

Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable

P. Sels^{a,b,c,*}, T. Dewilde^{a,1}, D. Cattrysse^{a,2}, P. Vansteenwegen^{a,3}

^a*KU Leuven, Leuven Mobility Research Centre, CIB, Celestijnenlaan 300, 3001 Leuven, Belgium*

^b*Logically Yours BVBA, Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium*

^c*Infrabel, Department of Traffic Management & Services, Fonsnylaan 13, 1060 Brussels, Belgium*

Abstract

Automatically generating timetables has been an active research area for some time, but the application of this research in practice has been limited. We believe this is due to two reasons. Firstly, some of the models in the literature impose artificial upper bounds on time supplements. This causes a high risk of generating infeasibilities. Secondly, some models that leave out these upper bounds often generate solutions that contain some very large time supplements because these supplements are not penalised in the objective function. The reason is that these objective functions often do not completely correspond to the true goal of a timetable. We solve both problems by minimising our objective function: *total passenger travel time, expected in practice*. Since this function evaluates and indirectly steers all time related decision variables in the system, we do not need to further restrict the ranges of any of these variables. As a result, our model does not suffer from infeasibilities generated by such artificial upper bounds for supplements.

Furthermore, some measures are taken to significantly speed up the solver times of our model. These combined features result in our model being solved more quickly than previous models. As a result, our method can be used for timetabling in practice. We demonstrate our claims by optimising, in about two hours only, the timetable of all 196 hourly passenger trains in Belgium. Assuming primary delay-distributions with an average of 2% on the minima of each activity, the optimised timetable reduces expected passenger time in practice, as evaluated on the macroscopic level, by 3.8% during peak hours. This paper demonstrates that we added two important missing steps to make cyclic timetabling for passengers really useable in practice: (i) the addition of the objective function of expected passenger time in practice and (ii) the reduction of computation time by addition of well chosen additional constraints.

Keywords: Optimal Cyclic Timetabling, Mixed Integer Linear Programming, Minimal Expected Passenger Time

*Corresponding author

Email address: sels.peter@gmail.com (P. Sels)

¹Email address: thijs.dewilde@kuleuven.be

²Email address: dirk.cattrysse@kuleuven.be

³Email address: pieter.vansteenwegen@kuleuven.be

1. Introduction and Literature Comparison

In railway operations, the construction of a train timetable, on the highest level of abstraction, means the determination of the arrival and departure times for each train in the system, in each station it serves or passes. This is a large combinatorial optimisation problem with, for example, 165000 constraints and about 200000 variables for the Belgian network of hourly passenger trains. It requires hours rather than minutes to solve. Many constraints should be respected. For example, minimum ride and dwell times should not be violated, two trains on the same infrastructure resource should be separated by some minimum time and passenger transfers should be assigned neither too much nor too little time. In this section, we introduce our timetabling approach and compare it with the approaches available from the literature. We take the approach of presenting in the order that research expanded. This is usually also the chronological order but some exceptions occur.

1.1. The Objective of a Timetable

Watson (2008) reports the story of the 'disaster timetable' that was set up in the United Kingdom after the government required in 1993 that train operator companies and the railway company, back then called Railtrack, operate separately. Since the European Community imposed a similar measure on train companies in all its member states in 2005, this story has a wider relevance. Watson (2008) concludes that '*The total train planning problem was no longer any one organisation's responsibility and, hence, inevitably, Railtrack concentrated on efficient network utilisation and train operators on efficient resource utilisation.*' As an example, Watson (2008) mentions that no stakeholder was interested in the synergetic network effects of properly planning the transfers between service lines of different operators. He suggests to resolve these problems by setting a *clear objective* for the whole timetable. Additionally, *software* should be developed to support shorter train timetable development iterations. We address both remedies in this paper. We believe that setting this objective to passenger service, for example by minimising the expected passenger time, will benefit passengers. Consequently, this increased experienced customer value will trickle down along the value chain from the passengers, to the train operator companies to the infrastructure company. This paper makes clear how this objective is used in both quick, automatic optimisation and evaluation of timetables.

So our objective is to design a timetable in an automated way. The main characteristic of this timetable should be that the passenger travel time in practice, for all passengers together, is as low as possible. We take into account small train delays, occurring on a daily basis in the current timetable. We suppose fixed minimum times for ride, dwell, transfer and turn-around activities and only determine and add supplements on top of these. According to our objective, these supplements should be large enough to compensate for typical delays, but the supplements should also not be too large in order to avoid needless extra waiting or travel time for the passengers. Since we focus on timetabling, we do not consider other measures that could improve the robustness or performance of the railway system such as network design, line design, halting pattern nor dispatching rules. We successfully apply our optimisation method to the network of all passenger trains in Belgium. However, our method is generally suited for quick, automated generation of robust, cyclic timetables for any passenger transport network.

1.2. Efficiency and Robustness of a Timetable

In their timetabling overview paper, Cacchiani and Toth (2012) make a distinction between *nominal* timetabling and *robust* timetabling methods. For the first, they indicate that objective functions vary between minimising the number of changes with respect to a preferred timetable, maximising customer satisfaction by minimising (planned) passenger time and minimising cost for the railway company. For robust timetabling, they describe that there is a trade-off between *efficiency*, the nominal objective - if one focusses on passenger or train travel time - and *robustness*. They mean that both cannot be optimal at the same time. Put otherwise, there is a *cost of robustness*. To the same effect, Schöbel and Kratz (2009) propose a pareto optimisation technique with these two separate criteria. Similarly, Cacchiani et al. (2012) aim for an efficient timetable via the objective function and add a Lagrangian heuristic to it to obtain some robustness. They state that the end user is responsible to make the trade-off between efficiency and robustness. We see this differently. A better solution to this apparent dilemma is to take *expected* passenger time *in practice* as the stochastic objective function of the timetable. It is clear that both efficiency and robustness have an effect on *expected* passenger time *in practice* but if one also carefully models both the effects on this objective function via a stochastic model of expected primary delays, the objective function will make this tradeoff. This total objective function was used in Sels et al. (2011a) and is also used in this paper. Dewilde et al. (2014) also proposed

this same objective function as the preferred approach to strike the right balance between efficiency and robustness in a timetable. Supplements will not be so small that the smallest primary delay cannot be absorbed. Likewise supplements will not be so large that the journey times become inefficiently large for passengers. In the timetables produced by our model, these two positive properties are merely the automatic result of minimising the expected passenger time in practice.

Kroon et al. (2006) focus on improving a timetable that is being used in practice, assuming that train orders have to remain fixed. This means that actual realisations that are obtained from logged train times are relevant. In The Netherlands, the type and granularity of the train logging measurement points in the infrastructure allows separation of primary and secondary delays from these logs (Goverde and Hansen, 2000; Daamen et al., 2009). Kroon et al. (2006) then perform a post-optimisation of arrival and departure times so that the average (primary plus secondary) delay is minimal with respect to the primary delay distributions which historically occurred for the considered trains. They call this a mixed simulation-optimisation approach. (Liebchen et al. (2009) call this a two-stage stochastic optimisation.) This method was applied on a corridor called 'the Zaanlijn' and the resulting timetable was put in practice. This experiment confirmed the results expected from theory. In practice, the optimised timetable proved to be more robust against small primary delays. The work in Kroon et al. (2006) is intended to improve an existing timetable already in operation, while our methodology intends to construct timetables from scratch. Because of this, our method differs in two aspects. Firstly, this means that we still must determine train orders. We also choose to determine all time supplements at the same time since these are interdependent with train orders. Secondly, our model estimates secondary delays as a function of chosen train orders and time supplements when a given set of primary delays is assumed. As such, it determines the best train order and time supplements between trains. These supplements are also the essential variables to be determined in the queuing model of Wendler (2007).

Apart from robustness, a further possible requirement for timetables is the criterion of *resilience*, meaning that, after a perturbation, the timetable is able to evolve back to the original one instead of diverging from it, and preferably must do so quickly. Semet and Schoenauer (2005) tackle this via an evolutionary algorithm.

Resilience is very close to *recoverable robustness*, which is defined by Liebchen et al. (2009). This technique requires that *for every likely input scenario* of a running train, including primary delays, that the timetable is able to recover from it. They state that two-stage stochastic optimisation is not able yet to produce, from scratch, robust timetables for instances of the size of an entire country. However, in our current paper, we show that with our PESP based model which includes a stochastic objective function, this can be done in a practical computation time. This is probably because we do not use the explicit scenarios present in two-stage stochastic optimisation, which naturally make the linear program model very large, but rather assume independent primary delay distributions on every activity in the graph. Note that this assumption covers a very broad range of scenarios of likely 'input' train runs, be it implicitly. It also delivers a simpler model. Our requirement is then that the negative influence of disturbances, weighted with the probability that they occur, on expected passenger time in practice is minimal. Liebchen et al. (2009) further present a case study of recoverable robust train platforming, but not on timetabling.

Fischetti and Monaci (2009), propose a technique called *light robustness*. It couples robust optimisation with a simplified two-stage stochastic programming and sometimes achieves results of comparable quality to stochastic or robust modelling while requiring lower modelling and computational efforts.

For a more complete overview of robustness definitions, we refer to Dewilde et al. (2011). They mention that passengers prefer a timetable which implies minimal expected passenger time under expected primary delays. The authors further demonstrate that this objective results in an amount of robustness that is ideal for passengers. By choosing this objective function, the authors follow and support the recent trend in passenger railway research to give a higher weight to the concerns of the passengers. We use the same objective function in this paper.

1.3. The Periodic Event Scheduling Problem (PESP) Approach

Cyclic timetabling formulated in the PESP way has been studied by, amongst others, Serafini and Ukovich (1989), Schrijver and Steenbeek (1993), Odijk (1996), Nachtigall (1994), Goverde (1998a, 1999) Peeters (2003), Liebchen (2006, 2007), Kroon et al. (2007, 2009) and Schöbel and Kratz (2009).

In PESP, an *event activity graph* $G(V,E)$ consisting of the event set V and the activity set or edge set E , is set up. Events are arrival and departure times of trains at stations and represent the vertices of the graph G while edges are train activities (ride, dwell or turn-around) or passenger activities (transfer) or just represent separation constraints

between events (headway-times). For a timetable with cyclic period T and model time resolution δ , the general vertex and edge constraints over the graph $G(V, E)$ as adapted from [Sparing et al. \(2013\)](#) are

$$\begin{aligned} \forall (i, j) \in E : \quad & w_{i,j} = x_j - x_i + z_{i,j} \cdot T, \\ \forall i \in V : \quad & 0 \leq x_i \leq T - \delta, \\ \forall (i, j) \in E : \quad & 0 \leq w_{i,j} \leq T - \delta, \end{aligned} \tag{1}$$

where x_i, x_j are cyclic begin and end times - to be considered as the minutes after the hour if T equals one hour - $w_{i,j}$ is the duration chosen for the activity (i,j) , $z_{i,j}$ is an integer, either 0 or 1, and when it is 1, it represents an overflow over the period T in the cyclic timetabling model. Each activity duration $w_{i,j}$ is subject to a lower bound l_i and an upper bound u_i .

Our model deviates from this traditional approach in a number of ways. Firstly, we do not constrain our train departure and arrival times (x_i in equation (1)) to be between 0 and $T - \delta$, but rather between a start hour (e.g.: 6am) and end time (e.g.: 11am). This has the direct advantage that for the planning of all ride and dwell activities, no single integer variable is necessary. Integer variables are what makes large MILP models hard to solve. But as a consequence, secondly, for inter-train edges, like transfers, turn-around and headway edges, we will need integer variables and the ranges of these variables will often even be larger than the $(0, 1)$ range of the $z_{i,j}$ variables here. However, we take measures to restrict these ranges. Thirdly, our variables corresponding to supplements on activities are all continuous, while many other publications describe PESP models where those supplements are integer variables ([Schrijver and Steenbeek, 1993](#); [Nachtigall, 1996](#); [Liebchen, 2006](#); [Peeters, 2003](#); [Großmann, 2011](#)). Often these variables have a resolution of 1 minute. The motivation for using integer variables is probably that the published timetable is traditionally required by the railway companies to have integer minutes. Note that the models formulated as a Satisfiability (SAT) problem ([Großmann, 2011](#); [Kümmling et al., 2013](#)) are essentially always integer variable models. We take the approach of calculating a timetable in the highest time resolution possible for internal use. For publication purposes to passengers, train arrival times can be rounded up to the minute level and train departure times can be rounded down to the minute level. The fourth difference of our PESP model compared to most other PESP models is that we restrict any supplement s only by $s < T$ while previous PESP models almost always use a constraint $s < x$, where x represents some small and relatively arbitrary amount of minutes like 2 to 10 or so. We can do without these arbitrary upper bounds thanks to our objective function which prevents very large supplements. This avoids any risk of creating infeasibilities due to upper bounds that are too tight. We give a proof of this in [Appendix A](#). Lastly, we also include all of the PESP constraints included in the traditional PESP models, except for the frequency edges and associated regularity constraints that spread alternative trains evenly over the timetable period. So, equally spreading of alternative trains is not considered in our model yet. We explain in section 7.4 why we prefer not to add these regularity constraints. However, we do *evaluate* the amount of temporal spreading of alternative trains in the original and generated timetable and compare these.

[Serafini and Ukovich \(1989\)](#) wrote the seminal paper on the PESP model defining the problem as the scheduling of periodically recurring events, which leads to cyclic time-window constraints. By their disjunctive nature - due to the two possible values of $z_{i,j}$ in equation (1) - the constraints may cause the exponential growth of the computational complexity of the problem, depending on its size. [Schrijver and Steenbeek \(1993\)](#) were the first to apply PESP to the Dutch railway system. They used a constraint programming approach without an objective function. Their efforts resulted in the DONS and CADANS systems which today are still used at the Dutch principal passenger train operator company, NS (Nederlandse Spoorwegen). [Odiijk \(1996\)](#) first described the generation of constraints based on cycles in the PESP graph, which can lower solver times. The helpfulness depends on the selection of the cycle set ([Odiijk, 1996](#); [Peeters, 2003](#)). With his model, [Odiijk \(1996\)](#) demonstrated that PESP allows to generate timetables for relatively large networks, like The Netherlands.

[Goverde \(1998a, 1999\)](#) studied scheduling of optimal connections between trains, making the trade-off between the effects of too little and too much planned time for a transfer under the occurrence of delays. [Goverde \(1999\)](#) defines timetabling as the optimisation problem with decision variables being buffer times located on edges, rather than decision variables being train arrival and departure times, located on the vertices. This reduces the number of model variables. We define decision variables on both edges and vertices. Of course, we add constraints where these variables are related. Defining vertex variables allows a user to impose absolute times, for example for an international train arriving or departing at boundaries of the network considered.

Peeters (2003) discusses many possibilities of cycle bases in what is called the *Cycle Periodicity Formulation* (CPF) of PESP. Up to now, none is proven to be the optimal set to reduce solver times and some cycle bases also take quite some time to be computed. We define and quickly calculate our own cycle sets. One cycle set is defined by the transfer activities where sufficient transfer passengers are present. Other sets are selected because their cycles have few edges and so induce simple constraints between only a few integer variables. We select the combination of cycle sets that reduces solver time the most for our experiments. More details are discussed in section 5.

Minimisation of dwell times has been used as the objective function by Lie (2006). He reports that some solutions have ride activity supplements that are very large. To avoid this and similar problems, our objective function contains expected passenger time terms corresponding to all activities that cause it: ride, dwell and transfer. Realising that a timetable needs to be robust against primary delays, an objective function based on stochastic delay distributions was investigated by Kroon et al. (2007). Improvement of the service for passengers has been reported by Vansteenwegen and Van Oudheusden (2006) and the minimisation of waiting time using linear programming has been studied by Vansteenwegen and Van Oudheusden (2007).

Liebchen (2007) produced the first cyclic timetable generated by optimisation that has also been put into practice. It concerns the Berlin Underground with 37 trains. Liebchen (2007) reports the snag of very large supplements for transfers that are not considered in his model, and so, do not occur in the objective function. However, these transfers are taken by some transferring passengers and their expected transfer time may be too high. To avoid this problem, the expected duration of all transfers should be present in the objective function, weighted by the number of people expected to take them. In our approach, we define a *potential transfer* whenever two trains stop in the same station, irrespective of their still unknown arrival and departure time and irrespective of the number of passengers we expect to take this transfer. The expected passenger travel time of all potential transfers is included in our objective function.

Sparing et al. (2013) report that 'The complete optimisation of a railway timetable of realistic size is an extraordinarily large mathematical problem. Some instances or similar subproblems of the timetable optimisation have been successfully solved before.' Here, the word subproblem means only considering a subset of trains of a network, or/and only considering a subspace of the full search space. An example of the latter is due to Kroon et al. (2009) who reported on a tool set that generated 20 different feasible solutions for the PESP problem for all Dutch periodic passenger trains. These solutions were feasible, not optimal, since no objective function was defined. Their system can find feasible solutions if they exist under the given initial parameters, indicating a subspace of the full search space. If no solutions exist, the system lists the critical constraints that cannot be met. Since, for a user, it is hard to be confronted with constraints and even harder to impossible to fix infeasibility among them, we want to build a system that guarantees to deliver a solution. Our objective function will automatically, but in a *soft way*, limit the length of the time supplements added to the timetable. Therefore, unlike most other research, we do not need any hard constraints that formulate hard upper bounds on the supplements to constrain the search space. We use large ranges for supplements which, unlike other research, makes that our model covers the whole solution space. Because of this, we see no reason why our model, for some problem instance and if enough solver time is allowed, would not find a solution if one exists. Practice confirms this, since for all our experiments in the last years, no single model was reported to be infeasible by the solver. This is a huge advantage in practice.

For more information on the above models we refer to the publications themselves and to Cacchiani and Toth (2012) for a comparison of the different constraints and objective functions between models in a common notation.

1.4. Other Macroscopic Approaches

Huisman et al. (2005) and Caimi et al. (2009) categorise the common approaches of train timetabling into two streams, the first one being a macroscopic approach where one abstracts from the detailed track topology and only determines train arrival and departure time per station. Within this stream, the mentioned Periodic Event Scheduling Problem (PESP) is the most common, but there are other macroscopic approaches. For the timetables produced by macroscopic models, naturally, feasibility on the microscopic level is not guaranteed. Microscopic models solve this issue, but typically cannot yet produce a timetable for a whole country. In this section we discuss some of these other macroscopic approaches and in the next section microscopic approaches are discussed.

Nachtigall (1996), early on, constructed an approach to tackle systems with vehicles that are to be planned with different frequencies. Galli and Stiller (2010) also propose a method for such systems, claiming that traditional PESP methods are underperforming for these. Nachtigall and Voget (1997) introduced an objective function equal to the

minimisation of waiting times and obtained schedules with a genetic algorithm. A different approach to macroscopic timetabling other than PESP is taken in [Nachtigall and Opitz \(2008\)](#), by using modulo simplex calculations. This publication presents an example application for 92 trains in Germany with periods of 20, 30, 60 and 120 minutes and considering 570 of the 1200 transfers that was solved in 20 minutes. However, in this paper, we show that it is paramount to evaluate over *all* transfers to know the real value of a timetable.

[Burdett and Kozan \(2010\)](#) use a job shop scheduling inspired technique to successfully schedule trains on sections and crossing loops. Their examples go up to the size of 55 sections, 52 crossing loops and 54 trains and calculation times stay below 8 minutes. [Liu and Kozan \(2011\)](#) extend this work to also allow scheduling for a mix of high- and low-priority trains, where high-priority trains are required to run without waiting. The method of this paper achieves a similar goal without having to resort to different train categories, by weighing time supplements by the number of passengers that experience it. [Großmann \(2011\)](#) proved that PESP can be polynomially reduced to the satisfiability problem (SAT). [Großmann et al. \(2012\)](#) and [Kümmling et al. \(2013\)](#) then used a satisfiability (SAT) solver instead of an integer linear programming (ILP) solver to solve PESP models, with satisfying results for large networks, like the high speed train network in Germany.

1.5. Microscopic and Decomposing Approaches

The microscopic stream of approaches concentrates on checking ([Bourachot, 1986](#); [Carey and Carville, 2003](#)) or constructing ([Zwaneveld et al., 1996](#); [Caimi et al., 2004, 2011](#)) microscopically feasible timetables. This means that for the occupancy of subsequent tracks for each train, a blocking time stairway is constructed and overlaps between any stairways are checked or are guaranteed to not occur. At this level, signals as well as train properties like, length, available acceleration and deceleration and weight are also considered in the timing calculations. Likewise, track properties like length and curvature are taken into account. Before putting a timetable into practice, the property of *conflict freeness* should be checked at this level.

One can suppose that macroscopic timetabling has been performed and then use the generated train arrival and departure times in a station to generate a platforming and routing plan for as many trains as possible without changing these times. Fixing these times, has the advantage that the macroscopic problem of timetabling and the microscopic problem of platforming and routing in stations can be decoupled ([Sels et al., 2014](#)). However, sometimes this can result in some trains not being able to be platformed or routed while in some cases this would be possible if these times would also be allowed to change. The microscopic models of [Caprara et al. \(2011a\)](#) and [Caimi et al. \(2011\)](#) also allow some limited shifts in arrival and departure times. When even within these allowed time shift windows, some trains cannot be scheduled, they hint at the importance of providing feedback to the macroscopic level that describes that arrival or departure times of these trains should be changed.

[De Fabris et al. \(2013\)](#) use a heuristic and model at the mesoscopic level to obtain timetables that are quickly computed, yet quite accurate. They do not check the resulting timetable for microscopic feasibility yet. [Schlechte et al. \(2011\)](#) construct a microscopic graph starting from routes in a station and then *aggregate* this graph into a macroscopic graph. They present an example of the Simplon pass train corridor which, in one micro-to-macro iteration happens to deliver a microscopic and macroscopically feasible timetable. Based on algorithms published in [Cacchiani et al. \(2010\)](#) and earlier work in [Bešinović et al. \(2013\)](#), [Bešinović et al. \(2015\)](#) extend the work of [Schlechte et al. \(2011\)](#) by also providing a *de-aggregation* algorithm from macro to micro. They also iterate between micro-level (obtaining a microscopically feasible timetable) and macro-level (obtaining a timetable satisfying a macroscopic objective function). They obtain a timetable which has some degree of robustness via the requirement that capacity consumption does not exceed the norms dictated by the UIC 406 leaflet ([UIC, 2004](#)). When this normative capacity consumption is exceeded, some trains are automatically cancelled. The practical value of the approach is demonstrated on a case study for the corridor Utrecht-Eindhoven in The Netherlands which contains 7 stations and 40 hourly trains pass or stop in these stations. A microscopically feasible and macroscopically robust timetable was reached in 20 minutes in 9 micro-macro iterations. No trains had to be cancelled since maximum capacity consumption in this corridor of 54.7% is below the 75% recommended by UIC. Compared to our approach in this paper, the method of [Bešinović et al. \(2015\)](#) has the advantage that accurate modelling of microscopic details is present which achieves microscopic feasibility. Our method has the advantage that we obtain a timetable for a whole country with 5 times more trains. Also, in our approach to robustness, we achieve optimal robustness for passengers, whereas [Bešinović et al. \(2015\)](#) follow the more crude robustness norm of UIC 406. Operators may sometimes insist on scheduling more trains than is recommended by this norm and the operator's question is then how to schedule the full set with minimal expected

interactions between train pairs. To solve this, we think that advantages of the microscopic method of [Bešinović et al. \(2015\)](#) and our macroscopic method in this paper could possibly be combined by adopting our expected passenger time function as objective function for the macroscopic part of the model of [Bešinović et al. \(2015\)](#).

Some authors *decompose* the problem of generating a microscopically feasible timetable for a whole country into subproblems. [Caimi et al. \(2009\)](#) differentiate between *condensation zones* which are large complex stations where train frequencies are high and *compensation zones* which have simple topologies and lower train frequencies. In condensation zones, a timetable can be independently constructed from the rest of the network. In the compensation zones, timetables are *stitched together* since they judge that in in these zones, more slack can be added. [Caimi \(2009\)](#) reports how the mentioned slack in compensation zones is determined and how the separately constructed timetables of condensation and compensation zones can be integrated. Any decomposing method can of course not guarantee to generate a timetable that is fully optimal on the global level. In this paper we do not decompose our network but strive for an optimal macroscopic timetable for a complete national network at once.

1.6. Timetabling Approach and Features Comparison

We summarise the differences in approach and features of the publications mentioned in the previous sections 1.2 to 1.5 in table 1. Note that feasibility under the macro section is usually only with respect to calculated minimum runtimes and given minimum dwell times and macroscopic minimum headway times which are usually set to 3 minutes. It does not include conflict-freeness with respect to microscopically calculated minimum headways nor does it guarantee that there will not be capacity issues inside stations. What becomes obvious from this table is that research on macroscopic models report a higher number of trains than microscopic models. Also, macroscopic models tend to have more evolved objective functions than microscopic models. Microscopic models produce solutions that respect the microscopically calculated headway constraints and station capacity limits and are thus truly conflict free. Micro-macro models try to combine benefits of both approaches. All models have their specific merits and disadvantages.

1.7. Paper Overview

In section 2, prior to describing our model, we discuss our assumptions about the input data and what is variable versus fixed. Section 3 refers to our previously published objective function, passenger travel time expected in practice, and to our earlier work on deriving passenger flows. These flows are required since we want to minimise passenger travel time. In section 4, we describe the mandatory constraints of our PESP Mixed Integer Programming Problem (MILP) model and in section 5 we present our cycle set and related constraints which decrease solver time. Section 6 summarises the model in a few tables. In section 7, we show that our constraints avoid infeasibility issues and that our MILP model, applied to all passenger trains in Belgium, results in a significantly better timetable for passengers, after only two hours of solver time. Section 9 concludes and mentions possible further work.

2. Assumptions

Before presenting our model, this section explains the general assumptions made in this paper. The distribution of the passenger demand in space is supposed to be known and available in an Origin-Destination (OD) matrix. This matrix is derived from seasonal ticket sales. Since this represents around 80% of all trips, this data can be considered as average demand. This also means that it is not representing any specific peak hour but rather the peak hours together as a whole.

As in many European countries (Austria, Denmark, Germany, Great-Britain, Norway, Switzerland, The Netherlands) ([Peeters, 2003](#)), the Belgian railway operator requires a cyclic timetable with a period of one hour. Larger countries typically also run some longer distance trains with lower frequencies. A reason sometimes given for cyclic timetabling, as in [Peeters \(2003\)](#) and [Cacchiani and Toth \(2012\)](#), is that cyclic timetables are constructed so as to make it easy for passengers to remember the train departure times. A stronger motivation is that a cyclic timetable is more compact and also simplifies the material handling and crew rostering planning problems. A-cyclic timetables are described by [Ford and Haydock \(1992\)](#) and are also called market-led timetables because these are more adapted

Table 1: Timetabling Approach and Features Comparison. n/r= nominal or robust, res. = resolution, CP = Constraint Programming, IP = Integer Programming, ILP = Integer Linear Programming, SAT = satisfiability. cond. & comp. = condensation and compensation. heur. = heuristic, mc flow = multi commodity flow. JSP = Job Shop Problem. sc = sections, cont. = continuous, disc. = discrete, NS = Nederlandse Spoorwegen (= Dutch Railways), NE = The Netherlands, BE = Belgium. GE = Germany.

publication	approach	time res.	n/r	objective (function)	#trains #stations #lines	tool name@ company
macro timetabling						
Serafini and Ukovich (1989)	PESP	1'	n	feasibility		
Schrijver and Steenbeek (1993)	PESP, CP	1'	n	feasibility		DONS&CAD- ANS@NS, NL
Nachtigall (1996)	multiple freq.	1'	n	feasibility	40l	
Nachtigall and Voget (1997)	genetic algorithm	cont.	r	min(wait t., cost)	28t,42s	
Peeters (2003)	PESP, IP	1'	n,r	various	37t,74s	
Semet and Schoenauer (2005)	evolutionary algorithm		r	min. total delay, resilience		
Lie (2006)				min. dwell time		
Liebchen (2006)				min. wait time		@Berlin Un- derground, GE
Liebchen (2007)	PESP, ILP	1'	n	some transfers	37t	
Kroon et al. (2007)	stochastic optimisation	cont.	r	robustness improvement	56t 40s	
Nachtigall and Opitz (2008)	modulo simplex calculations	1'	n	min. weighted sum of slack	92t ≥104s	
Kroon et al. (2009)	PESP, CP& ILP	1'	n	feasibility & improved transfers		DONS&CAD- ANS@NS, NL
Schöbel and Kratz (2009) (acyclic)	PESP pareto opt.		r	eff. & robust.		
Fischetti and Monaci (2009)	robust optim. & simplified stoch. programming		r	light robustness	1l	
Cacchiani et al. (2010) (acyclic)	Lagrangian heur.		r	eff. & robust.	1000t,54s	
Galli and Stiller (2010)	sharp trees		n	feasibility		
Großmann et al. (2012)	SAT, binary ..	1'	n	min. weighted sum of slack	1336t, 3027s	TAKT @DB Netz AG, GE
Kümmling et al. (2013)	.. search heuristic					
Sparing et al. (2013)	PESP ILP	cont.	r	min. cycle time (feas. & stab.)		
This paper: Sels et al.	PESP ILP	cont.	r	min. expected total passenger journey time	203t 847s	RhinoCeros @Infrabel, BE
micro timetabling: includes platforming and routing						
Odiijk (1996)	PESP Cut Generation	1'	n	feasibility	18t,1s	
Zwaneveld et al. (1996)	branch & cut heuristic	1'	n	feasibility	18t,1s	STATIONS @NS, NL
Caimi et al. (2004)	fixed point iter. heur. local search optim.		n r	feasibility delay tolerance	19t 1s	
Caimi et al. (2009) & Caimi (2009)	decomposition in condensation & comp. zones	disc.	r	max. planned train movements	32t 1s	
Burdett and Kozan (2010)	JSP, tabu search	cont.	n	feasibility	54t,	
Caprara et al. (2011a)	Node Packing ILP	disc.	r	max. trains planned	237t 1s	
Caimi et al. (2011)	time exp. graph mc flow, ILP	disc.	r	max. planned train movements	67t 1s	
Liu and Kozan (2011)	JSP, ILP	cont.	n	min. makespan	37t,43sc	
De Fabris et al. (2013) (meso)	heur., mc flow			meso feas.	,120s	
Sels et al. (2014) (acyclic and cyclic)	ILP, (fixed arr. & dep. times)	cont.	n	max # trains platformed	1028t, 1s	Leopard @Infrabel, BE
micro-macro timetabling: includes platforming and routing						
Schlechte et al. (2011) (acyclic)	aggregation	6s..30s	n	max. planned trains	196t,18s	
Bešinović et al. (2013)	aggregation & de-aggregation	5s	r	feas. & robust	40t,26s	

to variation in passenger demand between different hours or days. In cyclic timetables, this flow variation is typically accommodated for by adding some morning peak specific and evening peak specific trains that are essentially each others opposites. This is also our approach in this paper. The line network, representing the lines along which trains services will be run, also called *train relations*, is fixed by the Belgian railway operator and we considered it given. It is not altered in our optimisation, nor is the halting pattern of each train relation. We currently do not impose temporal spreading of alternative trains. These constraints are usually enforced to limit the inter-departure times and reduce the waiting times in the station for some categories of departing passengers. We prefer to impose temporal spreading of alternative trains in a softer way via the objective function in future research.

Minimal running times are calculated by the infrastructure manager based on the relevant input parameters: train material (performance of locomotive(s), load of car(s)), track parameters (slope, curvature) and this for the worst case value of all parameters, giving the longest time amongst all parameter values. To not underestimate the minimum, Infrabel adds an additional 5% to this longest time. This result is used as ride minimum time. The minimum dwell time per stop of each train is also specified by the operator. We assume a minimum of 3 minutes for all transfers. This is a parameter that can be further tuned to stations when walking times between station platform tracks become available. The minima required for ride, dwell, transfer, headway and turn-around activities are given as multiples of 6 seconds. All supplements in our timetable for all these activities are modelled as continuous variables. However, due to all minima being specified as multiples of 6 seconds, and the presence of cycles in the timetable graph, many supplements also take values of multiples of 6 seconds.

When a unit of rolling stock is serving a relation in one direction and also the relation in the opposite direction, a turn-around edge is added to our graph, creating a minimum turn-around time requirement. When this is not the case, we suppose that another rolling stock unit can serve the opposite relation, but our model does not decide if this is possible. The minimal turn-around time in our optimisations was set to 4 minutes. This is the default initial minimum value which Infrabel assumes for normal turn-around operations. Some trains, train drivers or train operator companies may demand more turn-time and for these cases this parameter can be changed easily in our model. Decisions on which train-sets serve which relations in both directions are taken by the train operator company and this information is typically not fully available to the infrastructure company. Full vehicle allocation is performed by the train operator after timetabling. The cases where the same train set serves both directions of a relation, are known to the operator and not the infrastructure company. For these cases, a minimal turn-around time chosen by the operator and not known to the infrastructure company is used. When this data is available, our model could be run again with this complete information, but this time by the train operator company. There is no reason to expect that the performance of our model would then significantly degrade.

We do not make a distinction between intermediate, transfer and end stations. Our model is general enough to treat any station in the same way and models all transfers in every station, whenever they can occur, meaning whenever two trains stop in the same station, irrespective of their - yet unknown - arrival and departure times.

A train can experience a knock-on delay (secondary delay) caused by a delay of the train in front of it. To be able to estimate these secondary delays between trains, we need an estimate of the primary delays as a known input to our system. Measured total delays are the sum of the primary and the secondary delays. Primary delays are timetable independent, called exogeneous, while secondary delays are timetable dependent (Sun et al., 2014), called endogeneous. However, since we currently lack the data in Belgium to separate the measured total delays into primary and secondary delays, we have to resort to another method of estimating primary delays. We currently suppose primary delay distributions of each separate activity (ride, dwell, transfer) to be independent and to have an average proportional ($a = 2\%$) to its activity minimum time.

3. Optimality: Model Objective Function

Our approach to optimise a timetable for passengers was first described in Sels et al. (2011b). It consists of the two steps we call *reflowing* and *retiming*. In the reflowing step we determine the number of passengers in each train on each part of the network (Sels et al., 2011b). This information is used in the objective function for the retiming step, also known as *timetabling*.

We will only give a qualitative description of our objective function, as the main focus of this paper is the derivation of the PESP constraints of our model. As derived formally in detail in Sels et al. (2013b) and Sels et al. (2013a), our

objective function consists of the sum of the expected passenger time for each edge in the event activity graph $G(V, E)$ that corresponds to a passenger activity. So, for each ride, dwell and transfer edge we model an expected passenger time. We express this expected passenger time of an edge as a function of its minimum time and its added *supplement time*. The *shape* of this function mainly depends on the expected primary delay distribution and consequently, so does the supplement that should be ideally added. The *scale* of this function depends on the number of passengers involved. This indicates the relative importance of the expected passenger time of one edge compared to that of another and these are balanced by the objective function.

As an example, the expected transfer cost function of a *transfer* supplement time, is a U-shaped function, with a typical minimum value around 3 to 6 minutes, depending on primary delays. This is the case because for this range of supplements, the trade-off of expected transfer time for all cases of a succesful versus all cases with a missed transfer is minimal. Probabilities of these cases are calculated from the expected primary delays and are correctly taken into account in this trade-off. The penalty for missing a transfer is the time it takes passengers to wait for the alighting train one hour later.

For the primary delays, we assume negative exponential distributions which is motivated by [Goverde \(1998b\)](#) and [Yuan \(2006\)](#) and is also common practice ([Vansteenwegen and Van Oudheusden, 2006](#); [Kroon et al., 2006](#)). These distributions have an average (=expected value) that can be set to a certain fixed percentage a of the minimum time for that activity. This average can in theory be determined by inspecting logs of trains as they are running in the current timetable. This has been described by [Goverde and Hansen \(2000\)](#) and [Daamen et al. \(2009\)](#) for the Dutch and by [Labermeier \(2013\)](#) for the Swiss infrastructure. For now, we assume the same value of a for all ride, dwell and transfer edges, for all trains and for all tracks. The value of a is typically chosen in the range of 1% to 5% ([Goverde, 1998a](#)).

As for secondary delays, or *knock-on delays*, our model already contains the graph edges associated to these. Indeed, they are the same edges as the headway edges, temporally separating pairs of trains that use the same infrastructure resource. So for each headway edge, we also add a term in the objective function that represents the knock-on time or secondary delay that passengers on the second train may experience in case the first train is delayed. In our model, as derived in [Sels et al. \(2013a\)](#), this time depends on the delay distributions of both trains and on the number of passengers on the second train. Obviously, the total knock-on time is proportional to the number of passengers on the second train. Also, the expected knock-on passenger time forms a decreasing function of the train separating supplement $s_{i,j}$, since the higher the separation, between two trains i and j , the lower the expected knock-on delay. Figure 1 shows an example of a knock-on delay cost function. The horizontal axis shows the supplement between 0 and 57 minutes and on the vertical axis the green curve is the actual cost function. The red piecewise linear approximation of it is used in our MILP model.

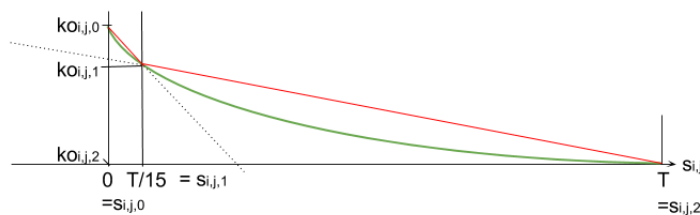


Figure 1: Knock-On Delay Cost Function

This concludes our qualitative discussion of the objective function of our PESP MILP model representing the timetabling problem. In the next sections, we derive all constraints for this model.

4. Feasibility: Mandatory Model Constraints

In section 4.1, we define our graph of train and passenger activities and then introduce some notation in section 4.2. This allows us to derive all constraints in sections 4.3, 4.4 and 5. Note that all notation is summarised in table 2 in section 6 for quick reference.

4.1. Definition of Graph, Vertices and Edge Types

We consider train and passenger activities. We set up a directed graph $G(V, E)$ with vertex set V and edge set E . Every vertex represents an arrival or departure of a train in the timetable. Every edge is an activity of a train or/and of passengers. Passenger activities are ride, dwell and transfer. Train activities without passengers are of one type only: turn-around. The respective edge sets are called E_r, E_d, E_{tr} and E_{ta} . We also introduce the set E_{hw} of headway-edges. Such an edge always contains a headway constraint, indicating that a pair of trains should be separated by a minimum time. Whenever a headway edge is modelled, we also model a knock-on delay on the same edge, which implies an expected passenger knock-on delay term in our objective function. Primary activities are activities that are related to only one train. We call the set of primary activity edges $E' = E_r \cup E_d$. Secondary activities are activities that occur between two trains. We call $E'' = E_{tr} \cup E_{hw} \cup E_{ta}$. It holds that $E = E' \cup E''$.

4.2. Function Notation

In the following sections, we will need the following notation. The *type* of an edge is either ride, dwell, transfer, headway (=knock-on) or turn-around. We use \neg to mean boolean negation. We also use the following functions on a graph $G(V, E)$:

$i : E \rightarrow V$ so that $i(e)$ equals the unique 'in' vertex of edge e (source) and
 $o : E \rightarrow V$ so that $o(e)$ equals the unique 'out' vertex of edge e (destination),
 $- : E \rightarrow E$ so that $-e$ equals the unique inverse edge of edge e , $-e$ is defined by $o(-e) = i(e) \wedge i(-e) = o(e)$.
 $\forall t \in \{r, d\} : i_t : V \rightarrow E$ so that $i_t(v)$ equals the unique type- t -edge that ends in vertex v and
 $\forall t \in \{r, d\} : o_t : V \rightarrow E$ so that $o_t(v)$ equals the unique type- t -edge that starts in vertex v ,
 $\forall t \in \{r, d\} : i_t : E \rightarrow E$ so that $i_t(e)$ equals the unique type- t -edge that precedes edge e and
 $\forall t \in \{r, d\} : o_t : E \rightarrow E$ so that $o_t(e)$ equals the unique type- t -edge that succeeds edge e .

Note that the word *unique* here, means that no two edges can satisfy the mentioned property and so at most one result can be returned by each of these functions. It is possible that no single edge satisfies the properties of the function definition. For example, $i_r(v_0)$ where v_0 corresponds to any vertex in the station where a train starts is not defined. The set of results of the function is then the empty set. So, for example, in a constraint generation context, no constraint will be generated for the occurrence $i_r(v_0)$. Also remark that, we use the same function name i_t both from the domain V and from the domain E . From the type of the argument to it, it will be clear which function is meant. The same holds for o_t . Lastly, note that for $t \in \{tr, ta, ko\}$, so for secondary edges, i_t and o_t are not defined. Indeed multiple 'in' or 'out' edges could then typically result.

$\bar{\cdot} : E_{hw} \rightarrow E_{hw}$ so that \bar{e} equals the unique *opposite* headway edge from train 2 to 1 when e goes from train 1 to 2.

For two trains riding in the same direction on an open track, a headway edge goes from the end of a ride activity of train 1 to the end of a ride activity of train 2 or goes from the beginning of a ride activity of train 1 to the beginning of a ride activity of train 2. So, for these headway edges e we can define its opposite headway edge \bar{e} by its vertices as $\forall e \in E_{hw} : i(\bar{e}) = o(e) \wedge o(\bar{e}) = i(e)$.

For two trains riding in the opposite direction on a single track, firstly, a headway edge goes from the end of a train 1 ride activity to the beginning of a train 2 ride activity. So, for these headway edges e we can define its opposite headway edge \bar{e} by its vertices as $\forall e \in E_{hw} : i(\bar{e}) = o(o_r(e)) \wedge o(\bar{e}) = i(i_r(e))$. Note that \bar{e} , just like e , will go from an end of ride vertex to a beginning of ride vertex.

For two trains riding in the opposite direction on a single track, secondly, due to the cyclical nature of our schedule, we also need to add headway edges between the beginning of a ride edge of train 2 to the end of the ride edge of (actually the cyclically next occurrence of) train 1. Definition of opposite edges is slightly more involved: $\forall e \in E_{hw} : i(\bar{e}) = i(i_r(i(e))) \wedge o(\bar{e}) = o(o_r(i(e)))$.

Note that for all headway edges, we require a headway of 3 minutes, both for same direction and opposite direction trains. These 3 minutes are a usual approximation of what is considered safe in many countries. If long trains operate at lower speed in interlocking areas and the set-up of routes is done manually, more headway time may be required. Considering these cases requires more microscopic data than directly available to us. In the future, we hope to extend our model with station and train-couple specific minimum headway times. We do not directly consider extra time corresponding to the passing time for the length of a train in our macroscopic timetabling. We suppose that these times are small since trains are riding fast outside stations. If this is a concern, the 3 minutes minimal headway time can be increased.

$\bar{\cdot} : E_{tr} \rightarrow E_{tr}$ so that \bar{e} equals the unique opposite transfer edge from train 2 to 1 when e goes from train 1 to 2, in the same station. A transfer edge however, goes from the end of a ride activity of train 1 to the beginning of the ride activity of train 2. So, for a transfer edge e we must define its opposite transfer edge \bar{e} by its vertices as $\forall e \in E_{tr} : i(\bar{e}) = i(i_d(o(e))) \wedge o(\bar{e}) = o(o_d(i(e)))$.

4.3. Mandatory Constraints

The following constraints are mandatory, meaning each constraint has to be part of our model to guarantee that each solution is a valid timetable.

4.3.1. Edge Constraints (Intra-Edge Constraints)

Since each edge $e' \in E'$ represents an activity, and each secondary edge $e'' \in E''$ a time difference with an enforced minimum, all $e \in E = E' \cup E''$ have a begin time b_e , minimum duration time m_e , supplement time s_e and end time e_e . So we add the obvious constraints to our model:

$$\forall e \in E : b_e + m_e + s_e = e_e \quad (2)$$

In our model, all b_e and s_e are variables that have to be determined by the solver and all e_e are expressions that are derived by equations (2) by the solver. The m_e are constants that are fixed for our model. For a ride edge, m_e is the minimum ride time that has been typically calculated by the infrastructure management company, taking into account train and infrastructure properties. For a dwell edge, m_e is the minimum dwell time that has typically been decided by the operator, considering minimal necessary times for passengers to embark on or alight from a train. For a transfer edge, we currently set m_e to three minutes, as we consider this as generally sufficient for the walking duration of passengers between the arrival time of their first train and the departure time of their second train. Note that the solver for our model, certainly for transfers taken by many passengers, will typically add some minutes for the supplement s_e to make the passenger transfer robust against transfer delays. For a turn around edge, we take m_e equal to four minutes as this is the preference of Infrabel's main operator, NMBS. For a knock on edge, m_e is set to three minutes as this is considered minimal but safe by Infrabel.

4.3.2. Node Constraints (Inter-Edge Constraints)

We define T as the period of our cyclic timetable. If $i(e)$ is the in vertex of the edge e and $o(e)$ its out vertex, it then holds that

$$\begin{aligned} \forall_{o(e_0)=i(e_1)}(e_0, e_1) \in E' \times (E' \cup E'') : e_{e_0} = b_{e_1} \\ \forall_{o(e_0)=i(e_1)}(e_0, e_1) \in E'' \times E' : e_{e_0} + d_{e_0} \cdot T = b_{e_1}. \end{aligned} \quad (3)$$

where, d_{e_0} is an integer decision variable, defined for secondary edges $e_0 \in E''$ only. Consider a transfer edge e . The idea is that, if the equations (3) that contain a variable d_e hold, then there is a valid transfer between *some* train instance of hourly train series t_1 and *some* instance of hourly train service t_2 . Since both train services occur every hour, it follows that there is also a valid transfer from *every* train instance from t_1 to *every* train instance from t_2 , with the same values for m_e and s_e . The same reasoning holds for all other secondary edge types. So the exact integer value of d_e does not matter much for feasibility. Peeters (2003) explains though, that its allowed range affects solver time. Section 4.3.4 explains our range choices for our integer variables.

The equations $\forall_{o(e_0)=i(e_1)}(e_0, e_1) \in E'' \times E'' : e_{e_0} + d_{e_0} \cdot T = b_{e_1}$ for the set-product $E'' \times E''$ are not needed, since every secondary edge $e \in E''$ in our graph, always occurs between the end and the beginning of a primary edge, so the constraints present in the constraints (3) already fix their begin and end times. Not adding the constraints for $E'' \times E''$ saved us some model setup time, and more importantly, also significant model solver time.

4.3.3. General Bounds on Continuous Variables

We define δ as the (smallest) time resolution unit of our system. For both primary and secondary supplements we allow the range $s_e \in [0, T - \max(m_e, \delta)]$. This guarantees that the length of a primary edge e being $m_e + s_e$ belongs to the interval $[m_e, T - \max(0, \delta - m_e)]$. Supplements of primary edges can be freely chosen. However secondary edges have to fit between them, so their lengths $m_e + s_e + d_e \cdot T$ are constrained by the difference between the vertices on primary edges it connects. One can see that any necessary length that can occur in our system, can be constructed by

this expression. Indeed all edge lengths in our system will be (positive or negative) multiples of δ . So, summarised, we add the bounds

$$\forall e \in E : 0 \leq s_e \leq T - \max(m_e, \delta) \quad (4)$$

and unlike tighter bounds, these will never cause infeasibilities.

4.3.4. General Bounds on Integer Variables

Each integer variable d_{e0} occurring in equations (3) represents the difference between the index i_1 of the train in train series t_1 and the index i_2 of the train in train series t_2 . We plan all trains, complete from begin time h_{lo} hours to end time h_{hi} hours. This means that the range of a difference of two arbitrary times in this interval is between $(h_{lo} - h_{hi})$ hours and $(h_{hi} - h_{lo})$ hours, so the range for any d_e on a secondary edge, which always indicates such a difference of time, is

$$\forall e \in E'' : (h_{lo} - h_{hi}) \leq d_e \leq (h_{hi} - h_{lo}). \quad (5)$$

Because currently all our selected trains are planned between 6am and 11am, we have an interval $[-5, +5]$ for each d_e . In some cases, as we will see in sections 4.4.1, 4.4.2, 4.4.3, 5.1, 5.3 and 5.4, these bounds will still be tightened.

4.3.5. All Trains Start in the First Hour

To make sure that every train is scheduled in every hour, we require that the first ride activity of each train should start in the first hour of the time interval we plan all trains in. The following constraints are added to the model

$$\forall \neg \exists i_d(e) e \in E_r : h_{lo} \cdot T \leq b_e < (h_{lo} + 1) \cdot T, \quad (6)$$

where \neg means negation and $\exists i_d(e)$ means that there exists a dwell predecessor edge to the edge e . For ride edges the whole condition only evaluates to true for the first ride edge of each train service.

4.3.6. All Passing Supplements are Zero

We define the function $s : E_d \rightarrow \{true, false\}$ so that $\forall e \in E_d : s(e)$ is true if dwell edge e represents a stopping activity and false if it represents a passing activity. In case a train passes a station, the minimum dwell time m_e is zero and we also set the corresponding dwell supplement to zero:

$$\forall \neg s(e) e \in E_d : s_e = 0. \quad (7)$$

Note that for any dwell activity that corresponds to a train that does not stop, no transfers to or from this train can exist. So we also do not define transfers for such dwell activities. For each pair of trains that stop in a station, we define transfers between them in both directions.

4.4. Odijks Rule for Integers Variables of Cycle Edges: Feasibility and Bounds

Before we go on we need to present two equations, derived by [Odijk \(1996\)](#). These concern constraints defined on undirected cycles of our graph. In each cycle, we make a weighted sum of edge lengths, traversing all its edges in a chosen loop direction. If we follow an edge e^+ pointing *in* this loop direction ($e^+ \in c^+$), we add its length. If we follow an edge e^- pointing *against* this loop direction ($e^- \in c^-$) we subtract its length. The result must be 0. In each undirected cycle c , it holds that

$$\begin{aligned} \forall c \in C(G) : & \left(\sum_{e \in (c^+ \cap E')} m_e + s_e \right) + \left(\sum_{e \in (c^+ \cap E'')} m_e + s_e + d_e \cdot T \right) \\ & = \left(\sum_{e \in (c^- \cap E')} m_e + s_e \right) + \left(\sum_{e \in (c^- \cap E'')} m_e + s_e + d_e \cdot T \right). \end{aligned} \quad (8)$$

Consider a graph $G = (V, E, l, u)$ of vertices V , activity-edges E , a vector l of integer lower bounds for the lengths of all edges and a vector u of integer upper bounds for the lengths of all edges. [Odijk \(1996\)](#) proved from equation (8) that a PESP instance defined by $G = (V, E, l, u)$ and the set of all cycles $C(G)$ and cyclic period T is feasible if and only if

$$\forall c \in C(G) : \forall e \in c'' : \exists d_e \in \mathbb{Z} : L_c \leq M_c \leq U_c, \text{ where } \begin{aligned} L_c &= \left\lceil \frac{1}{T} (\sum_{e \in c^+} l_e - \sum_{e \in c^-} u_e) \right\rceil, \\ M_c &= -(\sum_{e \in c^+} d_e - \sum_{e \in c^-} d_e), \\ U_c &= \left\lfloor \frac{1}{T} (\sum_{e \in c^+} u_e - \sum_{e \in c^-} l_e) \right\rfloor. \end{aligned} \quad (9)$$

Here $c'' = c \cap E''$ is the set of secondary edges of c . We will use equation (9) several times in sections 4.4.1 up to 5.5 to derive tighter bounds on integer decision variables and thereby reduce solver times. Note that Odijk actually defines a vector p where $p_e = -d_e, \forall e \in E''$. In equations (8), $+d_e$ would be replaced by $-p_e$ then as well.

Equations (8) and (9) together ensure that the total cycle duration is a multiple of T . One can see that this property is a necessary condition for any cycle in the system, if after some multiple n of T , the schedule is to be repeated. The question as to in how few multiples n of T the schedule can be repeated is not answered by the resulting values of m_e, s_e and d_e alone, but also depends on the available train resources. More specifically, for a cycle c that is carried out by r_c available trains of the same train relation, and takes time $n_c \cdot T$, the lowest possible time t_c to repeat it is $t_c = \lceil \frac{n_c}{r_c} \cdot T \rceil$. So for the whole schedule, its minimum repeat time is the smallest common multiple over all t_c over all cycles that are carried out by a single train set. If one also wants to ensure that all potential transfers between all train pairs take at most time T , one has to repeat every train service every time T by ensuring $r_c = n_c$.

4.4.1. Separate and Forbid Reordering of Same Direction Trains on the Same Open Track Section

Consider the situation shown in the left half of Figure 2. A pair of trains rides on the same open track section in the *same direction*. The headway time needs to be respected, both between the times that the trains enter and the times that they leave the resource. As described before, in our graph, for these train pairs and at the corresponding ride edges, we added headway edges both between the beginnings of these pairs of corresponding ride edges and also between the endings of these pairs of corresponding ride edges, and this in both directions. Bottom left, we see that

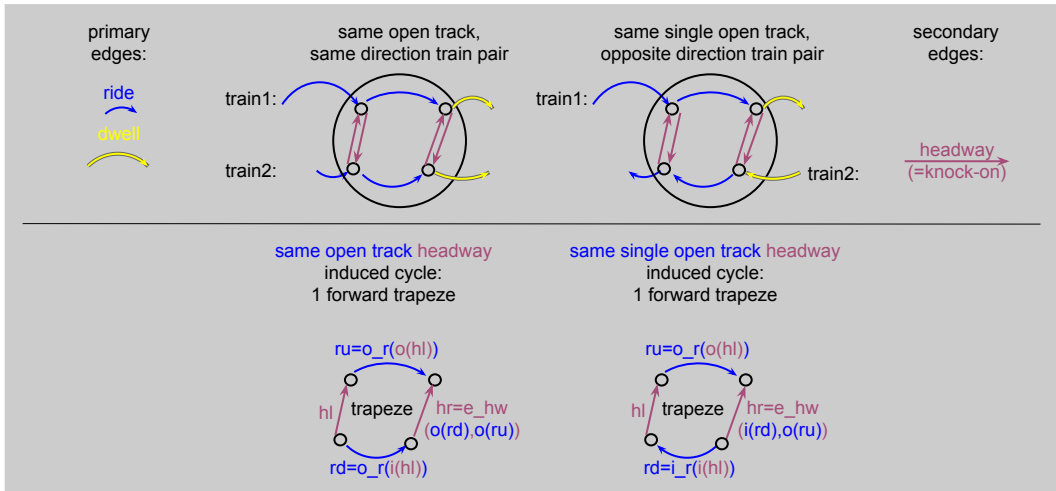


Figure 2: Mandatory Cycles for train pairs riding in same direction and in opposite direction on the same track.

a trapeze shaped cycle is formed with two upwards pointing headway edges and two ride edges. The set of all these cycles is defined by a *condition-definition-function* $cd(h_l; r_u, r_d, h_r)$. Here, h_l stands for the left headway time edge (knock-on edge), r_u stands for the upper ride edge, r_d for the down ride edge, and h_r stands for the right headway time edge. In the definition of $cd(h_l; r_u, r_d, h_r)$, we specify how we calculate the edges r_u, r_d and h_r , if they exist. Whenever we can derive all edges r_u, r_d and h_r , we will formulate a constraint over these edges. The condition under which all edges can be derived is

$$\begin{aligned}
 cd(h_l; r_u, r_d, h_r) &\equiv h_l < \overline{h_l} : \\
 &\exists r_u \in E_r : r_u = o_r(o(h_l)) : \\
 &\exists r_d \in E_r : r_d = o_r(i(h_l)) : \\
 &\exists h_r \in E_{hw} : h_r = e_{hw}(o(r_d), o(r_u))
 \end{aligned} \tag{10}$$

Any *condition-definition-function* is implemented in software as a series of nested if-statements. For example, for equation (10), 'if $(h_l < \overline{h_l})$ ' will restrict the inner body to cases where $(h_l < \overline{h_l})$ holds. Within this first if-statement,

the next statement will be 'if $(\exists r_u \in E_r : r_u = o_r(o(h_l)))$ ' and so forth. The condition $h_l < \overline{h_l}$, meaning h_l comes strictly before $\overline{h_l}$ in the ordered list of all edges we keep, is added to avoid that we add symmetric constraints, one for $(h_l, \overline{h_l})$ and one for $(\overline{h_l}, h_l)$, which we think would only deliver similar information to the solver. The expression $\exists r_u \in E_r : r_u = o_r(o(h_l))$ selects the ride edge that starts from the end of the headway time edge h_l , if it exists. It does not exist in the case that this headway edge h_l is pointing to the end instead of to the beginning of a ride edge. A similar expression is added to derive the ride edge r_d . Lastly h_r is calculated as the unique headway edge between the end points of the ride edges r_d and r_u . For the cases where $cd(h_l; r_u, r_d, h_r)$ succeeds in defining the three new edges, it defines a cycle with four edges.

As these small trapeze shaped cycles $c_{h_l, trap}$, we get $\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : c_{h_l, trap} = (h_l, r_u, -h_r, -r_d)$. We add the cycle constraints to our model:

$$\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : \underbrace{(m_{h_l} + s_{h_l} + d_{h_l} \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{r_u} + s_{r_u})}_{0 \leq \bullet < T} - \underbrace{(m_{h_r} + s_{h_r} + d_{h_r} \cdot T)}_{0 \leq \bullet < T} - \underbrace{(m_{r_d} + s_{r_d})}_{0 \leq \bullet < T} = 0. \quad (11)$$

Considering the ranges of terms marked in equation (11), applying Odijks equation (9) to these cycles, gives for the ranges of the integers: $\forall_{cd(h_l; r_u, r_d, h_r)} h \in E_{hw} : -1 = \left\lfloor \frac{0+0-(T-\delta)-(T-\delta)}{T} \right\rfloor \leq d_{h_l} - d_{h_r} \leq \left\lfloor \frac{(T-\delta)+(T-\delta)-0-0}{T} \right\rfloor = 1$. Instead of adding these bounds, we first also require that there are no reorderings between train 1 and 2. This is formally written as

$$(m_{h_l} + s_{h_l} + d_{h_l} \cdot T) \text{ div } T = (m_{h_r} + s_{h_r} + d_{h_r} \cdot T) \text{ div } T, \quad (12)$$

where 'div' means integer division. Considering the ranges in equations (11) this is equivalent with $d_{h_l} = d_{h_r}$. So, we enforce in our model:

$$\forall_{cd(h_l; r_u, r_d, h_r)} h \in E_{hw} : d_{h_l} = d_{h_r}. \quad (13)$$

4.4.2. Separate and Forbid Reordering of Opposite Direction Trains on the Same Single Open Track Section

Now, consider the situation shown in the right half of Figure 2. A pair of trains rides on the same single track section in *opposite direction*. The headway time needs to be respected, between the time that the first train leaves and the second train enters the track. In our graph, for these opposite train pairs, we added headway edges from end to beginning of the corresponding ride edges, but also from beginning to end of the ride edges to cover the cyclical train knock-on relation. Bottom right, we see that a trapeze shaped cycle is formed with two upwards pointing headway edges and two ride edges. The set of all these cycles is defined by a *condition-definition-function*,

$$\begin{aligned} cd(h_l; r_u, r_d, h_r) \equiv & h_l < \overline{h_l} : \\ & \exists r_u \in E_r : r_u = o_r(o(h_l)) : \\ & \exists r_d \in E_r : r_d = i_r(i(h_l)) : \\ & \exists h_r \in E_{hw} : h_r = e_{hw}(i(r_d), o(r_u)) \end{aligned} \quad (14)$$

and as small trapeze shaped cycles $c_{h_l, trap}$, we get $\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : c_{h_l, trap} = (h_l, r_u, -h_r, r_d)$. We arrive at the cycle equations:

$$\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : \underbrace{(m_{h_l} + s_{h_l} + d_{h_l} \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{r_u} + s_{r_u})}_{0 \leq \bullet < T} - \underbrace{(m_{h_r} + s_{h_r} + d_{h_r} \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{r_d} + s_{r_d})}_{0 \leq \bullet < T} = 0 \quad (15)$$

Considering the ranges of terms marked in equation (15), applying Odijks equation (9) to these cycles, gives for the ranges of the integers: $\forall_{cd(h_l; r_u, r_d, h_r)} h \in E_{hw} : -2 = \left\lfloor \frac{-(T-\delta)-(T-\delta)+0-(T-\delta)}{T} \right\rfloor \leq d_{h_l} - d_{h_r} \leq \left\lfloor \frac{-0-0-(T-\delta)-0}{T} \right\rfloor = 0$. Instead of adding these bounds, we first also require that there are no collisions between trains 1 and 2 riding in opposite directions on the same track. This boils down to equation (12) which effectively says that train order on the track cannot change. Imposing this and considering the ranges in equations (15) this is equivalent with $d_{h_l} = d_{h_r}$. So, we enforce in our model:

$$\forall_{cd(h_l; r_u, r_d, h_r)} h \in E_{hw} : d_{h_l} = d_{h_r}. \quad (16)$$

4.4.3. Separate and Forbid or Allow In-Station Reordering based on Infrastructure & Halting Pattern

From the previous section 4.4.1, we know how to forbid reordering of two trains on the same open track section. We can use the same system for train pairs that cannot overtake each other in a station. This is the case when none of the two trains makes a stop or also when the station doesn't have enough platform tracks to allow reordering. When one train stops and the other does not, the bounds interval of the expression $d_{h_l} - d_{h_r}$ is different. We define the function $iar : E_d \times E_d \rightarrow \{\text{true}, \text{false}\}$ so that $\forall e, e' \in E_d : iar(d)$ is true if the station infrastructure allows reordering between the dwell activities e and e' and false otherwise. Note that, by definition, $\forall e, e' \in E_d : iar(e, e') = iar(e', e)$. Currently, we only allow reordering in stations when (1) these possess at least 4 platform tracks (2 in both directions) and (2) a switch grid is present. This corresponds to the guideline currently used by the Belgian infrastructure manager. In Belgium this holds for 150 stations of the 780 stations, so 19% of them. The possibility of reordering will also depend on whether both trains, just one train or no trains stop or not.

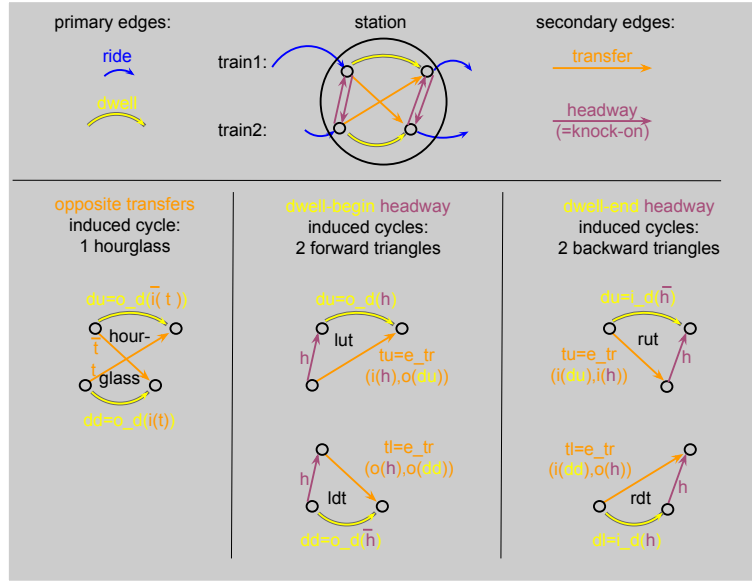


Figure 3: Optional Cycles for train pairs dwelling in a station.

Consider only the top part of Figure 3. It shows two station dwell activities and the transfer and headway edges defined by these. We derive the general bounds on the integer variables of the headway edges. The set of all the cycles containing the dwell activities of two trains in the same station is defined by another condition-definition-function

$$\begin{aligned}
 cd(h_l; d_u, d_d, h_r) &\equiv h_l < \bar{h}_l : \\
 &\exists d_u \in E_d : d_u = o_d(o(h_l)) : \\
 &\exists d_d \in E_d : d_d = o_d(i(h_l)) : \\
 &\exists h_r \in E_{hw} : h_r = e_{hw}(o(d_d), o(d_u))
 \end{aligned} \tag{17}$$

and as small cycles, we get the trapezes $\forall cd(h_l; d_u, d_d, h_r) h_l \in E_{hw} : c_{h_l, trap} = (h_l, d_u, -h_r, -d_d)$. The corresponding cycle constraints enforced in our model are

$$\forall cd(h_l; d_u, d_d, h_r) h_l \in E_{hw} : \underbrace{(m_{h_l} + s_{h_l} + d_{h_l} \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{d_u} + s_{d_u})}_{0 \leq \bullet < T} - \underbrace{(m_{h_r} + s_{h_r} + d_{h_r} \cdot T)}_{0 \leq \bullet < T} - \underbrace{(m_{d_d} + s_{d_d})}_{0 \leq \bullet < T} = 0. \tag{18}$$

Considering the ranges marked in equation (18), applying Odijks rule here gives $\forall cd(h_l; d_u, d_d, h_r) h_l \in E_{hw} : -1 = \left\lceil \frac{(0)+(0)-(T-\delta)-(T-\delta)}{T} \right\rceil \leq d_{h_l} - d_{h_r} \leq \left\lfloor \frac{(T-\delta)+(T-\delta)-(0)-(0)}{T} \right\rfloor = 1$, as the ranges of the integers. The specific cases of reordering

or not reordering now will tighten these bounds further, so we first enforce the following narrower intervals:

$$\forall_{cd(h_i; d_u, d_d, h_r)} h \in E_{hw} : \begin{array}{ll} \text{if } (\neg \text{iar}(d_u, d_d)) & : -0 \leq d_{h_i} - d_{h_r} \leq +0, \\ \text{else if } (\neg s(d_u) \wedge \neg s(d_d)) & : -0 \leq d_{h_i} - d_{h_r} \leq +0, \\ \text{else if } (s(d_u) \wedge \neg s(d_d)) & : -1 \leq d_{h_i} - d_{h_r} \leq +0, \\ \text{else if } (\neg s(d_u) \wedge s(d_d)) & : -0 \leq d_{h_i} - d_{h_r} \leq +1, \\ \text{else if } (s(d_u) \wedge s(d_d)) & : -1 \leq d_{h_i} - d_{h_r} \leq +1. \end{array} \quad (19)$$

We believe that equations (19) represent the tightest bounds on the integer d -variables that can be calculated. Since some other publications do not explicitly mention the ranges of integer variables in their models, comparison with other work is hard. This completes our derivation of all mandatory constraints that guarantee that our model will produce only valid timetables.

5. Computation Speed: Optional Model Constraints

In this section, we derive optional constraints which are only enforced to reduce solver times. Apart from the feasibility criterion in equation (9) mentioned above, Odijk (1996) also describes the generation of constraints based on cycles. Even though these cycle-based constraints are in fact linear combinations of the constraints (2) and (3), they can be helpful in practice to lower solver times (Peeters, 2003). Nachtigall (1994) showed that it is sufficient to impose the cycle periodicity constraints on a set of cycles which is an *integral cycle basis* of the whole cycle space. This means that all cycles in the cycle space are integer linear combinations of the cycles in this integral basis. A practical algorithm to arrive at such an integral basis, more specifically a *strictly fundamental cycle basis*, is to first construct a spanning tree of the graph and then to construct all cycles induced by the edges that do not belong to that tree. For a general graph, computing one of its spanning trees requires quite some time, but we realised that the set of primary edges of our graph is very close to a complete spanning tree of our graph and as a consequence each secondary edge can then simply be used to each induce one cycle. This avoids having to calculate a spanning tree at all. Using this insight, and after performing many experiments, we arrived at our own cycle sets, concentrating on secondary edges in our graph. Firstly, we focus on small cycles which contain only a few secondary edges. In this way few integer variables are present, so that the induced constraints are simple. Secondly, we also focus on secondary edges where many passengers are present and find a shortest cycle through these edges and generate the corresponding constraint. The combination of both types of cycle constraints, delivers a model that can be solved quickly. We describe our chosen cycle set and set of induced constraints in the following sections 5.1 through 5.5.

5.1. Opposite Transfer Pair Induced Small Cycles: Hourglasses

Consider the bottom left part of Figure 3. It shows a part of a graph where two trains stop in a station and transfers in both directions between these trains occur. Two opposite transfer edges t (from train 1 to train 2) and \bar{t} (from train 2 to train 1) are always defined between two dwell edges (of train 1 and of train 2) when these both represent a train stop. We will call d_u (dwell up) and d_d (dwell down), between their respective begin and end vertices, with which a 4-edged cycle can be formed. For both transfers t and \bar{t} , the same cycle is formed, so we restrict ourselves to the transfers t for which $t < \bar{t}$ holds, where $<$ can be any total order defined on (transfer) edges. The condition-definition-function is

$$cd(t; d_u, d_d) \equiv \forall_{t < \bar{t}} t \in E_{tr} : \begin{array}{l} d_u = o_d(i(t)), \\ d_d = o_d(i(\bar{t})), \end{array} \quad (20)$$

which in fact always succeeds in defining the edges d_u and d_d . The opposite transfer-pair induced cycles c_t are now defined as $c_t = (t, -d_u, \bar{t}, -d_d)$. The constraints corresponding with these hourglass cycles which we add to our model are

$$\forall_{cd(t; d_u, d_d)} t \in E_{tr} : \underbrace{m_t + s_t + d_t \cdot T}_{0 \leq \bullet < T} - \underbrace{(m_{d_u} + s_{d_u})}_{0 < \bullet < T} + \underbrace{m_{\bar{t}} + s_{\bar{t}} + d_{\bar{t}} \cdot T}_{0 \leq \bullet < T} - \underbrace{(m_{d_d} + s_{d_d})}_{0 < \bullet < T} = 0. \quad (21)$$

In this equation the minimum dwell times m_{d_u} and m_{d_d} must be strictly positive, since no transfers are possible on trains that do not stop. Applying Odijk's equation (9) to this *hourglass* cycle and taking into account the ranges of

terms marked in equation (21), gives $\forall_{t < \bar{t}} t \in E_{tr} : -1 = \left\lceil \frac{(0)+(0)-(T-\delta)-(T-\delta)}{T} \right\rceil \leq d_t + d_{\bar{t}} \leq \left\lfloor \frac{(T-\delta)+(T-\delta)-(0)-(0)}{T} \right\rfloor = 1$, for the ranges of the integers. Equations (21) can be rewritten as

$$\forall_{cd(t; d_u, d_d)} t \in E_{tr} : \quad + (d_t + d_{\bar{t}}) \cdot T = \underbrace{(m_{d_u} + s_{d_u})}_{0 < \bullet < T} + \underbrace{(m_{d_d} + s_{d_d})}_{0 < \bullet < T} - \underbrace{(m_t + s_t)}_{0 \leq \bullet < T} - \underbrace{(m_{\bar{t}} + s_{\bar{t}})}_{0 \leq \bullet < T}. \quad (22)$$

If we disallow dwell times to become extremely long and suppose that $(m_{d_u} + s_{d_u}) + (m_{d_d} + s_{d_d}) < T$, then according to equation (22), $d_t + d_{\bar{t}}$ can never equal 1. We could then reduce the upper bound to 0, giving

$$\forall_{t < \bar{t}} t \in E_{tr} : \quad -1 \leq d_t + d_{\bar{t}} \leq 0. \quad (23)$$

When we imposed the stricter equations (23), solver times reduced significantly. When we further reduce the bounds interval in (23) from $[-1, 0]$ to $[0, 0]$, we reduce the search space, but cannot guarantee feasibility. Even so, in practice, we still got solutions returned in all cases, but on average solver times increased. Guaranteed feasibility and lower solver times are two good reasons to keep the $[-1, 0]$ interval in (23).

5.2. Opposite Dwell-Begin-Headway Induced Small Cycles: Forward Triangles

The bottom middle part of Figure 3 also indicates the headway edges. Here, we only consider headway edges that connect the *beginnings* of two dwell edges. This condition can be formulated as the restrictions $\exists d_u \in E_d : d_u = o_d(h)$ and $\exists d_d \in E_d : d_d = o_d(\bar{h})$. We also restrict ourselves to the cases where both trains stop and transfers between them occur. The condition-definition-function becomes

$$\begin{aligned} cd(h; d_u, t_u, d_d, t_d) &\equiv h < \bar{h} : \\ &\exists d_u \in E_d : d_u = o_d(h) : \exists t_u \in E_{tr} : t_u = e_{tr}(i(h), o(d_u)) : \\ &\exists d_d \in E_d : d_d = o_d(\bar{h}) : \exists t_d \in E_{tr} : t_d = e_{tr}(o(h), o(d_d)). \end{aligned} \quad (24)$$

Altogether, this gives the definition of $c_{h,lut}$ (a left up triangle) and $c_{h,ldt}$ (a left down triangle) as defined cycles for the following headway edges h :

$$\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : \begin{cases} c_{h,lut} &= (h, d_u, -t_u) \\ c_{h,ldt} &= (h, t_d, -d_d). \end{cases} \quad (25)$$

The constraints corresponding with these forward triangle cycles are

$$\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : \begin{cases} \underbrace{(m_h + s_h + d_h \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{d_u} + s_{d_u})}_{0 \leq \bullet < T} - \underbrace{(m_{t_u} + s_{t_u} + d_{t_u} \cdot T)}_{0 \leq \bullet < T} = 0 \\ \underbrace{(m_h + s_h + d_h \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{t_d} + s_{t_d} + d_{t_d} \cdot T)}_{0 \leq \bullet < T} - \underbrace{(m_{d_d} + s_{d_d})}_{0 \leq \bullet < T} = 0. \end{cases} \quad (26)$$

Applying Odijks equation (9) to the cycles lut and ldt and taking into account the ranges of terms marked in equation (26), gives for the ranges of the integers

$$\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : \begin{cases} 0 = \left\lceil \frac{(0)+(0)-(T-\delta)}{T} \right\rceil \leq d_{t_u} - d_h \leq \left\lfloor \frac{(T-\delta)+(T-\delta)-(0)}{T} \right\rfloor = 1 \\ -1 = \left\lceil \frac{(0)-(T-\delta)-(T-\delta)}{T} \right\rceil \leq d_{t_d} + d_h \leq \left\lfloor \frac{(T-\delta)-(0)-(0)}{T} \right\rfloor = 0. \end{cases} \quad (27)$$

We add equations (26) as well as (27) as constraints to the model.

5.3. Opposite Dwell-End-Headway Induced Small Cycles: Backward Triangles

Similarly to the previous section, the bottom right part of Figure 3 now shows headway edges that connect the *beginnings* of two ride edges and we define

$$\begin{aligned} cd(h; d_u, t_u, d_d, t_d) &\equiv h < \bar{h} : \\ &\exists d_u \in E_d : d_u = i_d(\bar{h}) : \\ &\exists t_u \in E_{tr} : t_u = e_{tr}(i(d_u), i(h)) : \\ &\exists d_d \in E_d : d_d = i_d(h) : \\ &\exists t_d \in E_{tr} : t_d = e_{tr}(i(d_d), o(h)) \end{aligned} \quad (28)$$

and as small cycles, the two backward triangles

$$\forall_{cd(h;d_u,t_u,d_d,t_d)} h \in E_{hw} : \begin{cases} c_{h,rut} &= (h, -d_u, t_u) \\ c_{h,rdt} &= (h, -t_d, d_d). \end{cases}$$

The constraints imposed corresponding with these backward triangle cycles are

$$\forall_{cd(h;d_u,t_u,d_d,t_d)} h \in E_{hw} : \begin{cases} \underbrace{(m_h + s_h + d_h \cdot T)}_{0 \leq \bullet < T} - \underbrace{(m_{d_u} + s_{d_u})}_{0 \leq \bullet < T} + \underbrace{(m_{t_u} + s_{t_u} + d_{t_u} \cdot T)}_{0 \leq \bullet < T} = 0 \\ \underbrace{(m_h + s_h + d_h \cdot T)}_{0 \leq \bullet < T} - \underbrace{(m_{t_d} + s_{t_d} + d_{t_d} \cdot T)}_{0 \leq \bullet < T} + \underbrace{(m_{d_d} + s_{d_d})}_{0 \leq \bullet < T} = 0. \end{cases} \quad (29)$$

Considering the ranges of terms marked in equation (29), applying Odijks equation to these cycles *rut* and *rdt*, gives as imposed ranges of the integers

$$\forall_{cd(h;d_u,t_u,d_d,t_d)} h \in E_{hw} : \begin{cases} -1 = \left\lceil \frac{(0)-(T-\delta)-(T-\delta)}{T} \right\rceil \leq d_{t_u} + d_h \leq \left\lfloor \frac{(T-\delta)-(0)-(0)}{T} \right\rfloor = 0 \\ 0 = \left\lceil \frac{(0)+(0)-(T-\delta)}{T} \right\rceil \leq d_{t_d} - d_h \leq \left\lfloor \frac{(T-\delta)+(T-\delta)-(0)}{T} \right\rfloor = 1. \end{cases} \quad (30)$$

5.4. Opposite Headway Edges Integer Constraints

Every headway edge $h \in E_{hw}$ between train 1 and 2 has an opposite headway edge $\bar{h} \in E_{hw}$ from train 2 to train 1. The two-edged cycles formed by two opposite headways are $\forall_{h<\bar{h}} h \in E_{hw} : c_{h,opp} = (h, \bar{h})$ and the corresponding constraints which we enforce are

$$\forall_{h<\bar{h}} h \in E_{hw} : \underbrace{(m_h + s_h + d_h \cdot T)}_{0 < \bullet < T} + \underbrace{(m_{\bar{h}} + s_{\bar{h}} + d_{\bar{h}} \cdot T)}_{0 < \bullet < T} = 0. \quad (31)$$

Note that minimal headways m_h and $m_{\bar{h}}$ are strictly positive and s_h and $s_{\bar{h}}$ are zero or positive. This implies the bounds in equations (31). Odijks equation now gives

$$\forall_{h<\bar{h}} h \in E_{hw} : -1 = \left\lceil \frac{(\delta)+(\delta)-(T-\delta)-(T-\delta)}{T} \right\rceil \leq d_h + d_{\bar{h}} \leq \left\lfloor \frac{(T-\delta)+(T-\delta)-(\delta)-(\delta)}{T} \right\rfloor = 1. \quad (32)$$

In addition to Odijks rule, we see that the equation equivalent to (31)

$$\forall_{h<\bar{h}} h \in E_{hw} : (d_h + d_{\bar{h}}) \cdot T = \underbrace{-(m_h + s_h)}_{0 < \bullet < T} + \underbrace{-(m_{\bar{h}} + s_{\bar{h}})}_{0 < \bullet < T} \quad (33)$$

< 0

must have a strictly negative (integer) solution for $(d_h + d_{\bar{h}})$, so together with (32) this results in the improved tighter bounds, which we add to our model:

$$\forall_{h<\bar{h}} h \in E_{hw} : \quad d_h + d_{\bar{h}} = -1. \quad (34)$$

5.5. Dijkstra Generated Cycle Constraints

Over our graph, for every transfer edge e , we define a cycle c_e induced by that edge. It is calculated as the concatenation of itself with the edge path p that starts at $o(e)$ and ends at $i(e)$ and has minimal edge length, based on minima of edges only, so supposing the supplements are zero. We calculate the shortest path on the implied undirected version of our directed graph. So $c_e = e \oplus p : p = \text{shortestPath}(o(e), i(e))$, where \oplus stands for the concatenation of subsequent edges. We calculate p by a modified Dijkstra algorithm, which includes a priority queue for speedup. We did this but only for transfer edges. When also doing this on headway edges we did not notice any reduction in solver times. As do [Odijk \(1996\)](#) and [Peeters \(2003\)](#), we define c^+ and c^- as the subsets of edges in cycle c with opposite orientations. We add the following constraints:

$$\forall t \in E_{tr} : \begin{cases} \sum_{e \in (c_t^+ \cap E')} m_e + s_e + \sum_{e \in (c_t^+ \cap E'')} m_e + s_e + d_e \cdot T = \\ \sum_{e \in (c_t^- \cap E')} m_e + s_e + \sum_{e \in (c_t^- \cap E'')} m_e + s_e + d_e \cdot T. \end{cases} \quad (35)$$

Since (9) is a necessary condition for feasibility, we could try and impose, $\forall t \in E_{tr} : \left[\frac{1}{T} \left(\sum_{e \in c_t^+} l_e - \sum_{e \in c_t^-} u_e \right) \right] \leq - \left(\sum_{e \in c_t^+} d_e - \sum_{e \in c_t^-} d_e \right) \leq \left[\frac{1}{T} \left(\sum_{e \in c_t^+} u_e - \sum_{e \in c_t^-} l_e \right) \right]$, as companion equations to equations (35), without reducing the solution space. However, when we did so, solver times did not further reduce, so we removed them again.

For efficiency, the modified Dijkstra algorithm was parallelised both on the core-level (using [openMP](#)) and the machine-level (using [openMPI](#)). OpenMP (multi-platform shared-memory parallel programming) and openMPI (message passing interface) are open source, portable C++ (and Fortran) libraries that are created to allow programmers to parallelise their sequential code over respectively different cores per processor and over different machines on the network. This behaviour is obtained by inserting simple 'pragma'-statements, for example around for loops to parallelise these. This then directs the compiler to generate code that will execute Dijkstra algorithms for different OD pairs concurrently on all available cores and machines.

5.6. Limiting the Set E_{tr}

For description of our MILP model, we define the set E_{tr} as only containing transfer edges defining a passenger flow occurring on them of at least f_{min} . Once this model is solved, the timing of each transfer edge with lower flow than f_{min} can be derived from the timing of the two ride edges which it connects. Although these smaller flows of a transfer edge are not taken into account in our optimisation model, they are used during the simulation based evaluation of the new timetable that results from our optimisation model. We currently set f_{min} to 10 passengers per hourly transfer. This brought down solver time of our model from many hours to about two hours. When we set f_{min} to 50, we typically obtain one hour as solver time, but then the evaluation shows that almost no travel time reduction is obtained. The reason is that what is gained in the optimisation of the bigger flows, is almost entirely lost in the non-considered small flows again.

6. Model Summary

The notation used is defined in Table 2. In the previous sections, we derived the mandatory constraints for our MILP model as given in Table 3 and the optional, solver speed improving constraints in Table 4. Thanks to the natural, large ranges of all supplement variables, we avoid a cause of infeasibilities that do occur in some other models. We give proof of this in [Appendix A](#).

Table 2: Notation used. Note that d_e is defined $\forall e \in E''$, while b_e, m_e, s_e, e_e are defined $\forall e \in E' \cup E''$.

T	= cyclic timetable period	δ	= timetable time resolution		
h_{lo}	= first hour of schedule	h_{hi}	= last hour of schedule		
E'	= set of primary edges	r	= ride	b_e	= begin time of e
E'	= $E_r \cup E_d$	d	= dwell	m_e	= minimum time of e
E''	= set of secondary edges	hw	= headway	s_e	= supplement time of e
E''	= $E_{tr} \cup E_{hw} \cup E_{ta}$	ta	= turn-around	e_e	= end time of e
E	= set of all edges	tr	= transfer	d_e	= integer variable for e
E	= $E' \cup E''$	t	= edge type	V	= vertex set
$i(e)$	= in vertex of e	$i_t(v)$	= unique type t inbound edge of vertex v		
$o(e)$	= out vertex of e	$o_t(v)$	= unique type t outbound edge of vertex v		
$e_t(v_0, v_1)$	= unique type t edge from vertex v_0 to vertex v_1	$i_t(e)$	= unique type t predecessor edge of edge e		
		$o_t(e)$	= unique type t successor edge of edge e		
$s(d)$	= true iff dwell activity d is a stop	$iar(d_0, d_1)$	= true iff station infrastructure allows reordering between trains with dwell activities d_0 and d_1		
\bar{e} is the edge between trains t_2 and t_1 , that is the <i>opposite</i> of the edge e from train t_1 to t_2					
$cd(\dots)$ functions are <u>condition</u> defining functions that set the <u>conditions</u> under which rules apply and also <u>define</u> names for edges used in the further expression					

7. Results

Compared to our previous papers [Sels et al. \(2011b, 2013b,a\)](#), that focussed on our goal function, this paper focusses on the constraints. We discuss three main results of our work. Firstly, in section 7.1, we show that our MILP model avoids a cause of infeasibilities that does occur in other models. Secondly, in 7.2, for the case of all passenger

Table 3: Mandatory constraints, enforced to generate valid timetables. The notation used is described in Table 2.

<i>intra-process constraints:</i>	$\forall e \in E : b_e + m_e + s_e = e_e$
<i>inter-process</i>	$\forall_{o(e_0)=i(e_1)}(e_0, e_1) \in (E', E' \cup E'') : e_{e_0} = b_{e_1}$
<i>connection constraints:</i>	$\forall_{o(e_0)=i(e_1)}(e_0, e_1) \in (E'', E') : e_{e_0} + d_{e_0} \cdot T = b_{e_1}$
<i>continuous variable bounds:</i>	$\forall e \in E : 0 \leq s_e \leq T - \max(m_e, \delta)$
<i>integer variable bounds:</i>	$\forall e \in E'' : (h_{l_0} - h_{h_i}) \leq d_e \leq (h_{h_i} - h_{l_0})$
<i>all trains start in 1st hour:</i>	$\forall \neg \exists_{i_d}(e) \in E_r : h_{l_0} \cdot T \leq b_e < (h_{l_0} + 1) \cdot T$
<i>passing supplements are 0:</i>	$\forall \neg_{s(e)} \in E_d : s_e = 0$
<i>separate same</i>	$cd(h_l; r_u, r_d, h_r) = h_l < \bar{h}_l : \exists r_u \in E_r : r_u = o_r(o(h_l)) :$
<i>direction trains</i>	$\exists r_d \in E_r : r_d = o_r(i(h_l)) : \exists h_r \in E_{hw} : h_r = e_{hw}(o(r_d), o(r_u)).$
<i>on the same open</i>	$\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : ((m_{h_l} + s_{h_l} + d_{h_l} \cdot T) + (m_{r_u} + s_{r_u})$
<i>track section:</i>	$= (m_{h_r} + s_{h_r} + d_{h_r} \cdot T) + (m_{r_d} + s_{r_d}))$
<i>and forbid reordering:</i>	$\wedge (d_{h_l} = d_{h_r})$
<i>separate opposite</i>	$cd(h_l; r_u, r_d, h_r) = h_l < \bar{h}_l : \exists r_u \in E_r : r_u = o_r(o(h_l)) :$
<i>direction trains</i>	$\exists r_d \in E_r : r_d = o_r(i(h_l)) : \exists h_r \in E_{hw} : h_r = e_{hw}(i(r_d), o(r_u)).$
<i>on the same single</i>	$\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : ((m_{h_l} + s_{h_l} + d_{h_l} \cdot T) + (m_{r_u} + s_{r_u})$
<i>open track section:</i>	$-(m_{h_r} + s_{h_r} + d_{h_r} \cdot T) + (m_{r_d} + s_{r_d}) = 0)$
<i>and forbid reordering:</i>	$\wedge (d_{h_l} = d_{h_r})$
<i>forbid or allow</i>	$cd(h_l; d_u, d_d, h_r) = h_l < \bar{h}_l : \exists d_u \in E_d : d_u = o_d(o(h_l)) :$
<i>reordering</i>	$\exists d_d \in E_d : d_d = o_d(i(h_l)) : \exists h_r \in E_{hw} : h_r = e_{hw}(o(d_d), o(d_u)).$
<i>within a station</i>	$\forall_{cd(h_l; d_u, d_d, h_r)} h_l \in E_{hw} : ((m_{h_l} + s_{h_l} + d_{h_l} \cdot T) + (m_{d_u} + s_{d_u})$
<i>depending on</i>	$= (m_{h_r} + s_{h_r} + d_{h_r} \cdot T) + (m_{d_d} + s_{d_d}))$
<i>infrastructure:</i>	$\wedge (\text{if } (\neg \text{iar}(d_u, d_d)) : -0 \leq d_{h_l} - d_{h_r} \leq +0,$
<i>& halting patterns:</i>	$\text{else if } (\neg s(d_u) \wedge \neg s(d_d)) : -0 \leq d_{h_l} - d_{h_r} \leq +0,$
	$\text{else if } (s(d_u) \wedge \neg s(d_d)) : -1 \leq d_{h_l} - d_{h_r} \leq +0,$
	$\text{else if } (\neg s(d_u) \wedge s(d_d)) : -0 \leq d_{h_l} - d_{h_r} \leq +1,$
	$\text{else if } (s(d_u) \wedge s(d_d)) : -1 \leq d_{h_l} - d_{h_r} \leq +1)$

Table 4: Optional constraints, only enforced to lower solver times. The notation used is described in Table 2.

<i>opposite transfers induced</i>	$\forall_{r \leftarrow \bar{t}} \in E_{ir} : ((m_r + s_r + d_r \cdot T) + (m_{\bar{r}} + s_{\bar{r}} + d_{\bar{r}} \cdot T)$
<i>small cycles (hourglasses):</i>	$= (m_{d_u} + s_{d_u}) + (m_{d_d} + s_{d_d})) \wedge (-1 \leq d_t + d_{\bar{t}} \leq 0)$
<i>opposite</i>	$cd(h; d_u, t_u, d_d, t_d) = h < \bar{h} : \exists d_u \in E_d : d_u = o_d(h) :$
<i>dwell-begin-headway</i>	$\exists t_u \in E_{ir} : t_u = e_{ir}(i(h), o(d_u)) :$
<i>induced small cycles</i>	$\exists d_d \in E_d : d_d = o_d(\bar{h}) : \exists t_d \in E_{ir} : t_d = e_{ir}(o(h), o(d_d)).$
<i>(forward triangles):</i>	$\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : ((m_h + s_h + d_h \cdot T) + (m_{d_u} + s_{d_u})$
	$= (m_{t_u} + s_{t_u} + d_{t_u} \cdot T)) \wedge (0 \leq d_{t_u} - d_{h} \leq 1)$
	$\wedge ((m_h + s_h + d_h \cdot T) + (m_{t_d} + s_{t_d} + d_{t_d} \cdot T) = (m_{d_d} + s_{d_d}))$
	$\wedge (-1 \leq d_{t_d} + d_{h} \leq 0)$
<i>opposite</i>	$cd(h; d_u, t_u, d_d, t_d) = h < \bar{h} : \exists d_u \in E_d : d_u = i_d(\bar{h}) :$
<i>dwell-end-headway</i>	$\exists t_u \in E_{ir} : t_u = e_{ir}(i(d_u), i(h)) :$
<i>induced small cycles</i>	$\exists d_d \in E_d : d_d = i_d(h) : \exists t_d \in E_{ir} : t_d = e_{ir}(i(d_d), o(h)).$
<i>(backward triangles):</i>	$\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : ((m_h + s_h + d_h \cdot T) + (m_{t_u} + s_{t_u} + d_{t_u} \cdot T)$
	$= (m_{d_u} + s_{d_u})) \wedge (-1 \leq d_{t_u} + d_{h} \leq 0)$
	$\wedge ((m_h + s_h + d_h \cdot T) + (m_{d_d} + s_{d_d}) = (m_{t_d} + s_{t_d} + d_{t_d} \cdot T))$
	$\wedge (0 \leq d_{t_d} - d_{h} \leq 1)$
<i>opposite headway</i>	$\forall_{h \leftarrow \bar{h}} \in E_{hw} : ((m_h + s_h + d_h \cdot T) = -(m_{\bar{h}} + s_{\bar{h}} + d_{\bar{h}} \cdot T))$
<i>integer constraints:</i>	$\wedge (d_h + d_{\bar{h}} = -1)$
<i>transfer induced</i>	$\forall t \in E_{ir} : \sum_{e \in (e_t^+ \cap E')} m_e + s_e + \sum_{e \in (e_t^+ \cap E'')} m_e + s_e + d_e \cdot T$
<i>Dijkstra cycle constraints:</i>	$= \sum_{e \in (e_t^- \cap E')} m_e + s_e + \sum_{e \in (e_t^- \cap E'')} m_e + s_e + d_e \cdot T$

trains in Belgium, we show that we significantly reduce expected passenger travel time in practice. Thirdly, in 7.5, we show that we can do so in relatively short solver times.

7.1. Feasibility: A Solution is Always Returned in Practice

Since our model has an objective function that properly penalises the choice of big supplements in a soft yet passenger optimal way, there is no reason for us to add a hard constraint that restricts supplements to any arbitrary value lower than $T - \delta$. Other models (e.g. Sparing et al. (2013) and Kroon et al. (2009)) lack an objective function that automatically restricts *all* supplements and so have to enforce a more arbitrary upper bound as a hard constraint.

Usually, this upper bound, directly or indirectly, is chosen lower than our $T - \delta$. As a result, they may risk producing infeasibilities in their model. We believe we have resolved this issue. We indeed notice that, over the last years, in all optimisation experiments with our model described here, our solver did not report any infeasibility.

7.2. Quality: The Solution has Lower Expected Passenger Travel Time in Practice

We applied our model for all passenger trains in Belgium departing between 7 and 8am in the timetable of March, 13th, 2013. We assume primary delay distributions with an average of 2% of the minimum time on all activities, which is Infrabel's current best estimate for morning peak hours. This case is mentioned as the second line of Table 5 and detailed results are given in Figure 4. All six sub-figures represent the change of some measure of time from before to after optimisation. The top row represents planned time, the middle row represents linearised expected time, while the bottom row represents actual expected time (non-linearised). This means that the difference in results between row 3 and row 2 is entirely due to the difference between the non-linear cost curves (row 3) and our piecewise linear approximation of it (row 2). Using non-linear optimisation may totally erase these differences, but it is much harder to solve a non-linear model than our corresponding linear one. Of the two columns, the left column represents total train time. This is the sum over all hourly trains of the trains total trip time, independent of the number of passengers on each train. The right column represents time for all daily passenger streams, also the small streams not considered during optimisation. The colours (blue, yellow, green, orange, red, purple) each stand for a particular activity (ride, dwell, depart, transfer, arrive, knock-on). There are dark and light versions of some colours (yellow, green, orange, red, purple). The dark colour indicates minimum times, while the lighter version indicates the supplement times of the same activity type. The shading with blue lines indicates that these activities (all except knock-on) were convoluted with ride activities.

Considering planned *train* time in column 1, row 1, we see a negligible decrease of the total ride+dwell time supplements from 12.63% to 11.42%. In column 1, row 2, evaluation via the linear cost functions shows a decrease of 13.21% to 12.40%. Column 1, row 3, shows a very similar decrease from 13.24% to 12.50%. This small difference between row 2 and 3 indicates that our piecewise linear approximation performs well. It should be noted, however, that our objective (and objective function) is not about minimising the total train running times (column 1) but it is about minimising the total passenger travel time in practice (column 2).

Column 2, row 1 of Figure 4, planned *passenger* time, demonstrates a much larger reduction of the ride+dwell supplements than in planned *train* time, from 10.06% down to 2.76%. These are weighted by number of passengers. This same advantageous larger expected time reduction in column 2 compared to column 1, is also present in the expected time domains. Indeed, in column 2, row 2, representing evaluation of the linearised cost functions, ride+dwell supplements go down from 9.88% in the original timetable to 3.77% in our optimised timetable. In column 2, row 3, representing evaluation on the true, non-linear cost functions, supplement percentages go down similarly from 10.0% to 3.80%. But also for expected knock-on time we reduce time (row 2 (5.93% down to 1.63%) and row 3 (5.06% down to 1.86%)). The decrease of these expected ride, dwell and knock-on times is compensated only partially by the increase in expected transfer time (row 2 (10.43% up to 16.80%) and row 3 (10.18% up to 17.10%)). Note that this expected transfer time includes both successful and missed transfers, properly weighted with their respective probabilities to occur. Our calculations, both during optimisation and during evaluation, assume a penalty for a missed transfer of 1 hour. Since our timetable is cyclic the same train occurs with the same timing every next hour. In practice, a passenger could also take another train in the same direction without waiting a full hour. We did not model this effect. This means that the expected transfer time in our model is conservatively somewhat overestimated. Since missed transfers are penalised more in our model than in a model that would account for this effect, our model will generate a timetable that has fewer missed transfers. In column 2, row 2, the net total expected passenger travel time reduction is 6.03% for the approximate linear cost function evaluation and in row 3 it is still 3.81% for the actual cost function evaluation. So we conclude that 3.81% is our best prediction for reduction of expected passenger time for all passenger streams together. The average train passenger may expect this reduction amount in practice when the optimised timetable is used.

Because the assumed amount of primary delays has an effect on the actual timetable that is generated, we then varied these primary delay distributions. For each ride and dwell activity we still assumed the same negative exponential type of primary delay distributions, but we varied them by increasing their average (expected value) of $a\%$ of each activity's minimum time. This average is given in column 1 of Table 5 for cases from $a = 1\%$ up to $a = 20\%$.

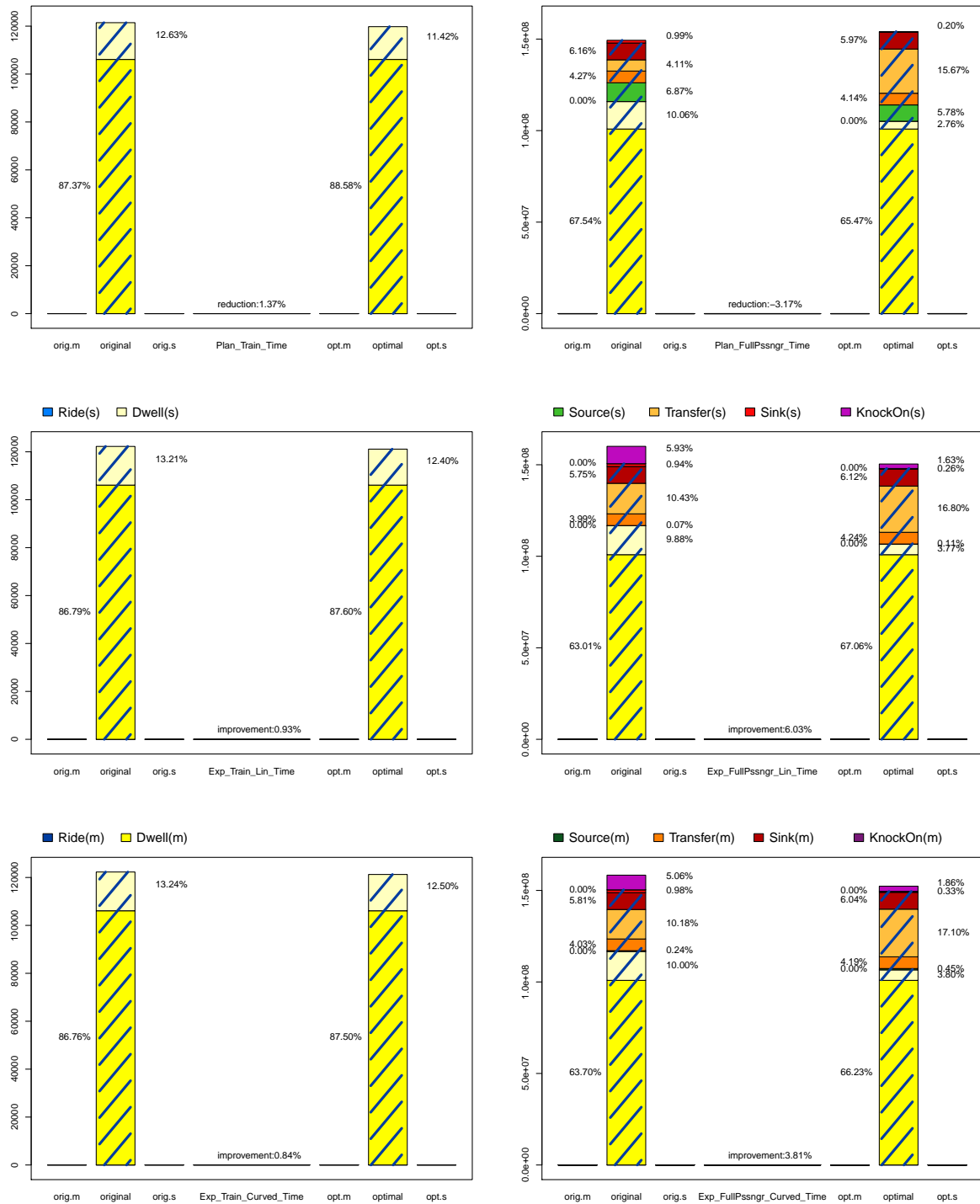


Figure 4: Reducing Train Time and Expected Passenger Time from Original to Optimised Timetable as also reported in line 2 of Table 5.

Note that for each value of a , a different timetable will be generated. Also, a value of $a = 20\%$ may seem unrealistically high, but this was chosen to show that our model can keep generating timetables at any value of a . At first, this may seem surprising if one realises that in some places in Belgium, as in the bottleneck in Brussels-Central station, leftover capacity is less than 20%. However, the constructed timetable for $a = 20\%$ will plan all trains in an hour, because these are hard constraints of our model. In practice, there will be delays, and the higher the value of a , the more statistical cases there will be that these trains will in practice not be able to all follow the prescribed timetable. However, these negative effects are allowed by the model since the penalties on them are only soft penalties present in the objective function. It remains therefore the responsibility of the user of our tool that not too many trains are forced to be planned. For example, the UIC 406 norm can be utilised to restrict the capacity consumption of the input train set to our model. The fact that our model allows scheduling of trains that consume up to 100% capacity has the advantage that it can also report total expected passenger time of a schedule at 80% and 95% and is likely to report that the second schedule has a higher total expected passenger time than the first because of its excessively higher knock-on delays. As such, our tool may be used to explore at which point adding more trains to the schedule stops being advantageous for passengers and should be discouraged.

Table 5: For (different) primary delay distributions, characterised by their average of $a\%$ of minimum dwell and ride times, properties of the (different) resulting optimised timetable are shown. Missed transfer probability of the original timetable is shown for easy comparison.

a	solver time	resulting	major	major	all	all	missed transfer	
		MILP gap	flows linearised time reduction	flows non-linearised time reduction	flows linearised time reduction	flows non-linearised time reduction	original timetable	optimised timetable
%	min.	%	%	%	%	%	%	%
1	107	83.0	6.25	4.19	5.64	3.58	13.6	2.42
2	123	79.2	6.69	4.48	6.03	3.81	13.9	2.60
3	123	77.0	5.49	3.10	4.85	2.45	14.2	2.31
4	139	74.2	5.00	2.55	4.41	1.94	14.5	2.14
5	128	70.2	5.66	3.19	5.04	2.57	14.8	2.37
6	125	68.8	4.45	1.89	3.91	1.35	15.1	3.44
7	96	67.2	3.64	1.17	3.08	0.61	15.4	2.61
8	117	64.0	4.07	1.60	3.52	1.05	15.7	2.35
9	118	61.0	4.50	1.87	3.96	1.33	16.0	2.31
10	133	59.9	3.74	1.30	3.21	0.78	16.3	1.91
11	126	57.6	3.90	1.26	3.36	0.74	16.6	2.06
12	127	55.2	4.20	1.57	3.63	1.00	17.0	2.68
13	121	53.5	4.32	1.70	3.75	1.13	17.3	2.44
14	130	51.9	4.18	1.57	3.65	1.04	17.6	2.85
15	120	51.8	3.38	0.61	2.85	0.09	17.9	2.92
16	117	49.8	3.74	1.06	3.21	0.56	18.2	2.69
17	100	47.0	4.59	1.60	4.05	1.07	18.5	2.35
18	441	48.9	2.78	0.01	2.31	-0.46	18.8	2.63
19	144	44.9	4.55	1.75	4.03	1.24	19.1	2.87
20	134	44.9	3.97	1.04	3.51	0.58	19.4	2.34

Column 2 and 3 in Table 5 show the solver time and the MILP gap achieved. For all rows, to guarantee a solution in a short time, the maximum desired MILP gap was set slightly above what was obtained as the gap of the first returned solution in earlier trials. We ran Gurobi 5.6.3 on a HP Z210 Workstation Xeon CPU E31240 at 3.3GHz with 16GB memory running Microsoft Windows 7 Enterprise. In table 6, we give some numbers that indicate the problem size that hold for all these cases. This allows comparison to other published models. Our graph contains 196 hourly trains, 5078 ride edges, 4882 dwell edges, 15495 major transfer edges, 41258 knock-on(=headway) edges and 168 turn-around edges. Our model contains 52067 b and 52067 s decision variables and 52067 e expressions. It has 42107 d integer decision variables, corresponding to one for each secondary edge, and 50481 objective function terms for major flows of which 41258 terms are functions of 1 s variable and 9223 are functions of 2 s variables. For final evaluation over all streams, 70909 function terms are added. After pre-solve, which only takes about 5 seconds, our model contains 165298 rows, 111576 columns and 464064 non-zero elements.

Compared to the timetable currently in operation, our optimised timetables have quite some advantages. Firstly, they respect all minimum ride- and dwell-times without exception. Secondly, they respect all headway time buffers

Table 6: Problem instance statistics.

# ride edges =	5078	# (continuous) b-variables =	52067
# dwell edges =	4882	# (continuous) s-variables =	52067
# turn-around edges =	168	# (continuous) e-expressions =	52067
# knock-on(headway) edges =	41258	# (integer) d-variables =	42107
# major transfer edges =	15495	# objective function terms for major flows =	50481
# model rows =	165298	# are functions of 1 s-variables =	41258
# model columns =	111576	# are functions of 2 s-variables =	9223
# model non-zero elements =	464064	# function terms in post-optimisation evaluation =	70909

of 3 minutes between all train pairs on the same track section. Thirdly, our calculations show that, over all primary delay assumptions of Table 5, the average chance of missing a transfer in the current timetable is at least 14.1% while in our optimised timetable it is at most 3.44%. For the case $a = 2\%$ in Table 5, the expected passenger time is 3.81% lower than in the original schedule. Further reduction of the maximum desired MILP gap below 79.2% did not give any new timetable solution within 12 hours. The decrease with 3.81% might seem small, but it should be noted that the fixed minimal ride and dwell times already consume 63.70% of the total passenger time in the original timetable and 66.23% in our optimised timetable.

There are at least three straightforward methods to further increase the reduction of 3.8%. Firstly, lowering the MIP gap can be attempted by simply allowing more solver time. However, in many cases we did not obtain an additional 1%, even in a few days of extra solver time. Secondly, differences of reduction percentages between evaluation over major flows versus all flows can be reduced by optimising also over the smaller flows. However, when trying this by lowering f_{min} below 7, sets with more than 50 trains did not return a solution within 12 hours anymore. Thirdly, looking at the case $a = 2\%$, columns 4 and 5 in Table 5, we see that we also still lose some of the 6.69% reduction achieved by the optimisation with the piecewise linear cost functions compared to the 4.48% reduction if we evaluate over the curved cost functions. An approximation with more than 2 segments could reduce the difference between those percentages, probably at the cost of a higher solver time. Alternatively, non-linear optimisation using the non-linear objective function directly, would completely eliminate the mentioned percentage difference. We also expect more computation time here. However, we have not tried these methods yet.

The different rows in Table 5 show the first feasible result obtained by Gurobi when a desired gap of 79.2% was set, for all cases of $a = 1\%$ to $a = 20\%$. When the assumed primary delays increase, our method, except for the case $a = 18\%$ where it does marginally worse, is still able to generate a timetable with decreased expected passenger travel time compared to the current one. However, roughly spoken, the larger the primary delays, the lower the obtained reduction for the optimised timetable becomes. Column 7 shows that it decreases from 3.58% at $a = 1\%$ down to 0.58% at $a = 20\%$. Computation times to the first feasible result remain similar. In column 8 we see that the original timetable, which is of course the same in all cases $a = 1\%$ up to $a = 20\%$, has an increasing missed transfer probability from 13.6% up to 19.4%. That this probability increases for any given timetable is only natural. However, in column 9, we see for our optimised timetables, which are different ones for every value of a , that the missed transfer probability remains roughly the same, always between 1.91% and 3.44%. It is remarkable to see that our method can calculate a (different) timetable with consistently low missed transfer percentages even for very large primary delays. This is due to the fact that these are also explicitly and proportionally penalised in our transfer cost terms in our objective function. More generally, it reflects that the produced timetables are optimised for the passengers, whatever the circumstances.

Since we always compare our optimised timetable to the original timetable, the impression may arise that we need a feasible timetable to start from. This is not the case. In fact, the original timetable is not feasible, since for some trains, it does not respect the minimum ride times in some places. Our optimised timetable does respect all those restrictions. In that sense the requirements on the optimised timetable are higher and positive results for the optimised timetable are to be seen as conservative.

7.3. Balance: Less Expected Knock-On Delay Time versus More Expected Transfer Time

As shown in figure 4, on the third row, right column, the total expected passenger time from original to optimised timetable decreases by 3.81%. Figure 4 also shows that the total expected knock-on delay time decreases while the expected transfer time increases.

7.3.1. Less Expected Knock-On Delay

Timetablers in Belgium, and many other European countries, construct a timetable with the goal to respect 'macroscopic headway minima' of 3 minutes. They do not use more accurate microscopic headway information nor more accurate estimations than those 3 minute minima. So do we in our optimised macroscopic timetable. So our headway minima are exactly the same and in that sense headway separations and associated knock-on time results are comparable.

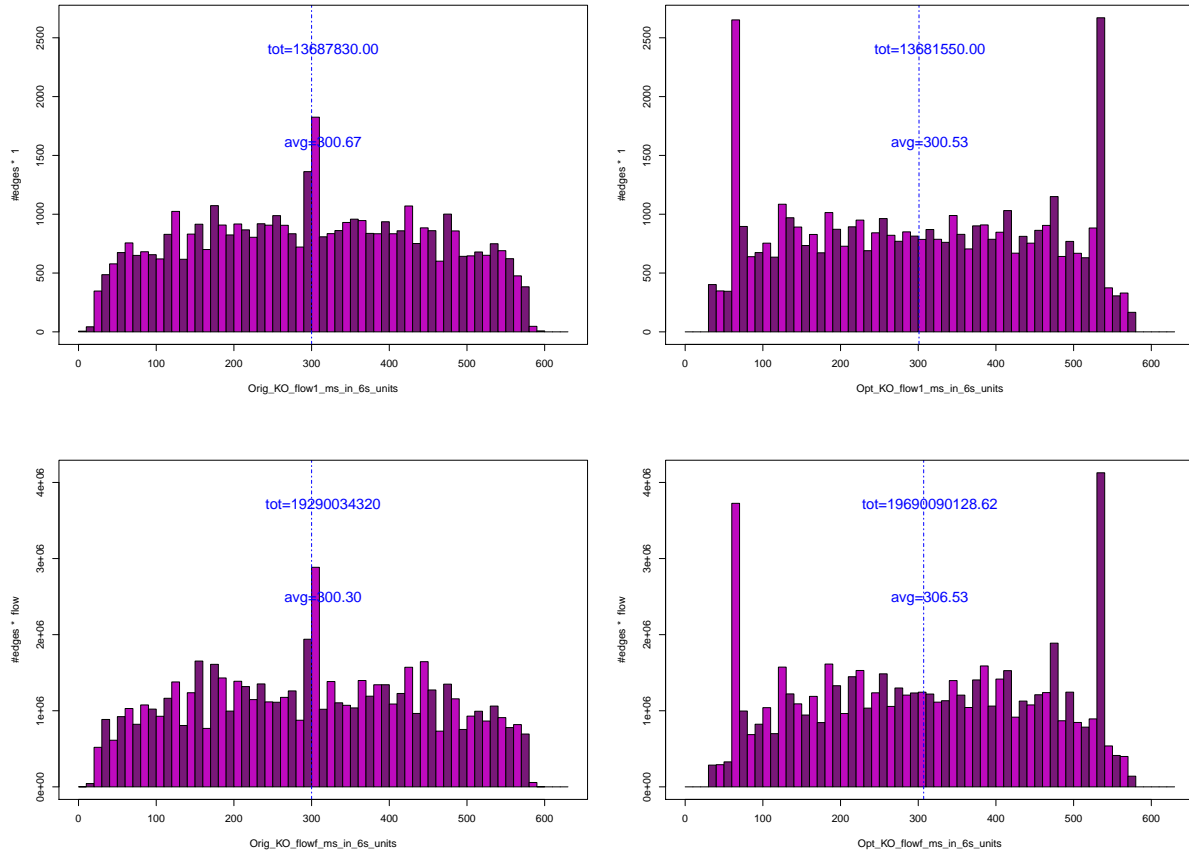


Figure 5: Planned headway times (minimum of 3 minutes + supplement) histograms, showing for each headway duration, for how many edges it occurs and how many passengers experience the knock-on time associated with this headway duration.

In figure 5, we show histograms of headway durations for both the original (in the left half) and the optimised timetable (in the right half) and in each case, both for the planned time domain, in terms of edges (top row) and in terms of passengers (bottom row). One can see that, for the optimised timetable, in the right half of figure 5, no headways are shorter than the imposed 3 minute minimum. Equivalently, none are longer than $T - 3 = 57$ minutes. In the original timetable, there are a few headways that are smaller than the minimal 3 minutes and some are larger than the 57 minutes maximum. There are 2 causes of this. Firstly, between some train pairs, less headway is planned by the planners. Secondly, for a few trains, for a few tracks, the original timetable violates minimum ride or dwell times. This happens precisely when timetablers are trying to fix headway violations. Since in practice, these runtime violations will mean that the train will be delayed, we also adapt the original timetable to reflect this, so arrival and departure times for the part of the train after the violation are increased so that the minimum ride and dwell times are respected. This means that headway times are again decreased. From the top row of figure 5, excluding the mentioned violations, it is evident that the average headway, if counted per edge, is always $T/2$. This is the case because every headway

edge e of length $m_e + s_e$ also has an opposite headway edge of length $T - (m_e + s_e)$. Small deviations in the average are entirely due to the discretisation in the histogram construction. When weighing with passenger numbers, we get a different situation. Indeed, we see that in our optimised timetable, many people experience a potential knock-on delay associated with a second train of a train-pair that is separated by 7 minutes and also of $T - 7 = 53$ minutes. Assigning big headways to trains riding before trains with many people on reduces total expected passenger knock-on time. This is the case because the cost function, as can be seen in figure 1, is a decreasing function. Note also that on the second row of figure 5, the optimised timetable, in the right column, indicates that fewer people will experience headways of between 3 and 6 minutes than in the original timetable. The first row, shows a less pronounced difference, and this can again be explained by the optimised timetable paying more attention to trains with many passengers than with few passengers via the objective function. This is consistent with the bar graphs shown in row 3, column 2 of figure 4.

7.3.2. More Expected Transfer Time

Our optimisation currently still results in a timetable with more expected transfer time and to improve on this is a topic for further study. So our software is already better at planning headways but humans are currently still better at planning transfers. Important questions now are if the reduction in expected headway time is coupled to the gain in expected transfer time or not, if a timetable with lower expected time for both knock-ons and transfers exists and whether it can be found in an automated way and how.

7.4. Balance: Less Journey Time Offsets More Excess Journey Time

Spreading alternative trains between the same origin and destination station over the period of the timetable is advantageous for some passengers. Indeed, consider passengers who can not adapt their arrival time at their station of departure to the departure time of a specific train of their choice, but arrive there at a uniformly distributed random time. [Welding \(1957\)](#), [Holroyd and Scraggs \(1966\)](#) and [Osuna and Newell \(1972\)](#) derived that, for this type of passengers, the expected waiting time until departure $E(w)$ can be expressed as a function of the *average vehicle planned heading time* $E(h)$ (over all heading times H_i in the cyclic timetable period T) and the *variation coefficient* $C_v(h)$ of the real time heading times:

$$E(w) = E(h)/2 \cdot (1 + C_v(h)^2). \quad (36)$$

Here, h is the heading time distribution and $E(h)$ is the expected heading time as it can be calculated from the planned timetable as

$$E(h) = \sum_{i=0}^{N-1} p_i \cdot H_i = \sum_{i=0}^{N-1} (H_i/T) \cdot H_i = \sum_{i=0}^{N-1} H_i^2/T, \quad (37)$$

where p_i is the probability of a passenger experiencing heading time H_i in period T . $C_v(h) = \sigma(h)/\mu(h)$ is the ratio of the standard deviation over the mean, both of the heading time distribution in real time. Substitution of the right hand side of equation (37) for $E(h)$ in equation (36) and multiplication by the number of considered randomly arriving passengers f delivers

$$E(f \cdot w) = \frac{f}{2T} \sum_{i=0}^{N-1} H_i^2 \cdot (1 + C_v(h)^2). \quad (38)$$

If we assume that there are no large deviations from the timetable, because it is robust, $C_v(h)$ can be approximated by 0 and we get

$$E(f \cdot w) \approx \frac{f}{2T} \sum_{i=0}^{N-1} H_i^2. \quad (39)$$

It can be shown that $E(f \cdot w)$ is minimised by setting all H_i equal to T/N . This is why many PESP papers use these equalities as hard constraints: the *regularity constraints* ([Peeters, 2003](#); [Kroon et al., 2007](#); [Liebchen, 2006, 2007](#); [Caprara et al., 2007](#); [Kroon et al., 2009](#); [Caprara et al., 2011b](#); [Sparing et al., 2013](#)). However, this can sometimes be overly strict, limiting other supplements in the network to be unnecessarily constrained and as such avoiding optimality in terms of the whole network and its expected total travel time for all passenger together.

So instead, for a given timetable and for a given set of alternative trains between stations O and D , we prefer to evaluate the expected inter-departure waiting time at O for the f passengers going from O to D in the timetable period T via equation (39). The inter-arrival waiting time at D is computed similarly. Together, inter-departure and inter-arrival time are called *excess journey time* (Zhao et al., 2013). It must be mentioned that currently, our objective function represents just total expected journey time for all passengers. This does not include the excess journey time yet and this addition would be most useful, but our first experiments of adding it to the objective function during optimisation show it is too computationally expensive. So for now, we will only evaluate this excess journey time for both the original and the optimised timetable.

Table 7 shows the computation of excess journey time for eleven different corridors between couples of neighbouring main cities, in both directions. For example, each day 1086 passengers go from Gent Sint-Pieter to Brussel-Noord on one of the three trains: IC:A (series 500-524), IC:E (series 1500-1525) and IR:i (series 3600-3624). IC stands for InterCity and IR for InterRegio. In the original timetable, departure times of these trains are planned at 7:03:00, 7:15:00 and 7:24:00. This leaves an interval of 39 minutes without a train departure which seems bad and gives $E(w) = 14.55$ minutes according to equation (39). In the optimised timetable, the departure times are changed to 7:39:42, 7:51:12 and 7:57:18. This leaves an interval of 42.4 minutes without a train departure which seems even worse and leads to a somewhat larger $E(w) = 16.39$ minutes. So the ratio of $E(w)$ for the optimised timetable over $E(w)$ for the original timetable is $16.39/14.55 = 1.13$. At arrival in Brussel-Zuid, the situation is similar, the $E(w)$ ratio is 1.36. However, for the opposite lines, from Brussel-Zuid to Gent-Sint-Pieter, there is an improvement from the original to the optimised timetable with ratios of 0.85 at departure and 0.78 at arrival. If, for this OD-pair, we take all excess journey time in the optimised timetable and divide it by all excess journey time in the original timetable, we get a factor 1.03. So excess journey time in the optimised timetable is increased with 3%. However, what goes together with the temporal spreading is of course the duration of these three train paths from origin to destination and back. We see that in the original timetable, the lines each took 31.0 minutes which has been changed in the optimised timetable to 27.6, 27.0 and 17.7 minutes, so on average 24.1 minutes, which is a reduction of the average passenger journey time of a factor $24.1/31.0 = 0.78$. Similarly, in the opposite direction, we reduce passenger journey time with a factor 0.87. The ratio of all journey time for this corridor is 0.82, so a reduction from original to optimised timetable of 18%. When we add up all time for this corridor, we have to weigh the excess journey time by the fraction ‘ r ’ of passengers who cannot adapt to the timetable departure times. When we assume $r = 100\%$, total time on the original timetable is 119.4 minutes and total time for the optimised timetable is 110.4, which means a reduction by a factor $110.4/119.4 = 0.92$ or 8%. So, for this corridor, in the optimised timetable, the excess journey time is 3% higher, but the time that is lost there for passengers is more than compensated by the decrease in journey time of 18% of going from origin to destination station. For these 11 OD-pairs together, the net time gained for passengers is 8%.

The situation is entirely similar for all other corridors shown in table 7 giving a net reduction of 1% to 12% per OD-pair, except for the OD-pairs (Namur,Brussel) (+3%), Oostende-Brugge (+2%) and (Enghien-Brussel) (+9%). The total reduction of total time over these 11 OD-pairs, so weighing with number of passengers as well as spent time per OD-pair, is 4.2%, assuming that $r = 100\%$. Table 8 show that for smaller values of ‘ r ’, the improvement is more significant, up to 15.6% for $r = 0$. This increase in improvement is logical, since the excess journey time part that is not controlled yet by our objective function is weighted by this factor ‘ r ’ while the journey time that is minimised via our objective function is not.

From the examples above, we can conclude that imposing regularity constraints has the potential drawback of increasing journey time with more time than is gained by the reduction of excess journey time. This is true, even for train time unweighted with passengers. This is why we want to avoid this approach.

7.5. Computation Speed: The Solution is Returned Quickly

As shown in Table 5, for any primary delay distributions assumed, our solver time stays below 2.5 hours, be it with one exception for the case $a = 18\%$ where it requires 7 hours 21 minutes. Obviously, this is a big improvement compared to the current manual timetable creation process that takes many human planners many months. Infrabel employs about 20 planners who work on a new planning for 4 to 6 months. Apart from this, the main passenger operator, NMBS, also employs their planners who also spend time on creating an initial planning, that is then passed on to Infrabel for verification and adaption. Infrabel communicates their changes back to NMBS, and this process is iterated over a few times. Note however, that in the current process, the planning time also includes checking and correcting routing and platforming on the microscopic level, which our method does not currently do.

Table 7: Evaluation of planned excess journey time (ejt) and comparison with journey time (jt) over OD pairs between neighbouring main cities. Change of both measures and combined measures (with $r=100\%$) from original to optimised timetables.

O & D	orig line 1	orig line2	orig (line 3)	orig E(.)	opt line 1	opt line 2	opt (line 3)	opt E(.)	opt/orig ratio E(.)	opt/orig totals
1	IC:F:1725	IC:A:525			IC:F:1725	IC:A:525				20.32
Liege-Guil[726]	7:08:00	7:00:00		23.1	7:19:48	7:51:12		15.0	0.65	
Leuven[715]	8:01:00	7:34:00		15.2	8:05:42	8:19:42		19.3	1.27	
E(jt)	53.0	34.0		43.5	45.9	28.5		37.2	0.86	
1	IC:F:1700	IC:A:500			IC:F:1700	IC:A:500				
Leuven[715]	7:59:00	7:26:00		15.2	7:33:18	7:49:30		18.2	1.20	
Liege-Guil[726]	8:54:00	8:00:00		24.6	8:35:24	8:20:54		19.0	0.77	0.92
E(jt)	55.0	34.0		44.5	62.1	31.4		46.8	1.05	0.95
				166.0				155.4		0.94
2	IC:M:2425	IC:J:2125			IC:M:2425	IC:J:2125				559.78
Namur[895]	7:51:00	7:21:00		15.0	8:24:24	9:35:48		20.8	1.38	
Brussel-Zuid[221]	8:48:00	8:18:00		15.0	9:12:06	10:25:36		19.5	1.30	
E(jt)	57.0	57.0		57.0	47.7	49.8		48.8	0.86	
2	IC:M:2400	IC:J:2100			IC:M:2400	IC:J:2100				
Brussel-Zuid[221]	7:12:00	7:42:00		15.0	6:53:54	6:43:42		21.5	1.44	
Namur[895]	8:09:00	8:39:00		15.0	7:44:30	7:31:36		19.9	1.32	1.36
E(jt)	57.0	57.0		57.0	50.6	47.9		49.3	0.86	0.86
				174.0				179.7		1.03
3	IC:N:4500	IC:I:2000			IC:N:4500	IC:I:2000				504.49
Charleroi-S[259]	7:37:00	7:08:00		15.0	6:59:54	6:32:42		15.1	1.01	
Brussel-Zuid[220]	8:25:00	7:54:00		15.0	7:46:18	7:13:42		15.1	1.01	
E(jt)	48.0	46.0		47.0	46.4	41.0		43.7	0.93	
3	IC:N:4525	IC:I:2025			IC:N:4525	IC:I:2025				
Brussel-Zuid[220]	7:34:00	7:06:00		15.1	8:13:18	7:46:54		15.2	1.01	
Charleroi-S[259]	8:18:00	7:52:00		15.3	8:55:54	8:25:36		15.0	0.98	1.00
E(jt)	44.0	46.0		45.0	42.6	38.7		40.7	0.90	0.92
				152.4				144.8		0.95
4	IC:F:1700	IR:j:3700			IC:F:1700	IR:j:3700				336.47
Mons[848]	7:43:00	7:07:00		15.6	7:29:54	7:15:12		18.9	1.21	
Brussel-Zuid[220]	8:27:00	7:55:00		15.1	8:05:30	7:56:18		22.2	1.47	
E(jt)	44.0	48.0		46.0	35.6	41.1		38.4	0.83	
4	IC:F:1725	IR:j:3725			IC:F:1725	IR:j:3725				
Brussel-Zuid[220]	7:33	7:04:00		15.0	7:40:00	7:10:12		15.0	1.00	
Mons[848]	8:16	7:52:00		15.6	8:14:24	7:55:54		17.2	1.10	1.20
E(jt)	43.0	48.0		45.5	34.4	45.7		40.1	0.88	0.86
				152.8				151.7		0.99
5	IC:H:1900	IR:d:3100			IC:H:1900	IR:d:3100				329.87
Enghien[360]	7:33:00	7:07:00		15.3	7:16:30	7:07:18		22.2	1.45	
Brussel-Zuid[220]	7:53:00	7:30:00		15.8	7:35:06	7:25:54		22.2	1.40	
E(jt)	20.0	23.0		21.5	18.6	18.6		18.6	0.87	
5	IC:H:1925	IR:d:3125			IC:H:1925	IR:d:3125				
Brussel-Zuid[220]	7:07:00	7:30:00		15.8	6:31:06	7:01:12		15.0	0.95	
Enghien[360]	7:26:00	7:53:00		15.2	6:52:42	7:20:24		15.1	1.00	1.20
E(jt)	19.0	23.0		21.0	21.6	19.2		20.4	0.97	0.92
				104.6				113.5		1.09
6	IC:A:500	IC:E:1500	IR:i:3600		IC:A:500	IC:E:1500	IR:i:3600			1086.56
Gent Sint-Pieter[455]	7:24:00	7:03:00	7:15:00	14.6	7:51:12	7:39:42	7:57:18	16.4	1.13	
Brussel-Zuid[220]	7:55:00	7:34:00	7:46:00	14.6	8:18:48	8:06:42	8:15:00	19.8	1.36	
E(jt)	31.0	31.0	31.0	31.0	27.6	27.0	17.7	24.1	0.78	
6	IC:A:525	IC:E:1525	IR:i:3625		IC:A:525	IC:E:1525	IR:i:3625			
Brussel-Zuid[220]	7:05:00	7:29:00	7:14:00	13.4	7:49:06	8:19:00	8:05:54	11.3	0.85	
Gent Sint-Pieter[455]	7:36:00	7:57:00	7:49:00	14.6	8:16:12	8:46:00	8:33:42	11.4	0.78	1.03
E(jt)	31.0	28.0	35.0	31.3	27.1	27.0	27.8	27.3	0.87	0.82
				119.4				110.4		0.92
7	IC:C:700	IC:G:1800	IC:P:3025		IC:C:700	IC:G:1800	IC:P:3025			393.16
Gent Sint-Pieter[455]	7:16:00	7:47:00	7:05:00	11.7	7:32:54	7:06:54	6:57:42	11.5	0.98	
Antwerpen-Centraal[37]	8:06:00	8:42:00	8:01:00	14.0	8:20:48	8:01:00	7:50:54	11.7	0.83	
E(jt)	50.0	55.0	56.0	53.7	47.9	54.1	53.2	51.7	0.96	
7	IC:C:725	IC:G:1825	IC:P:3000		IC:C:725	IC:G:1825	IC:P:3000			
Antwerpen-Centraal[37]	7:52:00	7:18:00	7:59:00	13.1	6:29:30	6:35:42	6:48:54	15.5	1.19	
Gent Sint-Pieter[455]	8:44:00	8:13:00	8:55:00	11.7	7:17:48	7:28:00	7:41:42	13.3	1.13	1.03
E(jt)	52.0	55.0	56.0	54.3	48.3	52.3	52.8	51.1	0.94	0.95
				158.5				154.8		0.98
8	IC:G:1800	IC:C:700	IC:A:500		IC:G:1800	IC:C:700	IC:A:500			1177.78
Oostende[929]	7:02:00	7:12:00	7:43:00	11.9	6:29:00	6:58:54	6:17:18	11.4	0.96	
Brugge[220]	7:18:00	7:27:00	7:54:00	11.6	6:41:06	7:15:42	6:29:18	12.7	1.10	
E(jt)	16.0	15.0	11.0	14.0	12.1	16.8	12.0	13.6	0.97	
8	IC:G:1825	IC:C:725	IC:A:525		IC:G:1825	IC:C:725	IC:A:525			
Brugge[220]	7:44:00	7:30:00	7:03:00	10.7	7:53:30	7:12:48	7:43:54	11.9	1.11	
Oostende[929]	7:58:00	7:44:00	7:17:00	10.7	8:06:00	7:25:24	7:56:12	11.8	1.11	1.07
E(jt)	14.0	14.0	14.0	14.0	12.5	12.6	12.3	12.5	0.89	0.93
				72.8				74.0		1.02
9	IC:N:4525	IC:I:2025	IC:Q:2625		IC:N:4525	IC:I:2025	IC:Q:2625			654.61
Antwerpen-Centraal[37]	7:47:00	7:17:00	7:42:00	12.9	7:33:48	6:59:54	7:21:36	10.8	0.84	
Mechelen[810]	8:07:00	7:37:00	8:03:00	13.3	7:48:42	7:22:30	7:37:24	12.4	0.94	
E(jt)	20.0	20.0	21.0	20.3	14.9	22.6	15.8	17.8	0.87	
9	IC:N:4500	IC:I:2000	IC:Q:2600		IC:N:4500	IC:I:2000	IC:Q:2600			
Mechelen[810]	7:53:00	7:23:00	7:57:00	13.3	7:12:54	7:39:30	8:22:00	12.5	0.95	
Antwerpen-Centraal[37]	8:13:00	7:43:00	8:18:00	12.9	7:27:42	7:55:18	8:37:48	12.2	0.94	0.92
E(jt)	20.0	20.0	21.0	20.3	14.8	15.8	15.8	15.5	0.76	0.82
				93.0				81.2		0.87
10	IC:R:3425	IC:N:4525	IC:I:2025		IC:R:3425	IC:N:4525	IC:I:2025			525.90
Mechelen[810]	7:57:00	7:10:00	7:40:00	11.3	7:00:00	7:49:42	7:24:30	11.2	0.99	
Brussel-Noord[221]	8:13:00	7:23:00	7:55:00	12.1	7:14:24	8:01:36	7:35:48	10.7	0.89	
E(jt)	16.0	13.0	15.0	14.7	14.4	11.9	11.3	12.5	0.85	
10	IC:R:3400	IC:N:4500	IC:I:2000		IC:R:3400	IC:N:4500	IC:I:2000			
Brussel-Noord[221]	7:47:00	7:37:00	7:05:00	12.1	7:15:54	7:58:36	7:25:18	12.5	1.03	
Mechelen[810]	8:03:00	7:50:00	7:20:00	11.3	7:30:42	8:10:48	7:36:06	13.6	1.20	1.03
E(jt)	16.0	13.0	15.0	14.7	14.8	12.2	10.8	12.6	0.86	0.86
				76.1				73.1		0.96
11	IC:G:1800	IC:A:500	IC:E:1500		IC:G:1800	IC:A:500	IC:E:1500			1169.78
Brugge[210]	7:18:00	7:59:00	7:36:00	10.1	6:42:12	7:31:24	7:18:42	13.4	1.33	
Gent Sint-Pieter[455]	7:45:00	8:22:00	8:00:00	10.3	7:05:54	7:50:12	7:37:42	11.8	1.14	
E(jt)	27.0	23.0	24.0	24.7	23.7	18.8	19.0	20.5	0.83	
11	IC:G:1825	IC:A:525	IC:E:1525		IC:G:1825	IC:A:525	IC:E:1525			
Gent Sint-Pieter[455]	7:42:00	7:01:00	7:24:00	10.1	7:29:06	7:17:18	7:48:00	11.3	1.12	
Brugge[210]	8:15:00	7:38:00	8:00:00	10.3	7:52:30	7:36:12	8:06:30	11.2	1.09	1.17
E(jt)	33.0	37.0	36.0	35.3	23.4	18.9	18.5	20.3	0.57	0.68
				100.9				88.5		0.88

Table 8: Reduction of expected passenger time, including both journey time and excess journey time for varying fractions ($r\%$) of passengers not adapting their arrival time at station of departure to train departure times in the timetable.

r (%)	0	10	20	30	40	50	60	70	80	90	100
reduction (%)	15.6	13.6	12.0	10.6	9.3	8.2	7.2	6.4	5.6	4.9	4.2

8. Validation and Check of Applicability by Simulation

8.1. Validation by Macroscopic Simulation

To validate the robustness of our timetable we simulated both the original timetable and the optimised timetable with the macroscopic simulator OnTime of TrafIT GmbH and VIA Consulting & Development GmbH. OnTime performs delay propagation calculations with given primary delays, minimal run times, minimal dwell times, minimal headway times at sections, junctions and stations. It can also take into account train priorities, minimal turn-around times and a list of transfers with synchronisation times. It does not take into account passenger numbers. This means that all reported measures are train related. OnTime reports train punctuality in various forms. In figures 6(a) to 6(n), the horizontal axis represents the time of the day and the vertical axis represents the number of events. An event is a departure or arrival of a train. Green pixels represent events that occur with a delay of less than 5 minutes compared to its planned time. Red pixels represent events that occur 5 or more minutes later than planned. Dark red pixels represent events that occur more delayed than events represented by light red pixels. Figures 6(a) to 6(b) show that without primary delays, there is no accumulated remaining delay in the system. We know that the original timetable has some run time violations, implying negative supplements, which inject delay. Since figures 6(a) to 6(b) show that there is no remaining delay, this implies that the original timetable is relatively stable, even with these negative supplements. This means that this timetable still can absorb these delays by compensating positive supplements at later activities of the same train. Note however that this result does not include any possible headway issues yet. Figures 6(c) to 6(d) show that even with some primary delays, the buffers present in the system can absorb them and no serious delays are accumulated. Only when the necessary headway minimum times are imposed, in figures 6(e) to 6(j), we see that some red pixels result, more so in the original than in the optimised timetable. This indicates accumulated delays. Especially the *section* headway minima are causing delays especially in the original timetable. This indicates that on top of these minima, larger buffers exist in the optimised than in the original timetable in places where it matters most. The optimised timetable can 'absorb' these problems better. Figure 6(k) to 6(l) show no significant change in accumulated delays when turn-around minima are imposed. Figure 6(m) shows that in the original timetable, when also transfers are considered, some more red pixels are added to the graph while for figure 6(n) no extra red pixels seem to be generated. Note that punctuality of all transfers are considered without any weighing by passenger numbers.

Figures 7(a) to 7(f) show the punctuality analysis for a whole day geographically in terms of the probability of being less than 0 minutes late in an approaching activity. This includes transfer punctuality. Note that passenger numbers are not considered in these graphs. In these figures, 'red/green $x\%$ ' means that a green dot will appear if for all $x\%$ smallest delays, the train still arrives in time. A light green, white, light red and dark red dot mean that this is almost to not at all the case. From figures for all values of x : 50, 80, 90, the punctuality for the optimised timetable is clearly better than for the original one. Indeed, the optimised timetable compares favourable to the original timetable for $x=50$ in that figure 7(b) has the only red dot in Brussels-Central and the only non-dark green dots between Antwerpen and Puurs, while figure 7(a) has quite some red dots at some stations at the Belgian boundaries and also a long string of white or only light green dots along the axis between Luxembourg and Brussels as well as between De Panne and Deinze. When requiring 80% up to 90% of trains to arrive on time in figures 7(c) to 7(f), of course more red dots appear for both timetables. However, on average over all locations, the situation is consistently and significantly better for the optimised timetable than for the original timetable. It should also be noted that a timetable that has truly minimal expected passenger time may still generate some red dots in an OnTime report for say $x=99\%$ because this means that only with a probability of 1% trains will be late. It may not be worth to provide a buffer against the delays causing this since this may make the timetable unnecessarily inefficient in 99% of the other cases. Our objective function makes this trade-off between robustness and efficiency for passengers. Note that figures 7(a) to 7(f) show lateness as defined with zero minutes delay, because that is the threshold at which passengers start

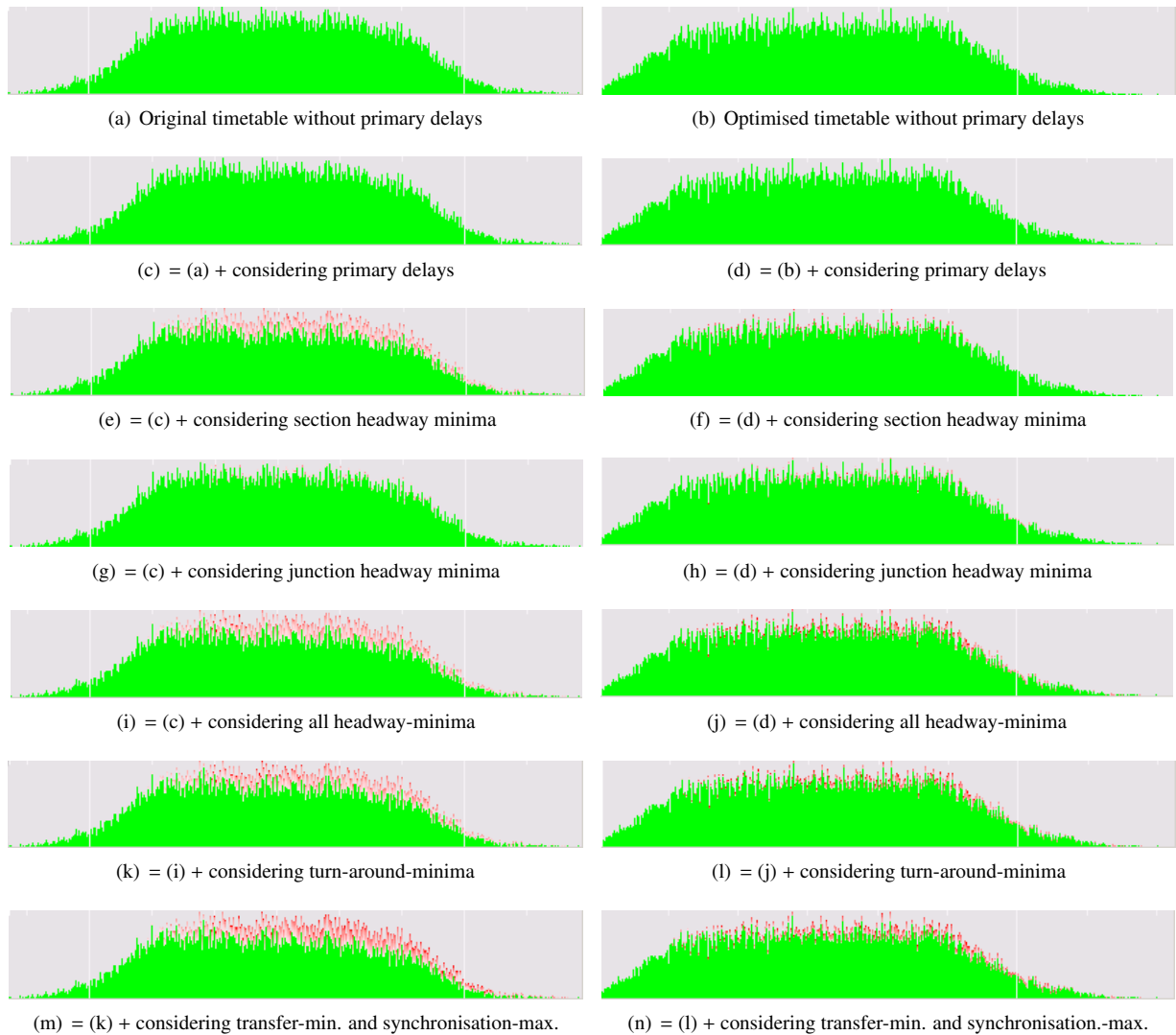


Figure 6: OnTime event punctuality analysis reports. The horizontal axis is the time of the day. The vertical axis is the amount of train events, where green pixels are on time events, light red pixels are slightly delayed events and dark red pixels are more delayed events. Passenger numbers are not considered in these graphs.

to experience delays. Train companies usually report punctuality numbers based on lateness definitions of 6 minutes late (in Belgium and Switzerland) or 3 minutes late (in The Netherlands).

We conclude that our OnTime macroscopic simulations and reports confirm that more punctuality and so more robustness against primary and secondary delays is present in the optimised timetable than in the original timetable. This robustness was achieved by minimisation of our objective function of expected passenger time. Note that our optimisation assumed an average of primary delay distribution of $a = 2\%$. With this assumption, we were able to generate a timetable that is more punctual according to OnTime simulations that use standard but train specific Infrabel primary delay distributions. This means that this $a\%$ assumption led to a timetable with both lower expected passenger journey time as well as better punctuality. When increasing the value of ‘ a ’, a generated timetable with higher punctuality against the same primary delays is expected, but expected passenger journey time may rise.

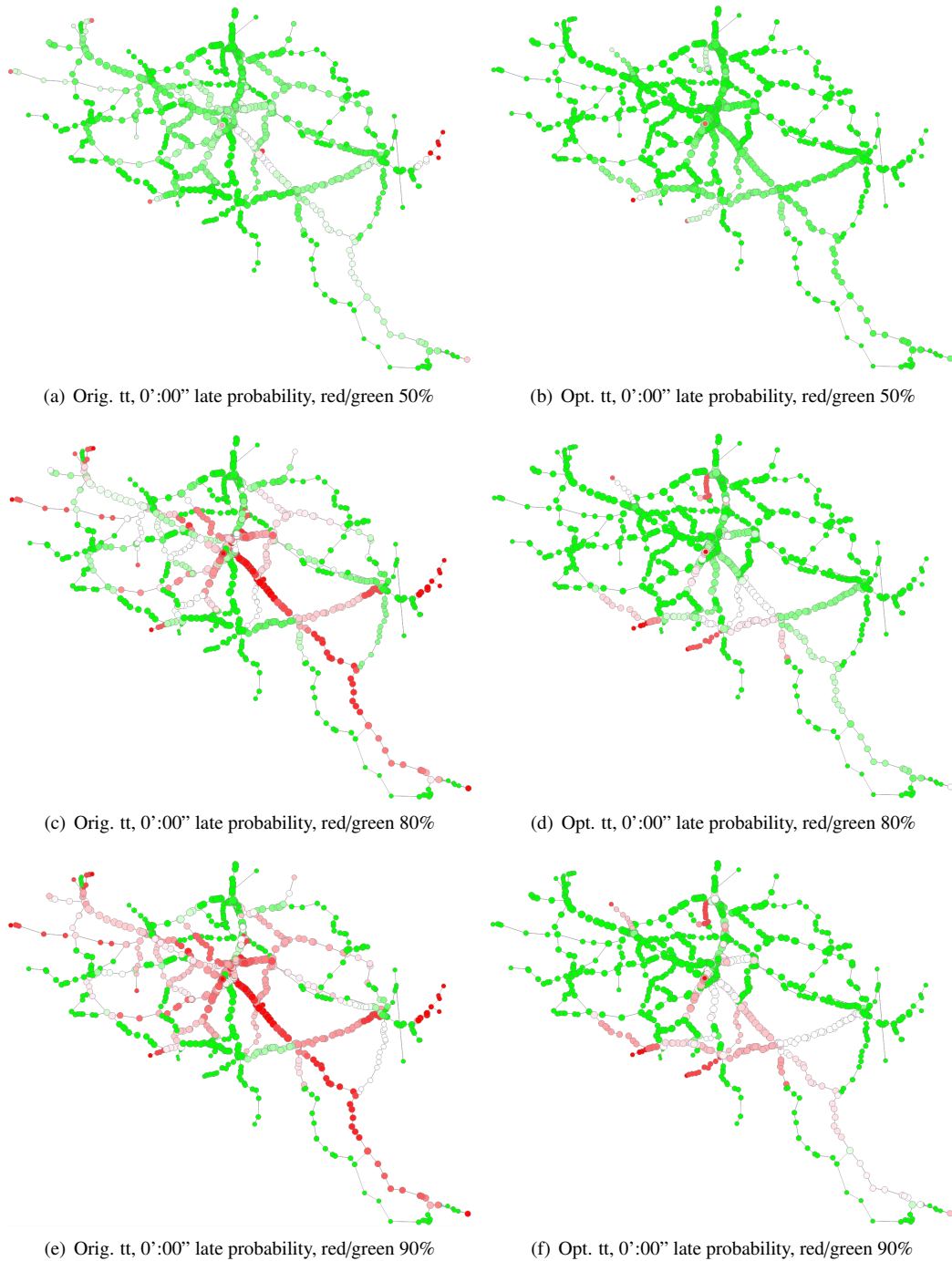


Figure 7: OnTime GeoView 0':00" late probability analysis for a full day, with consideration of transfers. For red/green $x\%$, a red dot means that in that location, less than $x\%$ of all trains arrives in time. Punctuality for the optimised timetable is clearly better than for the original one.

8.2. Check of Applicability by Microscopic Simulation

On a macroscopic level, we have decreased expected passenger journey time by creating a new timetable and validated this with a macroscopic simulator OnTime. We now want to check that we do not lose these benefits if we simulate the timetable on a *microscopic* level. We use the microscopic simulator LUKS of VIA Consulting & Development GmbH. LUKS performs delay propagation calculations on a microscopic level. It takes much more calculation time than OnTime for the same area and the microscopic routing of the trains is not available for many areas. Therefore, we are forced to restrict the simulations to a smaller area where the routings are known. The most relevant area is the bottleneck of train traffic in Belgium: the axis between Brussels-Kapelle, Brussels-Central and Brussels-Congres, because there, the most trains per track occur. This means that train knock-on effects will be the largest in this area. Additionally, this is also the area with the most passengers. Because of this, the timetable of this area affects the most passengers and requires the most attention in timetable planning.

In each of these 3 stations, only six tracks exist, all of them in underground tunnels. This means that it would be hard and expensive to increase the number of tracks in this area and so it will remain a bottleneck in the Belgian train network. In each of these 3 stations, for a chosen line, there is only one platform choice possible and also only one routing variant possible. This means that the platform and routing assignment plan is necessarily the same for the original and the optimised timetable. So the comparison of the delay propagation of the two timetables is fair and cannot be biased by a different platform and routing plan.

First of all, LUKS detected that the original timetable contained one conflict on the microscopic level but declared the optimised timetable conflict-free. Secondly, for primary delays that Infrabel deems typical, 100 simulations with LUKS of both timetables revealed that the average ratio of the realised train time to planned train time is 1.6 for the original timetable but only 1.25 for the optimised timetable. We conclude that, by separating trains on a macroscopic level, such that passengers benefit most in terms of low expected journey times, we generated a timetable that also has fewer punctuality issues - at least for the tunnel-tracks in Brussels - than the current one. We see no reason why this would not be the case in other areas.

9. Conclusions and Further Work

This paper has four main contributions. Firstly, our MILP model avoids infeasibility issues caused by artificial upper bounds on supplements as can be the case in other models. In practice our model has always returned a feasible solution. This assumes that the number of trains being scheduled does not exceed the available capacity, since naturally, all our generated timetables have to and do respect all minimal ride, dwell, transfer, headway and turn-around time rules. Secondly, our objective function results in timetables with minimised expected passenger time, meaning the total passenger travel time, including their ride, dwell and transfer activities, as well as the typical primary delays and their consequential knock-on delays in practice. This means that, on a macroscopic level, our generated timetables are both efficient and robust by construction. Thirdly, this timetable is also quickly generated. Computation times for the whole Belgian timetable are only about two hours. Fourthly, supposing primary delay distributions with an average of 2% of the minimal time of their corresponding activities, our improved timetable reduced expected passenger time for all passenger streams by 3.8% compared to the current one. We also show that the generated timetable is more punctual than the original timetable.

From our extensive modelling efforts, we also learned that while restricting the search space and using curtailed objective functions are the easy way to reduce solver times, searching the full solution space and defining an all-encompassing objective function can lead to more desirable results: a lowered risk on infeasibility, optimality and even satisfactory solver times. Our model and software tool is now available within Infrabel and we believe that together with DONS and CADANS at the Dutch railway operator NS and TAKT at the German railway operator DB, these are the only tools that are available to railway timetable practitioners that are capable to automatically generate a national timetable. In summary, this paper demonstrates that we added two important missing steps to make cyclic timetabling for passengers really useable in practice: (i) the addition of the objective function of expected passenger time in practice and (ii) the reduction of computation time by addition of well chosen additional constraints.

We envision some further useful extensions of our work. (i) The macroscopic timetable should be microscopically verified. In some stations the guaranteed 3 minutes between train-pairs on open lines may not be enough to also guarantee that all trains can be routed and platformed. Hopefully this can be done with small adaptations to train

arrival and/or departure times. (ii) In our timetabling process we adapted the timetable to the expected passenger flows. Once our timetable would be put in practice, passengers will reconsider which route they will take. To obtain an estimate of expected passenger time in practice that also considers this effect, one could add another passenger reflow pass *after* the described timetabling process and report expected passenger time again after that. A higher reduction percentage than the currently reported 3.81% is then expected, so this road should be taken. (iii) The calibration of the primary delay distributions with the total real delays measured in practice would give an even more fine-tuned timetable. (iv) Also refining the minimum transfer times, which are currently all set to 3 minutes, differentiating them by station or even - if likely platform assignments are known - by platform to platform walking time seems a task worth trying to make the presented model even more realistic. (v) The concern of *spreading in time* of alternative trains between the same source and destination has not been modelled yet. As for knock-on delays, this could be done as a soft constraint by addition of objective function terms, rather than the usual hard *frequency-arc* constraint approach.

Acknowledgements

We want to thank Infrabel for the project funding, for clearly specifying what the most desirable timetable would be for them, for their main operator and for the passengers, for delivering input data and for guiding integration in their process flow. Also, we are much obliged and thankful to the Gurobi team, for their uniquely performant MILP solver, which is currently the only one able to consistently solve our country sized model in a practical time. We thank TrafiT GmbH and VIA Consulting & Development GmbH for their great support of the simulators OnTime and LUKS.

AppendixA. Proof: Our supplement upper bounds never cause infeasibilities.

Unlike other PESP models we only restrict our supplements to be smaller than the timetable period T , formally, $\forall e \in E = E' \cup E'' : s_e < T$. We claim that this never introduces new infeasibilities compared to the model without these constraints. A proof is given here.

Let the model $M1$ be the model without any upper bounds on any supplement s_e . Let the model $M2$ be the model with an upper bound on any supplement s , so $M2$ is equal to the model $M1$ with only the additional constraints: $\forall e \in E : s < T$. Suppose that $M1$ is feasible. We then execute the following algorithm to construct a solution to $M2$ from the solution to $M1$.

- *Step 1: Modulo T Compression:* For any intra-train constraint, so $\forall e \in E' : b_e + m_e + s = e_e$, we call the solution to $M1$ for a specific $e \in E'$, $(b_{e,1}, s_{e,1}, e_{e,1})$ for which $s_{e,1} < T$ not necessarily holds. Given that $s_{e,1} \geq 0$, we can construct a solution to $M2$ as $(b_{e,2}, s_{e,2}, e_{e,2}) = (b_{e,1}, s_{e,1} - i_e \cdot T, e_{e,1} - i_e \cdot T)$, where the integer i_e is chosen so that $0 \leq s_{e,2} < T$. Said otherwise $s_{e,2} = s_{e,1} \bmod T$, so this is always possible. This solution can be applied for every $e \in E'$, performed train per train (independently), and in the chronological order of (ride, dwell) edges of a train. When an edge e is compressed by an amount $i_e \cdot T$, for all succeeding edges e of that same train, $b_{e,1}$ and $e_{e,1}$ are also decremented by $i_e \cdot T$. (Note that this maintains the satisfaction of the constraint: $b_{e,1} + m_e + s_{e,1} = e_{e,1}$.) Afterwards $(b_{e,2}, s_{e,2}, e_{e,2}) = (b_{e,1}, s_{e,1} - i_e \cdot T, e_{e,1} - i_e \cdot T)$ is applied on those succeeding edges. This is always possible since the set E' contains no cycles. Note that also for the minimum scheduling hour $h_{l_0} \cdot T$, $\forall e \in E' : h_{l_0} \cdot T \leq b_{e,2} \leq e_{e,2}$ still holds, since the beginning time $b_{e_{t,0},2} = b_{e_{t,0},1}$ of the first edge $e_{t,0}$ of each train t has not been decreased and all other times $b_{e,2}, e_{e,2}$ of that train t have not been decremented below $b_{e_{t,0},2}$. Since no single time b_e or e_e has been increased, it also still holds that $\forall e \in E' : b_{e,2} \leq e_{e,2} \leq h_{hi} \cdot T$. So we conclude that constraints for all intra-train edges in $M2$ as well as the boundary constraints to do with h_{l_0} and h_{hi} are satisfied.
- *Step 2: Inter-Train Constraint Matching:* For any inter-train constraint, so $\forall e \in E'' : b_e + m_e + s + d_e \cdot T = e_e$, we call the original solution to $M1$ for a specific $e \in E''$, $(b_{e,1}, s_{e,1}, d_{e,1}, e_{e,1})$ for which $s_1 < T$ not necessarily holds. Since step 1 above was carried out, $b_{e,2}$ and $e_{e,2}$ can now be different from $b_{e,1}$ and $e_{e,1}$. Indeed, after step 1, in general $(b_{e,2}, e_{e,2}) = (b_{e,1} - i \cdot T, e_{e,1} - j \cdot T)$ holds, with $0 \leq i \leq j$ and $b_{e,2} \leq e_{e,2}$. So the originally satisfied constraint from $M1$, $(b_{e,1}) + m_e + s_{e,1} + d_{e,1} \cdot T = (e_{e,1})$, can be converted to $(b_{e,2} + i \cdot T) + m_e + s_{e,1} + d_{e,1} \cdot T = (e_{e,2} + j \cdot T)$ or $b_{e,2} + m_e + s_{e,1} + (d_{e,1} + i - j) \cdot T = e_{e,2}$. By setting $s_{e,2} = s_{e,1} - k \cdot T$ with $k \geq 0$ and $0 \leq s_{e,2} < T$, so $s_{e,2} = s_{e,1}$

mod T , we get $b_{e,2} + m_e + (s_{e,2} + k \cdot T) + (d_{e,1} + i - j) \cdot T = e_{e,2}$ or $b_{e,2} + m_e + s_{e,2} + (d_{e,1} + i - j + k) \cdot T = e_{e,2}$. Next, we set $d_{e,2} = (d_{e,1} + i - j + k) \cdot T$ such that $b_{e,2} + m_e + s_{e,2} + d_{e,2} \cdot T = e_{e,2}$ results, which delivers a solution $(b_{e,2}, s_{e,2}, d_{e,2}, e_{e,2})$ for any edge $e \in E''$ for M2.

Note that $i - j + k$ can be both negative or positive so $d_{e,2}$ can be smaller, equal or larger than $d_{e,1}$, which initially can raise the concern that a larger range for the d_e variables would be needed for M2 compared to for M1 to be able to guarantee feasibility. However, since step 2 does not change any values $b_{e,2}$ or $e_{e,2}$ compared to step 1, all these values are still in the $[h_{lo} \cdot T, h_{hi} \cdot T]$ window as proven in step 1. Consequently, they are still at most $(h_{hi} - h_{lo}) \cdot T$ apart. This means that the range of d_e can be kept the same for M2 as for M1, more specifically: $\forall e \in E'' : d_e \in [h_{lo} - h_{hi}, h_{hi} - h_{lo}]$.

As for the cycle constraints, these are linear combinations of the other constraints, so if the other constraints will not cause infeasibilities, neither will the cycle constraints.

□

Appendix B. Conjecture: Activity lower bounds are the only potential causes of infeasibilities.

It is clear that the following situations cause our model to become infeasible.

1. *Lower bounds on ride and dwell times per train:* When the sum of the minima of subsequent ride and dwell actions of a train sum up to S and when the time window $[h_{lo} \cdot T, h_{hi} \cdot T]$ to schedule trains in has length $(h_{hi} - h_{lo}) \cdot T < S$, the model will be infeasible. These problems are caused by *lack of time*.
2. *Lower bounds on headway times per track:* When more than twenty trains are required to be planned on a single infrastructure resource in 60 minutes and also a lower bound of 3 minutes headway time is imposed, one is obviously asking the impossible. The model will then be infeasible. These problems are caused by *lack of capacity* which can also be seen as *lack of space*.

We think that there are no other causes of infeasibilities in our model. Note that the manually created timetable that we read in should also avoid these two causes of infeasibilities. If this is the case, a model will be created that has none of these supplement lower bound infeasibility issues either. Firstly, to avoid the infeasibilities of type 1, we derive the makespan of the original timetable, round it up to an integer multiple of hours and then add one extra hour to it. This forms the time window in which we try to schedule all trains for the optimised timetable. Secondly, in practice, we see that in the original timetables, no more than 17 trains per hour have been planned on any track, as such avoiding infeasibilities of type 2. This means that we will not have infeasibilities of that type in our model either. So, the guarantee of our model being protected against infeasibilities of type 1 and type 2 is controlled by the ‘sensitivity’ of the original input timetable. More particularly, in the original timetable, the sum of ride, dwell minima for every train should be at most the timetable’s makespan and the sum of the headway minima should be at most the timetable’s period. One could check for these conditions and report type 1 and type 2 infeasibilities even before trying to solve the model. If there are indeed no other infeasibility problems apart from type 1 and 2, this would then mean that the model only has to be solved when feasibility is guaranteed.

References

- Bešinović, N., Quaglietta, E., Goverde, R., 2013. [A simulation-based optimization approach for the calibration of dynamic train speed profiles](#). Journal of Rail Transport Planning and Management 3, 126–136.
- Bešinović, N., Roberti, R., Quaglietta, E., Cacchiani, V., Toth, P., Goverde, R., Mar. 2015. Micro-macro approach to robust timetabling. Proceedings of 6th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailTokyo2015, March 23-26, Tokyo, Japan., 1–21.
- Bourachot, J., 1986. Computer-aided planning of traffic in large stations by means of the afaig model. Rail Int, 2–18.
- Burdett, R., Kozan, E., 2010. [A disjunctive graph model and framework for constructing new train schedules](#). European Journal of Operational Research. 200, 85–98.
- Cacchiani, V., Caprara, A., Fischetti, M., 2012. [A Lagrangian Heuristic for Robustness, with an Application to Train Timetabling](#). Transportation Science, 124–133.
- Cacchiani, V., Caprara, A., Toth, P., 2010. [Scheduling extra freight trains on railway networks](#). Transportation Research Part B: Methodological 44(2), 215–231.
- Cacchiani, V., Toth, P., Nov 2012. [Nominal and robust train timetabling problems](#). Nominal and robust train timetabling problems. European Journal of Operational Research. 219, 727–737.

- Caimi, G., 2009. Algorithmic decision support for trains scheduling in a large and highly utilised railway network. Ph.D. thesis, ETH Zurich.
- Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., Zenkhusen, R., 2011. [A New Resource-Constrained Multicommodity Flow Model for Conflict-Free Train Routing and Scheduling](#). *Transportation Science*. 45 (2), 212–227.
- Caimi, G., Hermann, T., Burkolter, D., Chudak, F., Laumanns, M., Nov. 2009. [Design of a railway scheduling model for dense services](#). *Networks and Spatial Economics*. 9 (1), 25–46.
- Caimi, G., Herrmann, T., Burkolter, D., 2004. [Finding delay-tolerant train routing through stations](#). In: Fleuren, H., den Hertog, D., Kort, P. (Eds.), *Operations Research Proceedings*. Springer, Heidelberg, pp. 136–143.
- Caprara, A., Galli, L., Toth, P., 2011a. [Solution of the Train Platforming Problem](#). *Transportation Science* 45(2) (2), 246–257.
- Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 2007. *Passenger Railway Optimization*. *Handbooks in Operations Research and Management Science*. Vol. 14. Elsevier, Amsterdam.
- Caprara, A., Kroon, L., Toth, P., 2011b. *Optimization Problems in Passenger Railway Systems*. *Wiley Encyclopedia of Operations Research and Management Science* 6, 3896–3905.
- Carey, M., Carville, S., 2003. [Scheduling and platforming trains at busy complex stations](#). *Transportation Research Part A* 37(3) (1), 195–224.
- Daamen, W., Goverde, R., Hansen, I., 2009. [Non-Discriminatory Automatic Registration of Knock-On Train Delays](#). *Networks and Spatial Economics* 9 (1), 47–61.
- De Fabris, S., Longo, G., Medeossi, G., Pesenti, R., May 2013. Application and validation of a timetabling algorithm to a large italian network. *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013*, May 13-15, Copenhagen, Denmark.
- Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2011. [Defining Robustness of a Railway Timetable](#). *Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011*, February 16-18, Rome, Italy.
- Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2014. [Improving the robustness in railway station areas](#). *European Journal of Operational Research* 235, 276–286.
- Fischetti, M., Monaci, M., 2009. [Light Robustness](#). In: Ahuja, R. K., Möhring, R. H., Zarogialis, C. D. (Eds.), *Robust and Online Large-Scale Optimization, Models and Techniques for Transportation Systems*. Vol. LNCS 5868. Springer, pp. 61–84.
- Ford, R., Haydock, D., 1992. [Signalling and timetabling](#). In: Harris, N., Godward, E. (Eds.), *Planning Passenger Railways: A Handbook*, Chapter 11. *Transportation Publishing Company*, Glossop, pp. 117–129.
- Galli, L., Stiller, S., Oct 2010. [Strong formulations for the Multi-module PESP and a quadratic algorithm for Graphical Diophantine Equation Systems](#). *Proceeding EAS'10 of the 18th annual European conference on Algorithms: Part I*, 338–349.
- Goverde, R., 1998a. [Optimal Scheduling of Connections in Railway Systems](#). Paper presented at the 8th WTCR, Antwerp, Belgium.
- Goverde, R., 1998b. [The Max-Plus Algebra approach to railway timetable design](#). In: Mellitt, B., Hill, R., Allan, J., Sciuotto, G., Brebbia, C. (Eds.), *Computers in Railways VI*. WIT Press, Ashurst, pp. 339–350.
- Goverde, R., 1999. [Improving Punctuality and Transfer Reliability by Railway Timetable Optimization](#). *Proceedings of 5th TRAIL Annual Congress*, TRAIL Research School, Delft, The Netherlands.
- Goverde, R., Hansen, I., 2000. [TNV-Prepare: Analysis of Dutch railway operations based on train detection data](#). *International conference on computers in railways VII*, 779–788.
- Großmann, P., 2011. [Polynomial Reduction from PESP to SAT](#). Tech. rep., TU Dresden.
- Großmann, P., Hölldobler, S., Manthey, N., Nachtigall, K., Opitz, J., Steinke, P., 2012. [Solving Periodic Event Scheduling Problems with SAT](#). In: Jiang, H., Wei, D., Moonis, A., Xindong, W. (Eds.), *Advanced Research in Applied Artificial Intelligence*. Vol. LNAI 7345. Springer-Verlag Berlin Heidelberg, p. 166–175.
- Holroyd, E. M., Scraggs, D. A., 1966. [Waiting Times for Buses in Central London](#). *Traffic Engineering Control*. 8, 158–160.
- Huisman, D., Kroon, L., Lentink, R., Vromans, M., 2005. [Operations research in passenger railway transportation](#). *Statistica Neerlandica* 59 (4), 467–497.
- Kroon, L., Dekker, R., Gábor, M., Helmrich, M. R., Vromans, M., 2006. [Stochastic Improvement of Cyclic Railway Timetables](#). Tech. rep., Erasmus Research Institute of Management (ERIM).
- Kroon, L., Dekker, R., Vromans, M., 2007. [Cyclic Railway Timetabling: A Stochastic Optimization Approach](#). *Algorithmic Methods for Railway Optimization*. *Lecture Notes in Computer Science*, 41–66.
- Kroon, L., Huisman, D., Abbink, E., Fioole, P.-J., Fischetti, M., Maróti, G., Schrijver, A., Ybema, R., 2009. [The New Dutch Timetable: The OR Revolution](#). *Interfaces* 39, 6–17.
- Kümmeling, M., Großmann, P., Nachtigall, K., Opitz, J., Weiß, R., 2013. [The State-of-the-art Realization of Automatic Railway Timetable Computation](#). In: *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013)*, December 2-4, Dresden, Germany. pp. 1–10.
- Labermeier, H., 2013. [On the Dynamic of Primary and Secondary Delay](#). *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013*, May 13-15, Copenhagen, Denmark.
- Lie, M., 2006. [Optimization of the Timetable of the Dutch Railways](#). Master's thesis, Vrije Universiteit Amsterdam.
- Liebchen, C., 2006. [Periodic Timetable Optimization in Public Transport](#). Ph.D. thesis, Technische Universität Berlin.
- Liebchen, C., 2007. [Periodic Timetable Optimization in Public Transport](#). *Operations Research Proceedings 2006*, 29–36.
- Liebchen, C., Lübbecke, M., Möhring, R., Stiller, S., 2009. [The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications](#). In: Ahuja, R. K., Möhring, R. H., Zarogialis, C. D. (Eds.), *Robust and Online Large-Scale Optimization, Models and Techniques for Transportation Systems*. Vol. LNCS 5868. Springer, pp. 1–27.
- Liu, S., Kozan, E., 2011. [Scheduling Trains with Priorities: A No-Wait Blocking Parallel- Machine Job-Shop Scheduling Model](#). *Transportation Science*. 45 (2), 175–198.
- Nachtigall, K., 1994. [A branch-and-cut approach for Periodic Network Programming](#). Tech. rep., Hildesheimer Informatik-Berichte.
- Nachtigall, K., 1996. [Periodic network optimization with different arc frequencies](#). *Discrete Applied Mathematics* 69, 1–17.
- Nachtigall, K., Opitz, J., 2008. [Solving Periodic Timetable Optimisation Problems by Modulo Simplex Calculations](#). In: Fischetti, M., Widmayer, P. (Eds.), *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*. Vol. 9 of *OpenAccess*

- Series in Informatics (OASIs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.
- Nachtigall, K., Voget, S., 1997. [Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks](#). *European Journal of Operational Research* 103 (3), 610–627.
- Odiijk, M., 1996. [A constraint generation algorithm for the construction of periodic railway timetables](#). *Transportation Research Part B* 30 (6), 455–464.
- Osuna, E. E., Newell, G. F., 1972. Control Strategies for an Idealized Public Transportation System. *Transportation Science*. 6 (1), 57–72.
- Peeters, L., 2003. [Cyclic Railway Timetable Optimization](#). Ph.D. thesis, Erasmus Research Institute of Management (ERIM), Rotterdam.
- Schlechte, T., Borndorfer, R., Erol, B., Graffagnino, T., Swarat, E., Nov. 2011. [Micro-macro transformation of railway networks](#). *Journal of Rail Transport Planning and Management*. 1 (1), 38–48.
- Schöbel, A., Kratz, A., 2009. [A Bicriteria Approach for Robust Timetabling](#). In: Ahuja, R. K., Möhring, R. H., Zarogialis, C. D. (Eds.), *Robust and Online Large-Scale Optimization, Models and Techniques for Transportation Systems*. Vol. LNCS 5868. Springer, pp. 119–144.
- Schrijver, A., Steenbeek, A., 1993. [Spoorwegdienstregelingontwikkeling \(Timetable Construction\)](#). Technical Report, CWI Center for Mathematics and Computer Science, Amsterdam.
- Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2011a. [Automated, Passenger Time Optimal, Robust Timetabling, using Mixed Integer Programming](#). In: *Proceedings of the 1st International Workshop on High-Speed and Intercity Railways (IWHIR)*, Hong Kong. Vol. 2. pp. 87–92.
- Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., Feb. 2011b. [Deriving all Passenger Flows in a Railway Network from Ticket Sales Data](#). *Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011*, February 16–18, Rome, Italy.
- Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., Dec. 2013a. [A Passenger Knock-On Delay Model for Timetable Optimisation](#). In: *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013)*, December 2–4, Dresden, Germany. pp. 1–10.
- Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., May 2013b. [Expected Passenger Travel Time as Objective Function for Train Schedule Optimization](#). *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013*, May 13–15, Copenhagen, Denmark.
- Sels, P., Vansteenwegen, P., Dewilde, T., Cattrysse, D., Waquet, B., Joubert, A., Mar. 2014. [The train platforming problem: The infrastructure management company perspective](#). *Transportation Research Part B* 61, 55–72.
- Semet, Y., Schoenauer, M., 2005. [An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling](#). *Evolutionary Computation* 3, 2752–2759.
- Serafini, P., Ukovich, W., 1989. [A Mathematical Model for Periodic Scheduling Problems](#). *SIAM Journal on Discrete Mathematics* 2, 550–581.
- Sparing, D., Goverde, R. M., Hansen, I. A., 2013. [An Optimization Model for Simultaneous Periodic Timetable Generation and Stability Analysis](#). *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013*, May 13–15, Copenhagen, Denmark.
- Sun, L., Jin, J. G., Lee, D. H., Axhausen, K. W., Erath, A., 2014. [Demand-driven timetable design for metro services](#). *Transportation Research Part C: Emerging Technologies* 46, 284–299.
- UIC, 2004. UIC CODE 406: Capacity. Tech. Rep. 1, Union Internationale des Chemins de Fer (UIC), URL <http://www.uic.org/>.
- Vansteenwegen, P., Van Oudheusden, D., 2006. [Developing Railway Timetables that Guarantee a Better Service](#). *European Journal of Operational Research* 173(1) (1), 337–350.
- Vansteenwegen, P., Van Oudheusden, D., 2007. [Decreasing the passenger waiting time for an intercity rail network](#). *Transportation Research Part B* 41(4) (4), 478–492.
- Watson, R., Nov 2008. [Train Planning in a Fragmented Railway - A British Perspective](#). Ph.D. thesis, Loughborough University.
- Welding, P. I., 1957. The Instability of Close Interval Service. *Operational Research Society*. 8 (3), 133–148.
- Wendler, E., May 2007. [The scheduled waiting time on railway lines](#). *Transportation Research Part B*. 41 (1), 148–158.
- Yuan, J., 2006. [Stochastic modelling of train delays and delay propagation in stations](#). Ph.D. thesis, Technical University of Delft.
- Zhao, J., Frumin, M., Wilson, N., Zhao, Z., 2013. Unified Estimator for Excess Journey Time under Heterogeneous Passenger Incidence Behaviour using Smartcard Data. *Transportation Research, Part C*. 34, 70–88.
- Zwaneveld, P. J., Kroon, L., Romeijn, H., Salomon, M., Dauzère-Perès, S., Van Hoesel, S. P. M., Ambergen, H. W., 1996. [Routing Trains Through Railway Stations: Model Formulation and Algorithms](#). *Transportation Science* 30 (1), 181–194.