

Demand-Driven Timetabling for a Metro Corridor Using a Short-Turning Acceleration Strategy

Tommaso Schettini

Politecnico di Milano, tommaso.schettini@polimi.it,

Ola Jabali

Politecnico di Milano, ola.jabali@polimi.it,

Federico Malucelli

Politecnico di Milano, federico.malucelli@polimi.it,

The efficient management of metro lines is a major concern for public transport operators. Traditionally, metro lines are operated through regular timetables, i.e., timetables where trains have a constant headway between all stations. In this paper, we propose a demand-driven metro timetabling strategy, and elaborate exact solution methods for the case of a two-directional metro corridor. In doing so, we avoid imposing any predetermined structure to the timetable, and instead control the trains individually to best match passenger demand. We consider that trains may short-turn, i.e., trains are not required to serve the line from terminal to terminal, but instead, may reverse direction before reaching the terminal. We present a MILP formulation for the demand-driven timetabling problem of a two-directional metro corridor with short-turning. Furthermore, we develop an efficient exact algorithm using cut generation for an alternative formulation with an exponential number of constraints, and derive two classes of valid inequalities. We evaluate the proposed formulation and algorithm considering seven possible cut generation strategies on a number of test instances from artificially generated lines and on two test-beds derived from real-world lines. Through the computational experiments, we demonstrate the effectiveness of the developed algorithm and the added value of the proposed strategy in terms of passengers' waiting time.

Key words: Metro timetabling; Short-turning; Demand-driven

1. Introduction

With the recent growth of urban populations (UN 2015), efficiently managing transit systems has become a top priority for local authorities. Traditionally, metro timetabling is performed mainly at a tactical planning level. Initially, each day of the year is categorized under a set of typical days, e.g., winter-workday, winter-festive, summer-weekend. Such categorization aims at capturing the observed demand variations throughout the year. Each day is further divided into several time periods such as morning peak, off-peak, etc. Then, for each time period, for each typical day, a service frequency is determined to guarantee a certain level of service while minimizing costs. For each time period, the demand is assumed constant. Then, based on these frequencies, a regular

train timetable is established, which is then staffed by optimizing appropriate crew scheduling problems. This tends to produce highly regular timetables, which are generally effective in practice, but rather inflexible since the structure of the timetable is fixed. In recent years, some research has focused on developing demand-driven metro timetables (Sun et al. 2014, Barrena et al. 2014a,b). The models related to such settings frequently forego the periodicity assumption on the train timetabling in favor of more flexible timetables, which are tailored to passenger demand. In this paper, we investigate a demand-driven metro timetabling problem, and elaborate solution methods for the case of a two-directional metro corridor. We refer to the problem addressed in this paper as the Demand-Driven Timetabling Problem (DDTP). In particular, we propose explicitly matching the train timetabling to the line’s demand, so as to minimize the total waiting time experienced by the passengers. In doing so, the trains are individually controlled, without following predetermined frequencies or timetable structure.

To further improve the service, we also consider the possibility of using a short-turning acceleration policy. This implies that trains are not required to serve the line from terminal to terminal, but instead, may reverse their direction before reaching the terminal of the line. By doing so, one may shape the timetable to better fit the distribution of demand on the line, offering increased frequencies to high-demand segments. Short-turning is a rather common acceleration strategy in transit systems, in particular in a bus context (Furth 1987). A survey of operational practices in metro systems, conducted by Schmöcker, Cooper, and Adeney (2005) reports that over 80% of the surveyed metro lines utilize short turning, to some extent, to handle service disruptions. However, using short-turning in the daily timetable of the metro lines, as considered in this paper, is less common.

The concepts of explicitly optimizing train timetables to match passenger demand while allowing trains to short-turn are rather broad, and may be applied to other transit modes (e.g., bus lines). Nonetheless, we believe that, at a preliminary level, these concepts fit a metro line setting. This is due to the fact that metro lines tend to be operated at high frequencies, and as a result, passengers generally do not plan the specific service they intend to board, but instead simply reach the station and wait for the next service. Thus, we refer to a metro optimization problem setting in this paper. However, we note that the resulting timetables may complicate crew scheduling. Therefore, applying a demand driven timetables using short-turning suits automated lines since these line would not require the scheduling of drivers.

Similarly to Sun et al. (2014), Barrena et al. (2014a,b), we consider a given demand profile with the objective of minimizing the total waiting time experienced by the passengers over a finite planning horizon. Controlling the trains individually while allowing them to short-turn, as entailed by the DDTP, yields a rather complex problem. The purpose of this paper is to devise an exact

algorithm for the DDTP, and to evaluate the potential added value of the resulting timetables, compared to a traditional approach. Therefore, we address the problem from a system-optimum perspective, whereby we assume to have prior knowledge of the demand of the line. This is in line with other demand-driven timetabling approaches, where the demand is forecasted from historical data. In practice, such data is frequently collected by an automatic fare collection system, which is generally unsuitable to provide real-time demand data for a line.

The contributions of this paper are as follows: 1) Proposing a flow-based formulation and a formulation with an exponential number of constraints for the DDTP. 2) Developing an efficient exact cut generation algorithm and two classes of valid inequalities for the second formulation. 3) Devising seven possible cut generation strategies. 4) Demonstrating, through extensive experiments, the effectiveness of the developed algorithm and the added value of the proposed timetabling strategy.

The remainder of the paper is organized as follows: In Section 2, we present an overview of the relevant research on optimization approaches involving short-turning or timetabling of metro lines. In Section 3, we formally describe the DDTP, and detail all made assumptions. In Section 4, we provide a flow-based MILP formulation of the problem. Then, in Section 5, we propose an alternative formulation with exponentially many constraints and develop a cut generation algorithm for it, discuss the generation of violated constraints, and present two classes of valid inequalities. In Section 6, we present our computational results. Lastly, in Section 7, we summarize the results.

2. Literature Review

The optimization of transit networks has received wide interest over the years. In the most general sense, optimizing a transit network can comprise a very wide set of problems, from the design of the lines to the rostering of the crew and vehicles, e.g., Ceder (2016), Guihaire and Hao (2008), Farahani et al. (2013). Timetabling addresses the issues of determining the timetables (or schedules) of the vehicles serving the line. Thus, timetabling generally consists of planning the arrival and departure times of the vehicles at the various stations.

In practice, timetables are frequently constructed in a periodic manner. Periodic timetables are more easily memorized by passengers and have shown their effectiveness both in metros (Liebchen 2008) and railway (Kroon et al. 2009). Liebchen and Möhring (2007) investigate the periodic timetabling problem (PTP), which aims at generating a feasible periodic timetable, by determining the relative timing and synchronization of the arrival and departure of the trains at the various stations of the line. The authors present a graph-based model, adapting the periodic event scheduling problem, which was proposed by Serafini and Ukovich (1989). Goerigk and Liebchen (2017) devise a custom simplex algorithm to better address the periodic nature of the PTP. A heuristic

approach for the PTP is proposed by Pätzold and Schöbel (2016). The PTP has been extended in several ways over the years. Liebchen et al. (2010) extend the PTP to explicitly address the robustness of the resulting solutions, with respect to random delays. Kroon et al. (2008) solve the problem of improving periodic timetables to maximize the robustness of the timetable with respect to random disturbances, by allocating buffer times between pairs of consecutive trains. In the PTP, the frequencies of all lines in the network are assumed to be identical, Siebert and Goerigk (2013) experiment with several PTP models introducing different line frequencies in the formulation.

Caprara, Fischetti, and Toth (2002) define the train timetabling problem (TTP), which consists of determining a periodic timetable for a set of trains operating in a single one-way track, subject to track capacity and operational constraints. The objective of the TTP is to minimize the deviations from a given ideal timetable. The problem is formalized through a multi-graph formulation and solved with a Lagrangian relaxation. Subsequently, Caprara et al. (2006) extend the model to address several commonly encountered operational constraints. Cacchiani, Caprara, and Toth (2008) reformulate the TTP using Dantzing Wolfe decomposition and propose heuristic and exact algorithms.

Several authors have developed timetabling approaches with passenger-oriented objective functions such as the minimization of the total waiting time on the line. These models explicitly optimize the service quality provided by the timetable. In the context of a train transit network, Kang and Zhu (2016) and Kang et al. (2015) investigate the problem of scheduling the first and last trains of the day, so as to optimize the transit time of passengers through the synchronization of the different lines at the connection stations of the network. Sun et al. (2014) develop a demand-driven approach to optimize the headway of a one directional metro corridor. The authors utilize demand data from smart card based fare collection to determine the optimal dispatching time of the trains from the terminal station. The objective is to minimize the total waiting time experienced by the passengers on the line.

Canca et al. (2014) develop a non-linear integer programming model to optimize demand-driven timetabling for a one directional metro corridor, and propose different measures to evaluate the quality and benefits of the proposed timetables. Barrena et al. (2014a) address the timetabling of a rail corridor under a given passenger demand profile. The authors develop a non-periodic timetabling model, that is solved exactly through a branch-and-cut algorithm. Barrena et al. (2014b) propose a metaheuristic algorithm for the same problem and apply it to artificial and real-world instances. To cope with overcrowding at stations of a metro line after a service disruption, Gao et al. (2016) propose the usage of stop skipping patterns in the timetable of the trains. They optimize the timetable in the recovery period after the disruption, to minimize the delays with respect to the original timetable and the passengers' waiting time. Niu, Zhou, and Gao (2015)

propose a timetabling algorithm for a metro corridor with predetermined stop-skipping patterns to minimize the total passenger waiting time.

One acceleration strategy of particular interest to this paper is the short-turning acceleration strategy. Short-turning has been extensively studied for bus networks, e.g., Furth (1987), Tirachini, Cortés, and Jara-Díaz (2011) and Cortés, Jara-Díaz, and Tirachini (2011). Comparatively, its study on rail lines has been limited. Mesa, Ortega, and Pozo (2009) use short-turning and stop skipping in a multi-criteria framework to allocate the available fleet according to the passenger demand. Kiefer, Kritzing, and Doerner (2016) propose a disruption management method for rebuilding services under a severe breakdown, by adding replacement lines, rerouting existing lines and possibly short-turning a subset of the vehicles. Canca et al. (2012) discuss the introduction of additional metro services on overloaded segments of a railway line. They develop a model which uses short-turning as well as dead-heading, to deal with service disruptions, without altering the pre-planned timetable of the other trains. To handle disruptions in the demand of a transit system, Canca et al. (2016) optimize the design of short-turning services introduced alongside the pre-planned timetable at a tactical level. The proposed model addresses the case of disruptions in a railway network, and uses short-turning trains as an additional separate service, running in parallel with the regular timetable, which is left unmodified. Different from other approaches, the authors do not optimize the timetabling of the trains explicitly, instead, the starting and ending stations of the short-turning trains, as well as their frequency, are decision variables.

Unlike the majority of other demand-driven methods, in this paper we utilize short-turning as part of the daily timetable. Specifically, we jointly optimize the train timetabling while considering the short-turning decisions. Recently, a similar version of the problem addressed in this paper without optimizing train dwelling time has been studied by Yang et al. (2020). An approximate solution scheme was devised and tested on small artificial instances with four stations, and thirty time-steps achieving an optimality gap of approximately 4%. Further tests are conducted on a realistic instances taken from the Beijing subway to demonstrate the service benefits offered by the proposed approach. To the best of our knowledge, no efficient exact algorithms are available in the literature to tackle such problems.

3. The Demand-Driven Timetabling Problem

We consider a single bi-directional metro line with two tracks, one dedicated to the upstream direction and another one dedicated to the downstream direction. Let $S = \{1, \dots, m\}$ be an ordered set of stations on the line. A station serves both the upstream and downstream directions.

Given a demand profile and a planning horizon, the DDTP consists of constructing a timetable for Ψ trains with the objective of minimizing the total waiting time of the passengers. The planning

horizon, is divided into discrete time-steps. A train timetable is described by a sequence of actions over the planning horizon. Each action has a duration of an integer number of time-steps and can be categorized as one of the following: 1) moving to the next station in the current direction, 2) performing a short-turn at the current station or 3) idling at the current station for one time-step. Trains are not allowed to overtake each other at any point.

Similar to Barrena et al. (2014a,b), we assume that the capacity of the trains is sufficiently large to accommodate all the passengers that wish to board at any station. We denote by $\Gamma = \{-1, 1\}$ the set of directions, where 1 indicates the upstream direction, meaning that the train is visiting the stations in increasing order, and -1 indicates the downstream direction, meaning that the train visits the stations in decreasing order. Additionally, let F_γ be the first station in direction $\gamma \in \Gamma$, i.e., $F_1 = 1$ and $F_{-1} = m$. By extension, $F_{-\gamma}$ is the last station of the line in direction γ . Thus, given a direction γ and a station $i \in S \setminus F_{-\gamma}$, we can express the station that follows i in direction γ as $i + \gamma$. Lastly, note that given stations $i, j \in S$, the direction γ of a train departing from i headed to j , can be expressed as $\gamma(i, j) = \gamma \in \Gamma : \gamma \cdot (j - i) \geq 1$.

Let $T = \{1, \dots, h\}$ be the set of time-steps in the planning horizon. Each time-step is of length δ , thus each $t \in T$ corresponds to δt units of time since the start of the planning horizon. We denote by $\tau_{i,j}$ the number of time-steps required for a train to go from station i to station j without short-turning including dwelling time and boarding time. The latter is assumed to be fixed for all stations. Additionally, let ρ be the number of time-steps a train takes to perform a short-turn at any station.

We denote by D the set of Origin-Destination (OD) pairs, i.e., $D = \{(i, j) : i, j \in S, i \neq j\}$. The number of passengers that arrive at station $i \in S$ with destination $j \in S$ at time-step $t \in T$ is denoted by $a_{i,j}^t$. We use the shorthand (i, j, t) to refer to the passengers that arrive at station $i \in S$ at time $t \in T$, with destination $j \in S$. We impose an upper bound g on the amount of time that a passenger may wait. This applies to the time a passenger spends waiting for a train to arrive as well as the time she waits on an idle train. We note that the parameter g is introduced for modeling purposes, and in practice, can be set to a rather high value, as we have done in the computational experiments, so that a waiting time of g is not attained by any passenger.

The solution to the DDTP consists of the sequence of actions performed by the Ψ trains over the planning horizon T , describing the timetabling of the line that minimizes the total waiting time of the passengers $(i, j, t) \forall (i, j) \in D, t \in T$. For ease of reading, a complete list of the used notation is provided in Appendix A. To conclude, the assumptions of the DDTP are as follows.

- The planning horizon is discretized into time-steps of length δ and all train operations have a duration of an integer number of time-steps (i.e., the travel speed is not a decision).
- Short-turning takes ρ time-steps and is independent from the station.

- A train takes $\tau_{i,j}$ time-steps to move from station i to station j without short-turning including dwelling and boarding times.
- No passenger can wait more than g time-steps to reach her destination, considering both the time she spends waiting for a train to arrive, and the time she waits on an idle train.
- Short-turning is allowed in all stations. Note that restricting short-turning to be allowed only in certain stations can easily be implemented.
- Trains are not allowed to overtake each other at any point in the line and two trains cannot simultaneously short-turn at the same station.
- The line demand is known in advance for each time-step and for each OD pair.
- The position of the trains at the start of the planning horizon can be freely selected. Note that these can be easily set to predetermined initial locations.
- Lastly, similar to Barrena et al. (2014a,b), we assume that the capacity of the trains is sufficiently large to accommodate all the passengers that wish to board at any station.

4. Demand-Driven Timetabling Problem Formulation

In this section we present a MILP formulation for the DDPS which will be referred to as the Flow Formulation (FF).

We consider a space-time graph $G = \{N, A\}$ to represent the position and actions of the trains. The node-set N is the set of triples $(s, \gamma, t) \forall s \in S, \gamma \in \Gamma, t \in T$, where each node represents the position and direction of a train at each time-step. The arc-set $A = A^m \cup A^i \cup A^u$ represents the actions the trains can take. Arcs in A^m correspond to trains moving to the next station on the line, connecting node (i, γ, t) with $(i + \gamma, \gamma, t + \tau_{i,i+\gamma})$. Arcs in A^i correspond to the trains idling for one time-step. Therefore, an arc in A^i connects a node (i, γ, t) with $(i, \gamma, t + 1)$. Lastly, arcs in A^u correspond to short-turning actions taken by trains, thus connecting node (i, γ, t) with $(i, -\gamma, t + \rho)$. See Figure 1 for an example of the structure of G on a seven-station line, such that the travel time between adjacent stations is one.

Ultimately, the timetable of a train would be represented as a path on G originating from a node (i, γ, t) and terminating at node (i', γ', h) , with $i, i' \in S, t \in T$ and $\gamma, \gamma' \in \Gamma$. We note that trains cannot overtake each other, two trains (or more) may not simultaneously short-turn at a station, and that two trains (or more) cannot simultaneously occupy the same station in the same direction. Therefore, train paths are node-disjoint. Note that since the train paths are node-disjoint from one another, any feasible solution that is described using a flow representation can uniquely be decomposed into Ψ independent train paths.

We express the paths using a binary variable for each arc. Specifically, for each arc in A^m , we associate a binary variable $x_{i,\gamma}^t$, which takes value one if a train departs from station i in direction

γ at time-step $t \in T \cup \{-\tau_{i,i+\gamma} + 1, \dots, 0\}$, and zero otherwise. For each arc in A^i , we associate a binary variable $z_{i,\gamma}^t$, which is one if a train in direction γ idles at station i at time-step $t \in T \cup \{0\}$ for one time-step, and zero otherwise. For each arc in A^u , we associate variable $u_{i,\gamma}^t$, which takes value one if a train in direction γ short-turns at station i at time-step $t \in T \cup \{-\rho + 1, \dots, 0\}$, and zero otherwise. Variables $x_{i,\gamma}^t$, $u_{i,\gamma}^t$, $z_{i,\gamma}^t$ with $t < 1$ indicate the starting time of the trains required for initializing the paths of the trains. Thus, $x_{i,\gamma}^{-2} = 1$ entails a train arriving at station $i + \gamma$ at time-step $-2 + \tau_{i,i+\gamma}$. Furthermore, we use variable $y_{i,j}^t$ to measure the waiting time experienced by passengers (i, j, t) . We note that the value of $y_{i,j}^t$ refers to the waiting time experienced by a single passenger and not the total waiting time of all passengers (i, j, t) .

The formulation is as follows.

$$[\mathbf{FF}] \quad \min \sum_{t \in T} \sum_{(i,j) \in D} a_{i,j}^t y_{i,j}^t \quad (1)$$

$$u_{i,-\gamma}^{t-\rho} + z_{i,\gamma}^{t-1} = u_{i,\gamma}^t + x_{i,\gamma}^t + z_{i,\gamma}^t \quad \forall \gamma \in \Gamma, t \in T : i = F_\gamma \quad (2)$$

$$u_{i,-\gamma}^{t-\rho} + x_{i-\gamma,\gamma}^{t-\tau_{i-\gamma,i}} + z_{i,\gamma}^{t-1} = u_{i,\gamma}^t + z_{i,\gamma}^t \quad \forall \gamma \in \Gamma, t \in T : i = F_{-\gamma} \quad (3)$$

$$u_{i,-\gamma}^{t-\rho} + x_{i-\gamma,\gamma}^{t-\tau_{i-\gamma,i}} + z_{i,\gamma}^{t-1} = u_{i,\gamma}^t + x_{i,\gamma}^t + z_{i,\gamma}^t \quad \forall \gamma \in \Gamma, i \in S \setminus \{F_\gamma, F_{-\gamma}\}, t \in T \quad (4)$$

$$u_{i,\gamma}^t + x_{i,\gamma}^t + z_{i,\gamma}^t \leq 1 \quad \forall \gamma \in \Gamma, i \in S \setminus \{F_{-\gamma}\}, t \in T \quad (5)$$

$$u_{i,\gamma}^t + z_{i,\gamma}^t \leq 1 \quad \forall \gamma \in \Gamma, t \in T : i = F_{-\gamma} \quad (6)$$

$$(4\rho - 3)u_{i,\gamma}^t + \sum_{t' \in T: t \leq t' \leq t + \rho - 1} u_{i,-\gamma}^{t'} + \sum_{t' \in T: t < t' < t + \rho - 1} (u_{i,\gamma}^{t'} + x_{i,\gamma}^{t'} + x_{i+\gamma,-\gamma}^{t'}) \leq 4\rho - 3 \quad \forall \gamma \in \Gamma, i \in S \setminus \{F_{-\gamma}\}, t \in T \quad (7)$$

$$(\rho - 1)u_{i,\gamma}^t + \sum_{t' \in T: t < t' < t + \rho - 1} u_{i,\gamma}^{t'} \leq \rho - 1 \quad \forall \gamma \in \Gamma, t \in T : i = F_{-\gamma} \quad (8)$$

$$\sum_{i \in S} \sum_{\gamma \in \Gamma} (z_{i,\gamma}^0 + \sum_{t=1-\tau_{i,i+\gamma}}^0 x_{i,\gamma}^t + \sum_{t=1-\rho}^0 u_{i,\gamma}^t) \leq \Psi \quad (9)$$

$$y_{i,j}^t \leq g \quad \forall (i,j) \in D, t \in T \quad (10)$$

$$y_{i,j}^t \geq y_{i+\gamma,j}^{t+\tau_{i,i+\gamma}} \quad \forall t \in T, i \in S, j \in S : |i - j| > 1, \gamma = \gamma(i,j) \quad (11)$$

$$y_{i,j}^t \geq 1 + y_{i,j}^{t+1} - (g + 1)x_{i,\gamma}^t \quad \forall t \in T, i \in S, j \in S : i \neq j, \gamma = \gamma(i,j) \quad (12)$$

$$x_{i,\gamma}^t \in \{0, 1\} \quad \forall i \in S, \gamma \in \Gamma, t \in T \cup \{1 - \tau_{i,i+\gamma}, \dots, 0\} \quad (13)$$

$$z_{i,\gamma}^t \in \{0, 1\} \quad \forall i \in S, \gamma \in \Gamma, t \in T \cup \{0\} \quad (14)$$

$$u_{i,\gamma}^t \in \{0, 1\} \quad \forall i \in S, \gamma \in \Gamma, t \in T \cup \{1 - \rho, \dots, 0\} \quad (15)$$

$$y_{i,j}^t \geq 0 \quad \forall (i,j) \in D, t \in T \quad (16)$$

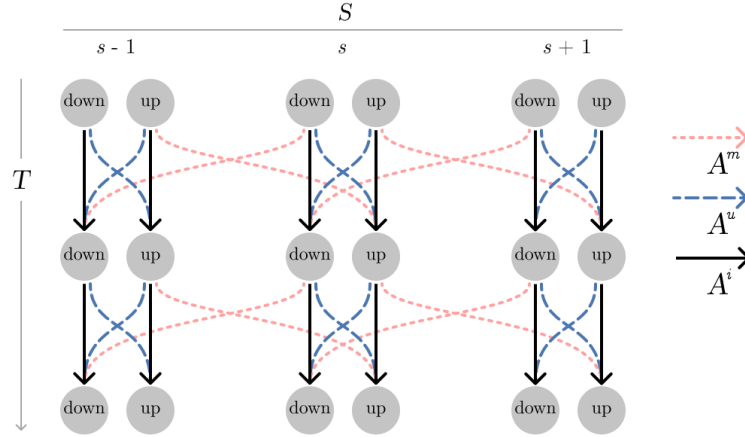


Figure 1 Example graph on which the train paths are defined.

The objective function (1) minimizes the total passenger waiting time. Constraints (2)–(4) balance the flow of trains at each station, accounting for incoming trains, departing trains, idling and short-turning. Specifically, constraints (2) express the balance of the trains on the initial stations in each direction (where there are no incoming trains), constraints (3) express the balance for the last stations in each direction (where there cannot be departing trains), and (4) express the constraint for each intermediate station. Constraints (5) and (6) ensure that at most one train can occupy a station at each time-step. Constraints (7) and (8) prevent scheduling conflicts while the trains are short-tuning. Specifically, they ensure that at most one train may be short-turning at a station at any given time. Additionally, while a train is short-turning at station i in direction γ , other trains cannot depart from i in direction γ , nor arrive at station i from direction $-\gamma$. The constant $4\rho - 3$ matches the maximum value of the left-hand side of constraint (7) when $u_{i,\gamma}^t = 0$, i.e., the constraint is deactivated if no train is short-turning at station i at time-step t . Similarly, the constant $\rho - 1$ matches the maximum value of the left-hand side of (8) when $u_{i,\gamma}^t = 0$. Constraint (9) ensures that the total number of trains in the line corresponds to at most the number of available trains. Formally, the constraint ensures that at most Ψ trains may be operating on the line at time-step 1 by ensuring that at most Ψ actions with starting time $t < 1$ and ending time $t \geq 1$ are selected in the timetable. The number of trains operating on the line after time-step 1 is preserved by constraints (2)–(4).

Constraints (10)–(12) keep track of the passenger waiting times. Constraints (10) force the waiting time experienced by each passenger to be smaller or equal to g . Constraints (11) force the waiting time of passengers (i, j, t) to be greater or equal to the waiting time of passengers $(i + \gamma, j, t + 1)$, with $\gamma = \gamma(i - j)$ (i.e., the direction in which the passengers are headed). In this case, passengers (i, j, t) can reach $i + \gamma$ as early as $t + \tau_{i,i+\gamma}$ without experiencing any waiting. At that point, they can reach their destination, experiencing the same amount of waiting time

as passengers $(i + \gamma, j, t + \tau_{i,i+\gamma})$. Thus, the waiting time experienced by passengers (i, j, t) is at least the waiting time experienced by passengers $(i + \gamma, j, t + \tau_{i,i+\gamma})$. Constraints (12) ensure that if there is no train departing at time t from station i towards station j , in direction $\gamma = \gamma(i, j)$, the waiting time of passengers (i, j, t) is one time-step greater than the waiting time of the passengers that arrive at the same station with the same destination, at time-step $t + 1$. Lastly, constraints (13)–(16) represent the domain of the variables.

Note that formulation FF has a quadratic number of big-M constraints (12), which, in practice, would render it not very effective at solving large instances.

5. Cut generation algorithm

In this section, we develop an alternative formulation for the DDTP coupled with a cut generation algorithm. In section 5.1 we present a formulation with an exponential number of constraints, which we refer to as EF. In section 5.2, we strengthen the constraints used in EF and subsequently present the strengthened exponential formulation (EFS). In section 5.3, we present algorithms for separating violated inequalities in the cut generation algorithm. Lastly, in section 5.4, we present two classes of valid inequalities and discuss how the cut generation algorithm can be modified to account for them. We note that various combinations of the proposed cuts and valid inequalities will be examined in the computational experiments in section 6.

5.1. Exponential formulation

For each (i, j, t) , we define the graph $G_{i,j}^t = \{N_{i,j}^t, A_{i,j}^t\}$. In what follows, for the sake of brevity, unless otherwise stated, we assume $\gamma = \gamma(i, j)$. The node-set is $N_{i,j}^t = \bar{N}_{i,j}^t \cup \bar{j}$ where \bar{j} is a termination dummy node. $\bar{N}_{i,j}^t$ is the set of pairs (i', t') with $i' \in i, \dots, j$ for $\gamma = 1$ and $i' \in j, \dots, i$ for $\gamma = -1$, and $t' \geq t$, $t' \in T$. Thus, $\bar{N}_{i,j}^t$ represents the possible positions of the passenger, i.e., passengers (i, j, t) will exclusively move between i and j in the appropriate direction. The arc-set $A_{i,j}^t = M_{i,j}^t \cup W_{i,j}^t \cup E_{i,j}^t$ describes the three actions the passengers can do during the trip, i.e., they can move to the next station ($M_{i,j}^t$), they can wait ($W_{i,j}^t$), or they can leave the line upon reaching their destination station j ($E_{i,j}^t$). Therefore, arcs in $M_{i,j}^t$ connect each node (s, t') to node $(s + \gamma, t' + \tau_{s,s+\gamma})$, arcs in $W_{i,j}^t$ connect each node (s, t') to node $(s, t' + 1)$, and arcs in $E_{i,j}^t$ connect each node (j, t') to node \bar{j} .

The trajectory of the passengers depends on the timetable of the trains. Given a timetable $\bar{x}_{i,\gamma}^t$, $\bar{u}_{i,\gamma}^t$, and $\bar{z}_{i,\gamma}^t$, $\forall i \in S, \gamma \in \Gamma, t \in T$, we define the auxiliary graph $G_{i,j}^t(\bar{x})$ to describe the available itineraries of passengers (i, j, t) . The auxiliary graph $G_{i,j}^t(\bar{x})$ has the same nodes and arcs as $G_{i,j}^t$, but includes arc capacities that depend on \bar{x} . In particular, all arcs in $W_{i,j}^t \cup E_{i,j}^t$ have a capacity of one, whereas the arcs $((s, t'), (s + \gamma, t' + \tau_{s,s+\gamma})) \in M_{i,j}^t$ have a capacity of $\bar{x}_{s,\gamma}^{t'}$. This implies that an arc in $M_{i,j}^t$ is only available if traversed by a train. Furthermore, a cost of one is associated with

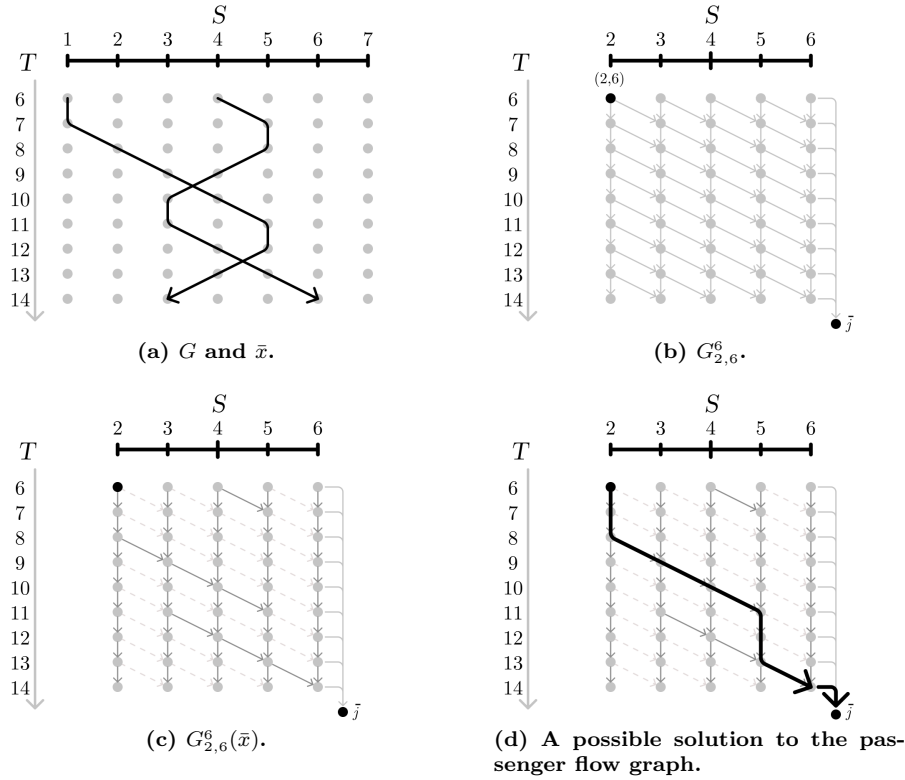


Figure 2 Example of the construction of the auxiliary graph for passengers (2, 6, 6). In the auxiliary graph, arcs drawn with a solid line have a capacity of 1, dashed arcs have a capacity of 0.

each arc in $W_{i,j}^t$, and the cost of the other arcs is zero. See Figure 2 for an example of the structure of graph $G_{i,j}^t$ and $G_{i,j}^t(\bar{x})$ on a seven-station line, such that the travel time between adjacent stations is one.

Given $G_{i,j}^t(\bar{x})$, the problem associated to passengers (i, j, t) corresponds to a minimum cost flow problem with a single source (i, t) and a single destination \bar{j} having a supply and demand of one unit. The flow on each arc represents the percentage of the passengers (i, j, t) that perform the action associated with the arc (i.e., wait or board a train). If the given timetable satisfies (13), the capacity of all arcs is integer, i.e., $\bar{x}_{i,\gamma}^t \in \{0, 1\} \forall i \in S, \gamma \in \Gamma, t \in T$, and the problem reduces to a shortest path problem. However, we note that in a cut generation algorithm, \bar{x} might be fractional.

The optimal flow describes one of the possible trajectories the passengers may take to reach their destination, and the cost of the flow corresponds to the waiting time experienced by each passenger if the timetable were to be operated. Thus, if $y_{i,j}^t = q$ it must be possible to construct a flow on $G_{i,j}^t(\bar{x})$ from node (i, t) to node $(j, t + \tau_{i,j} + q)$. By the same reasoning, if there is no path connecting nodes (i, t) and $(j, t + \tau_{i,j} + q)$, we can conclude that $y_{i,j}^t \geq q + 1$.

We denote by $\mathcal{CS}(i, j, t, q)$ the family of cut-sets separating nodes (i, t) and $(j, t + \tau_{i,j} + q - 1)$ in $G_{i,j}^t$, and by $\mathcal{K}(c, \bar{x})$ the sum of the capacities of the arcs contained in cut-set $c \in \mathcal{CS}(i, j, t, q)$ on $G_{i,j}^t(\bar{x})$. Using these definitions we now formulate the DDTP as follows.

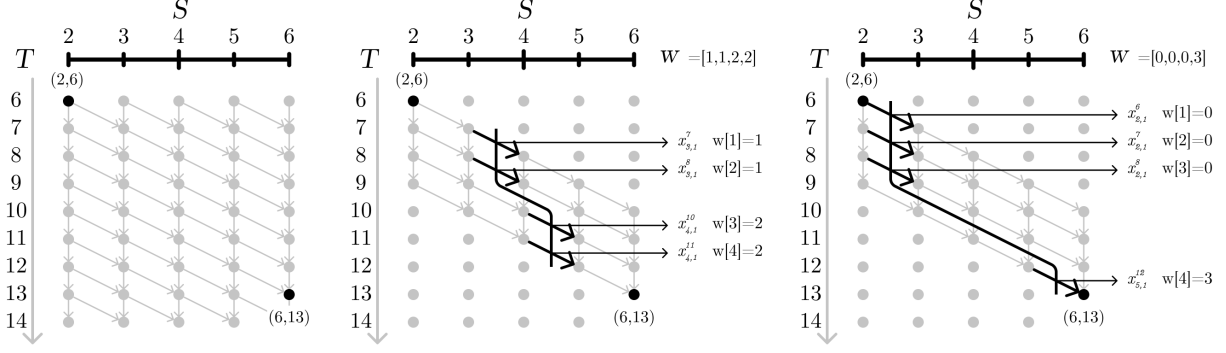


Figure 3 Example of the construction of a cut-set on $G_{i,j}^t$. On the left, the structure of $G_{2,6}^6$ referred to passengers $(2, 6, 6)$, in the center the cut-set associated to $q = 4$ and $\vec{w} = [1, 1, 2, 2]$, on the right the cut-set associated to $q = 4$ and $\vec{w} = [0, 0, 0, 3]$.

$$[\mathbf{EF}] \quad \min \sum_{t \in T} \sum_{(i,j) \in D} a_{i,j}^t y_{i,j}^t \quad (17)$$

$$(2)-(10), (13)$$

$$y_{i,j}^t \geq q(1 - \mathcal{K}(c, x)) \quad \forall (i, j) \in D, t \in T, q \in \mathbb{N}_0, c \in \mathcal{CS}(i, j, t, q) \quad (18)$$

$$y_{i,j}^t \in \mathbb{N}_0 \quad \forall (i, j) \in D, t \in T \quad (19)$$

Comparing FF to EF, constraints (11)–(12) are replaced by constraints (18). Constraints (18) ensure that if on $G_{i,j}^t(x)$ there is no path connecting (i, t) and $(j, t + \tau_{i,j} + q - 1)$ then $y_{i,j}^t \geq q$. Thus, when $\mathcal{K}(c, x) = 0$, for some $c \in \mathcal{CS}(i, j, t, q)$, the constraint reduces to $y_{i,j}^t \geq q$. Conversely, when $\mathcal{K}(c, x) \geq 1$, nothing can be concluded on $y_{i,j}^t$ based on c , thus, the inequality is inactive.

5.2. Strengthened exponential formulation

We now rewrite constraints (18) to be more easily treatable, and reduce the size of $\mathcal{CS}(i, j, t, q)$ by removing unnecessary elements. Given $c \in \mathcal{CS}(i, j, t, q)$, if $c \cap (A^i \cup A^u) \neq \emptyset$, then $\mathcal{K}(c, x) \geq 1$, since arcs in $A^i \cup A^u$ all have capacity one. Thus, in this case, constraint (18) associated to cut-set c is redundant.

Given $G_{i,j}^t$ and a parameter $q \in \mathbb{N}_0$, $q \leq g$, let us consider an integer vector \vec{w} with q elements, ordered in a non-decreasing manner. Furthermore, the values contained in \vec{w} are in the interval

$[0, |i - j| - 1]$. For convenience, we defined $\bar{\tau}(n) = \tau_{i, i+\gamma n}$, i.e., the travel time from station i to the n -th station in direction γ relative to i . The set of arcs:

$$\{((i + \gamma \vec{w}[k], t + k - 1 + \bar{\tau}(\vec{w}[k])), (i + \gamma \vec{w}[k] + \gamma, t + k - 1 + \bar{\tau}(\vec{w}[k] + 1))) \mid \forall k = 1, \dots, q\} \quad (20)$$

is a cut-set between nodes (i, t) and $(j, t + \tau_{i,j} + q)$ on $G_{i,j}^t$.

Example

Given a seven-station line such that the travel time between adjacent stations is one (see Figure 3), let us consider passengers $(2, 6, 6)$ (thus $\gamma(i, j) = 1$), i.e., passengers arrive at station 2 at time-step 6 headed to station 6. Additionally, let $\vec{w} = [1, 1, 2, 2]$. The cut-set associated to \vec{w} is as follows:

$$\{((3, 7), (4, 8)), ((3, 8), (4, 9)), ((4, 10), (5, 11)), ((4, 11), (5, 12))\}.$$

Each arc in the cut-set corresponds to a value in \vec{w} ; thus, the cut-set is composed of four arcs since $|\vec{w}| = 4$. The q^{th} element of \vec{w} corresponds to an arc in the cut-set (20) which originates from station $i + \gamma \vec{w}[q]$ at time-step $t + q - 1 + \bar{w}[q]$, i.e., time $t + q - 1$ plus the time required to reach station $i + \gamma \vec{w}[q]$ from i .

Given \bar{x} and \vec{w} , the sum of the capacities of the arcs in cut-set (20) in $G_{i,j}^t(\bar{x})$ can be computed as follows:

$$\sum_{k=1}^q \bar{x}_{i+\gamma \vec{w}[k], \gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])}.$$

For a given q , we denote by $C_{i,j}^{t,q}$ the set of all integer vectors $\vec{w} \in \mathbb{N}_0^q$, with non-decreasing values in the interval $[0, |i - j| - 1]$.

Proposition 1: Given a train timetable described by variables $\bar{x}_{i,\gamma}^t$, $\bar{u}_{i,\gamma}^t$, and $\bar{z}_{i,\gamma}^t$, $\forall i \in S, \gamma \in \Gamma, t \in T$, and passengers (i, j, t) . If there is no path in $G_{i,j}^t$ from (i, t) to $(j, t + \tau_{i,j} + q - 1)$ then there exists $\vec{w} \in C_{i,j}^{t,q}$ such that

$$\sum_{k=1}^q \bar{x}_{i+\gamma \vec{w}[k], \gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} = 0.$$

The proof of Proposition 1 is presented in Appendix B. We note that the cardinality of set $C_{i,j}^{t,q}$ is equal to:

$$|C_{i,j}^{t,q}| = \frac{(|i - j| + q)!}{|i - j|! q!}. \quad (21)$$

Using set $C_{i,j}^{t,q}$ and Proposition 1, we can replace constraints (18) with the following constraints:

$$y_{i,j}^t \geq q - q \sum_{k=1}^q x_{i+\gamma \vec{w}[k], \gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} \quad \forall (i, j) \in D, t \in T, q \in \mathbb{N}_0, \vec{w} \in C_{i,j}^{t,q} : t + q - 1 \leq h, q \leq g. \quad (22)$$

Proposition 2: Constraint (22) can be strengthened as follows:

$$y_{i,j}^t \geq q - \sum_{k=1}^q (q-k+1) x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}. \quad (23)$$

Proof. Let us consider $q' < q$, we can rewrite constraint (22) as follows:

$$y_{i,j}^t \geq q - q \sum_{k=1}^{q'} x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} - q \sum_{k=(q'+1)}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} \quad (24)$$

The expression:

$$\sum_{k=1}^{q'} x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}$$

represents the capacity of a cut-set between node (i, t) and node $(j, t + \tau_{i,j} + q' - 1)$. Thus, if

$$\sum_{k=1}^{q'} x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} = 0$$

we can conclude that $y_{i,j}^t \geq q'$, regardless of $\sum_{k=(q'+1)}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}$. Therefore, it holds the following:

$$\sum_{k=1}^{q'} x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} = 0 \quad \wedge \quad \sum_{k=(q'+1)}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} = 1 \quad \rightarrow \quad y_{i,j}^t \geq q'$$

We include this observation in constraint (22) as follows:

$$y_{i,j}^t \geq q - q \sum_{k=1}^{q'} x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} - (q - q') \sum_{k=(q'+1)}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}. \quad (25)$$

Constraint (25) yields tighter bounds when $\sum_{k=1}^{q'} x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])} = 0$. Selecting $q' = 1$ we tighten the constraint as follows:

$$y_{i,j}^t \geq q - q x_{i+\gamma \bar{w}[1], \gamma}^{t+\bar{\tau}(\bar{w}[1])} - (q-1) \sum_{k=2}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}. \quad (26)$$

Afterwards, if $q > 2$ we can rewrite (26) as:

$$y_{i,j}^t - 1 + q x_{i+\gamma \bar{w}[1], \gamma}^{t+\bar{\tau}(\bar{w}[1])} \geq (q-1) - (q-1) \sum_{k=2}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}. \quad (27)$$

Note that the right-hand-side of constraint (27) has the same structure as constraint (22). Thus, the constraint can be further tightened to:

$$y_{i,j}^t - 1 + q x_{i+\gamma \bar{w}[1], \gamma}^{t+\bar{\tau}(\bar{w}[1])} \geq (q-1) - (q-1) x_{i+\gamma \bar{w}[2], \gamma}^{t+1+\bar{\tau}(\bar{w}[2])} - (q-2) \sum_{k=3}^q x_{i+\gamma \bar{w}[k], \gamma}^{t+k-1+\bar{\tau}(\bar{w}[k])}. \quad (28)$$

Then, if $q > 3$ the process can be repeated again to further tighten the constraint. If this tightening strategy is applied for a total of $q-1$ times, we obtain strengthened constraint (23). \square

When considering an integer solution, if $\sum_{k=1}^q x_{i+\gamma\vec{w}[k],\gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} = 1$, constraint (23) reduces to the bound $y_{i,j}^t \geq q' - 1$, with q' achieving $x_{i+\gamma\vec{w}[q'],\gamma}^{t+q'-1+\bar{\tau}(\vec{w}[q'])} = 1$. Thus, the bound imposed by constraint (23) corresponds to the cost achieved by the optimal solution to the min cost flow problem associated to passengers (i, j, t) . Similarly, in case of a fractional solution \bar{x} , it can be shown that the bound imposed by constraint (23) corresponds to the cost achieved by the optimal solution to the min cost flow problem associated to passengers (i, j, t) if $0 \leq \sum_{k=1}^q x_{i+\gamma\vec{w}[k],\gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} \leq 1$.

Example

Consider a seven station line with unitary distance between stations (i.e., such that $\tau_{i,j} = |i - j| \forall i, j \in S$), $\rho = 1$, and passengers $(2, 6, 6)$ (thus $\gamma(i, j) = 1$). The passengers will arrive at station 2 at time-step 6 headed to station 6. On this, let us consider constraint (23) with $q = 4$ and $\vec{w} = [1, 1, 2, 2]$. See Figure 3 for the visual representation of the considered cut-set on the passenger graph. Following (23), the resulting constraint is as follows:

$$y_{2,6}^6 \geq 4 - 4x_{3,1}^7 - 3x_{3,1}^8 - 2x_{4,1}^{10} - x_{4,1}^{11}. \quad (29)$$

Considering this, we acknowledge the following six cases.

1. if $x_{3,1}^7 = 1$ and $x_{3,1}^8 = x_{4,1}^{10} = x_{4,1}^{11} = 0$ we cannot conclude anything on the state of the auxiliary graph $G_{i,j}^t(x)$ and the constraint reduces to $y_{2,6}^6 \geq 0$.
2. if $x_{3,1}^8 = 1$ and $x_{3,1}^7 = x_{4,1}^{10} = x_{4,1}^{11} = 0$ nodes $(2, 6)$ and $(6, 10)$ are not connected and thus $y_{2,6}^6 \geq 1$.
3. if $x_{4,1}^{10} = 1$ and $x_{3,1}^7 = x_{3,1}^8 = x_{4,1}^{11} = 0$ nodes $(2, 6)$ and $(6, 11)$ are not connected and thus $y_{2,6}^6 \geq 2$.
4. if $x_{4,1}^{11} = 1$ and $x_{3,1}^7 = x_{3,1}^8 = x_{4,1}^{10} = 0$ nodes $(2, 6)$ and $(6, 12)$ are not connected and thus $y_{2,6}^6 \geq 3$.
5. if none of the represented x variables are one, it results that the node $(2, 6)$ and $(6, 13)$ are not connected on the auxiliary graph $G_{i,j}^t(x)$, and thus $y_{2,6}^6 \geq 4$.
6. Lastly, if $x_{4,1}^{10} = x_{4,1}^{11} = 1$ and $x_{3,1}^7 = x_{3,1}^8 = 0$ the constraint reduces to $y_{2,6}^6 \geq 1$, which is valid.

Using the strengthened constraint (23), we rewrite the formulation [EF] as follows.

$$[\mathbf{EFS}] \quad \min \sum_{t \in T} \sum_{(i,j) \in D} (a_{i,j}^t y_{i,j}^t) \quad (30)$$

$$(2)-(10), (13)$$

$$y_{i,j}^t \geq q - \sum_{k=1}^q (q - k + 1) x_{i+\gamma\vec{w}[k],\gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} \quad \forall (i, j) \in D, t \in T, q \in \mathbb{N}_0, \vec{w} \in C_{i,j}^{t,q} : t + q - 1 \leq h, q \leq g, \gamma = \gamma(i, j) \quad (31)$$

$$y_{i,j}^t \in \mathbb{N}_0 \quad \forall (i, j) \in D, t \in T \quad (32)$$

In the rewritten formulation, constraints (18) are replaced by constraints (31). Constraints (31) are exponentially many (see equation (21)). Additionally, these constraints are frequently not active. Therefore, they should be added to the formulation through a cut generation framework. To do so, we relax constraint (18) to obtain formulation [EFM], which will be used as the master problem for the cut generation algorithm.

5.3. Separation of violated constraints

We now discuss how, for a given solution that satisfies constraints (2)–(10), violated constraints (31) can be separated. In the rest of this section, we will start by providing a general strategy to separate violated (31). We then develop efficient separation strategies for two special cases of (31).

Let us consider a solution $\bar{x}_{i,\gamma}^t, \bar{u}_{i,\gamma}^t, \bar{z}_{i,\gamma}^t, \forall i \in S, \gamma \in \Gamma, t \in T$, and $\bar{y}_{i,j}^t, \forall (i,j) \in D, t \in T$. We assume that the given solution satisfies constraints (13), i.e., the separation of violated constraints is only performed at integer nodes in the search tree. When the solution is feasible, $\bar{y}_{i,j}^t$ corresponds to the waiting time experienced by each passenger (i,j,t) . Thus, we compute the actual waiting time experienced by the passengers by solving the min-cost flow on $G_{i,j}^t(\bar{x})$ and compare the resulting waiting time with the value of $\bar{y}_{i,j}^t$. If the actual waiting time exceeds $\bar{y}_{i,j}^t$, we generate a cut by constructing a vector $\vec{w} \in C_{i,j}^{t, \bar{y}_{i,j}^t + 1}$ (i.e., $q = \bar{y}_{i,j}^t + 1$) such that:

$$\sum_{k=1}^{(\bar{y}_{i,j}^t + 1)} \bar{x}_{i+\gamma\vec{w}[k], \gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} = 0. \quad (33)$$

We now present the algorithm used to compute \vec{w} . This is done according to the integer min cost flow solution on $G_{i,j}^t(\bar{x})$. To aid the computation, we use integer λ to keep track of the number of moving arcs taken by the passengers (i.e., arcs in A^m), and integer q to keep track of the number of idle arcs taken (i.e., arcs in A^i). Thus, $\lambda = 3$ and $q = 1$ indicate that a total of three moving arcs have been used and the passengers have waited one time-step (i.e., the passengers are at node $(i + 3\gamma, t + 1 + \bar{\tau}(3))$).

We initialize the algorithm by setting λ and q to zero, and \vec{w} to an empty vector. Iteratively, we check the value of $\bar{x}_{i+\gamma\lambda, \gamma}^{t+q+\bar{\tau}(\lambda)}$. If $\bar{x}_{i+\gamma\lambda, \gamma}^{t+q+\bar{\tau}(\lambda)} = 1$ we increment λ . Otherwise, we set $\vec{w}[q] = \lambda$ then increase q by one unit. The process is repeated until $q > \bar{y}_{i,j}^t$.

By construction, it holds that $\vec{w} \in C_{i,j}^{t,q}$. Indeed, the procedure terminates when $q = \bar{y}_{i,j}^t + 1$ elements have been added to the vector. Note that all elements of \vec{w} are integer, monotonically increasing, and fall in the interval $[0, |i - j| - 1]$. Meaning that the algorithm never reaches a state where $\lambda = |i - j|$ before terminating. Indeed, reaching such a state would imply that there exists a trajectory for the passengers to reach their destination experiencing less than $\bar{y}_{i,j}^t$ time-steps. Additionally, the construction of \vec{w} has been performed in such a way that it satisfies (33). Thus, constraint (31) is guaranteed to be violated for the generated \vec{w} , and it is added to the master

Algorithm 1: The procedure to separate violated constraints (31) on a given integer solution, for passengers (i, j, t) .

```

 $\lambda \leftarrow 0, q \leftarrow 0, \vec{w} \leftarrow [];$ 

while  $q \leq \bar{y}_{i,j}^t$  do
  if  $\bar{x}_{i+\gamma\lambda, \gamma}^{t+q+\bar{\tau}(\lambda)} = 1$  then
     $\lambda \leftarrow \lambda + 1;$ 
  else
     $\vec{w}[q] \leftarrow \lambda;$ 
     $q \leftarrow q + 1;$ 
return  $\vec{w};$ 

```

problem [EFM]. Note that this procedure is guaranteed to generate a violated constraint for the considered passengers (i, j, t) , if any. However, by construction, the generated constraint is violated by exactly one unit.

In the following, we discuss two special cases of constraint (31). Furthermore, we present a case where it is possible to perform the separation on fractional solutions, i.e., solutions that violate (13).

We define the concept of the “order” of a cut (31) as the number of different values in \vec{w} . For instance, a cut of order 1 will have $\vec{w} = [i \dots q]$ (where $i \dots q$ is used to denote a series of q consecutive elements equal to i), order two cuts will have $\vec{w} = [i \dots q', j \dots q'']$ (i.e., \vec{w} is composed by a series of q' elements equal to i , followed by a series of q'' elements equal to j), for an example of this concept see Figure 4.

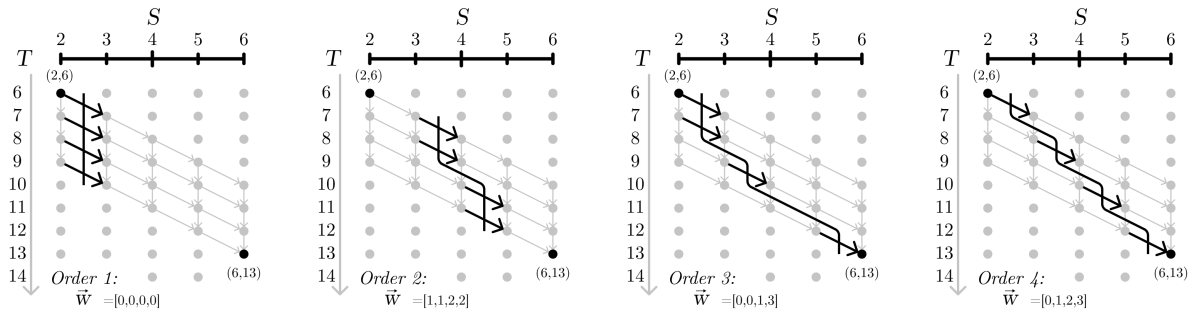


Figure 4 Example of different order cuts for $G_{2,6}^6$.

5.3.1. Order 1 cuts We now consider the case where \vec{w} is a vector of q identical elements. We rewrite an order 1 cut as follows:

$$\bar{y}_{i,j}^t \geq q - \sum_{k=1}^q (q - k + 1) \bar{x}_{i+\gamma\epsilon, \gamma}^{t+k-1+\bar{\tau}(\epsilon)}. \quad (34)$$

This corresponds to a constraint with $\vec{w} = [\epsilon \dots \epsilon]$ (i.e., a vector of q elements ϵ). We now discuss how violated order 1 cuts can efficiently be detected.

If the given solution satisfies (13), the separation of violated order 1 cuts can be done by inspection. Considering passengers (i, j, t) , for a fixed value of q , the number of order 1 cuts is $|i - j|$. Thus, we enumerate all possible cuts and verify their violation.

Proposition 3: Given $\bar{x}_{i,\gamma}^t$, $\bar{u}_{i,\gamma}^t$, and $\bar{z}_{i,\gamma}^t$, $\forall i \in S, \gamma \in \Gamma, t \in T$, and passengers (i, j, t) , if there exists a violated order 1 cut for the passengers, there exists a violated order 1 cut with $q = \bar{y}_{i,j}^t + 1$.

Proof. Choosing a value of $q \leq \bar{y}_{i,j}^t$ will not yield a violated constraint (34). If the solution satisfies (13), and if $q = \bar{y}_{i,j}^t + 1$, constraint (34) is violated if the following holds.

$$\sum_{k=1}^{(\bar{y}_{i,j}^t+1)} \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k+\bar{\tau}(\epsilon)} = 0 \quad (35)$$

Lastly, in the case of $q' > \bar{y}_{i,j}^t + 1$, if the solution satisfies (13), constraint (34) is violated if the following holds.

$$\sum_{k=1}^{q'} \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k+\bar{\tau}(\epsilon)} = 0 \quad (36)$$

Condition (36) implies condition (35), and therefore if a choice of $q > \bar{y}_{i,j}^t + 1$ produces a violated order 1 cut, a choice of $q = \bar{y}_{i,j}^t + 1$ also produces a violated cut. \square

As a consequence of Proposition 3, in the separation, it is sufficient to uniquely consider the value of $q = \bar{y}_{i,j}^t + 1$. We now present the procedure for separating violated order 1 cuts if the given solution does not satisfy (13).

Proposition 4: Given $\bar{x}_{i,\gamma}^t$, $\bar{u}_{i,\gamma}^t$, and $\bar{z}_{i,\gamma}^t$, $\forall i \in S, \gamma \in \Gamma, t \in T$, and passengers (i, j, t) , for a fixed value of ϵ the right hand side of (34) is a convex function of q .

Proof. Let us write the right hand side of (34) for three values of q : \bar{q} , $\bar{q} + 1$, and $\bar{q} + 2$.

$$\bar{q} - \sum_{k=1}^{\bar{q}} (\bar{q} - k + 1) \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} \quad (37)$$

$$\bar{q} + 1 - \sum_{k=1}^{\bar{q}} (\bar{q} - k + 2) \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)} \quad (38)$$

$$\bar{q} + 2 - \sum_{k=1}^{\bar{q}} (\bar{q} - k + 3) \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} - 2\bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+1+\bar{\tau}(\epsilon)} \quad (39)$$

The difference between (37) and (38) is as follows:

$$1 - \sum_{k=1}^{\bar{q}} \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)}. \quad (40)$$

The difference between (38) and (39) is as follows:

$$1 - \sum_{k=1}^{\bar{q}} \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+1+\bar{\tau}(\epsilon)}. \quad (41)$$

Since \bar{x} is non negative we can conclude the following:

$$1 - \sum_{k=1}^{\bar{q}} \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+1+\bar{\tau}(\epsilon)} \leq 1 - \sum_{k=1}^{\bar{q}} \bar{x}_{i+\gamma\epsilon,\gamma}^{t+k-1+\bar{\tau}(\epsilon)} - \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)}. \quad (42)$$

Thus, the right hand side of (34) is a convex function of q . \square

For a given value of ϵ , the separation is performed by checking the departing trains from station $i + \gamma\epsilon$ at time-step $t + \epsilon + q$, evaluating cut (34) for increasing values of q . As for Proposition 4, the right hand side of (34) is a convex function of q . Thus, the search is stopped when a decrement in the right hand side is observed, then we conclude that no violated constraint exists for the chosen value of ϵ . Furthermore, the search is also stopped whenever a violated constraint is found.

We now describe the algorithm used for the separation of fractional solutions in detail. The pseudocode is presented in Algorithm 2. We initialize the algorithm by setting three variables: $\alpha = 0$, $\bar{q} = 0$ and $\beta = 0$. The α variable is used to keep track of the fractional number of trains that have departed from station $i + \gamma\epsilon$. The \bar{q} variable is used to keep track of the number of time-steps considered by the algorithm. Lastly, β keeps track of the resulting right hand side of cut (34) for $q = \bar{q}$. In each iteration, we increase α by $\bar{x}_{i+\gamma\epsilon,\gamma}^{t+q+\bar{\tau}(\epsilon)}$. We then increase β by $1 - \alpha$ and increase \bar{q} by 1. Afterwards, if $\beta > \bar{y}_{i,j}^t$ we can conclude that a cut with $\vec{w} = [\epsilon \dots \bar{q}]$ is violated for the problem. If $\alpha \geq 1$ on the other hand, no order 1 cut with the given ϵ is violated.

Algorithm 2: The procedure to separate violated order 1 cuts on a given fractional solution, for passengers (i, j, t) with a fixed ϵ .

```

 $\alpha \leftarrow 0, \bar{q} \leftarrow 0, \beta \leftarrow 0;$ 
while  $\alpha < 1$  do
     $\alpha \leftarrow \alpha + \bar{x}_{i+\gamma\epsilon,\gamma}^{t+\bar{q}+\bar{\tau}(\epsilon)};$ 
     $\beta \leftarrow \beta + (1 - \alpha);$ 
    if  $\beta > \bar{y}_{i,j}^t$  then
        add order 1 cut with  $\vec{w} = [\epsilon \dots \bar{q}]$  to the problem;
        return;
     $\bar{q} \leftarrow \bar{q} + 1;$ 
return;
```

5.3.2. Order 2 cuts We now consider the case when \vec{w} is a vector containing elements q of two different values. Order 2 cuts can be rewritten as follows:

$$\bar{y}_{i,j}^t \geq q - \sum_{k=1}^{q'} (q-k+1) \bar{x}_{i+\gamma\eta,\gamma}^{t+k-1+\bar{\tau}(\eta)} - \sum_{k=(q'+1)}^q (q-k+1) \bar{x}_{i+\gamma\theta,\gamma}^{t+k-1+\bar{\tau}(\theta)}. \quad (43)$$

This corresponds to a constraint with $\vec{w} = [\eta \dots_{q'}, \theta \dots_{q-q'}]$.

In the case of integer solutions, the separation of violated order 2 is performed once the values of η and θ have been fixed. Similar to the case of order 1 cuts, setting q to $\bar{y}_{i,j}^t + 1$ allows finding a violated constraint, if one exists. Thus, q' needs to be determined. The possible values of q' range from 0 to q . We therefore enumerate them. Note that if $\sum_{k=1}^{q''} \bar{x}_{i+\gamma\eta,\gamma}^{t+k-1+\bar{\tau}(\eta)} \geq 1$, we can already conclude that any value of $q' \geq q''$ does not yield a violated inequality. Thus, in this case, we evaluate the values of q' in the range 0 to q'' .

5.4. Valid inequalities

Given the concept of ordered cuts, we now present two classes of valid inequalities, which will be referred to as chain constraints:

$$y_{i,j}^t \leq y_{i,j+\gamma}^t \quad \forall (i,j) \in D, t \in T : \gamma = \gamma(i,j), \quad (44)$$

$$y_{i,j}^t \leq y_{i-\gamma,j}^{t-\tau_{i-\gamma,i}} \quad \forall (i,j) \in D, t \in T : \gamma = \gamma(i,j). \quad (45)$$

These inequalities state that:

- the waiting time experienced by the passengers cannot decrease by selecting a destination station that succeeds their destination station.
- the waiting time experienced by the passengers cannot decrease by selecting an origin station that precedes their origin station (discounting the additional travel time).

We note that constraints (45) are derived from constraints (12).

For a given solution, let us consider passengers (i,j,t) . The order 1 cut that is associated to $\epsilon = 0$ is as follows:

$$y_{i,j}^t \geq q - \sum_{k=1}^q (q-k+1) \bar{x}_{i,\gamma}^{t+k-1}. \quad (46)$$

If constraints (44) are used in the formulation, we combine (46) with (44) to obtain the following:

$$y_{i,j+\gamma}^t \geq y_{i,j}^t \geq q - \sum_{k=1}^q (q-k+1) \bar{x}_{i,\gamma}^{t+k-1}. \quad (47)$$

This corresponds to the order 1 cut associated to passengers $(i,j+\gamma,t)$ with $\epsilon = 0$, which, therefore, does not need to be generated.

Similarly, let us consider the order 1 cut that is associated to $\epsilon = |i - j| - 1$. The cut is as follows:

$$y_{i,j}^t \geq q - \sum_{k=1}^q (q - k + 1) \bar{x}_{j-\gamma,\gamma}^{t+k-1+\bar{\tau}(\epsilon)}. \quad (48)$$

If constraints (45) are used in the formulation, we combine (48) with (45) to obtain the following:

$$y_{i-\gamma,j}^{t-\tau_{i-\gamma,i}} \geq y_{i,j}^t \geq q - \sum_{k=1}^q (q - k + 1) \bar{x}_{j-\gamma,\gamma}^{t+k-1+\bar{\tau}(\epsilon)}. \quad (49)$$

This corresponds to the order 1 cut associated to passengers $(i - \gamma, j, t - 1)$ with $\epsilon = |i - j|$ and $q = \bar{q}$, which, therefore, does not need to be generated. Therefore, in the implementation of the algorithm whenever we combine order 1 cuts with valid inequalities (44) and (45) we only generate order 1 cuts for passengers (i, j, t) such that $|i - j| = 1$. Similar considerations can also be made for higher-order cuts; however, in our experiments, we did not observe a significant benefit in expanding the generation of higher-order cuts.

6. Experimental Results

The objectives of our computational experiments are: 1) to validate the effectiveness of the developed formulations and cut generation algorithm, 2) to derive the most effective combination of cuts and valid inequalities, and 3) to assess the added value of the DDTP solutions, when compared to the frequently used regular timetables. In section 6.1, we discuss the implementation details, and the instance generation procedure. In section 6.2, we compare the computational performance of the FF formulation and the cut generation algorithm applied to the EFS formulation with different valid inequalities and separation strategies, and establish the best configuration of the algorithm. In section 6.3, we compare the solutions obtained using the DDTP to those obtained by the regular timetables. Lastly, in section 6.5, we apply the proposed cut generation algorithm on instances based on two real-world lines.

6.1. Implementation and Instance Generation

All experiments have been performed on a 2.10 Ghz Intel Xeon Silver 4116, with 16 GB of RAM and 16 cores, running Ubuntu 20.04.2 LTS. The formulation FF and the cut generation algorithm were solved using the CPLEX solver version 12.10.0.0 and has been implemented in Julia 1.5.3 using the JuMP modeling interface version 0.18.5. Each experiment was run on a single thread, with a time limit of 3600 seconds.

We implemented and tested several configurations of the developed algorithm employing different combinations of separation strategies and valid inequalities. For the separation of order 1 cuts, we consider two cases: $\epsilon = 0$, and $\epsilon = |i - j| - 1$. These two cases effectively correspond to focusing exclusively on the origin and destination stations of the passengers. In the first case, the value

imposed by the cut will depend only on the departing trains from station i . In the second case, the value imposed by the cut will depend only on the incoming trains at station j . Similarly, when separating order 2 cuts, we consider the case where $\eta = 0$ and $\theta = |i - j| - 1$. We now list the configurations of the algorithm that have been tested.

1. b – on integer nodes of the search tree, we separate violated cuts (31) using Algorithm 1.
2. o1 – on integer nodes of the search tree, for passengers $(i, j, t) \forall (i, j) \in D, t \in T$, we separate violated order 1 cuts (34) with $\epsilon = 0$ and $\epsilon = |i - j| - 1$, then, if no violated cuts were found, we separate violated cuts (31).
3. o2 – on integer nodes of the search tree, for passengers $(i, j, t) \forall (i, j) \in D, t \in T$, we separate violated order 2 cuts (43) with $\eta = 0$ and $\theta = |i - j| - 1$, then, if no violated cuts were found, we separate violated cuts (31).
4. o1-o2 – on integer nodes of the search tree, for passengers $(i, j, t) \forall (i, j) \in D, t \in T$, we separate violated order 1 cuts (34), followed by order 2 cuts (43), then, if no violated cuts were found, we separate violated cuts (31).
5. o1-c – we augment the formulation with valid inequalities (44) and (45). On integer nodes of the search tree, for passengers $(i, j, t) \forall (i, j) \in D, t \in T : |i - j| = 1$, we separate violated order 1 cuts (34) with $\epsilon = 0$ (note that since order 1 cuts are only separated if for passengers such that $|i - j| = 1$, $\epsilon = 0$ and $\epsilon = |i - j| - 1$ are equivalent). Then, if no violated cuts were found, we separate violated cuts (31).
6. o1-f – on fractional nodes of the search tree, for passengers $(i, j, t) \forall (i, j) \in D, t \in T$, we separate violated order 1 cuts (34) with $\epsilon = 0$ and $\epsilon = |i - j| - 1$ using Algorithm 2. On integer nodes of the search tree, we use configuration o1.
7. o1-c-f – we augment the formulation with valid inequalities (44) and (45). On fractional nodes of the search tree, for passengers $(i, j, t) \forall (i, j) \in D, t \in T : |i - j| = 1$, we separate violated order 1 (34) cuts with $\epsilon = 0$ using Algorithm 2. On integer nodes of the search tree, we use configuration o1-c.

Along with the previously reported cut generation configurations, we tested the flow formulation FF (Section 4). We note that numerous other configurations were considered such as combining order 2 cuts with chain constraints, but these were not competitive.

The formulations and cut generation algorithm are provided with an initial feasible solution. The initial solution is obtained by imposing a regular timetable on the entire line using all trains, without short-turning and idling. Specifically, trains are set to depart from the first station at constant time intervals, while respecting the discretization of the planning horizon, so as to be evenly spaced throughout the timetable. To obtain a complete feasible solution, we fix the timetable

variables and optimize the passenger related variables separately. We note that the computation of the initial solution requires negligible computational time.

We generated a set of instances with unitary distance between stations, i.e., such that $\tau_{i,j} = |i - j| \forall i, j \in S$, and $\rho = 1$. This assumption is relaxed in Section 6.4. We considered instances with $|S| = \{5, 10, 15, 20\}$ stations, and planning horizons of $|T| = \{10, 20, \dots, 100\}$ time-steps. The number of trains operating on the line is set to $\Psi = |S| - 1$. Additionally, g was set to 10. We note that this value serves as an upper bound, and is never reached in the experiments. Overall, the number of considered instances is 40.

The passengers arrive at their origin station at each time-step following a Poisson distribution. Each station of the line is assigned a popularity score. The mean of the Poisson distributions describing the passenger demand between two stations is the product of the popularity of the two stations. The popularity scores of the various stations were selected according to a bell distribution. The test instances can be found at:

<https://github.com/tommaso-schettini-polimi/DTP-Instances>

6.2. Algorithmic comparison

In this section, we evaluate the performance of FF and each of the seven configurations of the cut generation algorithm. Tables 1 and 2 report the related results. Each line corresponds to ten instances with a given number of stations, which is reported in the first column. For each tested method, we report the number of instances solved to optimality out of 10 ($\#s$), the average solution times for all ten instances in seconds ($t(s)$), the average objective value achieved (TWT), and the average optimality gaps in percent. Detailed results are found in Table 9, in Appendix C.

Compared to FF, the tables show overall better gaps and better average computational time for all configurations of the cut generation algorithm, with the exception of configuration b, which performed worse than FF. The results show the merits of explicitly considering order 1 and order 2 cuts before resorting to the general cuts. When the separation is performed exclusively on integer solutions (i.e., configurations o1, o2, and o1-o2) all configurations of the method outperform FF, and significantly improve the performance of the algorithm over configuration b. Combining order 1 and order 2 cuts further improves the average achieved gaps. The usage of chain constraints in configuration o1-c improves the algorithm in terms of the number of solved instances but leads to worse optimality gaps on larger instances, specifically on the 20 stations line. Performing the separation at fractional solutions in configuration o1-f improves upon the algorithm in terms of optimality gaps and the quality of the solutions. The usage of chain constraints in configuration o1-c-f leads to the best overall optimality gaps, and best solutions in terms of TWT. Overall, we conclude that o1-c-f is the most effective variant of the algorithm, achieving the lowest gaps with 24

instances solved to optimality. Using o1-c-f, the cut generation algorithm is able to solve instances with up to 60 time-steps on all line sizes within an optimality gap of 1.8% or less. Therefore, in the rest of the computational results, we will use configuration o1-c-f.

Table 1 Results of the formulation and cut generation algorithms.

Formulation FF, configurations b, o1, o2.

FF					b				o1				o2			
$ S $	#s	t(s)	TWT	gap(%)	#s	t(s)	TWT	gap(%)	#s	t(s)	TWT	gap(%)	#s	t(s)	TWT	gap(%)
5	6	1734	2114	1.46	6	1575	2130	1.82	7	1179	2111	0.50	8	1101	2116	0.53
10	4	2439	3129	2.73	4	2203	3110	3.22	7	1475	3072	0.87	6	1645	3070	0.89
15	3	2888	4510	6.37	3	2863	4496	10.88	4	2238	4310	2.76	4	2305	4321	3.48
20	2	2916	5205	11.87	2	2892	5234	16.48	4	2326	4894	4.56	4	2525	4945	5.14
all	15	2494	3740	5.61	15	2383	3743	8.10	22	1805	3597	2.17	22	1894	3613	2.51

Table 2 Results of the cut generation algorithms.

Configurations o1-o2, o1-c, o1-f, o1-c-f.

o1-o2					o1-c				o1-f				o1-c-f			
$ S $	#s	t(s)	TWT	gap(%)	#s	t(s)	TWT	gap(%)	#s	t(s)	TWT	gap(%)	#s	t(s)	TWT	gap(%)
5	7	1134	2114	0.51	8	914	2111	0.36	8	982	2110	0.34	8	900	2111	0.31
10	7	1507	3060	0.72	7	1484	3061	0.63	7	1546	3076	0.91	8	1470	3061	0.51
15	4	2305	4307	3.07	4	2285	4276	2.43	5	2174	4312	2.61	4	2244	4279	2.10
20	4	2409	4830	3.55	4	2376	5010	6.00	4	2360	4907	3.22	4	2328	4882	2.79
all	22	1839	3578	1.96	23	1765	3615	2.35	24	1766	3601	1.77	24	1736	3583	1.43

Table 3 Percentage improvement of the DDTP solutions over the regular timetables.

$ S $	$\frac{(TWT_R - TWT_D)}{TWT_R} (\%)$		$\frac{(MAX_OCC_R - MAX_OCC_D)}{MAX_OCC_R} (\%)$		$\frac{(AVG_OCC_R - AVG_OCC_D)}{AVG_OCC_R} (\%)$		$\frac{(\sigma^2(OCC_R) - \sigma^2(OCC_D))}{\sigma^2(OCC_R)} (\%)$		$\frac{(DIST_R - DIST_D)}{DIST_R} (\%)$	
	all	optimal	all	optimal	all	optimal	all	optimal	all	optimal
5(6)	2.27	2.70	0.50	-0.56	-3.44	-3.82	2.91	2.87	2.99	3.29
10(6)	12.07	12.63	22.90	24.50	-4.00	-4.34	34.97	35.86	3.84	4.16
15(2)	11.60	15.08	20.24	19.99	-4.70	-4.91	24.58	21.35	4.45	4.71
20(2)	9.59	12.81	10.11	3.14	-4.55	-4.34	4.82	8.73	4.38	4.16
all(16)	8.88	10.81	13.44	11.77	-4.17	-4.35	16.82	17.20	3.92	4.08

6.3. Comparison to a regular timetable

We compare the solutions obtained by the DDTP to the regular timetables. The regular timetable solutions were computed as detailed in section 6.1. For fairness, we now compare to a demand-adapted regular timetable. This was done by enumerating the possible starting times for the regular timetable and selecting the one that minimizes the total passenger waiting time. In the comparison, horizons of 10 and 20 time-steps are ignored as they would bias the results in favor of the DDTP. Indeed, on instances with a short planning horizon, the DDTP achieves disproportionately large improvements over the regular timetable. This is due to the interaction of the optimization with

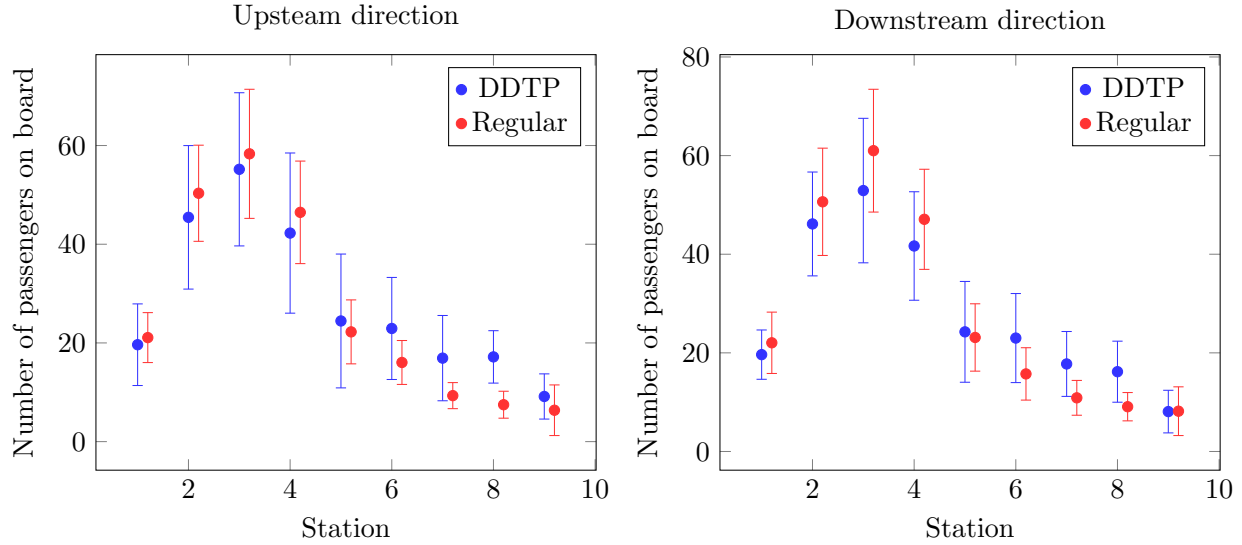


Figure 5 Number of passengers on board a train registered over the timetable of instance 10-50. Blue ticks represent the DDTP solution, red ticks represent the regular timetable. The plotted error-bars correspond to one standard deviation.

the end of the horizon effect. As an example, on the 20-10 instance, DDTP is able to achieve a reduction in TWT of 40.5%. However, on the 20-40 instance, the reduction in TWT is limited to 12.6%.

For a given line size, Table 3 reports the comparison of the total waiting time achieved by the DDTP timetables to the waiting time achieved by a regular timetable. Let (TWT_D) denote the total waiting time achieved by the DDTP and (TWT_R) denote the total waiting time achieved by the regular timetable, the value reported in the table is computed as $\frac{(TWT_R - TWT_D)}{TWT_R}$ (%). Columns denoted by “all” report the comparison over all considered instances, considering the best found feasible solution. Columns denoted by “optimal” report the comparison only considering instances that are solved to optimality using configuration o1-c-f. The number of solutions accounted for in the “optimal” columns is reported in parenthesis after the number of stations. Detailed results are found in Table 10, in Appendix C.

Considering all instances, the DDTP is able to improve the total waiting times of the passengers by 8.88%. Considering instances solved to optimality, the TWT is further reduced, achieving an average improvement of the passengers’ waiting time of 10.81%. However, we note that the benefits of the DDTP vary significantly depending on the considered line. In particular, we note that on the five-station instances, the improvement achieved by the DDTP is relatively low. This is due to the limited size of the line, which limits the advantages that can be gained by the DDTP.

We compared the DDTP solutions to the regular timetable also with respect to the number of passengers on board a train. This measure is a proxy for the realised occupancy of a train. Given

the best found solution, we assumed that passengers will board the first train headed towards their destination, and will alight a train at the station where it short-turns. Furthermore, alighting passengers leave the train immediately upon arrival at their destination station, even if the train idles afterwards. Boarding passengers will not board an idling train, they will board the train upon its departure. We then computed the number of passengers on board a train at each time-step. Table 3 reports the comparison between the DDTP solutions and the regular timetables in terms of the maximum number of passengers on board a train, the mean and the variance of the number passengers on board a train. The maximum number of passengers on board a train is denoted by MAX_OCC . The mean value of the passengers on board a train is denoted by AVG_OCC and its variance is denoted by $\sigma^2(OCC)$. The values reported in the table are computed as $\frac{(MAX_OCC_R - MAX_OCC_D)}{MAX_OCC_R}$, $\frac{(AVG_OCC_R - AVG_OCC_D)}{AVG_OCC_R}$, and $\frac{(\sigma^2(OCC_R) - \sigma^2(OCC_D))}{\sigma^2(OCC_R)}$.

We note that the DDTP does not explicitly address the number of passengers on board a train in its objective function. Despite this fact, the timetables produced by the DDTP affect the number of passengers on board in a consistent manner. Specifically, we observe that, on average, the maximum number of passengers on board a train is reduced by nearly 12% compared to the regular timetables. This suggests that the model reduces the number of passengers accumulating at the station for the purpose of minimizing the passenger total waiting time. Additionally, the average number of passengers on board a train increases by about 4%, compared to a regular timetable. Furthermore, we observe that the variance of the number of passengers on board is also reduced. This implies that the number of passengers on board a train tends to fluctuate less over the DDTP timetable, i.e., trains are less frequently completely empty or completely full. In Figure 5, we plot the average passengers on board a train at each station of instance 10-50 in both directions for the regular and the optimal DDTP timetable. We observe that the average number of passengers on board a train is reduced in stations with high demand and increased in stations with low demand.

Lastly, Table 3 reports a comparison of the total distance traversed by the trains during the timetable (DIST). Similarly to the other reported values, the value reported is computed as: $\frac{(DIST_R - DIST_D)}{DIST_R}$ (%). We observe that the DDTP timetables cause a decrease of around 3.9% in the traveled distance. This also implies a reduction in the operating costs of the trains.

The usage of short-turning implies that not all passengers will benefit uniformly across the line. Passengers at high demand stations will, on average, wait less than passenger at low demand stations. Considering instance 10–50, in Table 4, we report the total waiting time experienced passengers originating from each station for the optimal DDTP timetable and the regular timetable. We observe that in the DDTP timetable, passengers originating from stations 6–10 wait more, compared to the regular timetable. However, this increase is compensated by a significant reduction in the waiting times related to passengers originating from stations 1–5. Additionally, we note that

despite the increase in observed waiting times, in the DDTP timetable, no passenger waits for more than four time-steps, and only 0.26% of the passengers wait for precisely four time-steps.

In the DDTP, the number of available trains is given as an input. The optimization will use all available trains to minimize the passenger waiting time. We tested the difference in the waiting time achieved by a DDTP timetable using one train less than the original setting. In Table 5, we report the passenger waiting for the instances with ten stations, where the original number of trains was $\Psi = 9$. We denote by TWT_R^ψ the TWT achieved by the regular timetable using ψ trains, and by TWT_D^ψ the TWT achieved by the DDTP timetable using ψ trains. We report the TWT achieved by the regular timetable using nine trains, and the passenger waiting time achieved by the DDTP using eight and nine trains. We also report the comparison between the DDTP timetables and the regular timetables. The reported value is computed as $\frac{(TWT_R - TWT_D)}{TWT_R}$ (%), where TWT_D represents the waiting time achieved by the DDTP timetable, and TWT_R represents the waiting time achieved by the regular timetable. As it can be seen, the DDTP is frequently able to improve on the TWT of the regular timetable (with nine trains) using eight trains. Considering instances with 30 time-steps and above, using eight trains, the DDTP with eight trains achieves a waiting time that is on average 3.60% more than the waiting time achieved by the regular timetable using nine trains.

Table 4 The total passenger waiting time depending on the passenger's origin station.

	Origin station										Total
	1	2	3	4	5	6	7	8	9	10	
Regular Timetable	345	598	551	716	497	127	102	34	53	90	3113
DDTP Timetable	251	448	437	524	400	152	197	61	103	150	2723
Regular - DDTP	94	150	114	192	97	-25	-95	-27	-50	-60	390

6.4. Experiments with non-unitary travel times

In this section, we modified the generated instances by increasing the travel times between two consecutive stations to two time-steps in about one fourth of the cases. Furthermore, we set the short-turning time at all stations ρ to two time-steps. The full test instances can be found at:

<https://github.com/tommaso-schettini-polimi/DTP-Instances>

We used the developed cut generation algorithm in configuration o1-c-f. Table 6 reports the related results. Detailed results are found in Table 11, in Appendix C. Overall we observe that the cut-generation algorithm performs similarly on the modified instances solving 27 instances to optimality, and achieving an average optimality gap of 2.46%.

Table 5 The passenger waiting time depending on the number of trains used in the timetable. Instances marked with an * are not solved to optimality.

$ S $	$ T $	TWT_R^9	TWT_D^9	$\frac{(TWT_R^9 - TWT_D^9)}{TWT_R^9}(\%)$	TWT_D^8	$\frac{(TWT_R^9 - TWT_D^8)}{TWT_R^9}(\%)$
10	10	573	442	22.9	528	7.9
	20	1243	1049	15.6	1289	-3.7
	30	1889	1645	12.9	1935	-2.4
	40	2524	2214	12.3	2657	-5.3
	50	3113	2723	12.5	3192	-2.5
	60	3816	3335	12.6	3922	-2.8
	70	4512	3890	13.8	4597	-1.9
	80	5102	4507	11.7	5315	-4.2
	90	5681	5087*	10.5	5939	-4.5
	100	6371	5715	10.3	6699	-5.1

Table 6 Results on instances with non-unitary travel times using o1-c-f.

$ S $	#s	t(s)	obj	gap(%)
5	10	14	3408	0.00
10	8	1200	4797	0.29
15	5	1976	6768	2.50
20	4	2425	8022	7.06
all	27	1404	5749	2.46

Table 7 Comprehensive computational results on the realistic instances using o1-c-f.

City	$ T $	t(s)	TWT	gap(%)
Madrid	10	2	2846	0.0
	20	34	5976	0.0
	30	683	9551	0.0
	40	3600	12895	3.7
	50	3600	16427	5.8
	60	3600	19427	2.6
	70	3600	23534	7.5
	80	3600	27143	7.1
	90	3600	30757	9.2
	100	3600	34204	8.4
Milan	10	2	520	0.0
	20	20	1251	0.0
	30	420	1966	0.0
	40	3600	2665	0.2
	50	3600	3354	0.5
	60	3600	4044	1.8
	70	3600	4709	0.5
	80	3600	5633	3.1
	90	3600	6300	4.3
	100	3600	7092	4.4

6.5. Application to two realistic instances

In this section, we apply the developed cut generation algorithm to instances generated from data from real-world lines. For the first line, we used data from the Madrid metro line, introduced by Canca et al. (2016), where the authors provide the hourly demand matrix for the line and the station distances. The line has 10 stations and is served by nine trains. For the second line, we

Table 8 Comprehensive comparison of the DDTP timetable w.r.t. the Regular timetable on realistic instances.
Instances reported with * are not solved to optimality.

City	$ T $	$\frac{(TWT_R - TWT_D)}{TWT_R} (\%)$	$\frac{(MAX_OCC_R - MAX_OCC_D)}{MAX_OCC_R} (\%)$	$\frac{(AVG_OCC_R - AVG_OCC_D)}{AVG_OCC_R} (\%)$	$\frac{(\sigma^2(OCC_R) - \sigma^2(OCC_D))}{\sigma^2(OCC_R)} (\%)$	$\frac{(DIST_R - DIST_D)}{DIST_R} (\%)$
Madrid	10	17.6	1.5	-5.1	23.3	3.4
	20	16.7	26.7	-3.7	38.9	4.7
	30	13.7	33.7	-3.9	40.6	2.5
	40 *	12.7	25.3	-4.5	34.0	3.1
	50 *	9.9	28.1	-3.9	28.0	4.1
	60 *	10.8	31.3	-5.1	38.9	5.7
	70 *	10.0	25.4	-4.8	35.8	4.1
	80 *	6.6	27.7	-5.6	35.5	5.9
	90 *	4.8	28.5	-4.6	28.5	4.9
	100 *	8.6	28.5	-5.1	33.8	5.0
Milan	10	31.8	19.4	-2.7	56.1	4.6
	20	18.8	0.0	-1.2	37.6	2.4
	30	15.3	3.7	-1.7	29.9	2.9
	40 *	12.3	0.9	-1.1	27.1	2.7
	50 *	10.9	16.7	-1.6	26.8	2.9
	60 *	11.6	3.4	-2.7	26.7	3.1
	70 *	11.7	4.0	-1.7	28.3	2.1
	80 *	10.5	5.5	-2.1	30.4	2.5
	90 *	9.9	-4.0	-2.3	25.8	2.1
	100 *	3.8	-20.8	-2.0	23.6	2.3

generated a set of instances modeled after the Milan metro line M5, which is a 19 station metro line serviced by 21 trains. Distances were taken from the real line. The demand profiles for the line have been generated using the same procedure used for the first set of instances described in section 6.1. For both sets of instances, we used a time-step length of one minute and rounded the travel times between the stations to the nearest integer.

Table 7 reports the performance achieved by the cut generation algorithm (using configuration o1-c-f) on the realistic instances. For each instance, we report the solution time in seconds (t(s)), the objective value (TWT), and the achieved gap.

We observe that the realistic instances are significantly more complex to solve. On the Madrid instances, we solve to optimality instances with up to 30 time-steps and achieve a gap of 5.8% or less on all instances with up to 60 time-steps. Similarly, on the Milan instances, we are able to solve to optimality instances with up to 30 time-steps, and solve all instances up to 70 time-steps with a gap of 1.8% or less.

Table 8 compares the TWT and number of passengers on board a train achieved by the DDTP timetables to the TWT and number of passengers on board a train achieved by the regular timetables. With regards to the total waiting time, on the Madrid instances, the DDTP timetable improves the TWT of the passengers by 13.7% on the 30 time-step instance, and by 12.7% on the 40 time-step instance (which is solved with a 3.7% optimality gap). On the Milan instances, the DDTP improves the TWT of the passengers by 15.3% on the 30 time-steps instance, and by 12.3% on the 40 time-steps instance (which is solved with a 0.2% optimality gap).

With regards to the number of passengers on board a train, we observe a similar behavior to the instances analyzed in section 6.3. We observe an increase in the average number of passengers

on board a train, accompanied by a sharp decrease in the variance, and frequently a reduced maximum number of passengers on board a train over the horizon. However, we note that in some instances, the maximum number of passengers on board a train increases. Additionally, we observed a decrease of around 3% in the total distance covered by the trains.

7. Conclusions

We considered the demand-driven timetabling problem associated with a two-directional metro corridor. In the DDTP, no predetermined structure is imposed on the timetable, and trains are allowed to short-turn. In particular, all the timetabling decisions are determined directly by the optimization. Compared to traditional approaches, the resulting setting allows considerably more flexibility in planning the service. The DDTP can be further expanded to massive commuter railway systems involving more complex network structures.

We developed two mixed integer linear formulations for the DDTP. The latter included an exponential number of constraints, for which we developed an efficient exact algorithm using a cut generation scheme coupled with additional valid inequalities. Our computational results show the effectiveness of the cut generation algorithms considering various configurations. We also demonstrated the potential added value of the proposed strategy when compared to the frequently used regular timetables. The produced demand-driven timetables are shown to offer several operational advantages and improve the service quality in terms of the total waiting time of the passengers. Furthermore, the maximum number of passengers on board a train also decreases. Similar results were also observed in experiments on data derived from two real-world lines.

The ideas presented in this paper refer to a relatively new concept for demand-driven timetabling. The DDTP is optimized from a system optimum standpoint. Thus, the paper serves as a benchmark to give an insight regarding the potential benefits of applying the DDTP. Despite having devised strong algorithms for the problem, many steps still need to be taken before the strategies entailed by the DDTP can be applied in practice. As such, this paper constitutes a first and fundamental step to measuring the effectiveness of heuristics, which could account for more realistic features. For instance, elaborating the objective function to account for passenger transfers induced by the used short-turning strategy. Additionally, a more explicit treatment of dwelling time may be explored. Furthermore, explicitly modeling train capacities would be interesting in certain settings. Finally, developing algorithms to handle metro networks is a promising future research direction.

References

- Barrena E, Canca D, Coelho LC, Laporte G, 2014a *Exact formulations and algorithm for the train timetabling problem with dynamic demand. Computers & Operations Research* 44:66–74.

- Barrena E, Canca D, Coelho LC, Laporte G, 2014b *Single-line rail rapid transit timetabling under dynamic passenger demand. Transportation Research Part B: Methodological* 70:134–150.
- Cacchiani V, Caprara A, Toth P, 2008 *A column generation approach to train timetabling on a corridor. 4OR* 6:125–142.
- Canca D, Barrena E, Algaba E, Zarzo A, 2014 *Design and analysis of demand-adapted railway timetables. Journal of Advanced Transportation* 48:119–137.
- Canca D, Barrena E, Laporte G, Ortega FA, 2016 *A short-turning policy for the management of demand disruptions in rapid transit systems. Annals of Operations Research* 246:145–166.
- Canca D, Barrena E, Zarzo A, Ortega F, Algaba E, 2012 *Optimal train reallocation strategies under service disruptions. Procedia-Social and Behavioral Sciences* 54:402–413.
- Caprara A, Fischetti M, Toth P, 2002 *Modeling and solving the train timetabling problem. Operations Research* 50:851–861.
- Caprara A, Monaci M, Toth P, Guida PL, 2006 *A lagrangian heuristic algorithm for a real-world train timetabling problem. Discrete Applied Mathematics* 154:738–753.
- Ceder A, 2016 *Public transit planning and operation: Modeling, practice and behavior* (Boca Raton, USA: CRC press).
- Cortés CE, Jara-Díaz S, Tirachini A, 2011 *Integrating short turning and deadheading in the optimization of transit services. Transportation Research Part A: Policy and Practice* 45:419–434.
- Farahani RZ, Miandoabchi E, Szeto WY, Rashidi H, 2013 *A review of urban transportation network design problems. European Journal of Operational Research* 229:281–302.
- Furth PG, 1987 *Short turning on transit routes. Transportation Research Record* 1108:42–52.
- Gao Y, Kroon L, Schmidt M, Yang L, 2016 *Rescheduling a metro line in an over-crowded situation after disruptions. Transportation Research Part B: Methodological* 93:425–449.
- Goerigk M, Liebchen C, 2017 *An improved algorithm for the periodic timetabling problem. Leibniz International Proceedings in Informatics* (Schloss Dagstuhl-Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing).
- Guihaire V, Hao JK, 2008 *Transit network design and scheduling: A global review. Transportation Research Part A: Policy and Practice* 42:1251–1273.
- Kang L, Wu J, Sun H, Zhu X, Wang B, 2015 *A practical model for last train rescheduling with train delay in urban railway transit networks. Omega* 50:29–42.
- Kang L, Zhu X, 2016 *A simulated annealing algorithm for first train transfer problem in urban railway networks. Applied Mathematical Modelling* 40:419–435.
- Kiefer A, Kritzing S, Doerner KF, 2016 *Disruption management for the viennese public transport provider. Public Transport* 8:161–183.

- Kroon L, Huisman D, Abbink E, Fioole PJ, Fischetti M, Maróti G, Schrijver A, Steenbeek A, Ybema R, 2009 *The new dutch timetable: The OR revolution*. *Interfaces* 39:6–17.
- Kroon L, Maróti G, Helmrich MR, Vromans M, Dekker R, 2008 *Stochastic improvement of cyclic railway timetables*. *Transportation Research Part B: Methodological* 42:553–570.
- Liebchen C, 2008 *The first optimized railway timetable in practice*. *Transportation Science* 42:420–435.
- Liebchen C, Möhring RH, 2007 *The modeling power of the periodic event scheduling problem: railway timetables—and beyond*. *Algorithmic Methods for Railway Optimization*, 3–40 (Springer).
- Liebchen C, Schachtebeck M, Schöbel A, Stiller S, Prigge A, 2010 *Computing delay resistant railway timetables*. *Computers & Operations Research* 37:857–868.
- Mesa JA, Ortega FA, Pozo MA, 2009 *Effective allocation of fleet frequencies by reducing intermediate stops and short turning in transit systems*. *Robust and Online Large-Scale Optimization*, 293–309 (Springer).
- Niu H, Zhou X, Gao R, 2015 *Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints*. *Transportation Research Part B: Methodological* 76:117–135.
- Pätzold J, Schöbel A, 2016 *A matching approach for periodic timetabling*. *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik).
- Schmöcker JD, Cooper S, Adeney W, 2005 *Metro service delay recovery: comparison of strategies and constraints across systems*. *Transportation Research Record* 1930:30–37.
- Serafini P, Ukovich W, 1989 *A mathematical model for periodic scheduling problems*. *SIAM Journal on Discrete Mathematics* 2:550–581.
- Siebert M, Goerigk M, 2013 *An experimental comparison of periodic timetabling models*. *Computers & Operations Research* 40:2251–2259.
- Sun L, Jin JG, Lee DH, Axhausen KW, Erath A, 2014 *Demand-driven timetable design for metro services*. *Transportation Research Part C: Emerging Technologies* 46:284–299.
- Tirachini A, Cortés CE, Jara-Díaz SR, 2011 *Optimal design and benefits of a short turning strategy for a bus corridor*. *Transportation* 38:169–189.
- UN, 2015 *World urbanization prospects: The 2014 revision*. United Nations Department of Economics and Social Affairs, Population Division: New York, NY, USA .
- Yang L, Yao Y, Shi H, Shang P, 2020 *Dynamic passenger demand-oriented train scheduling optimization considering flexible short-turning strategy*. *Journal of the Operational Research Society* 1–19.

Appendix A: Summary of Notation

Parameters	$S = \{1, \dots, m\}$	the ordered set of stations
	$\Gamma = \{-1, 1\}$	the set directions. 1 is the upstream direction, -1 is the downstream direction
	$\gamma(i, j)$	the direction associated to stations pair i, j
	F_γ	the first station of the line in direction γ
	$T = \{1, \dots, h\}$	the set of time-steps
	$\tau_{i,j}$	time-steps required to go from station i to station j
	ρ	time-steps required to perform a short turn
	g	maximum waiting time experienced by the passengers
	D	the set of origin-destination pairs OD of passengers
	$a_{i,j}^t$	the number of passengers that arrive at station i with destination j at time-step t
Decision variables	Ψ	number of trains servicing the line
	$x_{i,\gamma}^t$	one if a train departs from station i in direction γ at time-step t ; and zero otherwise
	$u_{i,\gamma}^t$	one if a train in direction γ short-turns at station i at time-step t ; and zero otherwise
	$z_{i,\gamma}^t$	one if a train in direction γ idles at station i at time-step t ; and zero otherwise
	$y_{i,j}^t$	the waiting time of an individual passenger arriving at station i at time t , to reach their destination j
Graph-related notation	$G = \{N, A\}$	the spacetime graph of the line.
	$G_{i,j}^t = \{N_{i,j}^t, A_{i,j}^t\}$	the passenger graph
	$G_{i,j}^t(\bar{x})$	the auxiliary graph associated to timetable \bar{x}
	$\mathcal{CS}(i, j, t, q)$	the family of cut-sets separating nodes (i, t) and $(j, t + \tau_{i,j} + q - 1)$ on $G_{i,j}^t$
	$\mathcal{K}(c, \bar{x})$	the sum of capacities of the arcs contained in cut-set $c \in \mathcal{CS}(i, j, t, q)$ on $G_{i,j}^t(\bar{x})$
	$C_{i,j}^{t,q}$	the set of all integer vectors $\vec{w} \in \mathbb{N}_0^q$, such that the integer numbers contained in \vec{w} are non-decreasing with the index, and all the values are in the interval $[0, i - j - 1]$

Appendix B: Proof of Proposition 1

Proof. If (i, t) and $(j, t + \tau_{i,j} + q)$ are not connected, the path the passengers take will include at least $q + 1$ waiting time-steps.

Without loss of generality, assume the passengers will board any train that can bring them closer to their destination, even if this implies having to alight before reaching j . Therefore, waiting will only occur when the passengers reach a station such that $\bar{x}_{i+\gamma\eta,\gamma}^{t'} = 0$. Additionally, if the first waiting time-step of the passengers occurs at station $i + \gamma\eta$, we can conclude that that station has been reached as early as possible, i.e., at time-step $t + \eta$. Thus, let $i + \gamma\vec{w}[1]$ be the station where passengers (i, j, t) wait their first time-step, it holds: $\bar{x}_{i+\gamma\vec{w}[1],\gamma}^{t+\bar{\tau}(\vec{w}[1])} = 0$.

If $q \geq 1$, we can apply a similar reasoning to the second waiting time-step of passengers (i, j, t) . Let $i + \gamma\vec{w}[2]$ be the station where the second waiting time-step occurs (which may coincide with $i + \gamma\vec{w}[1]$), we can conclude that: $\bar{x}_{i+\gamma\vec{w}[2],\gamma}^{t+1+\bar{\tau}(\vec{w}[2])} = 0$. The same reasoning can be applied to the successive waiting time-steps if $q \geq 2$, and so on. Thus, let vector \vec{w} describe the position of the first q waiting time-steps of passengers (i, j, t) , by construction it holds:

$$\sum_{k=1}^q \bar{x}_{i+\gamma\vec{w}[k],\gamma}^{t+k-1+\bar{\tau}(\vec{w}[k])} = 0$$

It is always possible to identify q elements since there is no path between (i, t) and $(j, t + \tau_{i,j} + q)$. \square

Appendix C: Tables of results

In this section, we present the comprehensive tables of results for the computational experiments presented in section 6.

Table 9 Comprehensive results of FF and cut generation algorithms. Optimality gaps are reported in percent.

Instance	FF		b		o1		o2		o1-o2		o1-c		o1-f		o1-c-f	
S	T	t(s)	obj	gap(%)	t(s)	obj	gap(%)	t(s)	obj	gap(%)	t(s)	obj	gap(%)	t(s)	obj	gap(%)
5	10	1	366	0.0	1	366	0.0	2	366	0.0	1	366	0.0	2	366	0.0
	20	7	687	0.0	1	687	0.0	1	687	0.0	1	687	0.0	1	687	0.0
	30	11	1113	0.0	3	1113	0.0	3	1113	0.0	2	1113	0.0	2	1113	0.0
	40	186	1539	0.0	24	1539	0.0	21	1539	0.0	13	1539	0.0	15	1539	0.0
	50	121	1966	0.0	76	1966	0.0	14	1966	0.0	7	1966	0.0	10	1966	0.0
	60	2613	2308	0.0	1245	2308	0.0	350	2308	0.0	206	2308	0.0	171	2308	0.0
	70	3600	2648	1.5	3600	2645	0.6	565	2645	0.0	428	2645	0.0	326	2645	0.0
	80	3600	3050	3.9	3600	3041	0.7	2998	3041	0.0	3600	3041	0.4	1420	3041	0.0
	90	3600	3542	3.9	3600	3539	1.6	3600	3539	1.4	3600	3538	1.1	3600	3537	1.4
	100	3600	3923	5.4	3600	3905	2.6	3600	3951	3.8	3600	3938	3.2	3600	3909	2.5
10	10	1	442	0.0	1	442	0.0	0	442	0.0	0	442	0.0	0	442	0.0
	20	27	1049	0.0	5	1049	0.0	1	1049	0.0	1	1049	0.0	1	1049	0.0
	30	182	1645	0.0	78	1645	0.0	16	1645	0.0	11	1645	0.0	17	1645	0.0
	40	2578	2214	0.0	344	2214	0.0	48	2214	0.0	75	2214	0.0	61	2214	0.0
	50	3600	2723	1.6	3600	2723	0.6	468	2723	0.0	1259	2723	0.0	1581	2723	0.0
	60	3600	3393	3.9	3600	3363	2.6	2362	3335	0.0	3600	3341	0.8	1179	3335	0.0
	70	3600	3905	1.3	3600	3932	3.0	1037	3890	0.0	722	3890	0.0	821	3890	0.0
	80	3600	4570	3.6	3600	4678	7.0	3600	4527	1.2	3600	4518	0.7	3600	4526	1.2
	90	3600	5290	8.0	3600	5267	11.1	3600	5069	2.8	3600	5097	3.4	3600	5073	3.0
	100	3600	6057	8.9	3600	5788	8.0	3600	5825	4.6	3600	5784	4.0	3600	5706	2.9
15	10	5	495	0.0	1	495	0.0	1	495	0.0	1	495	0.0	1	495	0.0
	20	161	1383	0.0	45	1383	0.0	18	1383	0.0	11	1383	0.0	7	1383	0.0
	30	3515	2205	0.0	3381	2205	0.0	210	2205	0.0	211	2205	0.0	187	2205	0.0
	40	3600	3101	5.1	3600	3060	3.7	3600	3027	0.5	3600	3034	0.9	3600	3027	0.2
	50	3600	3916	4.8	3600	3884	7.3	640	3798	0.0	1224	3798	0.0	1043	3798	0.0
	60	3600	4651	6.3	3600	4653	11.9	3600	4540	2.3	3600	4545	2.7	3600	4536	2.4
	70	3600	5725	8.0	3600	5961	21.8	3600	5575	4.3	3600	5573	4.3	3600	5648	5.7
	80	3600	6445	7.9	3600	6495	18.9	3600	6450	7.0	3600	6311	6.0	3600	6330	5.3
	90	3600	8082	14.8	3601	7661	18.7	3600	7374	5.9	3600	7632	10.9	3600	7369	8.0
	100	3600	9097	16.8	3600	9163	26.5	3600	8253	7.5	3600	8234	10.1	3600	8281	9.0
20	10	3	552	0.0	2	552	0.0	2	552	0.0	2	552	0.0	1	552	0.0
	20	356	1446	0.0	119	1446	0.0	25	1446	0.0	26	1446	0.0	30	1446	0.0
	30	3600	2533	2.4	3600	2515	1.3	222	2504	0.0	321	2504	0.0	300	2504	0.0
	40	3600	3454	3.8	3600	3419	6.7	1416	3391	0.0	2146	3391	0.0	1823	3391	0.0
	50	3600	4411	6.9	3600	4520	20.8	3600	4272	2.2	3600	4292	3.2	3600	4275	2.3
	60	3600	5520	7.6	3600	5753	23.8	3600	5243	1.8	3600	5217	1.1	3600	5222	1.4
	70	3600	7065	14.5	3600	7065	25.0	3600	6311	6.2	3600	6294	4.6	3600	6288	2.9
	80	3600	8011	26.9	3600	8011	29.0	3600	7377	11.2	3600	7054	9.3	3600	7311	11.9
	90	3600	9049	28.8	3600	9049	28.8	3600	8367	7.0	3600	8369	16.9	3600	8349	7.0
	100	3600	10012	29.7	3600	10012	29.2	3600	9477	17.3	3600	9656	16.3	3600	9033	10.4

Table 10 Comprehensive results comparing the percentage improvement of the DDTP solutions with the regular timetables. Instances marked with an * are not solved to optimality.

$ S $	$ T $	$\frac{(TWT_R - TWT_D)}{TWT_R} (\%)$	$\frac{(MAX_OCC_R - MAX_OCC_D)}{MAX_OCC_R} (\%)$	$\frac{(AVG_OCC_R - AVG_OCC_D)}{AVG_OCC_R} (\%)$	$\frac{(\sigma^2(OCC_R) - \sigma^2(OCC_D))}{\sigma^2(OCC_R)} (\%)$	$\frac{(DIST_R - DIST_D)}{DIST_R} (\%)$
5	10	1.3	1.9	-1.6	12.7	5.7
	20	3.4	-17.3	-11.1	4.1	7.4
	30	4.2	0.0	-6.4	1.7	4.0
	40	3.1	-3.3	-4.0	2.3	4.5
	50	2.5	0.0	-5.0	2.7	3.0
	60	0.9	0.0	-3.2	1.7	2.6
	70	2.9	0.0	-1.3	1.7	1.8
	80	2.7	0.0	-3.1	7.2	3.8
	90 *	0.7	0.0	-1.6	4.1	2.1
	100 *	1.3	7.4	-3.0	2.0	2.2
10	10	22.9	9.5	-4.3	32.5	3.4
	20	15.6	12.9	-5.9	35.3	5.3
	30	12.9	36.7	-5.8	38.8	5.6
	40	12.3	26.0	-5.6	35.9	5.4
	50	12.5	24.1	-5.2	38.2	4.6
	60	12.6	19.0	-2.2	32.5	2.4
	70	13.8	26.4	-2.6	32.8	2.4
	80	11.7	14.8	-4.7	36.9	4.6
	90 *	10.5	18.6	-3.7	34.1	3.5
	100 *	10.3	17.6	-2.2	30.5	2.2
15	10	39.4	10.2	-6.2	47.1	4.2
	20	18.3	21.9	-3.4	26.0	3.3
	30	15.8	15.3	-5.8	19.8	5.7
	40 *	14.9	21.4	-4.4	22.5	4.3
	50	14.3	24.7	-4.0	22.9	3.8
	60 *	14.4	21.0	-4.1	30.3	3.8
	70 *	10.0	20.0	-5.3	26.2	5.2
	80 *	10.9	25.2	-5.2	28.4	4.7
	90 *	7.6	5.9	-5.0	21.2	4.7
	100 *	9.1	23.6	-4.9	20.6	4.7
20	10	40.5	31.6	-6.3	52.2	5.1
	20	22.6	11.9	-4.9	22.8	5.6
	30	13.1	2.9	-4.3	7.5	4.1
	40	12.6	3.4	-4.4	10.0	4.2
	50 *	12.3	7.0	-4.3	8.4	4.1
	60 *	12.0	18.8	-5.1	4.2	4.8
	70 *	7.3	8.9	-4.5	2.0	4.4
	80 *	8.3	19.0	-4.8	8.8	4.7
	90 *	6.7	18.6	-4.6	-2.7	4.4
	100 *	4.6	2.4	-4.4	0.4	4.3

Table 11 Comprehensive results on instances with non-unitary travel times using o1-c-f.

$ S $	$ T $	t(s)	obj	gap(%)
5	10	2	547	0.0
	20	0	1140	0.0
	30	1	1822	0.0
	40	1	2475	0.0
	50	6	3222	0.0
	60	9	3742	0.0
	70	8	4277	0.0
	80	23	4914	0.0
	90	41	5695	0.0
	100	45	6247	0.0
10	10	0	641	0.0
	20	4	1647	0.0
	30	32	2574	0.0
	40	35	3472	0.0
	50	335	4268	0.0
	60	1006	5233	0.0
	70	1167	6163	0.0
	80	3600	7162	0.9
	90	2218	7789	0.0
	100	3600	9022	2.0
15	10	0	747	0.0
	20	49	2164	0.0
	30	302	3472	0.0
	40	280	4703	0.0
	50	3600	6033	0.7
	60	3600	7158	1.0
	70	1125	8359	0.0
	80	3600	9899	3.8
	90	3600	12173	11.1
	100	3600	12976	8.4
20	10	1	914	0.0
	20	9	2177	0.0
	30	580	3842	0.0
	40	2065	5209	0.0
	50	3600	6622	2.3
	60	3600	8401	5.4
	70	3600	10121	7.7
	80	3600	12638	17.7
	90	3600	14293	18.5
	100	3600	16007	18.9