The Train Platforming Problem: the Infrastructure Management Company Perspective

P. Sels[a,b,c,*], P. Vansteenwegen[a], T. Dewilde[a], D. Cattrysse[a], B. Waquet[d], A. Joubert[d]

[a]*Katholieke Universiteit Leuven, Centre for Industrial Management/Traffic & Infrastructure,*
Celestijnenlaan 300a, 3001 Heverlee, Belgium
[b]*Logically Yours BVBA,*
Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium
[c]*Infrabel, Department of Rail Access,*
Frankrijkstraat 91, 1070 Brussels, Belgium
[d]*ICTRA, B-Holding,*
Tweestationsstraat 82, 1070 Brussels, Belgium

Abstract

If railway companies ask for station capacity numbers, their underlying question is in fact one about the platformability of extra trains. Train platformability depends not only on the infrastructure, buffer times, and the desired departure and arrival times of the trains, but also on route durations, which depend on train speeds and lengths, as well as on conflicts between routes at any given time. We consider all these factors in this paper. We assume a current train set and a future one, where the second is based on the expected traffic increase through the station considered. The platforming problem is about assigning a platform to each train, together with suitable in- and out-routes. Route choices lead to different route durations and imply different in-route-begin and out-route-end times. Our module platforms the maximum possible weighted sum of trains in the current and future train set. The resulting number of trains can be seen as the *realistic capacity consumption* of the schedule. Our goal function allows for current trains to be preferably allocated to their current platforms.

Our module is able to deal with real stations and train sets in a few seconds and has been fully integrated by Infrabel, the Belgian Infrastructure Management Company, in their application called Ocapi, which is now used to platform existing and projected train sets and to determine the capacity consumption.

Keywords: Train Platforming Problem, Mixed Integer Linear Programming, Train Station Capacity

[*] Corresponding author
Email address: sels.peter@gmail.com (P. Sels)

1. INTRODUCTION

Over the years, train operator companies require more and more trains to be added on the existing infrastructure. For infrastructure management companies, it is essential to determine a feasible platforming plan for all stations and junctions. Each of these train platforming problems (TPPs) deals with assigning all trains in a station to the available platforms, in a way to avoid conflicts on the platforms as well as on the routes from an incoming and to an outgoing line. Often, this platforming process is still done manually. This means it is error prone, takes a lot of time and the result is not optimal. At Infrabel, the Belgian Railway Infrastructure Manager, both cases of wrongly judging two routes as being in conflict and cases of wrongly considering two routes as not being in conflict, have been noted. De Luca Cardillo (1998) illustrates how in a particular case, platforming 242 trains in a station with 16 platforms requires 15 working days for an expert planner. In order to perform this platforming task better as well as faster, we want to develop software to automate this process.

In this paper, we focus on automatically platforming as many trains as possible from two train sets: an already operational set (current set) and a future set, based on the expected traffic increase. The way that infrastructure management companies work today, is that they first construct a timetable and then try to platform all planned trains in the stations they visit, respecting the planned platform time. So, for our platforming problem, we consider the timetable to be given. We don't allow automatic changes in platform arrival and departure times. Trains that cannot be platformed will be put on a *fictive platform* and arrive or depart from there via *fictive routes*.

Our platforming solution is a Mixed Integer Linear Programming MILP model that extends the model of Billionnet (2003) with consideration of route duration differences, as well as with the fictive route concept. Thanks to the MILP method, the fictive platform concept and our simple goal function, (i) our model always returns a usable solution, (ii) we can and do list which trains cannot be platformed, (iii) the gap also tells us how far our solution is removed from the optimal one, in meaningful units of number of trains that could not be platformed. Moreover, for realistic traffic, we obtain the optimal solution in very low solver times and our system is integrated and used at Infrabel. Also, our goal function can be tuned to prefer a solution close to the current one or otherwise optimize more progressively.

In the next section we summarize the existing literature about the Train Platforming Problem. In section 3, we first situate our own TPP version amongst the TPP in the literature. Our detailed mixed integer linear programming (MILP) model is described in section 4. The last sections discuss the results, summarize the conclusions and hint at some further work.

2. LITERATURE REVIEW

We refer to Caprara et al. (2007), Lusby et al. (2011a), Caprara et al. (2011b) and Caprara et al. (2011a) for some recent surveys about the Train Platforming Problem (TPP). An earlier survey, with a larger scope of problems, but also describing train platforming problems, is due to Cordeau et al. (1998).

An important categorization mentioned by Zwaneveld et al. (1996), Cordeau et al. (1998) and Lusby et al. (2011a) is whether the TPP considers a system which is intended for use at the strategic (S), tactical (T) or operational (O) level. The strategic level is concerned with platforming and possibly also routes for capacity estimation. At this level, one should evaluate the platform feasibility for a range of potential timetables and possibly also infrastructure change options. The tactical level of the TPP tries to determine platforms for the timetable that is already decided. The importance of deciding on routes becomes bigger here. The operational level TPP decides on platforms and routes in real time.

2.1. Strategic Level TPP

Zwaneveld et al. (1996); Zwaneveld (1997, 2001) and Kroon et al. (1997) consider the strategic level TPP where they allow multiple route variants for each direction-platform pair. They use as input, the current timetable with a set of possible time shifts in minutes on arrival and departure time couples. They construct triplets, each consisting of a train-platform, a time interval with a time shift and each dominating route (inbound, outbound or complete (for passing trains)). They pre-calculate if pairs of the mentioned triplets conflict with each other or not. The result is a MILP model representing a *node packing problem*. Reformulating some constraints as clique inequalities, they significantly reduce the number of constraints and also solution time. The goal is to maximize the number of trains platformed. All platforming experiments for the timetable without time shift variations could be solved in at most 85 seconds. Experiments including time shift variations required at most 400 seconds on a 1996 SUN LX workstation running CPLEX 2.1. The system was integrated in STATIONS, software used at the Dutch railways for timetable evaluation and capacity estimation. Some variants of the problem solved in Zwaneveld et al. (1996) are reformulated in Kroon et al. (1997) as fixed interval scheduling problems.

Delorme et al. (2001) use *constraint programming* and its constraint propagation technique to calculate the capacity of a railway subnetwork. The area studied is the Pierrefitte-Gonesse area north of Paris. Four combinations of mixed speed, high speed, IC and freight train traffic are generated and their effect on capacity consumption is reported. Even though calculation times, for the junction described, are in between 1000 and 10000 seconds, this approach is integrated into the capacity evaluation tool RECIFE at Sociéte Nationale des Chemins de fers Français (SNCF).

2.2. Tactical Level TPP

De Luca Cardillo (1998) used a graph coloring formulation and an efficient heuristic they call *Conflict-Direct Backtracking*, to quickly solve the feasibility problem at the tactical level. Between two endpoints, only a single route variant is considered. Additionally, a list of incompatible routes is used. They solve 5 out of the 6 real stations considered, in less than a second, one in 115 seconds.

Billionnet (2003) uses integer programming to first solve the same problem as De Luca Cardillo (1998), but also considers various goal functions, like maximum use of some platforms. He uses 20 randomly generated station infrastructure graphs and train sets. Solving times on these instances are from below one second to 80 seconds and one case had no solution in 1200 seconds. These are further reduced by an alternative model formulation and the addition of clique cuts. For the one reported real station of Abatone, with 5 platforms and 41 trains to platform, he obtains a very low solver time of 0.01 up to 0.03 seconds.

Caprara (2010) and Caprara et al. (2011a) also treat the tactical level TPP but then for the case of multiple routes where platform times can also vary in a discrete interval. They minimize a goal function, which is a quadratic function of among others: deviation from preferred platforms and deviation from platform times. Other terms concerning *platform choice* are related to the number of used platforms, used but not preferred platforms, never preferred, simultaneously used and dummy platforms. There are also *platforming quality* related terms like the total number of time shifts used, the number of dynamic conflicts used, the number of trains assigned to a non preference platform and the number of trains assigned to a dummy platform. They linearize the goal function and use clique inequalities which together reduce solver times. Thanks to the dummy platforms, the system always returns solutions and may prefer to not platform a train in order to be able to deviate less from the given timetable for the platformed trains. This is in contrast to for example, Zwaneveld et al. (1996) who do not penalize the deviations from platform times.

Carey (1994a,b), Carey and Lockwood (1995) and Carey and Carville (2000, 2003) treat the intermediate case of variable platform times with a unique route per line-platform combination. They don't intend to set up a methodology that can find a fully optimal solution, which they claim would be too complex and require too much computation time. Instead, their approach is to mimic the methods that planners use. This means for example that their tool also plans train by train sequentially, in decreasing order of importance. They capture other human planner concerns in an explicit lexicographic goal function that contains the three cost components, in this lexicographical order of decreasing importance: adjustment or knock-on costs, platform desirability costs and platform occupation (obstruction) costs. Even though this method does not obtain an optimal solution, nor does it intend to, it represents current human planner practices in a more formal way. Their results show that about 20% of the trains are allocated to a different platform than when allocated by the human planners. They mention that this is due to the fact that ”the platform preference data was elicited from discussions with individual schedulers, and are estimates of largely unwritten, changing and somewhat discretionary rules”. In a timetable which was finalized by human planners, Carey and Carville (2003) mention their tools found headways smaller than the *specified* minimum headways, yet still above some *absolute* minimum headway. Carey and Carvilles tools also, safely, never reduce the headway time below the *absolute* minimum headway. They mention their results are obtained quickly. Carey and Crawford (2005) spatially extend this work with the consideration of whole *corridors of stations* in the combined problem of scheduling and platforming.

Ghoseiri et al. (2004) use a multi-objective goal function composed of the passenger satisfaction criterion of lowering travel time, which they call *effectiveness* and the railway company interest of reducing fuel consumption, which they call *efficiency*.

2.3. Operational Level TPP

Because of the real time requirements and the higher level of detail at this level, computationally, the TPP at the operational level is the most demanding one. Chakroborty and Vikram (2008) treat this case, using a MILP model with a goal function that minimizes, weighted by train priorities, the summed inconvenience caused by (1) delay of trains waiting for a platform, (2) allocation of non-preferred platforms and (3) last minute reassignment of platforms. The largest problems solved optimally within 10 minutes is one with 110 trains and 9 platforms on a time horizon of 2 hours and where arrival times of trains are known at about an hour in advance.

Lusby et al. (2011a, 2013) also discuss the TPP research at the operational level. They argue that for this level, constraints based on (pairs of) paths as sequences of separate sections, which they call *resources*, are better replaced by constraints based on separate resources themselves. They report solution times in terms of iterations, but no absolute times in seconds are given.

Miao et al. (2012) consider a form of *stability* as main optimization criterium for their TPP. When the platform arrival or/and departure times of the given timetable cannot result in a feasible platform, they succeed in finding a feasible platform with minimal change to these platform times. They both minimize the changes the TPP requires for the timetable as well as the chance on platform conflicts in real time. The technique they use is ant colony optimization.

2.4. Summary

We summarize the characteristics from the TPP related literature in Table A.1 and their applications and obtained results in Table A.2 in AppendixA. Table cells without a value occur when we could not find any mention of the corresponding characteristic in the specific publication nor from correspondence with the authors. We also already contrast our own research with the state-of-the-art by including our work in this table. In this paper, we focus on the strategic and tactical level. The next section will describe our work in more detail and compare it to the previous work we described.

Further, more elaborate studies considering the TPP are the PhD dissertations of Burkolter (2005), Herrmann (2006), Lusby (2008), Galli (2009) and Caimi (2009) and the Masters thesis of Fuchsberger (2007).

3. OUR TPP VERSION

From the previous section, it is clear that there are many variants of the TPP problem. In this section, we describe the variant we consider and the assumptions we make. In section 4, we will derive our mathematical model for it.

3.1. TPP Level: Strategic and Tactical

The train platforming problems Infrabel wants to solve are initially at the strategic level, where changed or/and increased train traffic raises the question whether all affected stations have sufficient capacity to absorb this new traffic. Because this is a phase where many timetables are explored, Infrabel needs a quick and optimal platforming solution and ideally a fully automated one. Also, during actual timetabling, a verification of the existing platforming and routing solution, as well as an automatic improvement of it is needed. We cover both types of uses described here, strategic and tactical, in our model developed below.

3.2. Fixed Platform Times, variable Junction Times

For capacity studies of a junction (station area, including platforms and a grid on each side) at the strategical level, it is fine to assume that *platform* entry and exit times are fixed and *junction* entry and exit times can always be derived by subtracting respectively adding route durations of the chosen route. Indeed, in capacity studies of just one junction, junction entry and exit times should not be subject to any line or previous nor next station constraints.

In contrast, for the tactical level, the timetable also imposes constraints on neighbouring junctions and as such also on junction in and out times of the junction considered. Hard constraints for platform times on the inside of the routing grids, and hard constraints on junction times on the outer side of the routing grids, may then render the TPP infeasible. We believe it is best to allow some variability on both, but this is only useful if one can make a sensible quantitative decision about the value of adaptations. Lacking such a valuation method for now, we assume platform times as fixed and junction in and out times as freely adaptable.

3.3. Conflict Modeling Based on Train-Pairs including a Conflict Filter

Contrary to Lusby et al. (2011a, 2013) but like Zwaneveld et al. (1996); Zwaneveld (1997, 2001) and Caprara et al. (2007), we define conflicts as existing between pairs of trains. When deciding which train pair occupations overlap, there are three cases to consider.

- Some occupation time interval pairs are so far removed from each other in time that they will *never overlap* in time, independent of the platforms and routes chosen for them. We will not need to model constraints for these pairs.
- Other interval pairs *always overlap*, independent of the platform and routes chosen. We will model constraints for these pairs.
- Interval pairs of a third type can *potentially overlap* depending on the platform and routes chosen. This is the case because some platforms and routings take more time to traverse than others. Platform times are fixed, but as explained in detail later, due to platform lengths, train lengths and finite train speeds, actual platform usage intervals are extended with some part of the total routing times, and these time intervals are thus route choice dependent. We will model constraints for these pairs. The choice of platform and routes will be variables in these constraints. As long as the constraint contains these variables, it forbids all *potential conflicts*. Such a constraint forbids the potentially wrong choice for platforms and routings that would lead to time overlap and thus to an actual conflict. When, during solving of the model, all actual platform and routing choice constants are substituted for these variables, the resulting constraints forbid all *actual conflicts*.

Just like for pairs of (same) route-extended platform occupations, for pairs of (dependent) route occupations, we apply a similar strategy of setting up constraints.

3.4. Level of Detail in Timing and Continuous instead of Discretized Time

To choose routings properly, it is important to consider train speed, train length and their influence on route duration in the platforming problem. This is so, because the choice of a route for a train will influence its timing and whether the train is in conflict with other trains on dependent routes or not. Indeed, just supposing that all routes would take an equal time is wrong and will introduce inaccuracies in the planning, which will lead to route conflicts in practice. This simplification is present in De Luca Cardillo (1998) and Billionnet (2003), but we now take this important consideration into account.

However, contrary to Zwaneveld et al. (1996); Zwaneveld (1997, 2001), Caprara (2010); Caprara et al. (2011a) and Lusby et al. (2011b), but similar to De Luca Cardillo (1998) and Billionnet (2003), we assume that each section can be taken at maximum section speed, independent of the trains' speed on the previous sections. We believe that there is enough slack time margin in our timetables during which a train can decelerate on the open track before entering a station, to reduce its speed to what is required on a station track. (Routing or platform sections often have a speed limit of 40km per hour.) We also believe similar margins exist on the open track after leaving a station to get up to the maximum speed there. The above assumption also avoids the tree of discrete possibilities that Zwaneveld et al. (1996); Zwaneveld (1997, 2001), Caprara (2010); Caprara et al. (2011a) and Lusby et al. (2011b) set up and the discretization of time that is inherent to such a tree. Our platformer does not shift times because we and Infrabel consider time shifting to be part of the macroscopic plan. In the macroscopic plan, enough slack on open lines before respectively after the station should be provided so that slowing down to, respectively speeding up from, station maximum speed restrictions is possible. The macroscopic plan should also provide timing such that all trains can be platformed in each station. This may result in some iterations over macroscopic timetabling and platforming.

3.5. One Route Variant

As for route variants, for the time being, for most stations, Infrabel could only supply us with *default routes*. Consequently, we also currently assume that there is at most one route between each incoming (outgoing) line and each platform. Even so, we maintain both a platform choice variable as well as a route choice variable. The fact that a choice of platform directly implies the single route to/from the fixed line will be expressed by a separate connectivity-constraint. This method allows us to expand towards multiple routes more easily later. The option of generating multiple routes for each line-platform pair from the raw infrastructure data could also be taken. Dewilde et al. (2013) have done this for the station area of Brussels, using a MILP based approach to generate, simultaneously, route choices and corresponding schedule times with the objective of obtaining more robust schedules for stations. However, currently, not enough data is available to apply this to other station areas.

3.6. *Mixed Integer Linear Programming*

We choose MILP, since we like its flexibility and also appreciate its clear answer as to how far our results are away from the optimal one. Other heuristic algorithms usually do not provide these benefits. Now that we presented what version of TPP we want to tackle, we can also explain in more detail here what we mean by feasibility and optimality of our MILP model and the solver methods we use.

3.6.1. *Feasibility*

We will try to platform the currently planned trains, but also some expected future trains. When planning too many new trains, without any extra measures, it is possible that the TPP becomes infeasible. This would mean no solution is returned. To avoid this, we defined, next to the real platforms and routes, a fictive platform. This was also done by Joubert in the original MILP model for Infrabel in 2008. We also added fictive IN and OUT routes that allow any number of occupations and movements assigned to it. This guarantees feasibility of the model. The fictive platform will have all trains assigned to it that, due to capacity problems or dependent route constraints, cannot be assigned to real platforms. Caprara et al. (2007) already described and used this same concept as the *dummy platform*. They define an infinite amount of them, but we need just one, since we allow occupations on the fictive platform to overlap. Any of our fictive routes too can hold any number of movements overlapping in time. Also, no potential conflicts can arise between a fictive route and a real one, let alone between two fictive ones. The result is that our model is always feasible.

3.6.2. *Optimality*

As mentioned, we define a current train set and a supplementary one. The second set, on the tactical level, could represent the actual future set of trains or, on the strategic level, could be incrementally enlarged with the aim to determine practical station capacity. Zwaneveld et al. (1996); Zwaneveld (1997, 2001) and Kroon et al. (1997) also used this approach. Our goal function, both for the initial and for the supplementary traffic, consists of two types of cost terms. The first corresponds to a penalty for changing an occupation from a real to another real platform. The second type represents a, usually higher, penalty for moving an occupation from a real to the fictive platform. The total cost function is the sum of these penalties over all occupations and will be minimized. The fictive platform can only be used at a higher penalty. As such, the solver will try to keep trains on their current platform. If this is not possible, it will move a train to another real platform. If even that is not possible, it will have to move the train to the fictive platform.

If keeping the already planned trains on the current platforms is no longer a goal, the cost terms of the first type can be removed and then more platforming options become attractive. This will be especially beneficial if the original planning was not yet done with future traffic in mind, which is often the case. Compared to Caprara et al. (2011a), we use a simpler and directly linear goal function. In fact, we forbid all conflicts using hard constraints, while Caprara et al. (2011a) allow small conflicts up to some threshold, penalize them in their goal function and forbid larger conflicts with hard constraints. In order to penalize the number of minutes of conflicts, one needs a quadratic term. This is why Caprara et al. (2011a) use a quadratic objective function. Clearly, if in Caprara et al. (2011a) the threshold is set to zero, then everything is directly linear. Also, we do not model some other concerns present in Caprara et al. (2011a) like time shifts and minimization of the number of trains simultaneously present, because our company considers these to be a secondary concern.

3.6.3. *Solver Algorithms:*

We use Mixed Integer Linear Programming and used the three solvers CPLEX, Gurobi and XPRESS with their default settings and solver algorithms. These are branch-and-cut or branch-and bound for MILP and simplex or dual simplex for LP problems. Since we obtain low computation times, we do not need nor use techniques like clique cuts or column generation. However, it must be mentioned that only considering one route variant per in- or out-line and platform combination, as we do, reduces the complexity of the model and the necessary solving time.

4. TPP MILP Model

In this section, we derive our train routing and platform allocation model in detail. Essentially, we map train traffic on the infrastructure, so we define these terms first. We subsequently define constant time points, constant sets and mappings, variables, constraints and the goal function of our model.

4.1. A Note on Booleans and MILP

We will derive an integer mixed linear programming model for our TPP version. However, in our model formulation, we allow the introduction of boolean variables and boolean expressions and equations using the logical operators: OR ($\vee$), AND ($\wedge$), implication ($\Rightarrow$), equivalence ($\equiv$) and negation ($\neg$). We also use the operators equality ($=$) and non-equality ($\neq$) that map two integers to a boolean. All these logical expressions can be converted to regular MILP constraints on 0,1-integers as, for example, described in Williams (1994). An open sourced software layer (Sels, 2012) we wrote does this automatically for all these logical expressions. The derived model then becomes a regular MILP and can directly be solved by any MILP solver. We used CPLEX, Gurobi and XPRESS here. Allowing these logical expressions in our model description makes it more compact as well as easier to understand.

4.2. Infrastructure Definition

A station consists of a set of parallel platforms, with a routing grid on one or two sides. The goal of these route grids is to be able to connect (almost) any platform to any line going to or coming from other stations. This grid allows more flexibility in assigning trains to platforms, which increases practical station capacity.

4.3. Traffic Definition

A train entering or leaving a station through any route and a platform is called a *movement*. Each movement happens on a particular *IN*- or *OUT-route*. An *occupation* is a collection of (n IN $+m$ OUT) *movements* that describe arrival and respectively departure times of the same physical train units and consequently to/from the same platform. Each movement can occur on a different route, but all the movements within one occupation will pass or stop at the same platform. By these definitions, each train *movement* has to be mapped on a *route* whereas each train *occupation* has to be mapped on a *platform*. By defining occupations to have an unrestricted number of IN and OUT movements, we effectively allow any train merge and split pattern. A platform is occupied from the beginning of the earliest IN movement of the occupation up to the end of the latest OUT movement in the occupation. Grouping the movements in occupations allows us, per occupation, to enforce the allocation of all its movements to the same platform, real or fictive. Also, movements of different occupations cannot be interleaved in time on the same platform. For these two reasons, the concept of occupation is essential.

4.4. Resource Occupation Time Point Definitions

We follow the train in Figure 1 in increasing time and space dimensions, as it goes from the IN route over the platform to the OUT route. The top half of this figure shows two curved lines. The bottom curve describes the position and time of the head of the train, evolving along the vertical space axis and horizontal time axis. The top curve shows the position and time of the tail of the train. The vertical distance between those curves, along the space axis, represents the length of the train which is of course constant. The curved parts represent the deceleration and acceleration before and after the dwell time at a stations platform for a stopping train. The theory that follows will also be applicable to passing trains though. The bottom half of Figure 1 is the vertical projection of its top half. This leaves the time dimension only for the different resources, being the IN route, the platform and the OUT route here. By performing this projection we get time intervals for each resource.

There is a time overlap between each pair of subsequent resources the train travels through. This is due to the head of the train being on the next resource, while the tail of the train is still on the current resource.

If we analyze an IN movement, there are different time points to consider, as well as durations between them.

Note that the indices o,m,p,r respectively stand for occupation, movement, platform and routing indices.

- The time $trhi_{o,m,r}$ is the time when the routing has the head of the train at its beginning (in side), which is also the end of the IN line.

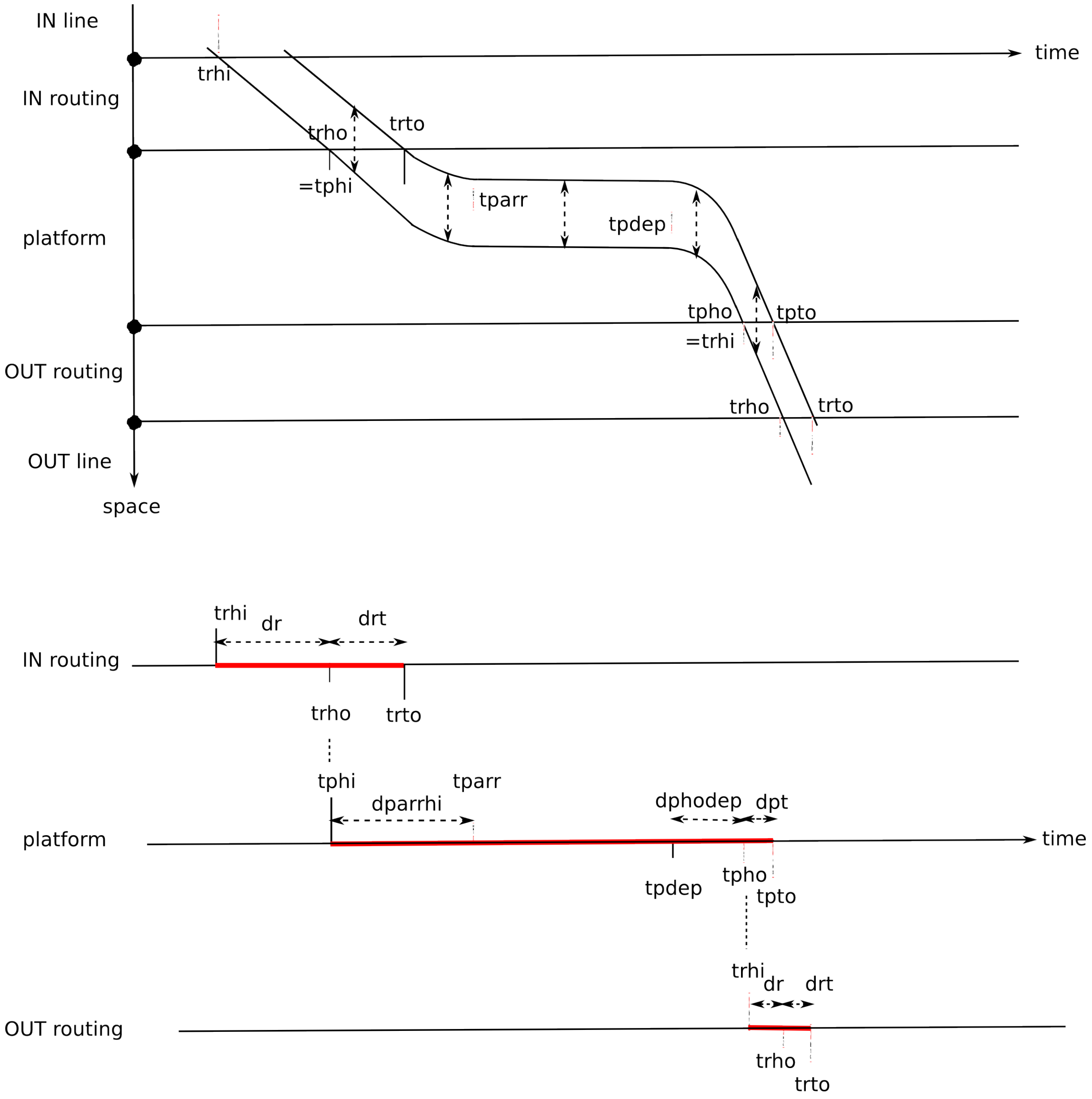

Figure 1: Resource Occupation Time Intervals and their Durations.

- The time $trho_{o,p,m,r}$ is the time when the routing has the head of the train at its very end (out side). This coincides with the in side of the platform, so it is equal to $tphi_{o,p,m}$

- The difference between the above times is the route to platform duration drs_r when we have a stop train and drp_r for a pass train. So $trho_{o,p,m,r} = trhi_{o,m,r} + drs_r$ and $trho_{o,p,m,r} = trhi_{o,m,r} + drp_r$ respectively. Note that these durations are supposed to be solely dependent on the route r and not on the train type. This is the case because we consider a conservatively high value for the slowest train, such that any train can satisfy these requirements.

- The time $tparr_{o,m}$ is the time when the middle of the train arrives at the middle of the platform. This is the given platform arrival time as specified in the planning.

- We define another time difference as $dparrhi_p$ where $tphi_{o,p,m} = tparrr_{o,m} - dparrhi_p$. Note that this value is dependent on the platform, more specifically on its length. It is also dependent on the speed of the train but again we take a conservative value for a slow train here.

Symmetrically for an OUT movement we define the following.

- The time $tpdep_{o,m}$ is the time where the middle of the train is still at the middle of the platform but the train is departing now.
- At the time $tpho_{o,m,r}$ the platform has the head of the train at its out side. This time also equals the time $trhi_{o,m,r}$ where the OUT routing has the head of the train at its in side
- The duration between the above two time points is $dphodep_p$ where $tpdep_{o,m} = tpho_{o,m,r} - dphodep_p$. As for the IN movement, this duration is considered to be only platform and not train dependent.
- The time $trho_{o,p,m,r}$ is the time where the OUT routing has the head of the train at its out side, which is also the starting point of the OUT line.
- The difference between the two above time points is the duration $dr_r = trho_{o,p,m,r} - tpho_{o,p,m}$. This solely depends on the route, since the train is considered the slowest one again.

On top of the above we have, for every resource, a duration between the leaving time of the head of the train and the leaving time of the tail of the train. Figure 1 shows these as

- drt_r for the duration of the routing of the train tail, which is dependent on the IN route it comes from, and
- dpt_p for the duration on the platform of the train tail, dependent on the OUT route it goes to. We take dpt_p as equal to the drt_r of the OUT route.

4.5. Basic Constant Sets and Mappings

- Infrastructure
  - L is the set of lines of both sides of a station, both in and out lines
  - P is the set of platforms of a station
  - R is the set of routes from lines towards the platforms, and from platforms to lines
  - $\forall p \in P : R_p$ is the set of routes that are connected to platform p
  - $r2p : R \to P : r \mapsto p$ is the mapping that for each route r, gives the platform p it is connected to
- Train activities
  - O is the set of occupations to be mapped on platforms
  - M is the set of all movements, where several movements can belong to the same occupation. M_{IN} is the set of IN movements M_{OUT} is the set of OUT movements
  - $\forall o \in O : M_o$ is the set of movements for an occupation o
  - $m2o : M \to O : m \mapsto o$ is the mapping that for each movement m, gives the occupation o it is belongs to
- Range of possible infrastructure items per train activity
  - P_o is the set of all platforms that are reachable for each of the movements belonging to occupation o.
  - $R_{o,m}$ is the set of all routes that are possible for movement m of occupation o.

4.6. Further Constants

4.6.1. Infrastructure properties

- $\forall p \in P : dps_p$ is the duration a train's head is on the platform p, when it will stop at the platform, excluding the dwell time
- $\forall p \in P : dpp_p$ is the duration a train's head is on the platform p when it will pass the platform, dwell time = 0
- $\forall p \in P : dparrhi_p$ is the duration a train's head is on the platform p up to the time when the middle of the train is at the middle of the platform. We take as approximation: $dparrhi_p = dps_p/2$ or $dparrhi_p = dpp_p/2$ for a stopping and passing train respectively.
- $\forall p \in P : dphodep_p$ is the duration a train's head is on the platform p after the time the middle of the train is at the middle of the platform. We take as approximation: $dpdepho_p = dps_p/2$ or $dpdepho_p = dpp_p/2$ for a stopping and passing train respectively.
- $\forall r \in R : drh_r$ as the duration of a train's head to go through the routing r
- $\forall r \in R : drt_r$ as the duration between a train's head and tail to go out of the routing r
- dep_{r_0,r_1} indicates whether two routes r_0 and r_1 are dependent (1) or not (0). (Two routes are dependent if they have an infrastructure resource (section, switch or signal) in common.)
- p_{FICT} is the fictive platform

4.6.2. Train activity properties and worst case time bounds

The entities defined here, are *constants* to the MILP model. (In some cases, similarly named *variables* are defined later. In those cases, we distinguish between these by adding a suffix *'C'* here.)

- $\forall o \in O : tparr_o$ is the fixed arrival/pass time at the platform (⇒ middle of train is at middle of platform)
- $\forall o \in O : tpdep_o$ is the fixed departure/pass time at the platform (⇒ middle of train is at middle of platform)
- To avoid that two trains use the same resource at any given time, be it a route or a platform, we will later forbid that their usage time intervals overlap each other. To allow this, we need to calculate the *worst case* (over all possible allocations) *C*onstant *Lo*w and *Hi*gh bounds on each time interval of movements and occupations, where each resource is potentially (assuming any allocation possible) in use. Some renaming is done here to make formulas later more consistent. We use ≡ to mean *defining equivalence* and define:
  - route movement time interval Low and High Constant times,
    - for IN movements:

$$\forall o \in O : \forall m \in M_{o,IN} : \forall_{p=r2p_r} r \in R_{o,m} :$$
$$\begin{cases} mtLoC_{o,m,r} \equiv trhi_{o,m,r} = tparr_{o,m} - dparrhi_p - dr_r \\ mtHiC_{o,m,r} \equiv trto_{o,m,r} = tparr_{o,m} - dparrhi_p + drt_r \end{cases} \tag{1}$$

    - for OUT movements:

$$\forall o \in O : \forall m \in M_{o,OUT} : \forall_{p=r2p_r} r \in R_{o,m} :$$
$$\begin{cases} mtLoC_{o,m,r} \equiv trhi_{o,m,r} = tpdep_{o,m} + dphodep_p \\ mtHiC_{o,m,r} \equiv trto_{o,m,r} = tpdep_{o,m} + dphodep_p + dr_r + drt_r \end{cases} \tag{2}$$

  - platform occupation time interval Low Constant time for each of its IN movements:

$$\forall o \in O : \forall p \in P_o : \forall m \in M_{o,IN} : otLoC_{o,p,m} \equiv tphi_{o,p,m} = tparr_{o,m} - dparrhi_p \tag{3}$$

  - platform occupation time interval High Constant time for each of its OUT movements:

$$\forall o \in O : \forall p \in P_o : \forall m \in M_{o,OUT} : otHiC_{o,p,m} \equiv tpto_{o,p,m} = tpdep_{o,m} + dphodep_p + dpt_p \tag{4}$$

- An important difference with De Luca Cardillo (1998); Billionnet (2003); Delorme et al. (2001), is that we take into account differences in route durations. So, depending on the route taken, the movement time intervals change. Consequently, the time interval $[otLoC_{o,p,m}, otHiC_{o,p,m}]$ when either route or platform is occupied, changes as well. So, in our model, time interval bounds are variables for the MILP model. In De Luca Cardillo (1998) and Billionnet (2003), *actual overlap* calculation of each pair of constant train (route) movement time intervals can be done, even prior to the setting up of the MILP model. We however, have to model in terms of *potential overlaps*. We do this, for both movements and occupations, using the worst case, constant, lower and upper bounds defined in equations (5), (6) and (7).

  - platform occupation time interval Low Lower bounds (Lb) and High Upper bounds (Ub) Constant times over all its IN respective OUT movements:

$$\forall o \in O : \forall p \in P_o : \left\{ \begin{array}{l} \forall m \in M_{o,IN} : otLoLbC_{o,p} \leq otLoC_{o,p,m} \\ \forall m \in M_{o,OUT} : otHiUbC_{o,p} \geq otHiC_{o,p,m} \end{array} \right. \tag{5}$$

  - Lower bounds (Lb) on Low time (Lo) and Upper bounds (Ub) on High time (Hi) of occupation platform time (ot) intervals, over all platforms possible for occupation o:

$$\forall o \in O : \forall p \in P_o : \left\{ \begin{array}{l} otLoLbC_o \leq otLoLbC_{o,p} \\ otHiUbC_o \geq otHiUbC_{o,p}. \end{array} \right. \tag{6}$$

    These platform choice independent bounds $otLoLbC_o$ and $otHiUbC_o$ will be used in equations (10).

  - Lower bounds (Lb) on Low time (Lo) and Upper bounds (Ub) on High time (Ht) of movement route time (mt) intervals, over all routes possible for movement m:

$$\forall o \in O : \forall m \in M_o : \forall r \in R_{o,m} \left\{ \begin{array}{l} mtLoLbC_{o,m} \leq mtLoC_{o,m,r} \\ mtHiUbC_{o,m} \geq mtHiC_{o,m,r}. \end{array} \right. \tag{7}$$

    The route choice independent bounds $mtLoLbC_{o,m}$ and $mtHiUbC_{o,m}$ will be used in equations (11).

- $\forall o \in O : pORIG_o$ is the original platform of the occupation o

- dt_S is the security time that separates all platform occupations pairs as well as all movement route pairs

- CF_{INI} the cost for assigning an *ini*tial train to the fictive platform

- CF_{SUP} the cost for assigning a *sup*plementary train to the fictive platform

- CR_{INI} the cost for assigning an *ini*tial train to a real platform. It is set to 0 for its preferred platform.

- CR_{SUP} the cost for assigning a *sup*plementary train to a real platform. It is set to 0 for its preferred platform.

4.7. Decision Variables

The entities defined here, are variables of the MILP model. To distinguish between possibly similarly named constants defined earlier, we make this explicit with a suffix 'V'. For every occupation $o \in O$, a variable $p_o \in P$ defines the variable platform that the occupation will be allocated to. Also, for every occupation $o \in O$ and for every movement $m \in M_o$, a variable $r_{o,m} \in R$ defines the variable route that the movement within this occupation will be allocated to. p_o and $r_{o,m}$ are related. Indeed, once the platform is chosen for an occupation, only routes connected to it are left as a valid choice for its movements. Also, since the lines are specified per movement, a movement will also only allow the routes that are connected to this line. We define:

- $\forall o \in O : \forall p \in P : op_{o,p}$, which is true iff platform p is chosen for occupation o,
- $\forall o \in O : \forall m \in M_o : \forall r \in R : mr_{o,m,r}$ which is true iff routing r is assigned to movement m,
- lower and upper bound variable times on occupation platform time intervals:

$$\forall o \in O : \begin{cases} otLoV_o \equiv \sum_{p \in P_o} otLoLbC_{o,p} \cdot op_{o,p} \\ otHiV_o \equiv \sum_{p \in P_o} otHiUbC_{o,p} \cdot op_{o,p}, \end{cases} \tag{8}$$

- lower and upper bound variable times on movement route time intervals:

$$\forall o \in O : \forall m \in M_o : \begin{cases} mtLoV_{o,m} \equiv \sum_{r \in R_{o,m}} mtLoC_{o,m,r} \cdot mr_{o,m,r} \\ mtHiV_{o,m} \equiv \sum_{r \in R_{o,m}} mtHiC_{o,m,r} \cdot mr_{o,m,r}. \end{cases} \tag{9}$$

- We have to impose that between two usages of the same resource, their time intervals do not overlap, so are *separated*. This is the case when the upper bound of one time interval comes *before* the lower bound of the other. In practice we also want to leave a certain *security time buffer* between each subsequent pair of trains on the same resource. This security time dt_S will typically range from 1 minute to 3 minutes. The lexical preceding operator $\prec$ is used to avoid unnecessary definitions and constraints.
  - Before and separated booleans for occupation platform interval pairs:

$$\forall_{\substack{o_0 \prec o_1 \\ [otLoLbC_{o_0}, otHiUbC_{o_0}] \cap \\ [otLoLbC_{o_1}, otHiUbC_{o_1}] \neq \phi}} o_0, o_1 \in O : \begin{cases} obef_{o_0,o_1} \equiv (otHiV_{o_0} + dt_S \leq otLoV_{o_1}) \\ obef_{o_1,o_0} \equiv (otHiV_{o_1} + dt_S \leq otLoV_{o_0}) \\ osep_{o_0,o_1} \equiv (obef_{o_0,o_1} \vee obef_{o_1,o_0}). \end{cases} \tag{10}$$

    In equation (10), based on the interval ordering bef booleans, we define separation booleans, which are used later in (17) in the filtering $((o_0 \prec o_1) \wedge [otLoLbC_{o_0}, otHiUbC_{o_0}] \cap [otLoLbC_{o_1}, otHiUbC_{o_1}] \neq \phi)$ of interval pairs that can potentially overlap. This filter will significantly reduce the number of constraints. So of course, here in equation (10), we also only define the ordering variables $osep_{o_0,o_1}$ that are needed in these constraints, by using exactly the same filter.
  - Before and separated booleans for movement route interval pairs:

$$\forall_{\substack{m_0 \prec m_1 \\ [mtLoLbC_{m_0}, mtHiUbC_{m_0}] \cap \\ [mtLoLbC_{m_1}, mtHiUbC_{m_1}] \neq \phi:}} m_0, m_1 \in M : \begin{cases} mbef_{m_0,m_1} \equiv (mtHiV_{o_0} + dt_S \leq mtLoV_{m_1}) \\ mbef_{m_1,m_0} \equiv (mtHiV_{o_1} + dt_S \leq mtLoV_{m_0}) \\ msep_{m_0,m_1} \equiv (mbef_{m_0,m_1} \vee mbef_{m_1,m_0}), \end{cases} \tag{11}$$

    where these $msep_{m_0,m_1}$ variables and the same movement filter is also used in (18).

- Changed Platform to Fictive Boolean, true iff for an occupation, the original platform was not the fictive one but the solver changes it to the fictive one.

$$\forall o \in O : cf_o \equiv ((pORIG_o \neq pFICT) \wedge (op_{o,p} = pFICT)). \tag{12}$$

- Changed To Other Real Platform Boolean: true iff for an occupation, the original platform is changed by the solver to another real one.

$$\forall o \in O : cr_o \equiv (op_{o,p} \neq pORIG_o) \tag{13}$$

4.8. *Constraints*

- For each occupation, exactly one platform has to be chosen:

$$\forall o \in O : \sum_{p \in P} op_{o,p} = 1 \tag{14}$$

- For each movement, exactly one route has to be chosen:

$$\forall o \in O : \forall m \in M_o : \sum_{r \in R} mr_{o,m,r} = 1 \tag{15}$$

- Platform-route connectivity implies a relation between the assignment of routes to movements and platforms to occupations. More specifically, if a route r is assigned to a movement m of occupation o (if $mr_{o,m,r} = true$), then the unique platform $r2p_r$ the route r is connected to, and the occupation $m2o_m$ the movement m belongs to, should also be assigned to each other (then $op_{m2o_m,r2p_r} = true$):

$$\forall o \in O : \forall m \in M_o : mr_{o,m,r} \implies op_{m2o_m,r2p_r} \tag{16}$$

The interval bounds resulting from the definitions (6) and (7) are now used as a filter which allows us to only have to impose the reduced set of the separation enforcing equations (17) and (18) below.

- A platform cannot be used by two trains at the same time. So, for each pair of different occupations which are potentially overlapping in time (domain: $[otLoLbC_{o_0}, otHiUbC_{o_0}] \cap [otLoLbC_{o_1}, otHiUbC_{o_1}] \neq \phi$), if they are assigned to *the same platform resource* ($p_0 = p_1$), then we enforce separation of their platform usage time intervals

$$\forall_{\substack{o_0 < o_1 \\ [otLoLbC_{o_0}, otHiUbC_{o_0}] \cap \\ [otLoLbC_{o_1}, otHiUbC_{o_1}] \neq \phi :}} o_0, o_1 \in O : \forall_{p_0 = p_1} (p_0, p_1) \in (P_{o_0}, P_{o_1}) : op_{o_0,p_0} \wedge op_{o_1,p_1} \implies osep_{o_0,o_1}. \tag{17}$$

  Note that, if two different platforms are chosen for o_0, o_1, no time separation is enforced between them.

- Two dependent routes cannot be used by two trains at the same time. So, for each pair of different movements which are potentially overlapping in time (domain: $[mtLoLbC_{m_0}, mtHiUbC_{m_0}] \cap [mtLoLbC_{m_1}, mtHiUbC_{m_1}] \neq \phi$), if they are assigned to *dependent route resources* (dep_{r_0,r_1}), then we enforce separation of their route usage time intervals

$$\forall_{\substack{m_0 < m_1 \\ [mtLoLbC_{m_0}, mtHiUbC_{m_0}] \cap \\ [mtLoLbC_{m_1}, mtHiUbC_{m_1}] \neq \phi :}} m_0, m_1 \in M : \forall_{dep_{r_0,r_1}} (r_0, r_1) \in (R_{m_0}, R_{m_1}) : mr_{o_0,m_0,r_0} \wedge mr_{o_1,m_1,r_1} \implies msep_{m_0,m_1}. \tag{18}$$

  Note that, if two independent routes are chosen for m_0, m_1, no time separation is enforced between them. Also, movements of each pair can belong to the same or to different occupations.

4.9. *Goal Function*

- Minimize penalties of moving assignments from real to fictive platform and of moving assignment from preferred (real) to non-preferred (real) platforms, for both initial and for supplementary train sets, possibly with different weights:

$$\begin{aligned} g(op_{o,p}) &= \textstyle\sum_{o \in O_{INI}} CF_{INI} \cdot cf_o + CR_{INI} \cdot cr_o \\ &+ \textstyle\sum_{o \in O_{SUP}} CF_{SUP} \cdot cf_o + CR_{SUP} \cdot cr_o. \end{aligned} \tag{19}$$

  For conservative optimization, where we have a preference for current platforms and also for current trains over future trains to be platformed, one can use $(CF_{INI}, CF_{SUP}, CR_{INI}, CR_{SUP}) = (8, 4, 2, 1)$. Actual capacity evaluating users at Infrabel also used $(100, 50, 10, 1)$. For progressive optimization one can use $(1, 1, 0, 0)$ instead. Here, there are no preferential platforms and both current and future traffic are equally preferred to be platformed.

This completes our high level model formulation. As mentioned before in section 4.1, we automatically convert the boolean expressions to 0,1-integers and linear constraints over them, so we end up with a MILP model. The constants defined in Section 4.4 to 4.6 are calculated prior to setting up the MILP model. For the ones in section 4.6, equations (1) to (7) are used to do this. The decision variables in section 4.7 are added to the MILP model and their definitions (8) to (13) become linear constraints in this model, as do the constraints (14) to (18) from section 4.8. Finally, the linear goal function (19) is added.

5. RESULTS

For a set of 10 stations and for current as well as current plus future traffic, during peak hours between 8am and 9am, we now present the results of our platforming tool. We do this for the conservative as well as progressive optimization scenario. Then, we mention our solver times. Lastly, we describe its integration and the context at Infrabel, in which it is being used.

5.1. *Platforming Results*

To measure the success rate of the platformer in a station, we count occupations and movements for both the original and the optimized schedule. Movement counts are a better measure for the platformers' performance in case of train splits and merges.

As for the columns of Table 1, with our goal function weights $(CF_{INI}, CF_{SUP}, CR_{INI}, CR_{SUP})$ set to $(8, 4, 2, 1)$, we consider the conservative optimization scenario (C-column) and with the same set to $(1, 1, 0, 0)$, the progressive optimization scenario (P-column).

Consider the rows in Table 1: for each station, results for occupations are given in the first 5 rows (1a-1e). A 6th row (1f) indicates the maximum theoretically possible number of occupations corresponding with 100% occupation of all platforms. The next 5 rows (2a-2e) indicate movement measurements and the last 5 rows (3a-3e) report a *platform use* percentage which is the fraction of time that all platforms together are used. Table 1 shows numbers for capacity increase relative to the current situation in percentages, for occupation counts (1c=(1b-1a)/1a), movement counts (2c=(2b-2a)/2a) and also platform occupied time (3c=3b-3a). These percentages can be seen as three measures for the relative, still available capacity.

The zeroes in all c rows (1c,2c,3c) in Table 1 indicate that the current traffic (C) of the current timetable can be platformed for 6 of the 10 stations considered. The negative numbers in the c rows, for the progressive scenario, indicate that for four stations, 5% up to 15% of occupations or 5% up to 20% of movements cannot be platformed. However, when we schedule current and future traffic (CF), we obtain the results in the d rows. The e rows then show the relative increase in occupations (1e=(1d-1a)/1a), movements (2e=(2d-2a)/2a) and relative platform occupied time (3e=3d-3a). The consistently positive numbers in the e rows (1e, 2e), show that, when also platforming future traffic (CF) (at other various platform times) an increase of the total number of platformed occupations, the total number of movements as well as the platform usage percentage is still possible. Row 3e has small negative percentages for Gent, Leuven and Oostende but this is caused by the solver choosing many occupations with shorter dwell times.

Remember that Table 1 shows the results for fixed platform times. It illustrates that for some stations the original schedule is infeasible. This often also means that the number of occupations on real platforms has to be reduced to obtain a feasible schedule. This is the case for the conservative scenario (Bergen: 19 → 18, Gent: 35 → 32, Leuven: 36 → 30, Oostende: 8 → 7), but also for the progressive scenario where the results are the same, except that for Leuven we can manage to platform just one more occupation. However, when we also add a future train set, we see that for the conservative scenario, the number of possible occupations increases (Bergen: 18 → 39, Gent: 32 → 36, Leuven: 30 → 56, Oostende: 7 → 11). The progressive optimization even adds 2 and 1 occupation more for Bergen and Gent respectively. This shows that the original infeasibility is not due to the current train set exceeding the absolute capacity of the station considered. It is due to the platform times fixed in the timetable. It is very likely that changing these platform times would allow to platform all trains in a feasible way.

For the stations where all current traffic could be placed, even in the conservative scenario, we also see that more future traffic can be additionally platformed. (Brugge: 19 → 43, Denderleeuw: 22 → 37, Aarschot: 6 → 34. Lichtervelde: 10 → 29). Only Mechelen (32→32) seems quite saturated but this is because no future traffic was foreseen there. For Brugge, Denderleeuw and Lichtervelde, the progressive optimization manages to add 4, 5 and 1 occupation more than the conservative option.

Table 1: Platforming **current train occupations only** (rows starting with 'C:') as well as **both, current and future train occupations** (rows starting with 'CF:'), each contrasting **Conservative** ('C' columns) and **Progressive Scenario** ('P' columns). Conservative: $(CF_{INI}, CF_{SUP}, CR_{INI}, CR_{SUP}) = (8, 4, 2, 1)$. Progressive: $(CF_{INI}, CF_{SUP}, CR_{INI}, CR_{SUP}) = (1, 1, 0, 0)$. (r) means original solution. (p) means optimal solution. $r \rightarrow p$ means the transition from original to optimized solution. Countings of occupations (Occup.) and movements (Mov.) for an original 'solution' which is infeasible are marked with a *.

Conservative or Progressive	C	P	C	P	C	P	C	P	C	P
Station **row**	Bergen		Brugge		Denderleeuw		Gent		Leuven	
# Original Platformed Occup. (r), **1a**	19*	.	19*	.	22	.	35*	.	36*	.
C: # Optimized Platformed Occup. (p) **1b**	18	18	19	19	22	22	32	32	30	31
C: Platformed Occup. % Inc. (r→p) **1c**	-5	-5	0	0	0	0	-9	-9	-17	-14
CF: # Optimized Platformed Occup. (p) **1d**	39	41	43	47	37	42	36	37	56	56
CF: Platformed Occup. % Inc. (r→p) **1e**	105	116	126	147	68	91	3	6	56	56
# Max. Reachable Platformed Occup. **1f**	168	.	240	.	216	.	288	.	336	.
# Original Routed Mov. (r) **2a**	33*	.	34*	.	39	.	64*	.	65*	.
C: # Optimized Routed Mov. (p) **2b**	31	31	34	34	39	39	57	56	51	55
C: Routed Mov. % Inc. (r→p) **3c**	-6	-6	0	0	0	0	-11	-12	-22	-15
CF: # Optimized Routed Mov. (p) **2d**	65	68	82	88	67	77	64	66	101	102
CF: Routed Mov. % Inc. (r→p) **2e**	97	112	141	159	72	97	0	3	55	57
Original Platform In Use % (r) **3a**	26	.	23	.	14	.	26	.	31	.
C: Optimized Platform In Use % (p) **3b**	24	24	21	21	14	14	20	21	21	23
C: Platform Extra % In Use (r→p) **3c**	-2	-2	-2	-2	0	0	-6	-5	-10	-7
CF: Optimized Platform In Use % (p) **3d**	49	58	29	26	20	21	23	23	31	30
CF: Platform Extra % In Use (r→p) **3e**	23	32	6	3	6	7	-3	-3	0	-1
Station **row**	Mechelen		Aarschot		Lichtervelde		Manage		Oostende	
# Original Platformed Occup. (r) **1a**	32	.	6	.	10	.	1	.	8*	.
C: # Optimized Platformed Occup. (p) **1b**	32	32	6	6	10	10	1	1	7	7
C: Platformed Occup. % Inc. (r→p) **1c**	0	0	0	0	0	0	0	0	-12	-12
CF:# Optimized Platformed Occup. (p) **1d**	32	32	34	34	29	30	41	41	11	11
CF: Platformed Occup. % Inc. (r→p) **1e**	0	0	467	467	190	200	4000	4000	38	38
# Max. Reachable Platformed Occup. **1f**	240	.	120	.	120	.	120	.	264	.
# Original Routed Mov. (r) **2a**	58	.	12	.	20	.	2	.	10*	.
C: # Optimized Routed Mov. (p) **2b**	58	58	12	12	20	20	2	2	8	8
C: Routed Mov. % Inc. (r→p) **2c**	0	0	0	0	0	0	0	0	-20	-20
CF: # Optimized Routed Mov. (p) **2d**	58	58	66	66	56	58	77	79	14	16
CF:Routed Mov. % Inc. (r→p) **2e**	0	0	450	450	180	190	3750	3850	40	60
Original Platform In Use % (r) **3a**	21	.	9	.	11	.	0.3	.	18	.
C: Optimized Platform In Use % (p) **3b**	21	21	9	9	11	11	0.3	0.3	12	12
C: Platform Extra % In Use (r→p) **3c**	0	0	0	0	0	0	0.0	0.0	-6	-6
CF: Optimized Platform In Use % (p) **3d**	21	21	28	24	26	25	27.0	27.0	13	8
CF: Platform Extra % In Use (r→p) **3e**	0	0	19	15	15	14	26.7	26.7	-5	-9

We have shown that for all the stations considered, more trains can be platformed. However, trying to platform all the current trains, with the current timetable is possible in some stations while not in others. For the stations where this is not possible, small manual shifts in platform times can lead to a feasible platforming solution.

5.2. Solver Response Times

Table 2 gives problem size and solver time data of the same test cases as reported in Table 1. We have tested the 3 commercial solvers CPLEX, Gurobi and XPRESS on comparable machines and mention their individual results.

Table 2: Solver times for platforming **current train occupations only** (rows starting with 'C:') as well as **both, current and future train occupations** (rows starting with 'CF:'). Each contrasting **Conservative** ('C' columns) and **Progressive Scenario** ('P' columns). Conservative: $(CF_{INI}, CF_{SUP}, CR_{INI}, C2OR_{SUP}) = (8, 4, 2, 1)$. Progressive: $(CF_{INI}, CF_{SUP}, CR_{INI}, CR_{SUP}) = (1, 1, 0, 0)$. Cplex v12.4.0.0 and Gurobi v5.0.1 run on Apple iMac Intel Quad core 2.6GHz. Fico Xpress v7.2.1 Solver run on HP Intel Xeon Quad core 3.3GHz. #P = number of real platforms, #R = number of routes, #O = number of occupations to be platformed. #C = number of constraints, #V = number of variables. Three cases only generated a solution at the two hours limit and this with a gap bigger than 0%, [1] : 5.80%,[2] : 33.48% and [3] : 2.24%. [4] Reported no result in two hours limit. All other cases reached a gap of 0%. When a solver is the fastest one to solve a problem instance, its solver time is marked in bold.

Conservative or Progressive	C	P	C	P	C	P	C	P	C	P
Station	Bergen		Brugge		Denderleeuw		Gent		Leuven	
C: #P #R #O	7,128,58		10,196,87		9,206,91		12,224,563		14,324,160	
C: #C * #V	7k, 5k		18k, 9k		13k, 8k		227k, 107k		46k, 24k	
C: Solver Time Cplex (s)	0.4	0.4	1.6	1.4	0.9	0.9	41.8	950.8	3.7	8.9
C: Solver Time Gurobi (s)	**0.2**	**0.2**	0.8	0.6	**0.4**	**0.4**	57.7	**55.4**	**2.2**	**3.0**
C: Solver Time Xpress (s)	0.3	**0.2**	**0.6**	**0.5**	0.5	0.5	**27.6**	[1] 7199.8	**2.2**	5.0
CF: #P #R #O	7,128,178		10,196,267		9,206,211		12,224,1069		14,324,256	
CF: #C * #V	96k, 54k		248k, 96k		140k, 58k		482k, 231k		101k, 56k	
CF: Solver Time Cplex (s)	138.1	847.4	**337.0**	[1]7200.7	27.7	78.2	**182.2**	[2]7204.8	36.2	183.6
CF: Solver Time Gurobi (s)	**78.3**	**502.4**	613.2	**2853.1**	27.5	37.5	442.8	**227.4**	49.7	218.2
CF: Solver Time Xpress (s)	138.3	1224.4	7199.6	[3]7201.4	**20.5**	**33.7**	[4]n.a.	[4]n.a.	**28.8**	**115.1**
Station	Mechelen		Aarschot		Lichtervelde		Manage		Oostende	
C: #P #R #O	10,170,121		5,48,25		5,94,55		5,66,26		11,78,15	
C: #C * #V	9k, 7k		1k, 0.9k		3k, 2k		1.0k, 0.8k		3k, 2k	
C: Solver Time Cplex (s)	0.6	0.5	0.2	**0.0**	0.5	**0.1**	0.6	**0.0**	0.2	**0.1**
C: Solver Time Gurobi (s)	**0.3**	**0.2**	**0.0**	**0.0**	**0.1**	**0.1**	**0.0**	**0.0**	**0.1**	**0.1**
C: Solver Time Xpress (s)	**0.3**	0.4	**0.0**	**0.0**	**0.1**	**0.1**	**0.0**	**0.0**	0.2	0.2
CF: #P #R #O	10,170,121		5,48,55		5,94,115		5,66,86		11,78,45	
CF: #C * #V	9k, 7k		3k, 4k		14k, 9k		4k, 6k		21k, 9k	
CF: Solver Time Cplex (s)	0.5	0.5	0.4	0.4	7.2	5.8	1.5	1.8	2.0	2.1
CF: Solver Time Gurobi (s)	**0.3**	**0.2**	**0.2**	**0.2**	**1.2**	**1.2**	2.3	2.2	**0.8**	**0.8**
CF: Solver Time Xpress (s)	0.4	0.4	0.4	0.7	1.5	1.9	**0.5**	**1.2**	1.1	1.0

As shown in Table 2, MILP matrices that describe the problem get quite big. With the latest solver versions, on 4 core-machines and the given clock frequencies, any problem can be solved in progressive mode in at most 9 minutes. Note that the Brugge and Gent cases are exceptions. CPLEX and Xpress don't return a result in the two hour limit for Brugge. For Gent, CPLEX does the same while Xpress reports 'no integer solution'. Only Gurobi solves all cases within the time limit. In fact all large solver times are due to a number of total traffic that largely exceeds the available capacity. From Table 1, one can see that the maximum number of occupations platformable in the progressive scenario are 47 for Brugge and 37 for Gent. In Table 2, one finds that the requested number of current plus future occupations is 267 for Brugge and 1069 for Gent, so this is 5.6 times and 28.9 times more than can be platformed. The number of feasible assignments to real and fictive platforms then strongly increases and hence it is logical that finding the best one can take much more time. To reduce computation times, it is better to gradually increase the future train set and stop adding trains as soon as, or short after not all trains can be platformed anymore.

Restricted to the more realistic current train set platforming problems, 6 minutes suffices to solve any of them. Gent and Brugge remain the hardest stations. For some stations, like Mechelen, Aarschot, Lichtervelde, Manage and Oostende solver times are below 1.2 seconds.

We can conclude that the solver response times certainly allow our tool to be used for the tactical platforming problem, with current traffic only. It is also well suited and still fast enough for the strategic level, where realistic capacity estimates, based on platforming current together with sensible future train traffic options, are required.

5.3. Integration and Use

Since Infrabel experienced that the manual platforming process takes considerable time and is error prone, they searched for commercially available automated platforming solutions. None were found, so they specified and created one in-house. Thanks to the full integration of Leopard with Infrabel databases containing infrastructure and train movements, the platforming of 2 hours of traffic in a station now typically only takes some seconds. This short time means that planners are not discouraged anymore to repeat this process during planning, at each change of the stations' platform times in the timetable. This should make the planners' jobs easier and the achieved results better.

Leopard has been integrated at Infrabel, in their Ocapi tool. The Ocapi GUI allows the user to select a station, to select current train traffic or not, and to select particular lines on which he wants to generate supplementary future train traffic. Ocapi then calls our platformer Leopard, which solves the corresponding model quickly and returns results as the ones described above. Ocapi then calculates more derived graphs and measures, like a capacity number. Infrabel was positively surprized by the amount of traffic our platformer can quickly platform without conflicts. The integration was successful and helps Infrabel to better and more quickly analyze the *possibility of platforming* current as well as extra future traffic and it also *generates an optimal platform schedule* which can be used in practice.

6. CONCLUSIONS

In this paper we define in detail a train platforming problem on the tactical and strategic level for the Belgian infrastructure manager Infrabel. A MILP model is developed and used to optimize the platforming problem for ten different stations. Based on these calculations the capacity of the different stations in practice can be evaluated. Both the current set of trains as well as a future set of trains, with fixed arrival and departure times, are considered. It handles normal train passes and stops, as well as train splits and merges from any n incoming to m outgoing trains. For each train, a preferred platform can be set, if so desired. For a realistic input problem, being the morning peak hours in a realistic station with current train traffic, the solver response time is less than 2.2 seconds with the fastest MILP solver. A large station with a whole day of traffic can require about 30 seconds to platform. This is fine for a tactical level planning tool. When evaluating timetable scenarios on the strategic level, including supplementary future traffic, calculation times rise but stay below 9 minutes for realistic scenarios. This is acceptable for capacity estimation studies at the strategic level. For the given future train sets for Brugge and Gent the solver times are much higher, but we consider these sets as highly unrealistic since they multiply the current traffic by 6 and 29 respectively.

For the stations considered, we were able to check and correct platforming assignments. We can also estimate the practical capacity of a station by adding a large future train set and see the amount of trains that can still be platformed. We proved that the stations Gent and Mechelen are quite saturated and that for the 8 other stations, there is still extra capacity available. The practical value of the developed methodology and tool is clear from the presented tests on real stations with real traffic. Our tool is also integrated at Infrabel and used for quickly evaluating future traffic scenarios and station capacity consumption.

As for the scientific value of this paper, the main improvements are that the use of MILP for TPP, introduced first by Billionnet (2003), is now extended with accounting for different route durations calculated from maximum route speeds and train lengths. This results in a time schedule which, apart from giving usage time intervals for platforms, also returns exact usage time intervals for default routes. Even with this extra complexity, which is reflected by a much bigger number of constraints and variables, solver times stay low. We find the optimal solution, with the goal function specified by the railway company. Moreover, since route parameters and their dependencies are already explicitly present in our model, we believe that our model is easier to extend with multiple route variants than Billionnet's model. Currently, no complex clique cuts are required to obtain acceptable solver times. We note that our relatively low solver times are partly caused by only considering one route variant for each combination of in- or out-line and platform track.

Compared to De Luca Cardillo and Mione, we reach similar low solver times but, contrary to them, also can guarantee optimality. Also, we see our use of MILP as more straightforward, flexible and extendable. The addition of the concept of fictive platform and fictive routes, together with the addition of the concept of supplementary traffic, and the possibility of treating existing and supplementary traffic differently in the goal function, allows to platform future scenarios and make studies about good use of the remaining capacity.

7. POSSIBLE EXTENSIONS

We envision some possible extension to our work. Firstly, to more closely approximate reality, we also want to model multiple route variants per line-platform combination. Ignoring this currently slightly underestimates real capacity.

Secondly, in the tactical context of planning timetables and solving the introduced platform problems, we want to allow the platform times and junction times to vary within certain limits, which increases the chance on a feasible solution. Of course, this would complicate the TPPs, but a timetable that guarantees that all its planned trains can also be platformed is a better one than a timetable that does not.

Thirdly, currently, our model only allows one train on a route at a time. Even though this is safe, in a real station, *liberation points* exist, which allow a second train to pass onto the same route, as soon as the first train has passed a liberation point. Modelling this too would improve the maximum capacity of the model and get it closer to reality. However, in practice, Infrabel considers liberation points as a method to be used in real time to solve cases of unexpected peaks in traffic, but they don't want the planning already to rely on it. This means that at Infrabel, this feature should only be used at the operational level. Zwaneveld et al. (1996); Zwaneveld (1997, 2001) and Kroon et al. (1997) already model the position and use of these liberation points, also at the strategic level.

Finally, our platforming tool is usable in other contexts than just capacity estimation. Indeed, for example, merely changing the goal function could result in a system also trying to lower transfer passenger average walking time or/and distance. This would indeed benefit passengers more than just trying to platform as many trains as possible, independently of the number of passengers present on them. This seems an interesting goal, given our previous timetabling research (Vansteenwegen and Van Oudheusden (2006, 2007); Sels et al. (2011, 2013)) that already optimizes a timetable for minimal expected passenger time. Our timetabling tool could propagate down the passenger numbers on transfers to our platformer tool, where they could then be used as weights in its goal function. We believe that the choice of platform - as well as of arrival, departure and its resulting dwell and transfer times - should ideally be steered by minimizing expected passenger time that is clearly a function of these choices. This should give a better result for passengers than artificial platform preferences which, as Carey and Carville (2003) mention, are also even difficult to obtain.

ACKNOWLEDGEMENTS

We thank both Infrabel for clearly specifying their requirements and supplying the necessary data as well as ICTRA for the integration of our train platforming module Leopard in the Infrabel interactive application Ocapi. We also wish to thank the reviewers for their reading and valuable feedback which helped to validate and improve this paper.

AppendixA. Comparison of Features in TPP Literature

publication	level	approach	technique	different route variants	platf. times var.	cont. time model	different		
							d_r	$v_{r,max}$	l_t
Zwaneveld et al. (1996) Zwaneveld (1997)	S	time & platform choice graph, NPP	MILP	Y	Y	N	Y	Y	Y
De Luca Cardillo (1998)	T	conflict graph SPP	backtrack heuristic	N	N	Y			
Delorme et al. (2001)	S	constraint programming	constraint propagation	N	N	Y			
Billionnet (2003)	T	conflict graph SPP	MILP	N	N	Y			
Carey (1994a) Carey (1994b)	S,T	human planner inspired B& B heur.	MILP	N	Y	N	Y	Y	N
Carey and Carville (2000) Carey and Carville (2003)	S,T	human planner inspired heuristic	greedy heuristic	N	Y	N	Y	Y	N
Carey and Crawford (2005)	S,T	human planner inspired heuristic	greedy heuristic	N	Y	N	Y	Y	N
Caprara et al. (2007) Caprara et al. (2011a)	T	time & platform choice graph, NPP	branch& price&cut	Y	Y				
Lusby et al. (2011b)	S,O	section based SPP	SPP	Y	Y	N	Y	Y	Y
Sels et al.	S,T	conflict graph SPP	MILP branch & cut	N	N	Y	Y	Y	Y

Table A.1: TPP Literature Publications: method and characteristics. When column ’different route variants’ contains a ’Y’, it means that multiple routes between an in or our line and a platform are considered. A ’N’ means that only one route is considered (when it exists). When column ’platf. times var.’ shows ’Y’, it means platform times are variable rather than fixed. The ’cont. time model’ column has a ’Y’ when time variables are modeled as continuous variables rather than modeled having a certain discrete resolution and coarseness. d_r = durations of routes, which are calculated from $v_{r,max}$ = maximum speed on routes and l_t = length of trains, SPP = Set Partitioning Problem, NPP = Node Packing Problem, S = Strategic level, T = Tactical level, O = Operational level.

publication	stations	traffic	optimal	solver times:		integrated:	
	tested	tested	solution	avg(s)	max(s)	company	in tool(s)
Zwaneveld et al. (1996) Zwaneveld (1997)	1 real Zwolle	20 real	Y	S:50	S:400	ProRail NL	STATIONS
De Luca Cardillo (1998)	1 real,Italian	6 real	N	T:0.49	T:115		
Delorme et al. (2001)	1 real, P.G.	4 projected	Y	S:≥1000	S:10k	SNCF FR	RECIFE
Billionnet (2003)	20 generated 1 real, Abatone	20 generated 2 real	Y	T:37.5 T:0.01	T:702 T:0.03		
Carey (1994a) Carey (1994b)	1 real, Leeds	1 real	N			N	N
Carey and Carville (2000) Carey and Carville (2003)	1 real Leeds	1 real	N			N	N
Carey and Crawford (2005)	1, adapted from real, Leeds	1, adapted from real, Leeds	N			N	N
Caprara et al. (2007) Caprara et al. (2011a)	4 real, Italian	4 real	Y	T:143	T:8935	RFI, IT	
Lusby et al. (2011b)	1 real, P.G.	20 generated	Y				
Sels et al.	10 real Belgian	10 real + 10 projected	Y	T: 3.15 S: 374	T:27.6 S: 2853	Infrabel BE	Ocapi Leopard

Table A.2: TPP Literature Publications: results, performance and use. projected = the future additional traffic being platformed in addition to the current one. Sels' results for T=Tactical correspond with Current Traffic with Conservative Approach while results for S=Strategic correspond with Current and Future Traffic with Progressive Approach. P.G. = Pierrefitte-Gonesse junction north of Paris.

AppendixB. Optimization Improvement Demonstration by Visual Verifications

The model developed in this paper describes how we optimize platform and route assignments. However, when trying to convince planners of our improved platforming and routing solution, we have to show both, that there are problems present in the current solution and that no such problems are present in the optimized solution. Typically, the best way to illustrate this to planners is to use a graphical presentation. So, for the first task, we use a picture as in Figure B.1. It shows the current planning, here for Brugge station. Platforms are arranged from I to X along the vertical axis and the horizontal axis represents time. Small brown rectangles show trains entering the station and their IN-route durations. Blue rectangles show trains leaving the station and their OUT-route durations. Between them, the yellow rectangles show the platform occupations and their durations. No overlap between rectangles should exist on any platform. Currently, at Infrabel, a similar figure is made with paper and pencil and double use of a platform can be visually checked already. However, new checks in Figure B.1, are the green, orange and red lines. Leopard draws these lines automatically between each couple of trains of which the second reuses the same or a dependent route within a time d less than 5 minutes. So each such line marks a potential problem which can become an actual problem if the first train gets delayed with a time d. If d is negative or 0, even without delay, we already have two trains on the same route at the same time. Figure B.1 shows 4 such cases ($d = -0.6, -0.5, -0.2, -0.01$) in red. Two cases with $0 < d \leq 1$ occur ($d = +0.2, +1.0$) in dark orange. Light orange means $1 < d \leq 2$ which is considered acceptable. Green means $2 < d \leq 5$ which is seen as robust enough.

To show planners that the optimized solution has no such $d \leq 1$ problems, we show a picture like Figure B.2. Dependent-route reuse lines now only have positive values for d. Here, the only ones below 2 minutes are $d = 1.8, 1.8, 1.9, 1.9$. However, one occupation ends up on the fictive platform due to route conflicts with other occupations for any assignment to a real platform.

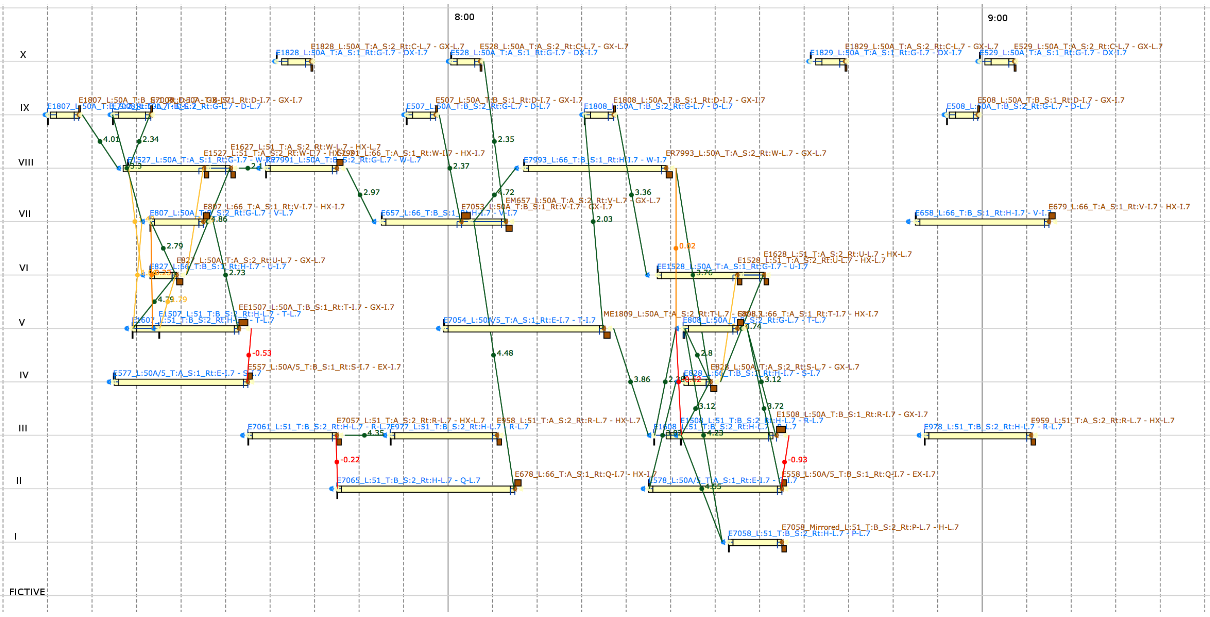

Figure B.1: Brugge station platform occupations. Original schedule. All occupations are assigned to real platforms, but 4 cases of simultaneous dependent-route use (red lines) and 2 cases of dependent-route reuse within 1 minute (dark orange lines) occur.

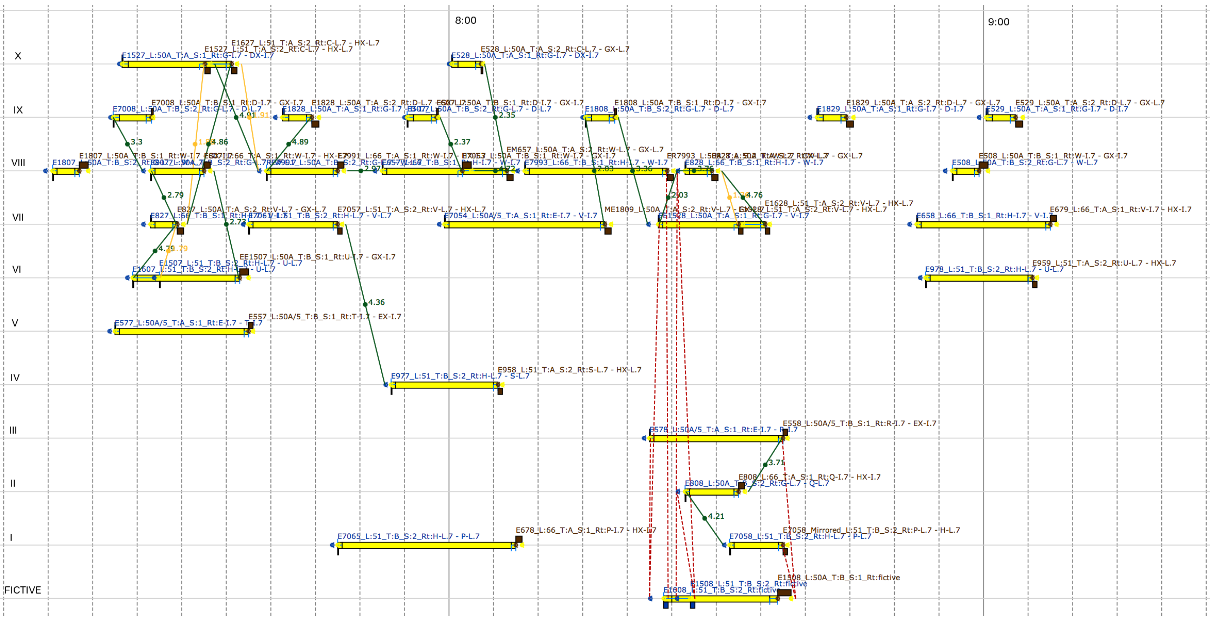

Figure B.2: Brugge station platform occupations. Progressively optimized schedule. No cases of simultaneous dependent-route use nor of dependent-route reuse within 1 minute occur. However, one occupation cannot be put on a real platform and ends up on the fictive platform.

In case an occupation is assigned to the fictive platform, dashed red lines indicate route occupations that occur at overlapping times. At least one of these represents an actual conflict if the occupation would be assigned to a real platform. In order to be able to assign this occupation to a real platform, for now, one has to manually adapt some platform times and run Leopard again. We shifted the platform time of one train with +1.5 minutes and of another

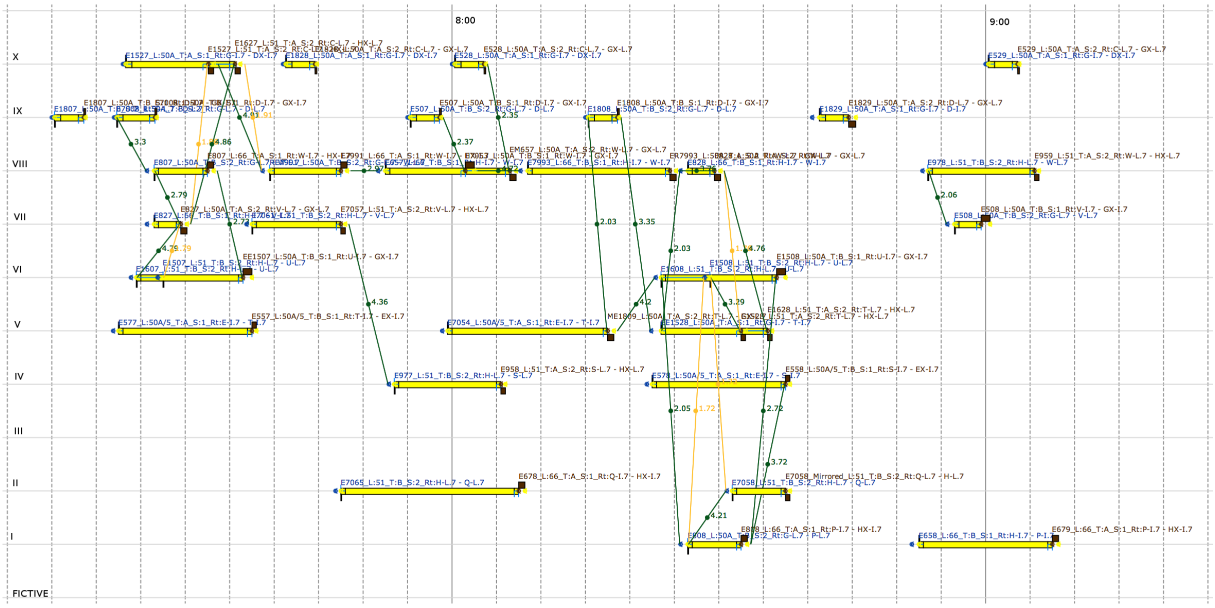

Figure B.3: Brugge station platform occupations with manually, only 2 slightly adapted platform times. Progressively optimized schedule. All occupations are assigned to real platforms and the fastest dependent-route reuse occurs with d_{min} = 1.7 minutes, which is acceptable.

with −1 minute. The result is shown in Figure B.3. This solution assigns all trains to real platforms, so is conflictless, but is also robust (d_{min} = 1.7) and hence a better platforming solution than the solution in Figure B.1.

We have shown, here for Brugge, with Figure B.1, that conflicts exist in the current schedule, with Figure B.2, that one must change platform times or not all trains can be platformed and with Figure B.3, that a conflictless and robust solution is found that platforms all trains when one cleverly and slightly changes a few platform times. Of course these station platform time changes must also take into account timetable constraints on the macroscopic level.

References

Billionnet, A., 2003. Using Integer Programming to Solve the Train Platforming Problem. Transportation Science 37 (2), 213–222.

Burkolter, D. M., 1 2005. Capacity of Railways in Station Areas using Petri Nets. Ph.D. thesis, Swiss Federal Institute of Technology Zürich.

Caimi, G., 1 2009. Algorithmic decision support for trains scheduling in a large and highly utilised railway network. Ph.D. thesis, ETH Zurich.

Caprara, A., 2010. Almost 20 Years of Combinatorial Optimization for Railway Planning: from Lagrangian Relaxation to Column Generation, In Thomas Erlebach and Marco Lbbecke (eds.), Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization and Systems (ATMOS), Schloss Dagstuhl, Germany. ATMOS 2010 1, 1–12.

Caprara, A., Galli, L., Toth, P., 2011a. Solution of the Train Platforming Problem. Transportation Science 45 (2), 246–257.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 1 2007. Passenger Railway Optimization. Handbooks in Operations Research and Management Science. Vol. 14. Elsevier, Amsterdam, 129-187.

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., 1994a. A model and strategy for train pathing with choice of lines, platforms and routes. Transportation Research Part B 28 (5), 333–353.

Carey, M., 1994b. Extending a train pathing model from one-way to two-way track. Transportation Research Part B 28 (5), 395–400.

Carey, M., Carville, S., 2000. Testing schedule performance and reliability for train stations. Journal of the Operational Research Society 51, 666–682.

Carey, M., Carville, S., 2003. Scheduling and platforming trains at busy complex stations. Transportation Research Part A 37 (3), 195–224.

Carey, M., Crawford, I., 2005. Scheduling trains on a network of busy complex stations. Transportation Research Part B 41 (2), 159–178.

Carey, M., Lockwood, D., 1995. A model, algorithms and strategy for train pathing. Journal of the Operation Research Society 46 (8), 988–1005.

Chakroborty, P., Vikram, D., 2008. Optimum assignment of trains to platforms under partial schedule compliance. Transportation Research Part B 42 (1), 169–184.

Cordeau, J.-F., Toth, P., Vigo, D., 1998. A Survey of Optimization Models for Train Routing and Scheduling. Les Cahiers du GERAD 1 (1), 1–42.

De Luca Cardillo, D., 1998. k L-List τ Colouring of Graphs. European Journal of Operational Research 106 (1), 160–164.

Delorme, X., Rodriguez, J., Gandibleux, X., 2001. Heuristics for railway infrastructure saturation. Electronics Notes in Theoretical Computer Science 50 (1), 39–53.

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2013. Improving the robustness in railway station areas. European Journal of Operational Research: accepted for publication.

Fuchsberger, M., 1 2007. Solving the train scheduling problem in a main station area via a resource constrained space-time integer multi-commodity flow. Master's thesis, Institute of Operations Research, ETH Zurich.

Galli, L., 1 2009. Combinatorial and Robust Optimisation Models and Algorithms for Railway Applications. Ph.D. thesis, University of Bologna.

Ghoseiri, K., Szidarovski, F., Asgharpour, M. J., 2004. A multi-objective train scheduling model and solution. Transportation Research Part B 38 (10), 927–952.

Herrmann, T. M., 1 2006. Stability of Timetables and Train Routings through Station Region. Ph.D. thesis, Swiss Federal Institute of Technology Zürich.

Kroon, L., Romeijn, H., Zwaneveld, P. J., 1997. Routing Trains Through Railway Stations: Complexity Issues. European Journal of Operational Research 98 (3), 485–498.

Lusby, R. M., 1 2008. Optimization Methods for Routing Trains through Railway Junctions. Ph.D. thesis, University of Auckland.

Lusby, R. M., Larsen, J., Ehrgott, M., Ryan, D., 2011a. Railway track allocation: models and methods. OR Spectrum 2011 33 (4), 843–883.

Lusby, R. M., Larsen, J., Ehrgott, M., Ryan, D., 2013. A set packing inspired method for real-time junction train routing. Computers and Operations Research 40 (3), 713–724.

Lusby, R. M., Larsen, J., Ryan, D., Ehrgott, M., 2011b. Routing Trains Through Railway Junctions: A New Set-packing Approach. Transportation Science 45 (2), 228–245.

Miao, J., Yu, Y., Meng, L., Yang, Z., 2012. Stability-oriented Optimization of Train Platforming Problem on High Speed Railway. Journal of Transportation Systems Engineering and Information Technology 12 (3), 115–121.

Sels, P., Sep. 2012. milp-logic: a C++ MILP Solver Abstraction Layer with a C++ Boolean Modelling Layer on Top.
URL http://code.google.com/p/milp-logic

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., Feb. 2011. 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.
URL http://4c4u.com/RR2011.pdf

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., May 2013. 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.
URL http://4c4u.com/RC2013.pdf

Vansteenwegen, P., Van Oudheusden, D., 2006. Developing Railway Timetables that Guarantee a Better Service. European Journal of Operational Research 173 (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), 478–492.

Williams, H., 1 1994. Model Building in Mathematical Programming, 3rd Edition. Vol. 1. John Wiley and Sons Ltd, England.

Zwaneveld, P. J., 1997. Railway Planning and Allocation of Passenger Lines. Ph.D. thesis, Rotterdam School of Management.

Zwaneveld, P. J., 2001. Routing Trains through a Railway Station based on a Node Packing Model. European Journal of Operations Research 128 (1), 14–33.

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.