6.90
TT-3-200-4-5	15.81	29428	29428	0.00	1	6.93
TT-3-400-1-5	44.05	102418	52149	49.08	10800	9.08
TT-3-400-2-5	43.50	98412	98412	0.00	476	9.72
TT-3-400-4-5	39.75	87260	87260	0.00	35	9.73
TT-3-600-1-5	44.99	167941	31144	81.46	10801	12.34
TT-3-600-2-5	43.33	162008	162008	0.00	7979	12.90
TT-3-600-4-5	39.44	150748	150748	0.00	906	13.07
TT-3-800-1-5	87.36	290061	29381	89.87	10802	17.01
TT-3-800-2-5	89.26	282996	65078	77.00	10801	17.60
TT-3-800-4-5	97.84	270468	270468	0.00	3480	17.85
TT-3-1000-1-5	112.70	393820	25130	93.62	10802	23.10
TT-3-1000-2-5	111.55	294992	50677	82.82	10801	18.30
TT-3-1000-4-5	108.02	283516	283516	0.00	4627	18.62
TT-3-200-1-10	6.48	28961	28961	0.00	153	4.86
TT-3-200-2-10	8.17	26366	26366	0.00	18	5.42
TT-3-200-4-10	8.17	20744	20744	0.00	2	5.48
TT-3-400-1-10	22.48	59715	29887	49.95	10801	5.29
TT-3-400-2-10	19.24	55714	54734	1.76	10800	5.94
TT-3-400-4-10	19.24	45592	45592	0.00	140	6.04
TT-3-600-1-10	24.47	99034	17418	82.41	10801	7.28
TT-3-600-2-10	21.75	83512	43803	47.55	10801	7.13
TT-3-600-4-10	21.75	70460	70460	0.00	8649	7.17
TT-3-800-1-10	40.43	167218	19581	88.29	10803	9.81
TT-3-800-2-10	43.65	168932	29814	82.35	10801	10.91
TT-3-800-4-10	43.66	127528	54541	57.23	10801	9.47
TT-3-1000-1-10	55.08	205169	16550	91.93	10805	12.03
TT-3-1000-2-10	52.76	158180	26658	83.15	10801	10.28
TT-3-1000-4-10	52.76	130460	53162	59.25	10801	9.65
Average	45.19	137808	64875	37.26	6656	**10.54**

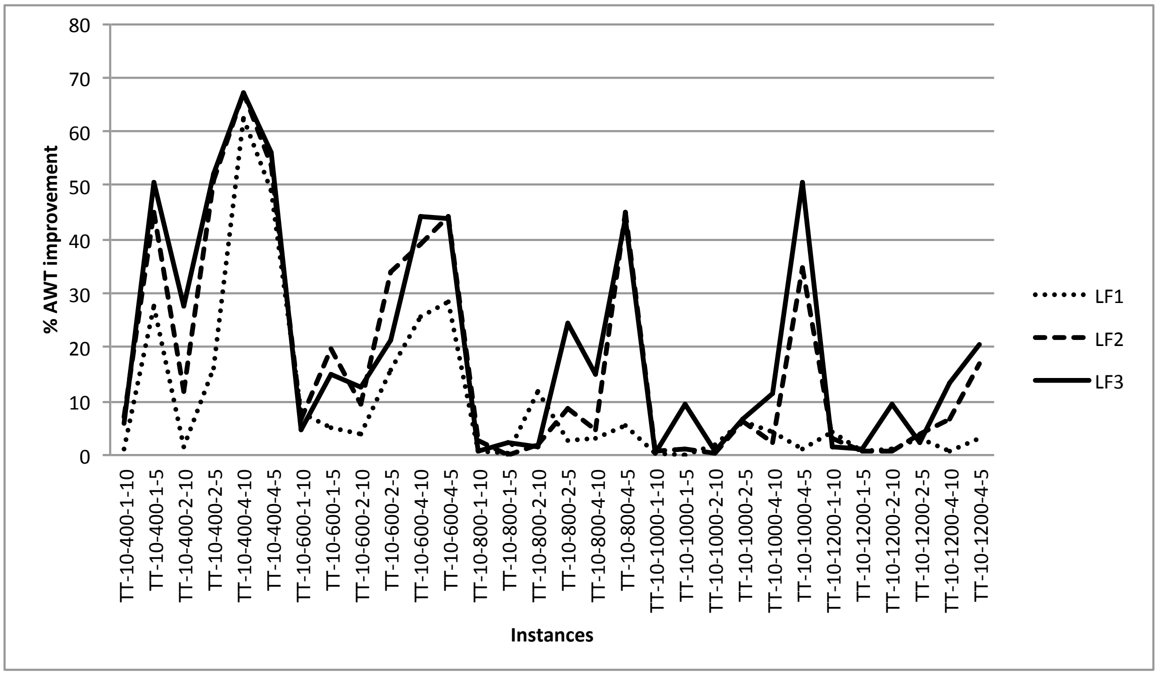


Figure 3: Percentage improvement of the AWT on instances with 10 stations with respect to a regular timetable

particular instances, the improvement yielded by all formulations with respect to regular timetable is over 50%.

In most cases the algorithm could not prove optimality within the allotted three hours of running time. For the instances where the optimal solution was proved, LF3 yields the shortest computing times, followed by LF2 and LF1. Due to the similarity of the upper bounds, we believe that the solutions we report are close to the optimum and that the larger gaps are probably due to the weakness of the lower bounds. For this reason, we have also fine tuned the CPLEX solver in order to obtain better lower bounds by setting appropriate parameters such as preprocessing, reductions, presolving, aggressive cuts generation, branching direction, node selection, etc, but these bounds did not improve significantly.

Table 5: Summary of computational results on instances obtained from the line C5 of Madrid Metropolitan Railway

Instance	AWT Regular	AWT LF1	AWT LF2	AWT LF3
$C5$-200-1-5	18.66	13.18	12.80	12.70
$C5$-200-2-5	20.21	13.93	13.68	13.68
$C5$-200-4-5	23.12	14.42	14.42	14.42
$C5$-400-1-5	32.04	28.83	28.17	28.07
$C5$-400-2-5	32.27	29.29	29.95	28.87
$C5$-400-4-5	32.17	29.51	29.40	29.33
$C5$-600-1-5	49.99	49.70	49.94	45.87
$C5$-600-2-5	50.26	50.25	46.66	44.74
$C5$-600-4-5	49.87	48.12	44.30	43.94
$C5$-800-1-5	69.22	69.22	68.91	69.22
$C5$-800-2-5	69.08	68.86	68.20	61.77
$C5$-800-4-5	68.03	64.22	62.65	61.57
$C5$-1000-1-5	84.98	84.90	84.98	84.94
$C5$-1000-2-5	85.44	83.83	83.78	78.77
$C5$-1000-4-5	84.63	80.92	80.95	76.83
$C5$-1200-1-5	109.00	109.00	109.00	109.00
$C5$-1200-2-5	108.70	108.57	107.78	108.70
$C5$-1200-4-5	107.73	107.70	106.48	96.02
$C5$-200-1-10	17.01	8.99	8.21	8.23
$C5$-200-2-10	20.73	9.82	9.44	9.44
$C5$-200-4-10	20.70	9.93	9.93	9.92
$C5$-400-1-10	17.56	17.51	16.32	16.37
$C5$-400-2-10	18.96	18.54	17.83	17.82
$C5$-400-4-10	18.98	17.40	16.59	16.45
$C5$-600-1-10	25.73	25.53	25.73	25.73
$C5$-600-2-10	26.26	26.26	26.23	25.31
$C5$-600-4-10	26.30	25.34	25.74	25.38
$C5$-800-1-10	35.13	35.13	35.13	35.13
$C5$-800-2-10	35.34	35.28	35.19	35.34
$C5$-800-4-10	35.44	35.41	34.77	34.00
$C5$-1000-1-10	44.44	44.44	44.44	44.44
$C5$-1000-2-10	44.68	44.68	44.68	44.68
$C5$-1000-4-10	44.79	44.67	44.57	44.57
$C5$-1200-1-10	54.50	54.50	54.50	54.50
$C5$-1200-2-10	54.43	54.43	54.43	54.43
$C5$-1200-4-10	54.63	53.75	52.13	51.62
Average	46.97	44.89	44.39	**43.38**

6 Conclusions

We have proposed four formulations for the train timetabling problem adapted to a dynamic demand pattern. The first one consists of a mixed integer non-linear programming model, while the other three are linear. The latter models clearly dominate the non-linear one. This was made possible by the introduction of flow variables, which allow a linear representation of the objective function. We have developed a branch-and-cut algorithm applicable to all models. Through extensive computational experiments on the real instances and on several randomly generated instances, we have shown that the three linear formulations yield improvements in the average waiting time per passenger with respect to the regular case of around 30% on average and 77% in the best case. We have observed that aggregating speed and waiting time variables into a single travel time variable helps generate better results in terms of the objective function value and computational time. These improvements are particularly important and hold even when the new travel time variable is eliminated and one controls launch times directly. A byproduct of this study is the generation of a large set of benchmark instances which are made available to the research community.

References

[1] M. R. Bussieck, T. Winter, and U. T. Zimmermann. Discrete optimization in public rail transport. *Mathematical Programming*, 79(1–3):415–444, 1997.

[2] V. Cacchiani, A. Caprara, and P. Toth. A column generation approach to train timetabling on a corridor. *4OR*, 6(2):125–142, 2008.

[3] V. Cacchiani, A. Caprara, and P. Toth. Non-cyclic train timetabling and comparability graphs. *Operations Research Letters*, 38(3):179–184, 2010.

[4] V. Cacchiani, A. Caprara, and P. Toth. Scheduling extra freight trains on

railway networks. *Transportation Research Part B: Methodological*, 44(2):215–231, 2010.

[5] X. Cai, C. J. Goh, and A. I. Mees. Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transactions*, 30:481–493, 1998.

[6] D. Canca, A. Zarzo, E. Algaba, and E. Barrena. Confrontation of different objectives in the determination of train scheduling. *Procedia Social and Behavioral Sciences*, 20:302–312, 2011.

[7] D. Canca, E. Barrena, E. Algaba, and A. Zarzo. Design and analysis of demand-adapted railway timetables. *Available online at arXiv:1307.0970 [math.OC]*, 2013.

[8] A. Caprara, M. Fischetti, and P. Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50(5):851–861, 2002.

[9] A. Caprara, M. Monaci, P. Toth, and P. L. Guida. A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, 154(5):738–753, 2006.

[10] A. Caprara, M. Monaci, P. Toth, and Guida. P. L. A Lagrangian heuristic approach to real-world train timetabling problems. *Discrete Applied Mathematics*, 154:738–753, 2006.

[11] A. Chierici, R. Cordone, and R. Maja. The demand-dependent optimization of regular train timetables. *Electronic Notes in Discrete Mathematics*, 17:99–104, 2004.

[12] J.-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404, 1998.

[13] R. Cordone and F. Redaelli. Optimizing the demand captured by a railway system with a regular timetable. *Transportation Research B*, 45(2):430–446, 2011.

[14] V. Guihaire and J. K. Hao. Transit network design and scheduling: A global review. *Transportation Research A*, 42:1251–1273, 2008.

[15] D. Huisman, L. G. Kroon, R. M. Lentink, and M. J. C. M. Vromans. Operations research in passenger railway transportation. *Statistica Neerlandica*, 59(7):467–497, 2005.

[16] L. Ingolotti, A. Lova, F. Barber, P. Tormos, M. A. Salido, and M. Abril. New heuristics to solve the CSOP railway timetabling problem. In M. Ali and R. Dapoigny, editors, *Advances in Applied Artificial Intelligence*, volume 4031 of *Lecture Notes in Artificial Intelligence*, pages 400–409. Springer. Berlin Heidelberg, 2006.

[17] L. G. Kroon, R. Dekker, and M. J. C. M. Vromans. Cyclic railway timetabling: A stochastic optimization approach. In F. Geraets, L. G. Kroon, A. Schoebel, D. Wagner, and C. D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes on Computer Science*, pages 41–66. Springer. Berlin Heidelberg, 2007.

[18] L. G. Kroon, G. Marti, M. R. Helmrich, M. J. C. M. Vromans, and R. Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research B*, 42(6):553–570, 2008.

[19] L. G. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, A. Schrijver, A. Steenbeek, and R. Ybema. The new Dutch timetable: The OR revolution. *Interfaces*, 39(1):6–17, 2009.

[20] R. C. Larson and A. R. Odoni. *Urban Operations Research*. Prentice-Hall. New York, 1981.

[21] C. Liebchen and R. Möhring. A case study in periodic timetabling. *Electronic Notes in Theoretical Computer Science*, 66(6):1–14, 2002.

[22] C. Liebchen and L. Peeters. Integral cycle bases for cyclic timetabling. *Discrete Optimization*, 6(1):89–109, 2009.

[23] K. Nachtigall. A branch and cut approach for periodic network programming. Technical Report 29, Hildesheimer Informatik-Berichte, Universität Hildesheim, 1994.

[24] K. Nachtigall and S. Voget. A genetic algorithm approach to periodic railway synchronization. *Electronic Notes in Theoretical Computer Science*, 23(5):453–463, 1996.

[25] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

[26] M. Voorhoeve. Rail scheduling with discrete sets. Technical report, Eindhoven University of Technology, The Netherlands, 1993.