Improving Punctuality and Transfer Reliability by Railway Timetable Optimization

TRAIL Research School, Delft, December 1999

Author:
Drs. Rob M.P. Goverde
Transportation Planning and Traffic Engineering Section,
Faculty of Civil Engineering and Geo Sciences, Delft University of Technology

Thesis supervisors:
Prof.Dr.-Ing. Ingo A. Hansen
Transportation Planning and Traffic Engineering Section,
Faculty of Civil Engineering and Geo Sciences, Delft University of Technology

Prof.Dr. Geert Jan Olsder
Department of Technical Mathematics,
Faculty of Information Technology and Systems, Delft University of Technology
# Contents

Abstract

1 Introduction 1

2 Modelling Timetable Constraints 4
   2.1 Introduction ................................................................. 4
   2.2 Time Window Constraints ............................................... 5
   2.3 Cycle Constraints ............................................................ 10
   2.4 Rolling Stock Constraints ............................................... 12

3 Railway Timetable Optimization 16
   3.1 Introduction ................................................................. 16
   3.2 The Periodic Railway Timetable Optimization Problem .......... 16
   3.3 Computational Complexity ............................................... 17
   3.4 Preprocessing: Integer Bounds ....................................... 18

4 The Network Synchronization Problem 20
   4.1 The Optimization Problem ............................................... 20
   4.2 Tight Connections ........................................................... 21
   4.3 Reliable Connections ....................................................... 22
   4.4 Transfer Connections ....................................................... 24

5 Conclusions 28

Acknowledgements 28

References 28
Abstract

The Dutch railway network is operated close to capacity with the current safety system. This leaves little space for control by process operators by which a delayed train can cause severe delay propagation. The NS currently invest a large amount in punctuality improvement of the railway operations. This paper shows how the effect of delays can already be incorporated in the timetable design process to obtain robust and optimized timetables for a punctual and reliable operation.

A detailed modelling approach is presented to compute periodic network timetables with an optimal distribution of buffer times at those points where they are needed the most without being too conservative. The variables are the buffer times and the proposed objective is to minimize a weighted sum of individual costs for each buffer time. Here, the weights define the relative importance of the buffer time costs. The following cost functions are proposed and proved to be convex for general running time distributions: buffer time (fast travel times for through passengers and small operating costs); expected delay due to buffer time failure of rolling-stock and crew connections (increasing punctuality) and train circulations (stability of train circulations); and expected transfer waiting time (trade-off between small transfer times and transfer reliability).

The constraints on a feasible timetable are modelled conveniently by time window constraints concerning interactions between two periodic events (arrivals and departures). This is compatible with the timetable design system DONS of the NS. For the optimization problem these constraints are reformulated using the buffer times as variables rather than the periodic event times. The resulting cycle constraints state that the oriented sum of process and buffer times on fundamental cycles of the constraint graph is a multiple of the cycle time. The buffer times are not periodic by which they can be used straightforwardly in the objective functions. The periodic time windows are thus replaced by periodic cycles.

Theoretical relevance

The periodic timetable optimization problem extends the associated feasibility problem, the Periodic Event Scheduling Problem (PESP), and improves several problem issues. The problem is formulated as an optimization problem with convex cost function and buffer times as variables rather than the periodic events which is most common in literature. Solving the resulting large-scale convex mixed-integer programming problem requires a synthesis of various techniques from linear algebra, network flows, integer programming, and convex programming.

Societal relevance

An optimal robust timetable contributes to improve performance and attractivity of the railway operation, increase service quality to passengers, and decrease costs to railway operators and passengers due to delays. Moreover, a punctual and reliable railway operation may attract more potential passengers who currently use other modes (cars). This reduces the congestion problem on the Dutch highways and the associated environmental pollution.
1 Introduction

Punctuality is a major concern to all transportation companies, especially to an underlying transportation network with lots of interdependent services and low service frequencies (up to 6 services per hour). The level of punctuality largely influences the reliability of connections, including crew, vehicle, and transfer connections. On the other hand scheduled connections are a main source for secondary (or knock-on) delays (for example if a connecting service waits for a delayed feeder service to secure a transfer connection) and hence on its turn affects punctuality. An optimal coordination or synchronization of connected services is thus important to improve punctuality whereas at the same time reliability of connections is obtained. Both service punctuality and connection reliability are main indicators for the performance and attractiveness of a transportation network. This paper concentrates on railway networks as they represent a particularly interesting case.

Railway networks have an additional difficulty due to the safety and signalling systems which increase the possibilities of conflicting movements. Moreover, the train service network of the Netherlands Railways (NS) is highly interconnected and is served near to its capacity (for the present safety and signalling system). Additionally, the railway timetable in the Netherlands is an integrated periodic timetable, i.e., all train services run in regular intervals. As a consequence, the railway planners are faced with an extreme hard and time-consuming task to design a feasible timetable that satisfies all constraints as imposed by for instance the line system, required connections and infrastructure limitations. Therefore, specific attention to the synchronization of connected services is currently very modest.

A feasible timetable is by definition realizable (conflict-free) for punctual operation. However, perturbations during operation result in variations of train running and dwell times. Perturbations are for instance varying train characteristics, varying driving behaviour, fluctuations in dwell times depending on the number of alighting and boarding passengers, and extreme weather conditions. A timetable has to be robust to small variations in running times which is achieved by adding running time margins to the minimum running times. This way the (scheduled) running times can compensate for most of the variations. The more severe perturbations result in exceeding the scheduled running times, which give primary delays. Moreover, once a train is delayed it may hinder other trains or propagate the delay to connecting trains, resulting in secondary delays. The secondary delays are indirectly the result of the robustness of the timetable design. A robust timetable design aims at minimizing secondary delays during operation and thereby improving punctuality and reliability.

A railway timetable consists of the scheduled arrival and departure times of all train services at each station. Implicitly this also gives the dwell times for each line and the transfer times between lines at transfer stations where lines meet. From an operational point of view the timetable also fixes the connection times for rolling stock and/or crew connections, and headways between trains (at e.g.
conflict points). All these process times consist of two components: a necessary minimum process time (e.g., minimum dwell time for alighting and boarding; minimum transfer time for alighting, walking to the departure platform and boarding; and minimum arrival headway between two arriving trains at a station) and a possible buffer time or recovery time which improves robustness with respect to delays.

We are concerned with a periodic railway timetable. Obviously, this imposes severe restrictions on the timetable. In combination with the network interdependencies of all train services, it is no surprise that there is an amount of slack in a timetable necessary to fit all process times in the regular interval pattern. The resulting buffer times lead to larger process times which at first glance is annoying. However, buffer times may be very desirable to improve robustness of the processes with respect to variations in process times. The challenge is to distribute the buffer times over the service network such that they are advantageous during railway operation. This is the periodic railway timetable optimization problem.

The development of the advanced design system DONS (Designer of Network schedules) [12, 25] is a great step forward to computing a feasible timetable that satisfies all constraints, or if no such timetable exists, giving information about conflicting constraints. DONS also features a timetable optimization option that can be used after having computed a feasible timetable. The optimization has then be understood with respect to the computed train (sequence) orders. So the train orders remain fixed which obviously results in local optimal solutions, i.e., better (feasible) timetables might be obtained if train orders are allowed to be swapped. Moreover, more advanced cost functions have to be defined to (sub)optimize the timetable with respect to operation reliability. Another improvement would be to obtain (nontrivial) multiple feasible solutions (if there are any). Currently, the computations stop when a feasible solution has been found. The existence of multiple solutions with perhaps more convenient train orders then remains unknown. This paper gives a solution to the above problems and thereby contributes to the further development of the DONS features. The problem of computing multiple feasible solutions to the system of time window constraints is also the subject of a parallel research project [24].

The optimization of railway network timetables has had some attention in the literature. The problem is typically modelled as a (linear) mixed integer program (MIP). Weigand [28, 29] reports a graphical oriented modelling approach. The MIP-formulation is based on cycles in a graph representing all stop and transfer connections. Infrastructure constraints are not dealt with explicitly. Nachtigall [19, 20] formalizes this approach and proves that the problem is \( \mathcal{NP} \)-complete. An alternative MIP-formulation is based on constraints imposed by interrelated train pairs resulting in time window constraints [15, 16, 18]. Here also infrastructure constraints (minimum headway between a train pair) are incorporated. The corresponding feasibility problem has been formalized by Serafini and Ukovich [26] and is known as the Periodic Event Scheduling Problem (PESP). They proved that PESP is \( \mathcal{NP} \)-complete, and hence also the associated optimization problem is \( \mathcal{NP} \)-complete. The PESP-formulation is also used
in the DONS-system [22, 23, 25]. A related modelling approach is the schedule synchronization problem in which the line schedules (running and dwell times) are fixed. The problem is to determine the optimal departure times of all lines at their starting terminal station such that the costs of the resulting transfer times at stations where the lines meet are minimized. This optimization problem can be formulated as a binary integer program (BIP) [13] or an integer program (IP) [2]. However, infrastructure constraints can not be handled trivially by these formulations. Domschke [7] proved that the problem is $\mathcal{NP}$-complete. Note that this problem can also be modelled as an MIP using the time window (or PESP) constraints as it is a special case for fixed dwell times. Then also infrastructure constraints can be incorporated.

Some literature is more or less devoted to reliability and performance measures of transportation services. Hallowell and Harker [10] studied expected delays due to interference at meet/pass points on a partially double-tracked railway line. Carey [3] and Ferreira and Higgins [8] consider the minimization of train running times and expected arrival delays at a single railway line including knock-on delays. Carey (and Kwieciński) [6, 5] and Higgins and Kozan [11] have generalized these models to networks of various lines. Carey [4] moreover studied the behavioural response of actual running times if the scheduled running time is increased to minimize arrival delays. Knoppers and Muller [14] studied the expected passenger transfer waiting time including the probability of missing a connection. Finally, Meng [17] developed a model to determine the required average dwell and transfer buffer times at a railway station for random (Gamma-distributed) inter-arrivals of trains, incorporating the number of trains and connections, the number of platform tracks, average secondary delays due to connections and conflicting train paths, and the risk of exceeding platform capacity.

The paper is organized as follows. Chapter 2 shows how to model the timetable constraints and get a constraint system of the buffer times. Chapter 3 defines the general periodic railway timetable optimization problem and considers its computational complexity. The network synchronization problem is addressed in Chapter 4 where various convenient cost functions for the individual buffer times are derived to obtain either tight connections, reliable connections, or optimal transfer connections, whatever is appropriate.
2 Modelling Timetable Constraints

2.1 Introduction

The train service network in the Netherlands is operated according to a regular interval or *periodic timetable*, which is also popular in train service networks in most densely populated countries in Europe, for long-distance railway networks as in Germany, and other public transportation systems. A periodic timetable is convenient for passengers but imposes severe restrictions on the timetable. All arrival and departure times must fit a regular interval pattern. Additional difficulties in finding a railway timetable are the network interdependencies and infrastructure limitations.

A (periodic) network timetable is a feasible solution to a (periodic) constraint system that specifies all interactions between the train services in the network. We assume that a line system is given in advance and that the process times are deterministic and given as well. Here a *line system* is defined as a set of lines, the train type of each line, and the connections between the lines at stations where the lines meet. A *line* is a sequence of train services on a particular route between two terminal stations, and the *train type* (for instance express train or local train) determines the intermediate served stations on the route. We define a *timetable point* as a node in the service network where interactions between trains are observed. Examples are transfer stations, level-crossings, and emerging sections. Then our input data contains

- train running times between timetable points including dwell times at intermediate stops;
- minimum dwell times at transfer stations;
- minimum turn-around times at terminal stations;
- minimum transfer times between train pairs at stations where lines meet;
- minimum crew and rolling stock connection times between train pairs at stations where a crew transfer or train coupling/decoupling is required;
- minimum headways between train pairs at timetable points.

Additionally maximum process times may be given to prevent too large dwell times, transfer times, etc. The crew schedules are usually planned after the timetable has been established. However, some crew connections may be known in advance when a timetable design is based on an existing timetable, with some adjustments due to for instance completed infrastructure projects. Typical crew connections are for example endpoints of lines. An extension of the planning process may include a feedback to the timetable design after crew schedules and rolling stock circulations have been obtained. Then slight adjustments of the...
timetable design may increase the robustness of the crew connections. Note that a crew connection does not have to occur at each timetable period.

A timetable is expressed in arrival and departure times of all services at each station. In the sequel it is more convenient to refer to an event which can be both an arrival or a departure. Generally, an event is the start or end of an activity. So a train departure from a station is an event that activates a train run to the next station, and an arrival at a station is the end of a train run from a preceding station. A periodic event is an event that repeats at a regular interval with cycle time $T$. For instance, a departure in an hourly timetable repeats every $T = 60$ min. An event time is the occurrence time of the event, e.g. the departure time.

If running times between stations are assumed deterministic (for planning purposes) then it suffices to consider departure events only: an arrival time can be given as the sum of a departure time at a preceding station and the train running time between the stations. This is convenient if large service networks are considered as this gives a reduction to half the number of variables. The variable $\tau_i \pmod{T}$ denotes the (departure) event time of the periodic (departure) event $i$. Below we express all timetable constraints in departure events. We also somewhat sloppy refer to $i$ as both the departure and the activated train service.

### 2.2 Time Window Constraints

Connections at transfer stations impose synchronization constraints on the train services. Also the track layout and the safety and signalling system of a railway network imposes limitations on train movements. For instance, trains can not cross a level-crossing at the same time, and a safety distance has to be respected for two trains running on joint track sections.

A train service is defined as a train trip between two stations. All timetable constraints can then be defined as an interaction (e.g. a connection) between two train services. For example, train 1 must depart no sooner than 2 to 5 min after the arrival of train 2 to allow a transfer. Such a constraint is called a time window constraint. Note that an interaction has a ‘direction’, i.e., train 1 waits for train 2, but the reverse does not have to hold unless also an interaction is defined from train 2 to train 1. An interaction of two services occurs for example at a stop, a transfer, or a conflict point (mutual track section). Each interaction produces a minimum process time and a buffer time. So a buffer time can for instance be a dwell buffer time, a transfer buffer time, or a headway buffer time.

Time window constraints specify that the time difference between two periodic event times is restricted to a particular time window. Let $\tau_i$ and $\tau_j$ be two event times. Then a time window constraint is

$$\tau_j - \tau_i \equiv l_{ij} + r_{ij} \pmod{T} \text{ and } r_{ij} \in [0, r_{ij}^{\text{max}}].$$

(1)

Here, $l_{ij}$ is the minimum time interval between event $i$ and $j$, and $r_{ij}$ is an additional nonnegative buffer time that is bounded from above by a given maximum
buffer time $r_{ij}^{\text{max}} \in [0, T]$. Equation (1) has to be evaluated modulo the cycle time $T$ which means that an integral multiple of $T$ can be subtracted where necessary. Without loss of generality we can write (1) as the MIP constraint

$$\tau_j - \tau_i = l_{ij} + r_{ij} + z_{ij}T \quad \text{and} \quad r_{ij} \in [0, r_{ij}^{\text{max}}]$$

or

$$l_{ij} \leq \tau_j - \tau_i - z_{ij}T \leq l_{ij} + r_{ij}^{\text{max}}$$

where $z_{ij} \in \mathbb{Z}$.

All timetable constraints can be modelled conveniently as time window constraints, see also Schrijver and Steenbeek [25]. The constraints can be classified in synchronization constraints, infrastructure constraints, and additional market constraints, which are considered sequentially below.

**Synchronization constraints** specify that a train service $j$ has to wait for a train service $i$ from a preceding station to guarantee a connection. A connection can either be a stop or transfer, and we thus distinguish between *stop constraints* and *transfer constraints*. Let $t_i^t$ be the running time of train service $i$ from the preceding station, and $t_{ij}^{\text{min}}$ be the minimum dwell/transfer time to the connecting train $j$. Then $l_{ij} = t_i^t + t_{ij}^{\text{min}}$ is the minimum time after the departure of train service $i$ from its preceding station before the connecting train service $j$ can depart. If $r_{ij}^{\text{max}}$ is the maximum (dwell or transfer) buffer time then the synchronization constraint is given as

$$\tau_j - \tau_i = t_i^t + t_{ij}^{\text{min}} + r_{ij} + z_{ij}T,$$

where $r_{ij} \in [0, r_{ij}^{\text{max}}]$ is the dwell/transfer buffer time, see Figure 1.

Crew and rolling stock connections give special synchronization constraints with respect to the logistics of crew and rolling stock schedules. First, a *turn* at a terminal station is a special stop where the minimum turn-around time does not just depend on alighting and boarding time, but also includes for instance the necessary time for cleaning the coaches and turning the locomotive. Second, a *crew connection* is a special transfer in which the crew of the feeder train is scheduled to transfer to the connecting train. In this case the connection can not be cancelled unless a reserve crew can be allocated to the connecting train. Third, a *rolling stock connection* is a connection in which trains are coupled/decoupled.

**Infrastructure constraints** describe headway restrictions due to the safety and signalling system and infrastructural limitations. Consider two trains services $i$ and $j$ that use a common track section some distance from the (distinct) departure stations. Let $t_i^r$ be the running time of train service $i$ to reach this point from its departure at the preceding station, and analog for $t_j^r$. Assume that a minimum headway $h_{ij}$ has to be respected from train service $i$ to $j$, and $h_{ji}$ if this sequence order is reversed. Then we have

$$h_{ij} \leq (\tau_j + t_j^r) - (\tau_i + t_i^r) - z_{ij}T \leq T - h_{ji}.$$

Rewriting in the form (2) gives the infrastructure constraint

$$h_{ij} + t_i^r - t_j^r \leq \tau_j - \tau_i - z_{ij}T \leq T - h_{ji} + t_i^r - t_j^r.$$  

(3)
Figure 1: Synchronization constraint: the time window between the departure time $\tau_i$ of the feeder train and the departure time $\tau_j$ of the connecting train at the next station consists of the running time $t'_i$, the minimum transfer time $t_{ij}^{\text{min}}$, and the transfer buffer time $r_{ij}$

So here $l_{ij} = h_{ij} + t'_i - t'_j$ and $r_{ij}^{\text{max}} = T - h_{ij} - h_{ji}$, see Figure 2. The associated headway buffer time at the junction is thus $r_{ij} \in [0, T - h_{ij} - h_{ji}]$. Also the headway buffer time between train service $j$ and the next train service $i$ is here determined as

$$r_{ji} = r_{ij}^{\text{max}} - r_{ij}$$

since the sum of the minimum headways and headway buffer times of a train pair must equal the cycle time $T$.

Infrastructure constraints also arise with respect to simultaneous arrivals and departures at transfer stations, due to limitations on available train paths at the station track layout. Moreover, train services that use the same route to the next station are restricted by the safety distance of trains on these common track sections. This results in minimum arrival headways between arriving train pairs and minimum departure headways between departing trains. The associated constraints are also given as (3), where for the arrival headway constraints the running times $t'_i$ and $t'_j$ correspond to the running times from the preceding stations to the mutual station or to overtaking points at an intermediate stop, and for the departure headway constraints $t'_i = t'_j = 0$, see Figure 2.

Finally, additional market constraints are restrictions with respect to market requirements. This models for instance relative restrictions on the departure times of two train services to force semi-regular interdeparture times. For instance, if a line has twice the frequency of the overall cycle time then this line is modelled by two sequences of train services. The frequency constraint may then specify that two train services have a semi-regular interdeparture time.
Figure 2: Infrastructure constraints: conflict point (left) and departure headway (right). The sum of the minimum headways in both directions ($h_{ij}$ and $h_{ji}$) and the headway buffer times $r_{ij}$ and $r_{ji}$ equals the cycle time $T$.

Train services $i$ and $j$ must be separated by half the cycle time with a tolerance of $\delta$ so that twice in a period a train service departs in a certain direction in a semi-regular fashion. Then the frequency constraint is

$$\frac{1}{2}T - \delta \leq \tau_j - \tau_i - z_{ij}T \leq \frac{1}{2}T + \delta.$$  

Here $l_{ij} = \frac{1}{2}T - \delta$ and $r_{ij}^{\text{max}} = 2\delta$. The interpretation of the buffer time is in this case obsolete. It can now be interpreted as a tolerance where $r_{ij}^{\text{max}}$ is the absolute tolerance. Market constraints may also specify exact departure times for a train service. This models prefixed departure times of for instance international (high speed) trains. For this a fictitious train service 0 is defined that has a predetermined departure time $\tau_0 = 0$. Assume that a train service $j$ has to depart at exactly $\tau_j = \tau$ then

$$\tau \leq \tau_j - \tau_0 - z_{0j}T \leq \tau.$$  

The maximum buffer time is 0 min and so $r_{0j} = 0$ as well.

The time window constraint system can also be given in matrix notation. Let $n$ be the total number of events, and $m$ be the number of time window constraints. Number all constraints from 1 to $m$. Suppose that the $k$th constraint is $\tau_j - \tau_i = l_{ij} + r_{ij} + z_{ij}T$. Then define the matrix $M \in \mathbb{Z}^{n \times m}$ as $(M)_{jk} = 1$, $(M)_{ik} = -1$ and $(M)_{lk} = 0$ for all $l \in \{1, \ldots, n\} \setminus \{i, j\}$. So a constraint corresponds with a column of two nonzeros. Let $\tau \in [0,T]^n$ be the vector of all event times, and define vectors $l \in \mathbb{R}^m$ and $z \in \mathbb{Z}^m$ according to their constraint (arc) number, so e.g. $l_k := l_{ij}$. Furthermore, denote the vector of buffer times as $x \in \mathbb{R}^n_+$ defined
Figure 3: Constraint graph

as \( x_k := r_{ij} \), and equivalently \( \bar{x} \in [0,T]^m \) is defined as \( \bar{x}_k := r_{ij}^{\text{max}} \). Then the constraint system is

\[
M^T \tau - T z = l + x \\
x \in [0,\bar{x}], \quad z \in \mathbb{Z}^m.
\] (6)

Note that we use the transpose matrix \( M^T \) rather than directly defining a matrix \( N = M^T \). This is motivated by the interpretation of the matrix \( M \) as the node-arc incidence matrix of a directed graph, the so-called constraint graph.

The constraint graph is a directed graph (or digraph) \( G = (V, E) \), where \( V \) is the set of \( n \) nodes and \( E \) is the set of \( m \) arcs, see Figure 3. The nodes \( i \in V \) correspond to the periodic events \( \tau_i \) and for each pair \( ij \) in a time window constraint there is a directed arc \( e_k = (i, j) \in E \) from \( i \) to \( j \). Each arc has an associated lower and upper capacity given as \( l_{ij} \) and \( u_{ij} = l_{ij} + r_{ij}^{\text{max}} \), respectively. Note that the constraint graph can have multiple arcs between two adjacent nodes, if several time window constraints are defined for one pair of event times. The columns of \( M \) define the arcs of the constraint graph: an arc \( e_k \) starts at node \( i \) for which \( (M)_{ik} = -1 \) and terminates at node \( j \) for which \( (M)_{jk} = 1 \).

In the sequel we assume that the constraint graph is strongly connected, i.e., there is a directed path between any two nodes. Otherwise, the problem can be decomposed into subproblems for which the constraint graphs are strongly connected. These subproblems can then be solved separately, and the solutions can afterwards be linked to one overall timetable. Note that the constraint graph contains all interactions between train services, including infrastructure restrictions. Second, we assume that the constraint graph does not contain loops, i.e., cycles of one arc. Note that a loop can not be defined by a node-arc incidence matrix. A loop would imply a relationship between an event to itself. This occurs for instance if a train circulates on a route without interactions to other trains but for one station. The loop constraint then specifies that the sum of the (circulation) running time and the dwell time at the one station where interactions occur has to be a multiple of the cycle time. Clearly, this constraint can be solved independently and does not influence the remaining constraint system.

The time window constraint system (6) contains both the event times \( \tau \) and the buffer times \( x \). Note that these variables are redundant: if we know the event times then we also know the resulting buffer times, and vice versa. Since we are interested in optimizing the buffer times it is desirable to use these buffer times
as decision variables. Note that since each buffer time \( x_k \in [0, T] \), these variables do not suffer from modular arithmetic and can be used as variables in objective functions without causing discontinuities and thereby disturbing convexity. The next section derives an equivalent constraint system in the variables \( x_k \) (\( k = 1, \ldots, m \)) only.

### 2.3 Cycle Constraints

Consider a simple railway network of two stations and trains circulating between the stations. Then the sum of running times and dwell times on this circulation must be a multiple of the cycle time since after returning at its origin station the train has to wait for its periodic departure time, which is exactly a multiple of an hour (or more generally, the cycle time) after its former departure, before starting a new circulation. So the sum of running times, minimum dwell times, and dwell buffer times has to be a multiple of the cycle time. Since the running times and minimum dwell times are given, this results in a constraint with respect to the buffer times. This is an example of a cycle constraint.

The above example can be generalized to circuits in a complex railway network with lots of interactions, represented by the constraint graph. From graph theory it is known that any circuit can be obtained by a linear combination of fundamental cycles [1]. A cycle is a sequence of adjacent arcs (a path) where the initial and final node is the same. The arcs on a cycle may have different directions. So it is not necessary that a cycle is an ordered sequence of successive processes (a circuit). Thus, more abstract, on each cycle in the constraint graph the oriented sum of the process times (running times, minimum dwell times, minimum transfer times, and minimum headway) and buffer times must be an integral multiple of the cycle time \( T \). It is sufficient to consider only fundamental cycles of a cycle basis from which all other cycles (and all circuits) can be obtained. The (finite) number of fundamental cycles in the cycle basis, the so-called cyclomatic number, is \( \nu = m - n + 1 \), where \( m \) and \( n \) are the number of arcs and nodes in the graph, respectively. This can be clarified as follows: a spanning tree of a digraph \( G \) is a subgraph that contains all nodes of \( G \) but no cycles. The number of arcs of a spanning tree is \( n - 1 \) because any additional arc gives a cycle. The number of nontree arcs is therefore \( m - (n - 1) = \nu \). Each nontree arc generates a fundamental cycle (together with the path on the spanning tree from the tail of the nontree arc to its head). The resulting \( \nu \) cycles form a cycle basis. Note that a cycle basis is not unique.

The cycle constraint system defines all interactions between services given in terms of buffer times and taking into account the network structure. The periodicity of the event times is replaced by the periodicity of circulation times on cycles. Thus, the system of \( \nu = m - n + 1 \) cycle constraints is equivalent with the system of \( m \) time window constraints (all interactions). Note that less cycle constraints are necessary to model all interactions as a result of incorporating the network structure. This implies that the number of integer variables in the constraint system is reduced by \( n - 1 \).
A cycle constraint system can be obtained by using the cycle matrix representation of the constraint graph. The cycle (-arc) matrix $\Gamma$ is defined as the $\nu \times m$ matrix where each row is a vector representation of a fundamental cycle. That is, $(\Gamma)_{ij} = 1$ if arc $j$ is a forward arc on cycle $i$, $(\Gamma)_{ij} = -1$ if arc $j$ is a backward arc on cycle $i$, and $(\Gamma)_{ij} = 0$ otherwise. The direction of an arc on a cycle depends on the orientation of the cycle. Using the cycle matrix the constraint system becomes

$$\Gamma x - Tz = b$$

$$0 \leq x \leq \bar{x}, \quad z \in \mathbb{Z}^\nu,$$

(7)

where $\Gamma \in \mathbb{R}^{\nu \times m}$ is the cycle matrix, $b \in \mathbb{R}^m$ is the vector of the negative oriented sum of minimum process times of the constraint graph/time window constraints, $\bar{x} \in [0, T]^m$ is the vector of maximum buffer times, and $T \in \mathbb{N}$ is the cycle time. Note that the right-hand side can be computed as $b = -\Gamma l$, where $l = (l_1, \ldots, l_m)^T$ are the minimum process times of the constraint graph/time window constraints.

If a solution $x$ to (7) has been computed then we can also simply compute the associated event time vector $\tau$ by solving $M^T \tau = l + x$ for $\tau$, and subsequently set $\tau := \tau \pmod{T}$. Since $l + x$ is now a given vector $M^T \tau = l + x$ can be computed efficiently using standard numerical algorithms such as LU decomposition or Gauss-Jordan elimination. Moreover, note that we in fact only need $n - 1$ independent rows of $M^T$ to completely determine $\tau$ up to a constant (one degree of freedom). The event time vector $\tau$ is then completely determined by fixing one event time. Note that if absolute constraints (5) are imposed then $\tau$ is uniquely determined by fixing the fictitious event time $\tau_0 = 0$. On the other hand, if we have found a solution $\tau$ to the time window constraint system (6) then the associated buffer time vector is simply obtained as $x = M^T \tau - l \pmod{T}$. This shows that the time window constraint system (6) and the cycle constraint system (7) are equivalent.

Above we gave a graph-theoretical construction of a cycle basis based on a spanning tree in the graph. This gives an algorithm to compute the cycle matrix. Below we present an alternative algorithm for computing the cycle matrix based on linear algebra.

Recall from linear algebra that a null space of a matrix $M$ is the space spanned by vectors $x$ such that $Mx = 0$. A null space can be computed efficiently by Gaussian elimination with back-substitution (or by Gauss-Jordan elimination) in $O(m^3)$ time for general matrices, see e.g. Strang [27]. The following theorem states that this algorithm can also be applied to compute a cycle matrix from the incidence matrix. Note that since the incidence matrix is very sparse ($m = 2n$), the efficiency of the algorithm is better than $O(m^3)$.

**Theorem 1** Let $G$ be a directed graph and $M$ the associated (node-arc) incidence matrix. Then

1. a cycle basis of $G$ is also a basis of the null space of $M$,
2. if a basis of the null space of $M$ is given by vectors in $\{0, \pm 1\}^{m-n+1}$ then it is also a cycle basis,
the matrix of which the rows are the basis vectors of the null space of $M$ computed by Gaussian elimination with back-substitution, is a cycle matrix.

Proof. Consider the incidence matrix $M$ in more detail. The rows correspond to nodes and the columns to arcs in the associated digraph. At each row a 1 implies that the arc is an incoming arc of the node and a $-1$ implies an outgoing arc. On the other hand, each column in $M$ can be viewed as a vector representation of an arc, where the entry $-1$ denotes the tail and the entry 1 denotes the head. Now, observe that on a cycle the number of incoming arcs over all nodes equals the number of outgoing arcs. Thus, a cycle can be found as a linear combination of all columns (arcs) giving the zero vector, that is, by $x \in \{0, \pm 1\}^\nu$ solving $Mx = 0$. These vectors $x$ are just elements of the null space of $M$. On the other hand, the range of an incidence matrix $M$ is the space spanned by the vectors corresponding to the arcs in the graph, i.e., all possible combinations $Mx$ of the columns of $M$. A basis of the range is a set of independent vectors spanning this vector space. This corresponds to a spanning tree in the graph. The dimension of the range of a node-arc incidence matrix equals its rank, which is $n - 1$. The dimension of the null space then is $m - (n - 1) = m - n + 1$. Recall that a cycle basis also contains $\nu$ fundamental (independent) cycles. It follows that a cycle basis is also a basis of the null space of $M$. This proves the first statement of the theorem. Reversely, since a cycle is represented as a vector with entries in $\{0, \pm 1\}$ this is an additional requirement for a basis of the null space to be a cycle basis, which proves the second statement.

Finally, we prove the third statement. Assume that we compute $\nu$ linearly independent solutions $x$ to $Mx = 0$ by Gaussian elimination with back-substitution (choosing unit vectors for the free variables in the back-substitution process). Then $x \in \{0, \pm 1\}^m$. This can be proven as follows. The independent columns correspond to a spanning tree on $G$. Taking one free variable $x_k = 1$ at a time and the remaining free variables zero, and solving for the independent variables by back-substitution, gives the free variable (non-tree arc) in terms of the independent variables which corresponds exactly to the associated path from the tail, across the spanning tree, to the head and closing the cycle. Or equivalently, the non-tree arc combined with the reversed path on the spanning tree is zero. The tree arcs are traversed only once in the reversed direction and so $x = \{0, \pm 1\}^m$. □

2.4 Rolling Stock Constraints

The amount of buffer time to be scheduled in the timetable may be bounded by a maximum amount of rolling stock. This imposes additional constraints to the timetable constraint system. These constraints can be formulated as linear (in-)equalities of the buffer times and do not contain additional integer variables. An alternative way to include rolling stock constraints is to express them in terms of the integral multipliers $z$. In this way the rolling stock constraints restrict the feasible region of the integral variables. This results in a smaller search space and can thus decrease the computational effort necessary to find the optimal solution.
Assume that trains circulate on fixed sequences of lines exclusively. An example is the popular case that trains are assigned to two lines corresponding to running a specific route in both directions. In this case a train arriving at its terminating station turns around to run in its opposite direction and is not assigned to a different line. Also more complex routings are possible covering a certain sequence of different lines that give a circuit in the service network. Such a circuit is called the circulation of a train.

A rolling stock constraint is modelled as follows. Let \( \bar{w}_i \) be a prescribed maximum number of trains on circulation \( i \). Then the rolling stock constraint is given as

\[
\sum_{k=1}^{m} u_{ik} x_k + d_i \leq \bar{w}_i T,
\]

where \( T \) is the cycle time, \( m \) is the number of connections/arc's in the constraint graph, \( x_k \) is the buffer time of stop \( k \) on the circulation, \( d_i \) is the minimum circulation time of circulation \( i \) (the sum of running times and minimum dwell and turn-around times), and \( u_{ik} = 1 \) if circulation \( i \) contains stop \( k \) and \( u_{ik} = 0 \) otherwise. Note that a train circulation is a circuit.

These constraints can also be written in vector notation. Let the number of circulations be denoted as \( \nu \) and define the matrix \( U \in \{0,1\}^{\nu \times m} \) as \( (U)_{ij} = u_{ij} \). Then the rolling stock constraints are given as

\[
U x + d \leq T\bar{w}.
\]

Here the inequality is defined componentwise. The vector \( d \) is given as \( d = Ul \), where \( l \) is the vector of minimum process times (or lower arc capacities in the constraint graph). The rolling stock constraints can now be taken into account by adding (8) to the periodic constraint system (7).

An alternative way to include the rolling stock constraints in the constraint system is to express the maximum number of trains of all circulations in the integral multipliers \( z \). This restricts the feasible region of the integral variables and the constraint system (7) is not expanded. The rolling stock constraints are now implicitly present in the feasible region of the integral variables.

Recall that any circuit can be written as a combination of fundamental cycles. The rows of the matrix \( U \) are the vectors representing the circulations, i.e., \( U = (u_1, \ldots, u_\nu)^T \), where \( u_i = (u_{i1}, \ldots, u_{im}) \). The cycle basis consists of the rows of the cycle matrix. A circulation \( i \) represented by the row vector \( u_i \) can thus be written as

\[
u_i = c_i \Gamma, \quad i = 1, \ldots, \nu,
\]

where \( c_i = (c_{i1}, \ldots, c_{iv}) \) is a row vector representing the scalar multiples of the rows of the cycle matrix \( \Gamma \) (the fundamental cycles) which generate the circulation \( i \). Note that (9) can be written in matrix notation as \( U = C\Gamma \), where \( C = (c_1, \ldots, c_\nu)^T \in \mathbb{Z}^{\nu \times \nu} \) is an unknown matrix. This matrix \( C \) can be found by solving (9) (or in standard form \( u_i^T = \Gamma^T c_i^T \)) for all \( i = 1, \ldots, \nu \) using Gaussian elimination.
Now the feasible set of integral variables $z$ can be restricted as follows.

\[
U x + d \leq T \bar{w} \quad \iff \\
CT x + d \leq T \bar{w} \quad \iff \\
C(b + T z) + d \leq T \bar{w} \quad \iff \text{(by (7))} \\
Cz \leq \bar{w} - (Cb + d)/T.
\]

This results in the following constraint on the feasible region:

\[
Cz \leq \lfloor \bar{w} - \frac{1}{T}(Cb + d) \rfloor.
\]

In the same vein, a given minimum number $w_i$ of trains on all circulations $i$ results in the following constraint on the feasible region of the integral variables:

\[
Cz \geq \lceil w - \frac{1}{T}(Cb + d) \rceil.
\]

Note that a minimum number of trains on a circulation can be obtained if all buffer times are zero. By (8) this gives $w \geq \lceil d/T \rceil$. So if no explicit minimum number of trains is given than this lower bound can be used to restrict the feasible integer region. Note that it is now easily checked whether or not a given minimum number of trains on a circulation is feasible by comparing it with this lower bound. If the number of trains $w$ on all circulations are fixed in advance then the feasible region is restricted by

\[
Cz = w - (Cb + d)/T
\]

Of course also combinations of the above three rolling stock circulation constraints are possible.

A different kind of a rolling stock constraint is obtained when the maximum amount of trains on the entire service network is given. In this case the allocation of trains to routes is still free. The optimization problem then also determines the optimal number of trains taking the given upper bound into account. Let $\bar{v} \in \mathbb{N}$ be the maximum number of trains. In the same notation as introduced above, the rolling stock constraint becomes

\[
\sum_{i=1}^{v} \sum_{k=1}^{m} u_{ik}(x_k + l_k) \leq \bar{v}T.
\]  

(10)

This constraint implies that the sum of the minimum circulation times and buffer times on all circulations must not be larger than the maximum number of trains to be allocated multiplied by the cycle time. If a minimum amount of rolling stock is desired then the analog to (10) holds with a $\geq$ inequality, and for a fixed number of rolling stock the analog to (10) holds with equality.

Note that these rolling stock constraints may result in infeasibility if the number of trains is not enough to operate the service network according to a desired cycle time $T$. This is easily checked by considering inequality (10) with all $x_k = 0$. The minimum necessary number of trains is thus at least

\[
v \geq \left\lceil \frac{\sum_{i=1}^{v} \sum_{k=1}^{m} u_{ik}l_k}{T} \right\rceil.
\]
Again the rolling stock constraint can also be expressed in the integral variables. In the same vein as before we then obtain for a given maximum amount of rolling stock

\[
\sum_{i=1}^{v} \sum_{j=1}^{\nu} c_{ij} z_j \leq \left[ \bar{u} - \frac{1}{T} \left( \sum_{i=1}^{v} \sum_{j=1}^{\nu} (c_{ij} b_j) + \sum_{i=1}^{v} \sum_{k=1}^{m} (u_{ik} l_k) \right) \right].
\]

This constraint gives an upper bound to a linear combination of the integral multipliers \( z_j \) associated with the \( j \)th cycle constraints. For a given minimum amount of rolling stock the analog to (11) holds with a \( \geq \) inequality (and rounding to the nearest integer above), and for a fixed amount of rolling stock the analog to (11) holds with equality. Note that the right-hand side of (11) is a fixed integer.
3 Railway Timetable Optimization

3.1 Introduction

The railway timetable optimization problem can be viewed as computing a feasible network timetable with an optimal allocation of buffer times. Buffer times are the variables in this optimization problem. The objectives differ between the various buffer times. Dwell buffer times should be as small as possible to obtain small travel times. Transfer buffer times can be used to guarantee reliable connections and so some buffer time has a positive impact. Turn-around buffer times should guarantee the stability of train circulations. Buffer times in rolling stock and crew connection times are convenient for improved punctuality. Headway buffer times reduce the probability of conflicts and are desired as large as possible.

This chapter presents the general periodic railway optimization problem based on the cycle constraints as derived in the previous chapter. These constraints agree with the DONS interface, that is, the generated files of variables and constraints used by CADANS (the solver of DONS) to compute a feasible timetable are also sufficient to obtain the variables and constraints in the optimization problem presented in this paper. So the optimization problem is compatible with DONS.

3.2 The Periodic Railway Timetable Optimization Problem

Let \( m \) be the total number of buffer times (or interactions), and let \( x_k \) \((k = 1, \ldots, m)\) be the buffer time with respect to an interaction \( k \). Then the periodic railway timetable optimization problem can be given as

\[
\min F(x_1, \ldots, x_m)
\]

\[
\sum_{k=1}^{m} \gamma_{ik} x_k - Tz_i = b_i \quad i = 1, \ldots, \nu
\]

\[
0 \leq x_k \leq \bar{x}_k \quad k = 1, \ldots, m
\]

\[
z_i \in \mathbb{Z} \quad i = 1, \ldots, \nu.
\]

Here, \( F : \mathbb{R}^m \rightarrow \mathbb{R} \) is a cost function of the buffer times \( x_k \) \((k = 1, \ldots, m)\), \( \bar{x}_k \in [0, T] \) \((k = 1, \ldots, m)\) are the maximum buffer times, \( T \) is the cycle time (or period length), \(-b_i\) is the oriented sum of minimum process times on a fundamental cycle \( i \) of the constraint graph, \( \nu \) is the number of fundamental cycles in the constraint graph, \( z_i \) \((i = 1, \ldots, \nu)\) are integers, and \( \gamma_{ik} = 1 \) if interaction \( k \) corresponds to a forward arc on cycle \( i \), \( \gamma_{ik} = -1 \) if interaction \( k \) corresponds with a backward arc on cycle \( i \), and \( \gamma_{ik} = 0 \) otherwise. Note that the (cycle) constraints state that the oriented sum of process times and buffer times on a fundamental cycle of the constraint graph equals an integral multiple of the cycle time \( T \).
The optimization problem can also be formulated in vector notation as

$$\min F(x)$$

$$\Gamma x - Tz = b$$

$$0 \leq x \leq \bar{x}, \quad z \in \mathbb{Z}^\nu,$$

where $\Gamma \in \mathbb{R}^{\nu \times m}$ is the cycle matrix, $b \in \mathbb{R}^m$ is given as $b = -\Gamma l$ where $l \in [0, \infty)^m$ is the vector of the minimum process times of the constraint graph/time window constraints, $\bar{x} \in [0, T]^m$ is the vector of maximum buffer times, and $T \in \mathbb{N}$ is the cycle time.

Additionally, the integers can be constrained by a convex set $\Omega \subseteq \mathbb{Z}^\nu$ defined by constraints on the amount of rolling stock, see Section 2.4. Also integer lower and upper bounds can be computed, see Section 3.4. These integer bounds can also be incorporated in $\Omega$.

The function $F : \mathbb{R}^m \rightarrow \mathbb{R}$ models the ‘cost’ of any allocation of buffer times over the timetable. From a computational point of view the cost function $F$ has to be convex implying that any local optimal solution is also the global optimum. For nonconvex $F$ the problem is practically unsolvable. A convenient cost function is a (weighted) sum of the relative costs of all buffer times, i.e.,

$$F(x) = \sum_{k=1}^{m} c_k F_k(x_k)$$

where $c_k \geq 0$ are positive real numbers, and the functions $F_k : \mathbb{R} \rightarrow \mathbb{R}$ are smooth convex functions of a single buffer time. The weights $c_k$ give the relative importance of the individual cost contributions and thereby tune the priorities of the buffer times. This particular form of $F$ is a convex separable function of the buffer times. Note that a linear cost function $c^T x$ is a special case of (14). This separable formulation is convenient for modelling the individual cost contributions of all buffer times. In the next chapter several appropriate cost functions are presented representing the buffer time costs to passengers and/or railway operators.

Another possible formulation of cost functions is $F(x) = \max_{k=1,\ldots,m} [F_k(x_k)]$, resulting in a minimax problem. The objective is here to minimize the maximum of all buffer time costs. For example, if $F_k(x_k) = x_k$ $(k = 1, \ldots, m)$ then the objective is to minimize the maximal buffer time over all buffer times. However, for our purposes this objective is not felt appropriate as the individual buffer times vary in interpretation. Only for special cases of uniform variables this objective may be interesting. For instance, if we are only interested in minimizing the transfer times over the network. Also the priorities of the individual buffer times can not be modelled in a minimax problem.

### 3.3 Computational Complexity

The minimization problem (12) is a convex mixed-integer programming problem, with linear mixed-integer constraints and a convex (separable) cost function. Note
that although the arrival/departure times are periodic events, the buffer times 
\(x_k \in [0, T]\) are not periodic. Therefore the cost function is not affected by per-
riodicity. If the optimization problem would have been formulated directly with re-
spect to the time window constraints and the cost function as a function of the 
periodic arrival/departure times then the cost function would have been noncon-
vex by the modular arithmetic (arrival and departure times have to be evaluated
modulo the cycle time).

For fixed integers \(z\) (and \(F\) convex in \(x\)), the optimization problem (12) becomes
a convex programming problem, and if \(F\) is linear in \(x\) then the problem is a
linear programming problem. These problems can be solved efficiently.

However, the integer variables \(z\) in the mixed-integer constraints make the prob-
lem difficult to solve. In fact, the optimization problem (12) is \(NP\)-complete
which follows from the \(NP\)-completeness of the underlying feasibility problem of
finding solutions to the cycle constraint system (or the time window constraint
system). For small networks the railway optimization problem can be solved
exactly by a branch-and-bound algorithm [21] using convex programming relax-
ations.

For large-scale networks more effort is necessary to solve the problem. Both for
solving the convex relaxation problem and the mixed-integer part. In particular
the structure of the timetable constraint system (the timetable polyhedra) can
be examined for preprocessing, and to tighten the feasible region by which the
branch-and-bound procedure becomes more effective. Nachtwall [20] examined
the timetable polyhedron and found two classes of facet defining valid inequalities.
Moreover he developed two polynomial separation algorithms which can be used
to generate strong cutting planes.

\[\sum_{j=1}^{m} \gamma^+_{ik} x_k \leq b_i + z_i T \leq \sum_{k=1}^{m} \gamma^-_{ik} x_k.\]

Let the \(\nu \times m\) matrices \(\Gamma^+\) and \(\Gamma^-\) be defined accordingly as 
\((\Gamma)^+_{ij} = \gamma^+_{ij}\), and

\[\sum_{j=1}^{m} \gamma^+_{ik} x_k \leq b_i + z_i T \leq \sum_{k=1}^{m} \gamma^-_{ik} x_k.\]
Then we obtain the lower bound vector
\[
z := \left\lceil \frac{1}{T}(\Gamma^- \bar{x} - b) \right\rceil
\]  
and upper bound vector
\[
\bar{z} := \left\lfloor \frac{1}{T}(\Gamma^+ \bar{x} - b) \right\rfloor.
\]  
Here \( \lceil a \rceil \) is the nearest integer above \( a \) (the ceiling operator) and \( \lfloor a \rfloor \) is the nearest integer below \( a \) (the floor operator), defined componentwise. An additional constraint to the cycle constraint system (7) is thus
\[
z \leq z \leq \bar{z}.
\]  
Note that the choice of the cycle basis influences the lower and upper bounds on the integral vector \( z \), and thus also the solution algorithm efficiency. The above derived integer bounds depend on the decision variable upper bounds \( \bar{x} \). These upper bounds (maximum buffer times) may be taken equal to the cycle time \( T \). In this way the overall optimal solution results. Note that the upper bounds have become somewhat superfluous as large buffer times are penalized by the cost functions. However, removing the buffer time upper bounds magnifies the gaps between the lower and upper bounds on \( z \) given by (15) and (16). In a branch-and-bound scheme this implies larger computation times. On the other hand, tight upper bounds \( \bar{x} \) on buffer times to avoid large waiting times may result in infeasibility of the constraint system.
4 The Network Synchronization Problem

4.1 The Optimization Problem

The aim of this paper is to find an optimal network synchronization. So we are concerned with buffer times between train connections. These include buffer times at stops, transfers, rolling stock connections, crew connections, and turns. The resulting optimization problem will be referred to as the network synchronization problem.

We consider the following optimization model as obtained in the two previous chapters:

\[
\begin{align*}
\min & \sum_{k=1}^{m} c_k F_k(x_k) \\
\sum_{k=1}^{m} \gamma_{ik} x_k - T z_i &= b_i & i = 1, \ldots, \nu \\
0 &\leq x_k \leq \bar{x}_k & k = 1, \ldots, m \\
z_i &\in \mathbb{Z} & i = 1, \ldots, \nu.
\end{align*}
\]

where \( c_k \geq 0 \) are positive real numbers, and the functions \( F_k : \mathbb{R}_+ \to \mathbb{R} \) are smooth convex functions of a single buffer time \( x_k \geq 0 \). Additionally, the integer vector \( z \) may be constrained by a feasible region \( \Omega \) defined by the rolling stock constraints (Section 2.4) and the integer bounds (Section 3.4). In this chapter we will present some appropriate cost functions \( F_k \) for the network synchronization problem.

We are concerned with an optimal allocation of buffer times over the network. Therefore, it is convenient to express cost in terms of time (minutes). Of course other cost units are possible as for instance the cost equivalent in cash. The following cost functions are proposed:

- **Buffer time**: minimizing dwell buffer times to obtain small travel times.
- **Expected departure delay**: maximizing buffer time reliability of logistics connections (crew connections, rolling stock connections, and turns of trains at terminal stations (stability of train circulation)).
- **Expected transfer waiting time**: optimizing a trade-off between a small transfer buffer time and a high transfer reliability.

The cost function \( F \) in the optimization problem is a (weighted) sum of these cost functions. The weights give the relative importance of the individual cost contributions. This tunes the priorities of circulation stability, crew and rolling stock connections, transfers, and travel times. The resulting network synchronization problem aims at finding a timetable with small travel times and transfer waiting times, and a punctual and reliable railway operation.
Also other choices of cost functions are possible representing the specific objectives of the problem at hand. Recall that we are concerned with connections only and so we did not consider cost functions for headway buffer times. Of course, appropriate cost functions can be derived that penalize small headway buffer times between a pair of trains in both directions. The subsequent sections consider the individual cost functions for the network synchronization problem.

4.2 Tight Connections

An obvious measure for the quality of a timetable is the total (synchronization) buffer time, or the total passenger waiting time resulting from these buffer times. Optimizing this measure results in a linear cost function to the network synchronization problem: the weighted sum of buffer times. So the linear cost function is

$$F(x) = \sum_{k=1}^{m} c_k x_k,$$

(19)

where $c_k \geq 0$ is a constant weight reflecting the priority of dwell/transfer connection $k$, for instance an estimate of the (relative) passenger flow, and $c_k = 0$ for headway buffer times.

The solution to the linear network synchronization problem gives an optimal timetable for punctual operation. However, with respect to arrival delays the resulting timetable may be far from optimal. The cost function (19) tends to give tight transfer times for high priority connections. In case of arrival delays this implies that either the connecting train is frequently delayed as well or, in the case of transfer connections, large amount of transferring passengers miss their connection. The linear cost function for all buffer times should therefore be considered only if the arrival times have a large punctuality percentage.

However, for dwell buffer times $x_k$ the linear cost function $F_k(x_k) = c_k x_k$ is convenient as it minimizes passenger travel times and train circulation times. Note that train running times include running time margins to neutralize the stochastic driving behaviour. Dwell buffer times also compensate arrival delays but this is just a (positive) side-effect. In principle the dwell times should be as small as possible and possible variation in running times should be compensated by running time margins.

An interesting observation is that the linear network synchronization problem (where $F(x) = c^T x$) has integral optimal buffer times as a result of a special property of the cycle matrix and the node-arc incidence matrix (total unimodularity).

**Theorem 2** Consider the (linear) network synchronization problem (18) with linear cost function (19). If all parameters $T$, $b_k$, $\bar{x}_k$ $(k = 1, \ldots, m)$ are integral then the linear network synchronization problem has integral solutions, i.e., the optimal buffer times and corresponding event times are also integers.

**Proof.** Note that $b = \Gamma l$ is integral if $l$ is, since then $b$ is just a linear combination
of integral vectors. Moreover, since $z \in \mathbb{Z}^n$ and $T \in \mathbb{Z}$ also $zT \in \mathbb{Z}^n$. From integer programming theory it is known that for a totally unimodular (TU) matrix $A$ the solution (if any) of the linear programming problem $\{\min x | Ax \leq b, x \in \mathbb{R}^m_+\}$ is integral [21]. So in particular the relaxation problem of an MIP has integral solutions, and thus also the MIP itself. Now, the cycle time is TU which can be proven from its construction of which all operations are closed under total unimodularity. Note that the proof also holds for the network synchronization problem with time window constraints. The matrix in this case is just the transpose of the incidence matrix which is totally unimodular.

Note that it is convenient that all arrival and departure times in a timetable are given in minutes (integers). Theorem 2 then states that this is automatically obtained by the optimal solution of the linear network synchronization problem.

4.3 Reliable Connections

A major cause of (secondary) delays is represented by crew connections in which a driver or conductors have to change trains, and by rolling-stock connections for, e.g., coupling of coaches. These connections are hard, that is, a connecting service can only depart after the processes of the connection have been completed.

For these connections the reliability of the buffer time between the arriving and departing services is hence very important. A third category to which buffer time reliability is of major concern is the stability of train circulations. The desire of a punctual initial departure time of a new train circulation has to be assured by a buffer time in the turn-around time at the terminating station. Note that running time margins and dwell buffer times during the circulation also compensate for possible delays at the cost of increased (passenger) travel times. A relative large buffer in the turn-around time does not influence travel times but assures a punctual departure at the start of a new circulation. In general, a reliable connection has to be assured for logistics constraints.

A measure for the reliability of a buffer time is the expected departure delay of the connecting service due to failure of the buffer time. The expected departure delay is formally given as follows. Consider a connection $k = (i, j)$ at a railway station from a train service $i$ to a subsequent train service $j$. Let $t_i^r$ be the scheduled running time of service $i$ from the preceding station, and $x_k \geq 0$ be the connection buffer time. Furthermore, let $T_i$ be a random variable denoting the stochastic running time of service $i$, with density function $g_i$ and associated distribution function $G_i(t) = \int_0^t g_i(\tau)d\tau$, see Figure 4. Common distributions are the shifted Weibull, lognormal, Erlang, or Gamma distributions, but also empirical density functions may be used. Denote the stochastic departure delay of train service $j$ as $Y_j$. Then the probability that train $j$ is delayed is $P(Y_j > 0) = P(T_i > t_i^r + x_k)$. Thus, the expected departure delay is $E[Y_j] = E[(T_i - t_i^r - x_k)^+]$ with $(\cdot)^+ = \max(0, \cdot)$, that is

$$E[Y_j] = \int_{t_i^r + x_k}^{\infty} (\tau - t_i^r - x_k)g_i(\tau)d\tau$$

$$= \int_0^{\infty} (p - x_k)^+ g_i(p + t_i^r)dp.$$
Note that the integration variable $p$ corresponds to the arrival delay of service $i$. So instead of the running time distribution, also the distribution of the induced stochastic arrival delay $P_i$ of service $i$ is sufficient. Note that the probability that train service $i$ arrives on-time is given as $P(T_i \leq t_i^*) = G_i(t_i^*)$. So typically, a random arrival delay $P_i$ has a density function

$$f_i(p) = \begin{cases} G_i(t_i^*) & \text{if } p = 0 \\ g_i(p + t_i^*) & \text{if } p > 0, \end{cases}$$

where $g_i$, $G_i$, and $t_i^*$ are interpreted as above.

Above we silently assumed that the arrival delay of a feeder train is the result of a stochastic running time only. Thus, a possible departure delay at the start of the train run is neglected. The underlying idea is that at each connection the buffer time is computed to cope with the primary delays as good as possible. But this also implies that the secondary delays are controlled as good as possible. Therefore, this approach can be interpreted as the minimization of a first-order approximation of the delay propagation with respect to primary delays.

Instead of looking at the expected departure delay we can more generally look at the expectation that an event is $\delta$ minutes late, with $\delta \geq 0$. The probability that train $j$ departs $\delta$ minutes late is given as $P(Y_j \geq \delta) = P(T_i \geq t_i^* + x_k + \delta)$ and the expectation of a departure delay more than $\delta$ minutes is $E[(T_i - t_i^* - x_k - \delta)]$, i.e.,

$$E[Y_j - \delta] = \int_{t_i^* + x_k + \delta}^{\infty} (\tau - t_i^* - x_k - \delta)g_i(\tau)d\tau = \int_{0}^{\infty} (p - x_k - \delta)^+ g_i(p + t_i^*)dp.$$

The expected departure delay is now a special case for $\delta = 0$. The following theorem states that the expectation $E[Y_j - \delta] = E[P_i - x_k - \delta]$ is convex in the buffer time $x_k$ for all $\delta \geq 0$. 

Figure 4: Buffer time failure of running time density function
Theorem 3 Let \( g : [0, \infty) \to [0, \infty) \) be a density function, and \( t, \delta \in [0, \infty) \).
Then
\[
\int_0^\infty (p - x - \delta)^+ g(p + t) \, dp
\]
(20)
is a convex function of \( x \) on the domain \([0, \infty)\).

Proof. Recall that \( \max(0, x) \) is a convex function. Denote the integral function (20) as \( h(x) \). Let \( \alpha \in [0, 1] \). Then we have to prove that \( h(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha h(x_1) + (1 - \alpha)h(x_2) \). We have
\[
\begin{align*}
h(\alpha x_1 + (1 - \alpha)x_2) &= \int_0^\infty (p - \alpha x_1 - (1 - \alpha)x_2 - \delta)^+ g(p + t) \, dp \\
&= \int_0^\infty \left( \alpha(p - x_1 - \delta) + (1 - \alpha)(p - x_2 - \delta) \right)^+ g(p + t) \, dp \\
&\leq \int_0^\infty \left[ \alpha(p - x_1 - \delta)^+ + (1 - \alpha)(p - x_2 - \delta)^+ \right] g(p + t) \, dp \\
&= \alpha \int_0^\infty (p - x_1 - \delta)^+ g(p + t) \, dp + (1 - \alpha) \int_0^\infty (p - x_2 - \delta)^+ g(p + t) \, dp \\
&= \alpha h(x_1) + (1 - \alpha)h(x_2).
\end{align*}
\]
This cost function can also be applied to optimize running time margins. So let the running time also consist of a constant minimum running time and a variable running time margin that has to be determined in the optimization. Then the running time margins are also decision variables in the optimization problem. Now the running time margin and the dwell buffer time (or turn-around buffer time, etc.) are jointly optimized. An appropriate cost function for the running time margin is then for instance the expectation of arriving 2 min late, and for the dwell buffer time just the buffer time. Then the optimization problem aims at optimizing the trade-off between a reliable running time (margin) and a small dwell (buffer) time. Note that more flexibility of the process times represented by the variable (buffer time/margin) part, also reduces the chance of infeasible constraints.

A further extension of the optimization model is to include knock-on delays along the route of a train. Carey and Kwiciński proved that the resulting expected delays along the route are convex in the successive buffer times/margins. However, the evaluation of these cost functions may take a lot of computation time. So here is a trade-off between a more accurate model and tractability of the optimization problem. A better approach may be to use these stochastic models in a post-evaluation of the solution to the optimization model.

4.4 Transfer Connections

The effect of arrival delays on transfer reliability can be dealt with by incorporating the risk and significance of missing connections in the objective function. This can be accomplished by considering the mean or expected transfer waiting time. Note that a missed connection results in large waiting times. This section
derives a model for the expected transfer waiting time based on a modified version of the approach in Knoppers and Muller [14]. Knoppers and Muller derive a periodic function of the transfer waiting time, whereas here we derive a smooth convex function of the transfer buffer time, see also Goverde [9].

Consider a connection \( k = (i, j) \) from a train service \( i \) to \( j \). The transfer waiting time is a function of the arrival delay \( p_i \) of the feeder service \( i \). If the feeder service arrives on time or within the transfer buffer time \( x_k \) then the transferring passengers have to wait for the scheduled departure time. For intermediate arrival delays of the feeder service the connecting service \( j \) may wait to secure the connection, and departs right after the arrival of the transferring passengers. If the feeder service has an arrival delay for which securing the connection would result in exceeding the maximum admissible train waiting time, the synchronization control margin \( \bar{s}_j \), then the connection is cancelled and the transferring passengers miss the connection. The passengers then have to wait on the next train of the connecting line. Assuming that the connecting train departs on-time, the transfer waiting time can be given as

\[
w_k(p_i) = \begin{cases} 
  x_k - p_i & \text{if } 0 \leq p_i \leq x_k \\
  x_k + \omega_j - p_i & \text{if } x_k + \bar{s}_j < p_i \leq x_k + \omega_j \\
  0 & \text{otherwise},
\end{cases}
\]  

(21)

see Figure 5. Here \( x_k \) is the transfer buffer time of connection \( k = (i, j) \), \( \omega_j \) is the interdeparture time (the reciprocal of the frequency) of the connecting service \( j \), and \( \bar{s}_j < \omega_j \) is the synchronization control margin of train \( j \).

The above derivation of the transfer waiting time assumes that transferring passengers who miss the connection are able to catch the next train of the connecting line and also do not leave the station in a train of an alternative line. Moreover it is assumed that the connecting train and its successor depart on time. These assumptions are reasonable from a scheduling point of view and result in a considerable reduction of complexity.

To obtain an expression for the expected transfer waiting time the distribution of the arrival delays is required. In the sequel it is assumed that early arrivals

![Figure 5: Transfer waiting time](image)
do not contribute to transfer waiting time costs, i.e., early arrivals are treated as on-time arrivals.

Let \( g_i \) be the density function of the stochastic running time \( T_i \) of the feeder train service \( i \) and \( G_i(t) = \int_0^t g_i(\tau) d\tau \) the corresponding distribution function. Let \( t_i^r \) be the scheduled running time. The probability that the train arrives on-time is simply \( \mathbf{P}(T_i \leq t_i^r) = G_i(t_i^r) \). The delay density function is therefore

\[
f_i(p) = \begin{cases} G_i(t_i^r) & \text{if } p = 0 \\ g_i(p + t_i^r) & \text{if } p > 0. \end{cases}
\]

The following theorem gives a sufficient condition for which the resulting expected transfer waiting time is a convex function.

**Theorem 4** Consider a transfer connection \( k = (i, j) \). Let \( \omega_i > \bar{s}_j \geq 0 \), the function \( w_k(p) \) defined as in (21), and \( P \geq 0 \) be a stochastic variable with a nonincreasing differentiable density function \( f_i \) with a possible discontinuity at \( p = 0 \). Then the expectation

\[
\mathbf{E}[w_k(p)] = x_k \mathbf{P}(p = 0) + \int_{0^+}^{\infty} w_k(p)f_i(p)dp
\]

is a convex function of \( x_k \geq 0 \).

**Proof.** Recall that \( 0^+ = \lim_{\epsilon \downarrow 0} \) is necessary because of a possible discontinuity of \( f_i \) at \( 0 \). Let \( \bar{w}_k(x_k) = \mathbf{E}[w_k(P)] \). Below it is shown by construction that the second derivative of \( \bar{w}_k \) exists and is nonnegative which proves convexity. Let \( F_i \) be the distribution function associated with \( f_i \). Note that \( f_i \) is integrable as it is a density function and so \( F_i \) exists. Moreover note that \( F_i(0) = \mathbf{P}(p = 0) \). Using partial integration the expectation of \( w_k(P) \) can be rewritten as

\[
\bar{w}_k(x_k) = x_k \mathbf{P}(p = 0) + \int_{0^+}^{x_k} (x_k - p)f_i(p)dp + \int_{x_k + \bar{s}_j}^{x_k + h_j} (x_k + h_j - p)f_i(p)dp
\]

\[
= x_k \mathbf{P}(p = 0) + \lim_{\epsilon \downarrow 0} \left( [(x_k - p)F_i(p)]_{x_k}^{x_k + \epsilon} + \int_{x_k + \epsilon}^{x_k + h_j} F_i(p)dp \right)
\]

\[
+ [(x_k + h_j - p)F_i(p)]_{x_k + \bar{s}_j}^{x_k + h_j} + \int_{x_k + \bar{s}_j}^{x_k + h_j} F_i(p)dp
\]

\[
= x_k \mathbf{P}(p = 0) - x_k F_i(0) + \int_{0^+}^{x_k} F_i(p)dp - (h_j - \bar{s}_j)F_i(x_k + \bar{s}_j) + \int_{x_k + \bar{s}_j}^{x_k + h_j} F_i(p)dp
\]

\[
= \int_{0^+}^{x_k} F_i(p)dp - (h_j - \bar{s}_j)F_i(x_k + \bar{s}_j) + \int_{x_k + \bar{s}_j}^{x_k + h_j} F_i(p)dp.
\]

The first derivative of \( \bar{w}_k \) can be computed using the fundamental theorem of calculus, i.e., \( \frac{d}{dx} \int_a^x F(p)dp = F(x) \) for any fixed \( a \leq x \). Hence,

\[
\bar{w}_k'(x_k) = F_i(x_k) - (h_j - \bar{s}_j)f_i(x_k + \bar{s}_j) + F_i(x_k + h_j) - F_i(x_k + \bar{s}_j).
\]
By the assumption of \( f_i \) being nonincreasing, we obtain for the second derivative of \( \bar{w}_k \)

\[
\bar{w}''_k(x_k) = \frac{f_i(x_k) - f_i(x_k + \bar{s}_j) - (h_j - \bar{s}_j) f'_i(x_k + \bar{s}_j) + f_i(x_k + h_j)}{\geq 0 \quad \geq 0 \quad \geq 0} \geq 0.
\]

Typically, railway planners (unconsciously) schedule the running time after the top of its density function by which most trains (e.g. 87%) arrive on-time, see Figure 4. The expected transfer waiting time is then a convex function by Theorem 4. Examples of convenient distributions are the Weibull and Erlang distribution, or any empirical density function with a decreasing tail, see Figure 4. The expected transfer waiting time can be computed numerically. Goverde [9] considers a negative-exponential density for the arrival delays with an additional punctuality rate. This can be viewed as approximating the tail of the running time density by a negative exponential density. For this case an analytical expression for the expected transfer waiting time can be computed [9], which is convenient in the overall optimization problem.

Figure 6 shows a typical expected transfer waiting time function of the transfer buffer time for some values of the synchronization control margin. Note that the function is convex. For large buffer times (larger than 8 minutes) the influence of the synchronization control margin is negligible: any arrival delay can be compensated by the buffer time. On the other hand, the effect of the synchronization control margin is considerable for small buffer times. If the synchronization control margin is large (about 5 minutes and larger) then the minimum of the expected transfer waiting time is obtained for zero transfer buffer time. The synchronization control can then compensate all (mainly occurring) arrival delays.
The computation of a periodic railway timetable can be modelled conveniently as an optimization problem, where the buffer times are the decision variables and the cost function is convex, by taking the network structure of the (periodic event time window) constraints in account. The resulting MIP constraint system of the buffer times is compatible with the DONS system of the NS, whereby the optimization problem is suitable to extend the DONS features.

An appropriate convex cost function of the buffer times can be derived representing the individual required performance of a tight connection, a reliable connection, or a transfer connection, and evaluating the priorities between the individual buffer time performance costs. The convex cost function guarantees that a local optimal solution gives the global optimum as well.

Using the network structure of the constraints in the problem formulation significantly reduces the required number of integer variables as opposed to the conventional periodic event (time window) constraint system. The network structure is also used in CADANS to solve the time window constraint system [25]. However, modelling the network structure explicitly gives opportunities to restrict the feasible integer region by integer bounds, rolling stock constraints, and strong cutting planes. This is advantageous for solving large-scale problems.

Another progress is made by using cost functions to penalize large buffer times at stops, connections, and turns, making upper bounds on these buffer times superfluous. This considerably increases the opportunity to find feasible solutions, and thus implicitly solves the problem of finding conflicting constraints. Moreover, also running time margins can be included as decision variables in the optimization problem which improves the feasibility issue as well as timetable efficiency even more.

Current research concentrates on efficient solution algorithms to solve the presented optimization problems. Moreover, the optimization approach will be validated by a case study. Experiments with small networks give promising results.

The proposed optimization model contributes to design efficient and robust railway timetables to improve performance and attractivity of railway operations, increase service quality to passengers, and decrease costs to railway operators and passengers due to delays.

Acknowledgements

This publication is a result of the research programme Seamless Multimodal Mobility (SMM), carried out within the TRAIL Research School for Transport, Infrastructure and Logistics, and financed by Delft University of Technology.
References


