An iterative optimization framework for delay management and train scheduling

Twan Dollevoet[1,2,*] Francesco Corman[3] Andrea D'Ariano[4]
Dennis Huisman[1,2]

[1] ECOPT and Econometric Institute - Erasmus University Rotterdam
Burgemeester Oudlaan 50, NL-3000 DR Rotterdam, the Netherlands
dollevoet@ese.eur.nl, huisman@ese.eur.nl

[2] Process quality & Innovation - Netherlands Railways
Laan van Puntenburg 100, NL-3500 HA Utrecht, The Netherlands

[3] Centre for Industrial Management - Katholieke Universiteit Leuven
Celestijnenlaan 300a, 3001 Heverlee, Belgium, Francesco.Corman@cib.kuleuven.be

[4] Dipartimento di Informatica e Automazione - Università degli Studi Roma Tre
Via della Vasca Navale 79, 00146 Roma, Italy, a.dariano@dia.uniroma3.it

Econometric Institute Report EI2012-10

May 23, 2012

Abstract

Delay management determines which connections should be maintained in case of a delayed feeder train. Recent delay management models incorporate the limited capacity of the railway infrastructure. These models introduce headway constraints to make sure that safety regulations are satisfied. Unfortunately, these headway constraints cannot capture the full details of the railway infrastructure, especially within the stations. We therefore propose an iterative optimization approach that iteratively solves a macroscopic delay management model on the one hand, and a microscopic train scheduling model on the other hand. The macroscopic model determines which connections to maintain and proposes a

*Corresponding author

disposition timetable. This disposition timetable is then validated microscopically for a bottleneck station of the network, proposing a feasible schedule of railway operations. This schedule reduces delay propagation and thereby minimizes passenger delays. We evaluate our iterative optimization framework using real-world instances around Utrecht in the Netherlands.

Keywords public transportation, railway operations, event-activity network, alternative graph

1 Introduction

Most regular train passengers will recognize the frustration of missing a connecting train when their feeder train arrives at the transfer station with a small delay. In low-frequency railway systems, missing a connection can have a severe impact on the travel time of the passengers, even if the delay of the incoming train is only small. In such cases, an alternative would be to delay the departure of the connecting train, so that passengers from the delayed train can transfer to the connecting one. If a train waits for passengers from a delayed feeder train, the punctuality will be reduced; if it does not wait, passengers need to wait for the next service connecting to their destination. Determining whether a train should wait for a delayed feeder train or should depart on time is the subject of Delay Management (DM). Netherlands Railways, the largest passenger operator in the Netherlands, endorses the importance of a reliable railway system and has recently introduced the passenger punctuality as a new performance indicator. The passenger punctuality measures the ratio of passengers who arrive at their destination with a delay smaller than a certain threshold value.

We propose in this paper an innovative approach that computes a connection plan that solves the DM problem, when the limited capacity of the railway infrastructure is considered. This latter is modeled as the Train Scheduling (TS) problem at a microscopic level, i.e. modeling the status of the signals and safety system. In our optimization framework the DM solution and TS solution iteratively set boundary conditions for each other. By coupling the two models, a solution is found that is locally feasible. Furthermore, by iteratively solving the DM and TS problems, delays for trains and passengers are reduced. The objective is multi-fold: (i) the computation of a feasible train schedule inside the stations, (ii) the minimization of train delays in station areas, (iii) the minimization of travel times for passenger flows at the network level.

We now review the main contributions on the DM and TS problems. In Schöbel (2001), a first integer programming formulation for the DM problem is given. This formulation is further developed in De Giovanni et al. (2008) and Schöbel (2007). In these models, it is assumed that passengers will wait for one cycle time whenever they miss a connection. Dollevoet et al. (2012) relax this assumption and assume that passengers take the fastest route to their destination. They present an integer programming formulation that allows for passenger rerouting and show that the delay is reduced significantly with respect to earlier models. Dollevoet and Huisman (2011) develop several

heuristics to solve the DM problem with passenger rerouting.
Other extensions of the classical DM model incorporate the limited infrastructure capacity. Schöbel (2009) proposes to apply headway constraints to model the limited capacity on the tracks. An integer programming formulation that includes these headway constraints and several computational tests are given in Schachtebeck and Schöbel (2010). In Dollevoet et al. (2011), a first attempt to take the limited number of platforms in a station into account is presented.
The DM models described so far are all macroscopic. The detailed characteristics of the railway infrastructure are abstracted to make sure that large parts of the network can be considered at once. Such a global scope is necessary for DM, as the passengers travel through large parts of the network. However, as a consequence, some of the complications arising from the infrastructure layout cannot be taken into account.
On the contrary, the train scheduling (TS) problem is to compute precisely the effects of delay propagation and the adjustments of train speed profiles at a microscopic level, by considering the capacity of the infrastructure and the behavior of the signaling system. This requires the definition of a microscopic scheduling problem, in which detailed information about the tracks and the switches is taken into account. This way, all characteristics of the infrastructure can be modeled.
Simulation models (see e.g. Hansen and Pachl (2008) for an overview) proved to be a suitable tool to represent the dynamics of train operations, but they are still limited especially when large stations and heavy traffic are considered, and are based on myopic rules that might result in large delays.
Concerning the optimization models for the TS problem, Törnquist (2012) resorts on heuristic procedures for computing schedules in a short time, compatible with operations. To this end, microscopic detail is considered for the most complex stations. Studies on a test case in Sweden report that for a time horizon of traffic prediction of 90 minutes, a feasible schedule is found within 30 seconds, even for instances where commercial optimization suites fail in finding a solution.
A fully microscopic model is used in Corman et al. (2011) to model train traffic over a complex and dense area of the Dutch railway network, with up to hundreds of trains. A truncated branch and bound procedure (D'Ariano et al., 2007) is used that achieves very often optimal solutions, substantially reducing delay propagation, compared to practice or simple dispatching rules. Building on that result, a bi-level programming is introduced in Corman et al. (2012) that allows control over very large instances, divided into many local areas. A coordination level is in charge of defining constraints at the border of the local areas to ensure a feasible global solution. The bi-level formulation allows to check feasibility and optimality at local and global network levels, leading to a branch and bound procedure that achieves quickly a good solution for up to one hour of traffic prediction.
Only recently, the DM problem has attracted some attention in the train scheduling literature. In Corman et al. (2010), a bi-objective TS model is developed that minimizes the delay of the trains on the one hand and the number of missed connections on the other hand. However, as only the connections and trains within a station area are

considered, the global behavior of the passenger flows cannot fully be captured.
For the existing TS approaches, the size of instances solvable within a real-time computation horizon is still smaller (in time horizon or geographical extent) compared to the macroscopic DM models. Moreover, typical objectives of TS models regard the reduction of (possibly weighted) delays and delay propagation, and generally exclude passenger flows. Inclusion of continuous passenger flows would increase further the complexity by taking into account multiple objectives.

This paper presents an iterative optimization framework based on DM and TS models. It closes the gap between the theory on DM on the one hand, and on TS on the other hand. In doing so, the global scope of DM is combined with the high level of detail from TS. This way, we can model both the passenger flows at a network level and the detailed infrastructure locally at the stations. To the best of our knowledge, this is the first attempt to consider both levels in an integrated approach.
In a macroscopic DM model, we first determine which connections to maintain and derive the departure and arrival times of trains at the stations. Given the connections that should be maintained, these departure and arrival times are then validated using a microscopic TS model. Given the outcome of this microscopic validation, the process is repeated until a feasible solution is found. Doing so, we find solutions to the DM problem that respect the limited capacity of the station infrastructure, even for some of the most complicated and densely occupied stations in The Netherlands.

The remainder of this paper is organized as follows. First, Section 2 describes the macroscopic DM model and Section 3 the microscopic TS model. Section 4 gives an illustrative example for both models. Then, Section 5 shows how these models are coupled in our iterative optimization framework. Section 6 reports the experimental setup to evaluate the framework. Section 7 concludes the paper with remarks on the framework and on the computational results. Further research directions are also outlined for practical applications of the proposed methodology.

2 Delay management model

The central question of the DM models is which connections to maintain in case the railway system faces delays. It is assumed that the original timetable and the passenger demand are known. The passenger data is represented as a set of origin-destination pairs (OD-pairs) $\mathcal{P}$. Each OD-pair $p \in \mathcal{P}$ represents a group of n_p passengers who want to travel from a common origin station to a destination station at a specified time. Given a set of initial delays, the aim is to determine for each connection whether it should be maintained or not. Besides, a so-called disposition timetable is determined that prescribes the expected departure and arrival times of the trains at each station. Finally, for each OD-pair we determine a *passenger route* that connects their origin and destination, possibly including transfers at intermediate stations.

The DM problem is commonly modeled with an event-activity network. In this directed graph, the nodes correspond to the *events* that have to be scheduled and are

denoted by $\mathcal{E}$. We distinguish between departure events $\mathcal{E}_{\text{dep}}$ and arrival events $\mathcal{E}_{\text{arr}}$, that correspond to the departure from and the arrival at a station, respectively. For each event $e \in \mathcal{E}$, we denote the time when the event is planned to take place by π_e. The variables π thus denote the timetable as it was planned to be operated. For each event $e \in \mathcal{E}$, the initial delay is denoted by d_e.
The arcs in the graph, denoted by $\mathcal{A}$, represent precedence constraints (or *activities*) between these events and ensure that a minimal time between the events is respected. We distinguish between driving arcs, waiting arcs and changing arcs. Driving arcs in $\mathcal{A}_{\text{drive}}$ connect a train's departure from one station to its arrival at the next station. Waiting arcs connect the arrival of a train at a station to its departure from that same station and make sure that time is available for the passengers to get off and on the train. We denote the set of waiting arcs by $\mathcal{A}_{\text{wait}}$. Finally, changing arcs, contained in $\mathcal{A}_{\text{change}}$, allow passengers to transfer from one train to another. Driving and dwell arcs correspond to operational constraints that cannot be neglected. On the contrary, transfer arcs model possible transfers for the passengers. In case of delays, the railway operator can decide to not maintain a transfer. For each activity $a \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{change}}$, we denote the minimal time required for that activity by L_a.

In order to compute the delay for the passengers correctly, Dollevoet et al. (2012) propose to determine a passenger route for each OD-pair explicitly. In order to do so, the event-activity network is extended with auxiliary events and activities. For each OD-pair $p \in \mathcal{P}$, both an origin event $Org(p)$ and a destination event $Dest(p)$ are added to the event-activity network. These auxiliary events act as a source and a sink in a unit flow problem. The origin event is connected to the departure events from the station where the passengers in p want to start their trip. Similarly, all arrivals at the destination station are connected to the destination event. The set of origin and destination arcs for OD-pair $p \in \mathcal{P}$ are denoted by $\mathcal{A}_{\text{origin}}(p)$ and $\mathcal{A}_{\text{destination}}(p)$, respectively. For notational convenience, we define

$$\mathcal{A}(p) = \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{change}} \cup \mathcal{A}_{\text{origin}}(p) \cup \mathcal{A}_{\text{destination}}(p).$$

In the extended event-activity network, a possible passenger route corresponds to a unit flow from the origin event to the destination event.

We are now ready to present an integer programming formulation for the DM problem with passenger rerouting. The main decision is which connections to maintain. We therefore introduce a binary decision variable

$$z_a = \begin{cases} 1 & \text{if connection } a \text{ is maintained,} \\ 0 & \text{otherwise.} \end{cases}$$

For each event $e \in \mathcal{E}$, we determine the actual time x_e that is in general equal to the planned time π_e plus a delay; the set x_e thus defines the disposition timetable.
For each OD-pair $p \in \mathcal{P}$, we determine a passenger route through the event-activity network. This corresponds to determining a unit flow from the origin event $Org(p)$ to the destination event $Dest(p)$ in the event-activity network. To model this, we

introduce for each activity $a \in \mathcal{A}(p)$ a binary decision variable

$$q_{ap} = \begin{cases} 1 & \text{if OD-pair } p \text{ uses activity } a, \\ 0 & \text{otherwise.} \end{cases}$$

For each OD-pair $p \in \mathcal{P}$, we introduce an auxiliary variable T_p that represents the arrival time for passengers in OD-pair p.
The integer program then reads as follows (see Dollevoet et al. (2012)).

$$\min \sum_{p \in \mathcal{P}} n_p T_p \tag{1}$$

such that

$$\begin{aligned}
x_e &\geq \pi_e + d_e, && \forall e \in \mathcal{E}, && (2)\\
x_e &\geq x_{e'} + L_a, && \forall a = (e', e) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}}, && (3)\\
M(1 - z_a) + x_e &\geq x_{e'} + L_a, && \forall a = (e', e) \in \mathcal{A}_{\text{change}}, && (4)\\
q_{ap} &\leq z_a, && \forall p \in \mathcal{P}, a \in \mathcal{A}_{\text{change}}, && (5)\\
1 &= \sum_{a \in \delta^{\text{out}}(Org(p))} q_{ap}, && \forall p \in \mathcal{P}, && (6)\\
\sum_{a \in \delta^{\text{in}}(e) \cap \mathcal{A}(p)} q_{ap} &= \sum_{a \in \delta^{\text{out}}(e) \cap \mathcal{A}(p)} q_{ap}, && \forall p \in \mathcal{P}, e \in \mathcal{E}, && (7)\\
\sum_{a \in \delta^{\text{in}}(Dest(p))} q_{ap} &= 1, && \forall p \in \mathcal{P}, && (8)\\
T_p &\geq x_e - M(1 - q_{ap}), && \forall a = (e, Dest(p)) \in \mathcal{A}_{\text{destination}}, && (9)\\
x_e &\in \mathbb{N}, && \forall e \in \mathcal{E}, && (10)\\
z_a &\in \{0, 1\}, && \forall a \in \mathcal{A}_{\text{change}}, && (11)\\
q_{ap} &\in \{0, 1\}, && \forall p \in \mathcal{P}, a \in \mathcal{A}(p), && (12)\\
T_p &\in \mathbb{N}, && \forall p \in \mathcal{P}. && (13)
\end{aligned}$$

The objective function (1) is to minimize the weighted sum of the passengers' arrival times. The planned arrival times are fixed, so this is equivalent to minimizing the average or total passenger delay. Constraints (2) incorporate the initial delays and make sure that no train departs earlier than planned. Constraints (3) propagate the delay along driving and waiting activities. For maintained connections, Constraints (4) propagate the delay from the arriving to the departing train. Constraints (5) make sure that a connection can only be used by passenger if it is maintained. Constraints (6)-(8) determine a unit flow from the origin event $Org(p)$ to the destination event $Dest(p)$, for each OD-pair $p \in \mathcal{P}$. Here $\delta^{\text{in}}(e)$ and $\delta^{\text{out}}(e)$ denote the set of arcs into and out of node $e \in \mathcal{E}$, respectively. Finally, Constraints (9) linearize the arrival times of the passengers. In Constraints (4) and (9), the parameter M is a sufficiently large number. We refer to Dollevoet et al. (2012) for more details on the integer programming formulation.

3 Train scheduling model

Given the actual train delays, the train scheduling problem is to compute a new feasible schedule compatible with the status of the network, with the signaling system, and the dynamics of trains. Potential conflicts between train paths are detected by a conflict detection procedure for a given period of traffic prediction. In case of fixed block signaling, tracks are divided into block sections; each block section cannot host two trains at the same time. A potential conflict occurs whenever two or more trains require the same block section and a decision on the train order has to be taken. The train that will traverse the block section as second will be held outside the block section by the signaling system. In fact, while this train approaches the occupied block section, first a yellow signal will be shown, prescribing to slow down to an approaching speed (e.g. 40 km/h); and finally the signal just before the block section will show a red signal that prescribes a complete stop before the block section, as long as the preceding train has not exited the block section and a minimum setup time has elapsed. A set of ordering decisions might furthermore result in a deadlock. A deadlock is the situation in which a set of trains is mutually waiting for a train in the set to move, and no movement for the trains in the set is possible.

To model those situations, a microscopic model is required, that has a precision of seconds in modeling the travel times and considers train movements at the level of block sections. This is the level of detail required to model properly the triggers of the safety system and represent the signal aspects of the signaling system. The final outcome is a detailed schedule of train movements, without deadlock situations, where all potential conflicts have been solved. In this way, precise times can be predicted and delays are estimated accurately.

We use a job shop scheduling model of the TS problem that can be represented as an event-activity network with additional constraints. Mascis and Pacciarelli (2002) show that this so-called alternative graph is a suitable model for the job shop scheduling problem with additional constraints, such as blocking, also occurring in the railway context. The main value of this formulation is the detailed representation of the train traffic, the network topology and the signaling system.
This formulation requires that a sequence of successive block sections is defined for each train. The time required by each train to traverse each block section can be computed in advance, except for a possible additional waiting time between operations in order to solve train conflicts. In the alternative graph model, this results in a chain of *operations* (passage of a train on a block section, modeled by nodes $n \in N$) and associated precedence constraints (modeled by fixed arcs in Fix), similarly to the event-activity network of the DM problem.

For every potential conflict, a passing order must be defined between the trains, which is modeled in the graph by introducing a suitable pair of alternative arcs (in the set Alt) for each pair of trains traversing a block section, that define each of the two possible orders between the trains. Those arcs result in minimum headways between different trains, according to the signaling system.

A deadlock-free and conflict-free schedule is finally obtained by selecting one alternative arc from each pair, and updating the speed profile of the trains to the actual aspects of the signaling system (Corman et al., 2011). Formally, the TS problem corresponds to a particular *disjunctive program*, i.e., a linear program with logical conditions involving operation "or" ($\vee$, disjunction), as follows.

$$\min t_n - t_0 \tag{14}$$

such that

$$t_j - t_i \geq w_{ij}, \qquad (i,j) \in Fix, \tag{15}$$

$$(t_j - t_{\sigma(i)} \geq w_{\sigma(i)j}) \vee (t_i - t_{\sigma(j)} \geq w_{\sigma(j)i}), \qquad ((\sigma(i),j),(\sigma(j),i)) \in Alt. \tag{16}$$

In Problem (14)-(16), a variable t_i, for $i = 1, \ldots, n-1$, is the start time of operation i and corresponds to the entrance time of a train in the associated block section, similarly to x_e in the DM model. We use $\sigma(i)$ to refer to the successor of operation i on the route followed by a particular train, i.e. the operation on the block section after i. Moreover, operation 0 is a dummy operation that precedes all the other operations, to give a common temporal reference; and operation n is a dummy operation that follows all the other operations, and is used to keep track of delays, as explained later. In the scheduling model, all t_i are expressed in seconds, while the precision of x_e in the DM model is in minutes.

Fixed constraints in Fix are a general family of constraints associated to characteristic processes of railway operations, as follows.

- Running constraints naturally define a chain of driving operations between operation i of a train, and its successor $\sigma(i)$ on the path followed by the train. For such driving process, we consider precedence relations of the form $t_{\sigma(i)} \geq t_i + w_{i\sigma(i)}$, where $w_{i\sigma(i)} > 0$ is the time required to traverse the block section associated to that operation, at its actual speed profile.

- Dwell constraints at a station model the boarding and alighting of passengers, where $w_{i\sigma(i)}$ is the minimal time required between the arrival operation and the departure operation of the same train.

- Release constraints of the form $t_i - t_0 \geq w_{0i}$ relate to operation 0 and represent moreover minimal start time for operation i, i.e. model the entrance time of a train into the area. This is analogous to the π_e in the DM model.

- Due date constraints of the form $t_n - t_i \geq w_{in}$ relate to operation n and represent a due date for operation i. Such constraints are used to compute the delay associated to train traffic.

- Connection constraints, as defined in the DM problem, fix the departure time of a connected train to be larger than the arrival of a feeder train, plus a given minimum connecting time. These constraints are the changing constraints specified

by the DM problem (the variables z_a). Such connections are normally associated to an arrival event of a train at a station platform, and a successive departure of another train at another platform of the same station.

Differently, the set Alt is disjunctive, i.e., is composed by pairs of alternative constraints, each of them representing an ordering decision between trains. For each pair i and j of operations associated with the entrance of two trains in the same block section, we introduce the disjunction $(t_j - t_{\sigma(i)} \geq w_{\sigma(i)j}) \vee (t_i - t_{\sigma(j)} \geq w_{\sigma(j)i})$, where $w_{\sigma(i)j} > 0$ and $w_{\sigma(j)i} > 0$ are the minimum headway times. Those headway times are a function of a variety of factors, such as the length of the block section, the speed profile of the train, the driver behavior, the length of the train, as specified by the blocking time theory (see e.g. Hansen and Pachl (2008)). Finally, running and headway times are a function of the speed profiles of trains, that again depend on the ordering decisions taken. The solutions computed are fully compliant with the operational rules, the dynamics of trains, and the actual signal aspects shown.

A TS solution corresponds to fixing the start time of each operation. The schedule is feasible if it satisfies all conjunctions in Fix and exactly a constraint for each disjunctive pair in Alt, and does not result in positive length cycles. Due to the structure of the arcs (i, n), the (positive) train delay can be computed at a set of relevant points (scheduled stops and the exit of the network). It is interesting to consider the consecutive delay only, i.e. the delay introduced when solving conflicts in the dispatching area, caused by the propagation of the initial delays of late trains to the other trains in the railway area. The objective function of the TS problem is the minimization of the maximum consecutive delay, that corresponds to the length of the longest path between the dummy nodes 0 and n, i.e. $t_n - t_0$.

4 Illustrative example

Figure 1 gives an illustrative example of the two models of Section 2 and 3. In the top part of Figure 1, two trains V and T are running on a line connecting station P with station Q. Train T stops at both stations, while train V stops only at station Q; thus, at this latter station there is a possibility to enforce a connection between the two trains. The dotted line defines the station area, i.e., a region in which switches connect different tracks, that merge and cross each other. In fact, train T follows the lower path in the network, while train V follows the upper path in the network; both are using the block section b just before station Q. To ensure minimum train separation and safe movements over the network, the fixed-block signaling system is used.

The middle part of Figure 1 refers to the macroscopic model used for the DM problem. Events are represented as nodes, and activities in the set $\mathcal{A}$ as arcs connecting them; train V is represented as the upper chain of events (including arrival event $A3$ and departure event $D3$), and train T as the lower chain (including $A1$, $D1$, $A2$, $D2$). More in detail, the graph shows a $Wait$ activity at station P for train T, a $Drive$ activity between station P and station Q for train T, and waiting activities at station

Q for both trains (reported as $Wait3$ and $Wait2$). A connection activity in $\mathcal{A}_{\text{change}}$ is also considered, resulting in the arc labeled $Connection\ 3 \rightarrow 2$.

The bottom part of Figure 1 considers instead the microscopic model as used for the TS problem, only for the area around station Q. The trains considered in the example define two chain of nodes and arcs (again, the upper chain for train V and the lower one for train T), plus the two dummy nodes 0 and n. Successive nodes of each train are connected by arcs representing Run activities, plus the two $Dwell$ activities at station Q. The ordering decision on the block section b is modeled by a pair of alternative, dotted arcs, representing the two possible orders between trains. The same connection constraint ($Connection\ 3 \rightarrow 2$) as in the DM model is included, constraining train T not to leave station Q before train V has arrived and a minimal time has passed. There are four release arcs, that connect dummy node 0 to the first node of a train, representing the entrance time of the train in the area considered, and to the departure from the stations, modeling the published departure time. Finally, two due date arcs connect the exit from the area considered to dummy node n.

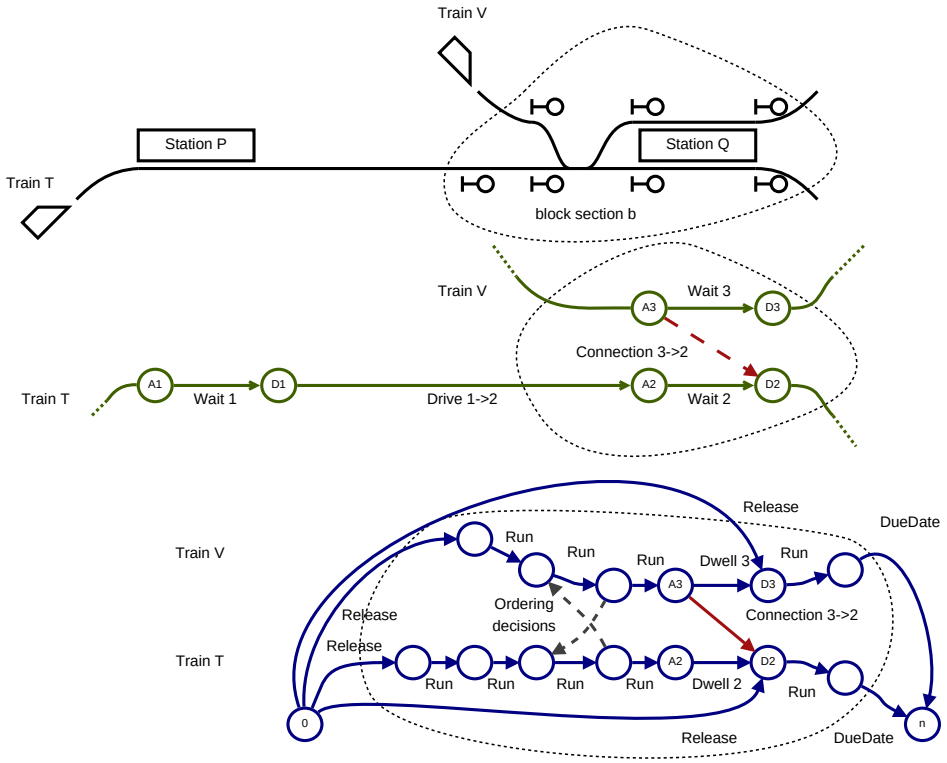

Figure 1: (top) Network of the illustrative example; (center) Macroscopic model used in the DM problem; (bottom) Microscopic model used for the area of station Q, in the microscopic model for the TS problem

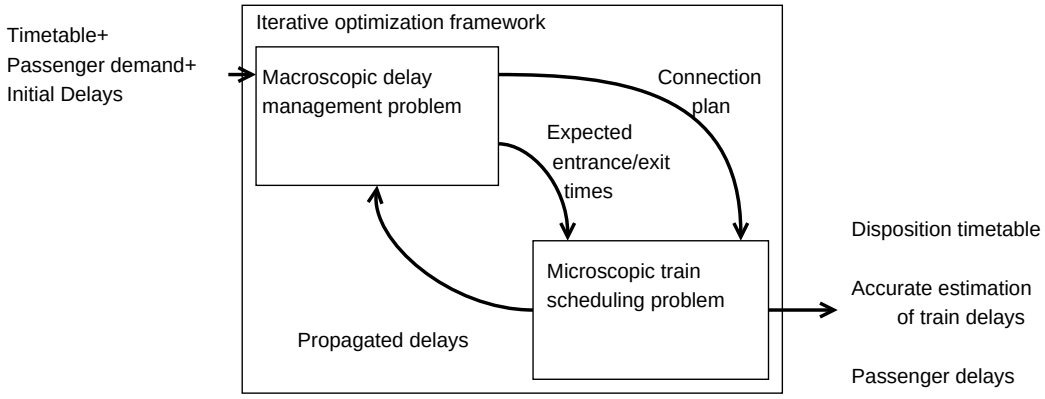

Figure 2: Schematic representation of the optimization framework.

5 Iterative DM and TS optimization approach

The previous sections presented the DM and TS models individually. We now introduce the optimization framework that iterates between solving a macroscopic DM problem on the one hand and a microscopic TS problem on the other. We will first give a general overview of the combined system and then an example is presented.

A schematic outline of our optimization framework is presented in Figure 2. The original timetable and the passenger demands are used as input for the algorithm. The passenger demand is given as a set of OD-pairs $p \in \mathcal{P}$, each of them representing n_p passengers who want to travel from an origin to a destination at a specified time. The timetable prescribes for each arrival and departure at a station at what time and at which platform it should take place. Furthermore, a set of initial delays is given. We assume that only the arrival events in the network have an initial delay. Equivalently, we assume that the initial delays are zero for all departure events.

The upper (macroscopic) part solves a DM problem to determine the connections to be maintained and computes a macroscopic disposition timetable. The DM solution minimizes the total delay for the passengers. In doing so, it allows the passengers to change their routes through the network.

The DM solution results in a set of passenger connections that should be maintained (i.e. a set of variables z_a) and an expected macroscopic timetable (corresponding to a set of event times x_e for all events e). Those variables are used to define a TS problem. To this end, we focus on those stations in the network where the infrastructure capacity is a bottleneck, and the possibility of facing conflicts for the scarcely available infrastructure is the highest.

Each TS problem considers part of the railway network around a station, in order to represent most of the potential train conflicts. The release time for an arriving train $e \in \mathcal{E}_{\text{arr}}$ into the area is computed based on the expected arrival time x_e of that train in the DM solution, minus a fixed time τ_e that corresponds to the minimal running

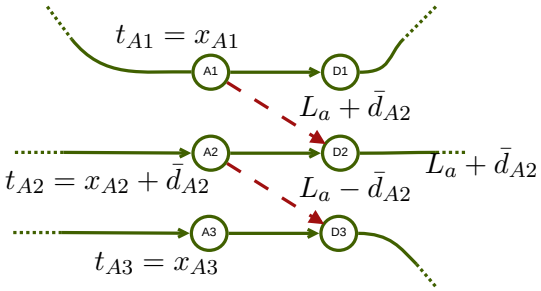

Figure 3: Part of the event-activity network within a station. $\bar{d}_{A2}$ is the extra delay computed by the TS model.

time between the entrance of the microscopic network and the arrival at the station platform. Similarly, we associate due dates to departing trains $e \in \mathcal{E}_{\text{dep}}$, based on x_e, the expected departure time computed by the DM solution, plus a time τ_e equal to the minimal running time from the platform until the exit of the microscopic network. The set of connections to be maintained, i.e. those for which $z_a = 1$, is also used in the TS problem. These transfer activities are added as fixed arcs to the set Fix.
The solution to the TS problem is a set of starting times of all operations, that are feasible with regard to the signaling system and the dynamics of trains. In particular, the solution contains starting times for the arrival and departure events $e \in \mathcal{E}$, that are considered in the DM problem. We will denote these starting times by t_e for all $e \in \mathcal{E}$.

This updated plan of operations will in general have conflicts in the station area and propagate some of the delays. The actual arrival and departure times t are going to be different from those original times x considered in the DM model. We thus find additional delays $\bar{d}_e = t_e - x_e$ for each event $e \in \mathcal{E}$ that is considered in the DM problem of the next iteration. To take these deviations into account, we update the minimal duration of the process times L_a for activities $a \in \mathcal{A}_{\text{change}} \cup \mathcal{A}_{\text{drive}}$, while avoiding to explicitly fix variables in the DM model.
To explain how these additional delays are incorporated, consider a train that departs later from a station than it was expected in the previous iteration. In that case, more passengers are able to transfer to that departing train. Furthermore, the train will probably arrive later at the next station in the macroscopic network.
We explain how we incorporate these intuitive ideas using Figure 3 that refers to a DM model. Part of an event-activity network is shown, that contains the arrivals and departures of three trains (respectively, A1, D1; A2, D2; A3, D3). The diagonal lines connect events of different trains and represent possible transfers for the passengers. We assume that the solution computed by the TS model contains some propagated delays $\bar{d}_{A2}$ and $\bar{d}_{D2}$, i.e. the actual times t_{A2} and t_{D2} are different from the plan x_{A2} and x_{D2}, respectively. All other events e have $\bar{d}_e = 0$, i.e. they occur at their planned time x_e. There are two possible connections represented (between $A1$ and $D2$; and between $A2$ and $D3$). Recall that L_a denotes the minimal transfer time for a transfer

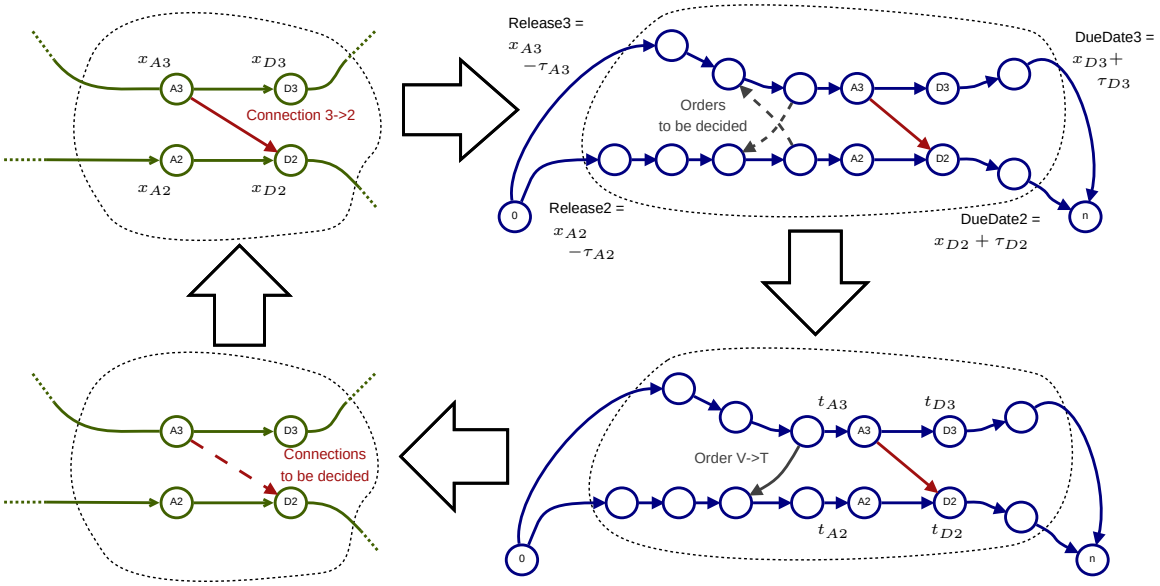

Figure 4: Iterative solution approach for the combined DM (left column) and TS (right column) problems. Iterations increase clockwise.

activity $a = (e, e') \in \mathcal{A}_{\text{change}}$. This means that the connection is maintained if and only if $x_{e'} - x_e \geq L_a$. Our aim is now to anticipate the delays from the TS model in the DM model. In the microscopic timetable, the transfer time for passengers equals $t_{e'} - t_e$. Incorporating the delays from the TS model, we thus find that the connection is maintained, if and only if

$$L_a \leq t_{e'} - t_e = x_{e'} + d_{e'} - x_e - d_e \quad \Leftrightarrow \quad L_a - d_{e'} + d_e \leq x_{e'} - x_e.$$

This suggests to use $L_a - d_{e'} + d_e$ as the minimal transfer time in the next iteration. For the transfer to the delayed train (i.e. $A1 \rightarrow D2$), the transfer time is thus decreased by the propagation of delays, as more passengers will be able to transfer. For the transfer from the delayed train (i.e. $A2 \rightarrow D3$), the transfer time is increased by the amount of delay. Finally, for the driving activity that connects the departure $D2$ to the arrival at the next station, the minimal driving time L_a is increased with the amount of delay.

We next illustrate the steps graphically, referring to Figure 4. We start from the top-left of Figure 4, which shows a solution to the DM problem, corresponding to a decision to maintain connection $3 \rightarrow 2$, and a proposed disposition timetable computed at macroscopic level, that corresponds to expected arrival and departure times (respectively, x_{A3}, x_{D3}, x_{A2}, and x_{D2}) for the two trains of the example reported in Figure 1. We use this solution to define a TS problem, in which only some station area is considered. This is reported in the top-right of Figure 4. The two trains enter the network at their release times ($Release3$ for the upper path corresponding to train V, and $Release2$ similarly for the lower path and train T), that are computed based on the expected

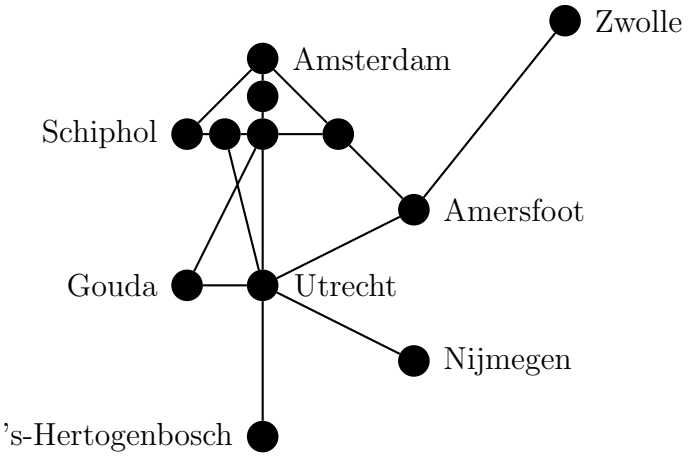

Figure 5: A large and busy part of the Dutch railway network

arrival time (x_{A3} and x_{A2} respectively) and the fixed times τ_{A3} and τ_{A2} related to running between the entrance of the microscopic network and the arrival at the station platform. Similarly, due dates are computed based on the expected departure time (x_{D3} and x_{D2}) and fixed times τ_{D3} and τ_{D2} related to running time from the platform to the exit of the microscopic network.
The TS problem is to compute the times of each operation, and orders between trains on shared infrastructure elements, that are represented by alternative arcs. The connections defined by the DM solution are included in the TS problem as fixed arcs. A solution to the TS problem is shown in the bottom-right figure, showing the order $V \to T$ chosen (i.e. train V precedes train T on block section b). This defines a microscopically feasible arrival time of the trains at the platform (respectively t_{A3} and t_{A2}), and similarly feasible departure times from the platform (t_{D3} and t_{D2}, respectively).
We then use the microscopically feasible times of the TS solution to define a new instance of the DM problem in the bottom-left of Figure 4. In general there will be a difference between the actual times t and the expected times x that were considered at the previous iteration, as trains might face yellow or red signals to avoid potential conflicts. Those differences result in propagated delays, that define new process times for driving and changing activities. Based on these updated data, the DM solution might keep the same set of connections as in the iteration before, or choose for new ones. The resulting solution would be the one shown on the top-left of Figure 4, leading to another iteration.

6 Computational experiments

We assess the performance of our optimization framework using real-world instances from the Netherlands. Railway activities in the Netherlands are split between an in-

Characteristics of the macroscopic model	
Time horizon	4 hours
Stations	46
Trains	377
Train driving activities $\lvert\mathcal{A}_{\text{drive}}\rvert$	1221
Dwell activities $\lvert\mathcal{A}_{\text{wait}}\rvert$	844
Connections activities $\lvert\mathcal{A}_{\text{change}}\rvert$	9643
OD-pairs $\lvert\mathcal{P}\rvert$	7086

Table 1: Some characteristics of the delay management model

frastructure manager (ProRail) on the one hand and several railway operators on the other hand. We obtained detailed information on the infrastructure from ProRail and the timetable and passenger information from Netherlands Railways. Netherlands Railways is the largest passenger operator in the Netherlands and transports over a million passengers per day.
We now first describe the instances that we used to evaluate our optimization framework. Then we present the computational results. In all our experiments, our main objective will be to minimize the total passenger delay.

6.1 Instances

The instances consider the railway network that is depicted in Figure 5. This picture shows a dense part of the railway network that contains Utrecht Central Station, which is in the centre of the Netherlands. The dots in the picture represent larger stations, where passengers have the possibility to transfer from one train to another. Two stations are connected by a line if there is a direct train between them. On most lines, both long distance trains and regional trains are operated with a high frequency. The long distance trains stop at the stations in the picture only. On the contrary, regional trains stop on smaller stations along the line, too. In total, we consider 46 stations. Because there are both long distance trains and regional trains with a high frequency, the station infrastructure in major stations is utilized heavily, especially in Utrecht Central Station.
In order to assess the performance of the iterative approach, we generate a set of delay scenarios and solve the corresponding delay management problem with the proposed optimization framework. We generate two samples: one sample with small initial delays and one with large initial delays. Both samples contain ten scenarios. We have generated the delay scenarios as follows. In all scenarios, each arrival of a train at a station has a probability of 10% to be delayed. If an arrival is delayed, the initial delay is uniformly distributed between 1 and 5 minutes in the sample with small initial delays. Similarly, in the sample with large initial delays, the initial delay is uniformly distributed between 1 and 15 minutes.

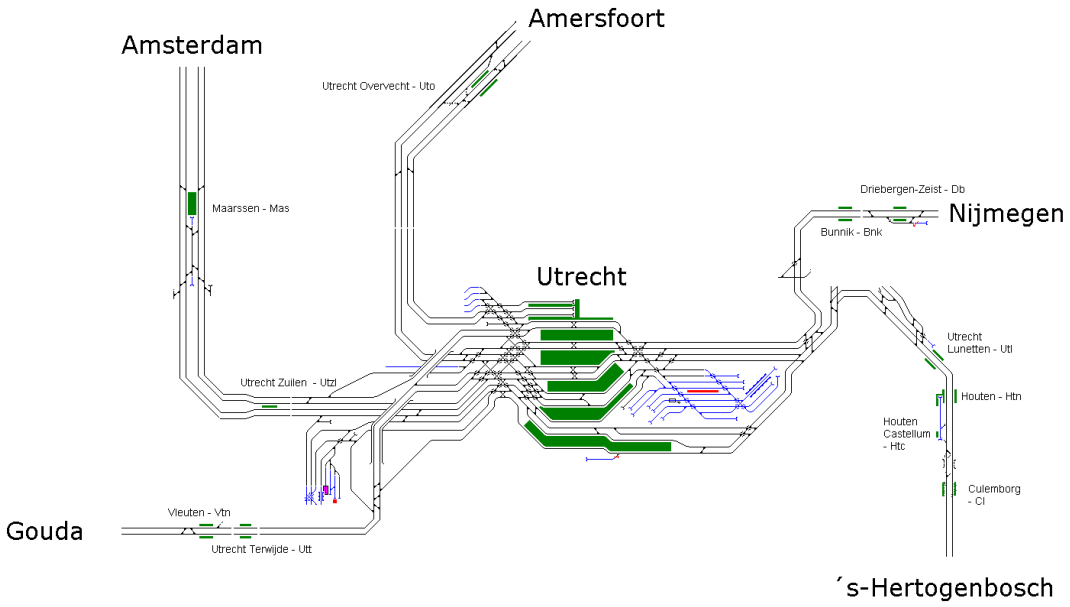

Figure 6: Microscopic detail in the area around Utrecht Central Station, and location of the borders.

In Table 1, we first present some characteristics of the resulting delay management problem. In total, 377 trains are considered. Together, these correspond to 1221 departure events, 1221 arrival events and 1221 driving arcs. Besides, there are 844 dwell arcs, leading to 2065 operational activities. Furthermore, the network contains 9643 possible connections. We consider 7086 OD-pairs, of which 1732 have a direct train from their origin to their destination. This shows that 76% of the OD-pairs should transfer at least once. It turns out that OD-pairs with a direct trip attract much more passengers: Only 20% of the passengers in the railway network have to transfer.

Considering the microscopic validation of the solution of the DM model, we focus on the bottleneck of Utrecht Central Station, that is the station in which the infrastructure is used most heavily. In fact, five main lines arrive and depart from the 14 platforms of Utrecht Central Station, passing through two large interlocking areas at the sides of the station with a total of about a hundred switches. The TS model refers to a railway network that includes the station area of Utrecht Central Station, and about 10 kilometers of the railway lines, as in Figure 6.

The network considered results in train scheduling problems with the characteristics reported in Table 2. On top of the main station of Utrecht, 10 more minor stations are considered along the lines. Compared to the DM problem, for the same time horizon, only the trains passing through the area are considered; anyway, the microscopic detail leads to more individual operations considered, with about 22 operations considered

Characteristics of the microscopic model	
Time horizon	4 hours
Stations	11
Block sections	531
Trains	257
Operations $\|N\|$	5681
Ordering decisions $\|Alt\|$	52600

Table 2: Some characteristics of the train scheduling model

for each train, on average. The amount of ordering decisions increase exponentially with the amount of trains running on the block sections, resulting in more than 52,000 variables defining the order of trains.

6.2 Results for instances with small delays

Typical behavior of the iterative optimization framework for instances with small delays is presented in Figure 7. This figure shows the objective value in each iteration for a single case. Along the vertical axis is the total delay for the passengers in minutes. The solid line gives a lower bound on the optimal objective value, obtained by solving the delay management problem without considering the station capacity. The objective values from the DM model in each iteration are represented by asterisks and connected by the lower dashed line. Recall that the corresponding solutions are in general microscopically infeasible. In order to obtain feasible solutions, we apply the TS model, obtaining a set of consecutive delays for the trains in Utrecht Central Station. By propagating these delays through the network, we obtain a solution to the DM problem that is microscopically feasible. The objective value for this solution can be found by computing for each OD-pair the earliest arrival time and the corresponding delay. Adding these delays over all OD-pairs gives the objective value for this solution. These objective values are indicated by the crosses in the figure.
We start the iterative approach with a solution in which no connections are maintained. In the second iteration, the possibility to maintain a connection is included and the solution value for the DM problem decreases significantly. However, the gap to the solution of the TS problem is rather large. In the next iteration, the consecutive delays found by the TS algorithm are anticipated, leading to a solution that is slightly better. From then onwards, the algorithm oscillates between two solutions.
The average objective values over 10 scenarios are presented in Table 3. In the second column we report the objective value that is found in a specific iteration. The iterative procedure does not improve the solution in every iteration. The third column therefore contains the best objective value that is obtained *until* that iteration. Finally, we present the best normalized passenger delays in the last column. As can be seen, in the second iteration the delay is reduced with 26% with respect to the first iteration.

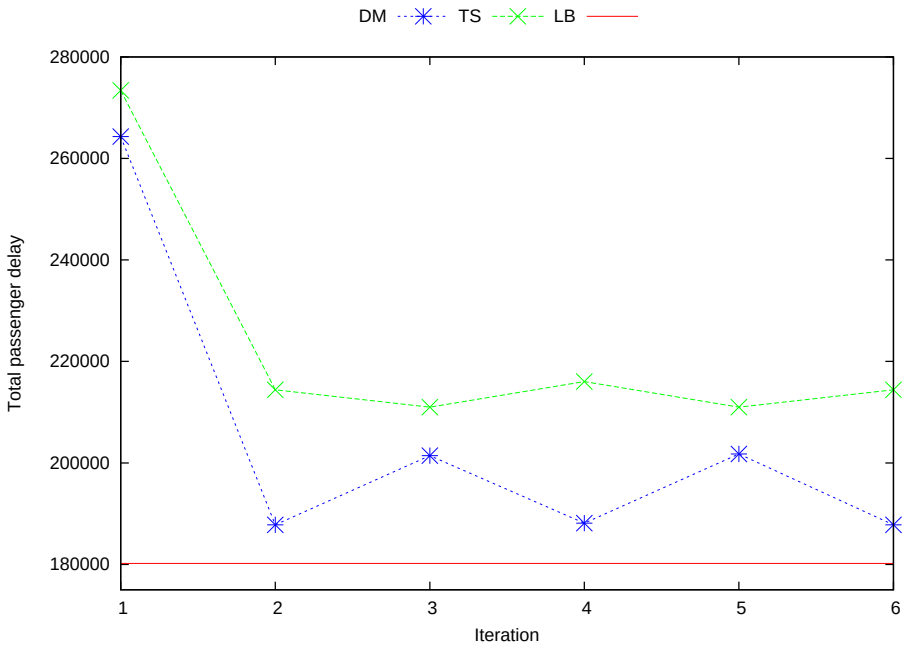

Figure 7: The total delay for the passengers in each of the iterations for a scenario with small delays

In the next iterations, the total passenger delay is reduced by another 1.0%. The best solution is found in the second iteration for 3 instances, in the third iteration for 5 instances and once in the fourth and sixth iteration.

Characteristics of the solution procedure for instances with small delays are reported in Table 4. Solutions to the DM problem can be found in 125 seconds on average. Solving an instance of the TS problem takes on average 240 seconds of computation time. The resulting solutions have a maximum consecutive delay of 212 seconds and an average consecutive delay of 4 seconds. In the first iteration, the solution value after solving the TS problem is about 6% worse than the objective value from DM. In other iterations, the solution value is increased by about 9%. The gap between the final

Iteration	Objective	Best objective	Best normalized objective
1	324734	324734	100
2	240836	240836	74.2
3	241027	238985	73.6
4	241205	238793	73.5
5	242023	238793	73.5
6	239075	237670	73.2

Table 3: The average objective value over 10 instances with small delays

Characteristics of the solution procedure	
Computation time for one DM iteration (seconds)	125
Computation time for one TS iteration (seconds)	240
Average consecutive train delay in Utrecht (seconds)	4.0
Gap between DM and TS solution (total passenger delay)	8%
Gap to the lower bound (total passenger delay)	15%
Difference between first and best solution (average train delay)	20%

Table 4: Some characteristics of the solution procedure for the instances with small delays

Iteration	Objective	Best objective	Best normalized objective
1	900634	900634	100
2	903771	878029	97.5
3	907546	874899	97.1
4	892270	871588	96.8
5	897342	871588	96.8
6	909596	871588	96.8

Table 5: The average objective value over 10 instances with large delays

solution and the lower bound is on average 15%. Recall that the lower bound is obtained by solving the DM problem from Section 2 without considering the station capacity. We also compare the average train delay between the first iteration and the iteration in which the best solution is found. For instances with small delays, the average train delay is increased with 20%.

6.3 Results for instances with large delays

For the scenarios with larger delays, the algorithm behaves less consistently. In Figure 8, we show the solution values for an instance where the iterative approach improves over the first solution. Again, we start the process with a solution that maintains no connections. In the following three iterations, the solution value decreases. After that, worse solutions are found. Such behavior is observed for 40% of the scenarios.
For the other instances we find worse solutions after the first iteration. A typical example is given in Figure 9. The course of the solution values for these instances is very unstable. Furthermore, the iterative approach does not improve over the start solution.
In Table 5 we report the average objective values in each iteration. Only in the third and fourth iterations, the average objective value is better than that of the start solution. In the fourth column we report for each iteration the best relative solution value obtained until that iteration. Here we see that, on average, the solutions in the final iteration are

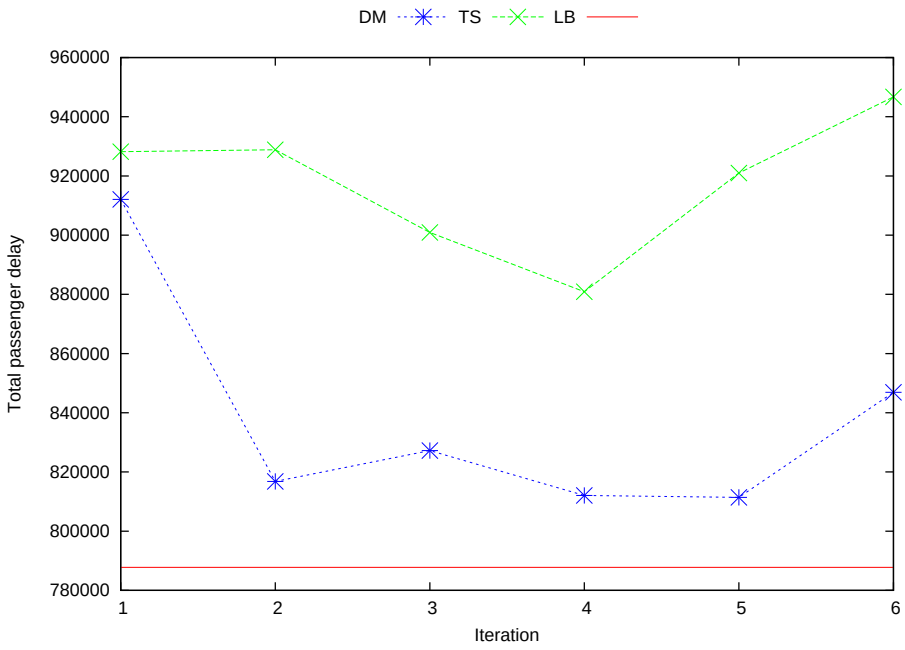

Figure 8: The total delay for the passengers for an instance with large delays where the iterative approach improves over the start solution

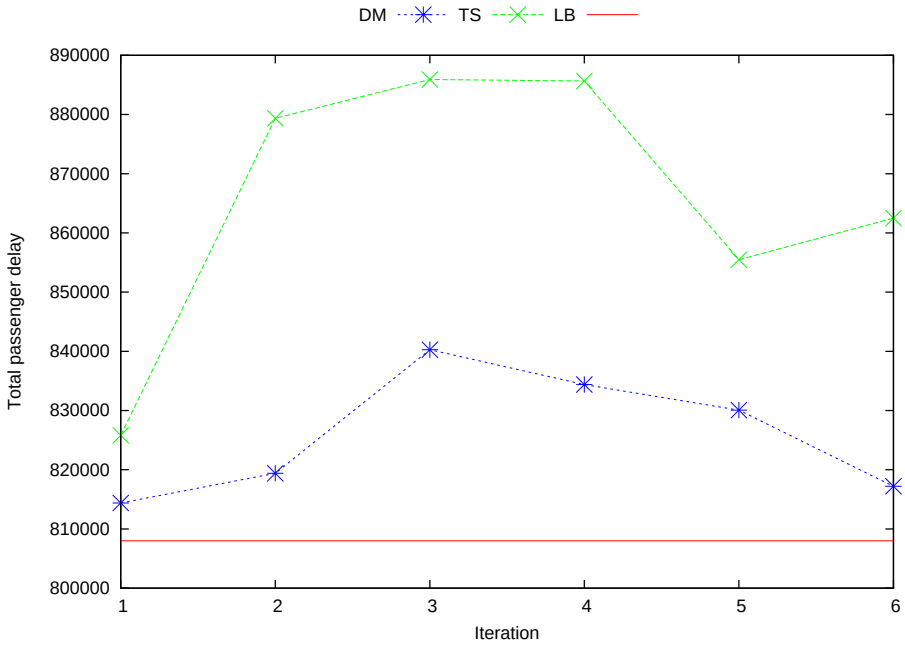

Figure 9: The total delay for the passengers for an instance with large delays where the iterative approach cannot improve over the start solution

Characteristics of the solution procedure	
Computation time for one DM iteration (seconds)	-
Computation time for one TS iteration (seconds)	300
Average consecutive train delay in Utrecht (seconds)	8.7
Gap between DM and TS solution (total passenger delay)	10 %
Gap to the lower bound (total passenger delay)	13 %
Difference between first and best solution (average train delay)	21%

Table 6: Some characteristics of the solution procedure for instancs with large delays

3.2 % better than the solutions found in the first iteration. The gap between the DM and TS solution is 4% in the first iteration and on average 11% in the other iterations.

In Table 6, some characteristics of the solution procedure are reported. Solving an instance of the train scheduling problem takes on average 300 seconds of computation time for large delays. The resulting solution has a maximum consecutive delay of 354 seconds and an average consecutive delay of 8.7 seconds. Solving the delay management problem to optimality takes much time. Therefore, we limit the time for the DM problem for each iteration to 20 minutes. Within this time, solutions are found that are close to optimal, with gaps smaller than 1% for all instances. Furthermore, the best solution is found within several minutes. Comparing the average train delay between the first and the best iteration, we observe an increase of 21%.

7 Conclusions

In this paper we developed an iterative optimization framework for delay management and train scheduling. We propose a mechanism to incorporate consecutive delays from the train scheduling solution in the delay management problem. By combining the global scope of delay management and the local scope of train scheduling, we were able find solutions to the delay management problem that respect the limited capacity of the station infrastructure. Besides these wait-depart decisions, the solution framework provides a feasible train schedule at the stations, where the infrastructure is used heavily. This train schedule allows for the precise evaluation of train delays, and thus also the passenger delays.

We first consider scenarios with small initial delays. For those scenarios, our framework obtains a solution to the DM problem that is microscopically feasible. In the iterative optimization procedure, the delay for the passengers is reduced by 27% with respect to a naive approach where only one iteration is performed. For scenarios with larger delays, the behavior of the solution procedure is less consistent. However, we are able to compute and evaluate a solution to the delay management problem that is feasible at the station level.

Several directions for further research are available. First, the interaction between

the models should be investigated in more detail for scenarios with large delays. Considering a more general feedback mechanism could potentially lead to better solutions. For example, one could define weights on the connections in the train scheduling model or penalize changes in the wait-depart decisions in the delay management model. Second, the iterative framework could be tested on a railway network with more bottlenecks. For each station where the infrastructure is scarce, a local scheduler could be applied to compute a feasible train schedule. As the updates from different station can be conflicting, these updates should be fully coordinated.

References

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C*, 20 (1):79–94, 2010.

F. Corman, A. D'Ariano, M. Pranzo, and I. Hansen. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, 34(4):341–362, 2011.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Optimal inter-area coordination of train rescheduling decisions. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):71–88, 2012.

A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2): 643–657, 2007.

L. De Giovanni, G. Heilporn, and M. Labbé. Optimization models for the single delay management problem in public transportation. *European Journal of Operational Research*, 189(3):762–774, 2008.

T. Dollevoet and D. Huisman. Fast heuristics for delay management with passenger rerouting. Technical Report EI2011-35, Econometric Institute Report Series, 2011.

T. Dollevoet, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. In A. Caprara and S. Kontogiannis, editors, *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, pages 88–99. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of passengers. *Transportation Science*, 46(1):74–89, 2012.

I. A. Hansen and J. Pachl, editors. *Railway Timetable and Traffic: Analysis, Modelling and Simulation*. Eurailpress, Hamburg, 2008.

A. Mascis and D. Pacciarelli. Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143(3):498–517, 2002.

M. Schachtebeck and A. Schöbel. To wait or not to wait and who goes first? Delay management with priority decisions. *Transportation Science*, 44(3):307–321, 2010.

A. Schöbel. A model for the delay management problem based on mixed-integer programming. *Electronic Notes in Theoretical Computer Science*, 50(1):1–10, 2001.

A. Schöbel. Integer programming approaches for solving the delay management problem. In *Algorithmic Methods for Railway Optimization*, number 4359 in Lecture Notes in Computer Science, pages 145–170. Springer, 2007.

A. Schöbel. Capacity constraints in delay management. *Public Transport*, 1(2):135–154, 2009.

J. Törnquist. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Research Part C: Emerging Technologies*, 20(1):62–78, 2012.