Automated Platforming & Routing of Trains in All Belgian Railway Stations

P. Sels[a,b,c,*], D. Cattrysse[a,1], P. Vansteenwegen[a,2]

[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 train to platform assignments has been an active research area for some time, but systems implementing this research are still not readily available to practitioners. However, now, our train platforming model has been implemented as the tool *Leopard* inside Infrabel, the Belgian railway infrastructure manager.

In practice, initial macroscopic timetables are often not yet feasible inside stations on the microscopic level. This means that a platforming tool must be able to handle cases where not all trains can be platformed or routed. Our model provides a platforming and routing plan for as many trains as possible and puts the remaining trains on a fictive platform. Contrary to the manually made platforming plans, the optimised platforming plans have no platform conflicts nor routing conflicts. Our model assigns as much trains as possible, given the timetable and the available infrastructure. Our tool can solve the platforming problems for all 530 stations in Belgium together in about 10 minutes. This means (i) it saves many man months of planning time compared to the still common manual practice to platforming and (ii) it achieves higher quality results leading to significantly less in-station train delays in practice.

Keywords: Train Platforming Problem (TPP), Mixed Integer Linear Programming (MILP)

[*] Corresponding author
Email address: `sels.peter@gmail.com` (P. Sels)
[1] *Email address:* dirk.cattrysse@kuleuven.be
[2] *Email address:* pieter.vansteenwegen@kuleuven.be

1. Introduction and State of the Art

In railway operations, planning all operations accurately and reliably is important since train passengers rely on the timetable to plan their journeys. A sequence of four main planning stages is identified (Zwaneveld et al., 1996; Cordeau et al., 1998; Lusby et al., 2011). Firstly, train lines with certain frequencies and stopping patterns are decided upon. Secondly, train arrival and departure times are fixed for each train in each station and for each train, platforms and routes are also determined. Thirdly, train material is assigned to the lines and lastly, crew rostering is performed. This paper focusses on the platforming and routing of trains in stations. Narayanaswami and Rangaraj (2014) mention that problems occurring in railway planning typically contain complex interdependencies between multiple components and are operationally critical and that human resolutions of these problems are inconsistent, scale inefficient, and potentially infeasible. These are the reasons why we look for computer assisted resolutions.

1.1. Decomposition into Timetabling and Platform Planning

The train platforming problem (TPP) is essentially a vehicle routing problem that includes timing constraints. Vehicle routing problems are usually solved using local search, possibly augmented with techniques to escape from local minima like meta-heuristics (Barbucha, 2012). However, the subcategory of TPP has attracted special attention and different versions resulted in dedicated solution approaches. Quite some research papers discuss the problem of train platforming (TPP) as the problem of trying to fit all trains on platforms and decide on routings while also allowing to change arrival and departure times (Zwaneveld et al., 1996; Zwaneveld, 1997; Caprara et al., 2007, 2011; Carey, 1994a,b; Carey and Carville, 2000, 2003; Dewilde et al., 2013). While this solves the problem for one station, any changed arrival or departure time for a certain train in the considered station will require changes to train times in the previous and next station as well if not enough ride time buffer is present between these stations to absorb that change. This means that platform models with variable train arrival and departure times cannot be solved independently from the platforming problems of neigbouring stations. This is why the problem of solving platforming and timetabling of corridors of stations or small networks of neighbouring stations has been addressed by, amongst others, Carey and Crawford (2005) and Bešinović et al. (2015). These publications present examples of solutions for corridors of stations or small research networks of stations but do not report studies nor results for entire countries. We built an expert system to assist Infrabel in platforming all 530 stations of Belgium.

1.2. Improving the Platforming Approach

We believe our approach of platforming optimisation is an improvement to the current practice in railway planning, where we think platforming does not get the right amount of attention. We explain this by contrasting the current approaches to plaforming in Section 1.2.1 and our approach in Section 1.2.2.

1.2.1. The Conflict Detection and Train Delay Simulation Approach

To the best of our knowledge, *no commercially available tool* exists for the *automatic generation* of optimal platforming solutions. Infrabel possesses the tool Artemis which can *check* existing platforming solutions on the signal and block section level. However, Artemis cannot generate a platforming solution by itself. The situation at railway infrastructure companies in most other countries is not better. Goverde and Hansen (2013) mention that in France and Italy in 2013, a timetable was constructed, but no platform nor routing choices were made in the planning phase. This makes it impossible to know if two trains will be simultaneously present on the same platform or on the same or crossing routes until this happens in practice. Most countries choose a platform track and route for every train during the planning stage, so that they can detect conflicts as well as simulate the effect of these choices on the total network behaviour before the plans are put in practice. RMCon's RailSys and ViaCon's LUKS used in Germany and elsewhere (Rail Management Consultants GmbH (RMCon), 2016; ViaCon GmbH, 2008b) and OpenTrack used in Switzerland and many other countries (OpenTrack Railway Technology Ltd., 2012) are microscopic simulators that are used to evaluate a timetable - including the chosen platforms and routes in stations - with respect to total propagated train delay. A LUKS user can indicate chosen platform tracks along with routes (ViaCon GmbH, 2008a). However, currently, automatic platform and route plan generation is not possible with any of these simulators. Also, the routes indicated can only depend on the line a train entering a station comes from and the line a train leaving the station goes to. In busy traffic, other platform tracks than the default ones may have to be considered to be able to platform all trains.

The mentioned simulators propagate train delays through a train network model. These simulators detect platform and routing conflicts and robustness issues that result from a given planned timetable that includes default or explicitly chosen platform tracks and routings. The simulators can indicate these conflicts by writing out a list of conflicts. Also, in their simulations, these conflicts and robustness issues cause train delays which are then propagated through the train network in the calculations. As such, a simulator user can evaluate whether one timetable causes more total propagated delay than another. This approach of conflict detection and simulation which was not preceded by a phase of optimal platform assignment that guarantees conflict avoidance has two major drawbacks. Firstly, there is no back-annotation from the resulting train delays to what they were caused by. So, it remains difficult to know how much of this total delay is caused by the macroscopic timetable itself and how much is caused by any of the platform and route choices of any of the trains in the system. As a consequence, it is sheer impossible to know how to 'repair' the timetable or platforming plan to avoid or reduce total train delays, should they be considered too high. Secondly, these microscopic simulations take a lot of computer time. At Infrabel, the use of LUKS on the complete Belgian train network is considered impractical. As a consequence, simulations are restricted to subareas of about a tenth of the country. Obviously, these simulations could ignore some important dependencies between these subareas.

1.2.2. *The Conflict Avoidance by Optimisation Approach*

Practice shows that manually constructed platform plans sometimes still possess train conflicts and robustness issues. Conflict detection tools and train delay simulation tools indicate these problems but do not tell a planner how to fix these issues. Solving one issue can be easy, but coming up with a platform plan that solves all issues simultaneously can be a large combinatorial problem that is hard to solve for human planners. We consider it more efficient to directly try to construct timetables and platforming plans that are guaranteed to be conflict free. Conflict detection and train delay propagation can and should still be performed afterwards, but the whole process will then have to be iterated over less often. This should save a lot of time in the total timetabling and platforming process.

For this to work, macroscopic timetabling should guarantee the absence of 'macroscopic conflicts' and platforming should avoid all platform and routing conflicts inside stations. In earlier research (Sels et al., 2016), we constructed a method for macroscopic timetabling that produces timetables without these 'macroscopic conflicts', assuming headways of at least 3 minutes are feasible and assuming that the number of trains on each track section does not exceed capacity constraints. In this paper, we test our platforming method that plans trains on routes and platforms without generating conflicts inside stations. By adopting our two step method, we believe better timetabling and platforming plans would be more quickly obtained and, as a consequence, a lot of conflict detection and simulation time of 'wrong' timetable or platforming plans would be saved.

This idea is not new. The Dutch Railways (NS) possess a platforming tool called STATIONS (Zwaneveld et al., 1996; Zwaneveld, 1997) that does allow generation of platforming and routing solutions. STATIONS even allows some shifts on train arrival and departure times. When NS produces a new timetable, STATIONS is used on checking feasibility of platforming and routing in the larger stations. The French railway operator SNCF (Société National de Chemins de Fer) together with IFSTTAR (Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux) have developed a platforming solution called RECIFE (Delorme et al., 2001). RECIFE has mainly been used on the Pierrefitte-Gonesse junction near Paris and the Lille-Flandres station. None of these tools is commercially available. Infrabel also believes that even after negotiation about adopting these solutions and in the case of a positive outcome, too much adapting and integration work would have to be carried out. This means a new platform plan generating tool had to be developed.

1.3. *Time Constraints of a Platform Plan: Fixed versus Variable Arrival and Departure Times*

Because no expert systems exist yet that solve countrywide timetabling and platforming for all stations at once, Infrabel, like many other railway companies, prefers to decompose the planning work sequentially into macroscopic timetabling and train platforming in stations. This means that for the platforming problem, train arrival and departure times are supposed to be fixed. So, we take the following approach. Firstly, the nationwide macroscopic timetable is automatically constructed, taking care that all minimal ride times, dwell times and headway times are respected. This work was described in Sels et al. (2013, 2016). In the second phase, the construction of the platform plan is performed for each station separately, with arrival and departure times for all trains in all stations considered fixed. Some other train platforming research also considers the train arrival and departure time to be fixed and just chooses platforms and routes for all or as many as possible train movements (De Luca Cardillo, 1998; Delorme et al., 2001; Billionnet, 2003). To the best of our knowledge, these tools have not been applied yet to all stations in a country to verify the complete feasibility in all stations. Our platforming and routing model with fixed arrival and departure times is described in (Sels et al., 2014). If in a station, due to these fixed times, not all trains can be platformed, it is then considered the job of the human timetabler to slightly adapt some train arrival and/or departure times. Kroon et al. (1997) mention that often just a few small changes of the order of a minute or half minute results in a few more trains being able to be platformed. The appendix in Sels et al. (2014) gives an example of this. In our expert system, we have also taken care that the acquisition of routing information, which is typically specific to the railway company, is decoupled from the actual platforming module itself, which is the general problem that all railway companies face. As such, the platform module can be run independently on input generated at other railway companies. This should increase the potential use cases at other railway companies.

1.4. *The Objective of a Platform Plan*

Caprara et al. (2011) discuss many components that can be part of the objective of a platforming plan. They minimize an objective 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 compared to the macroscopic timetable that was needed in the platforming solution, the number of dynamic conflicts allowed, the number of trains assigned to a non preference platform and the number of trains assigned to a dummy platform. Infrabel currently prefers to just platform as many as possible of the planned trains in each station. Therefore, our objective function is the number of trains that are platformed on real platforms. Like other research, as for example Caprara et al. (2011), we model a *fictive platform* to hold any train movements that cannot be placed on real platforms. Naturally, we will penalise the assignment of a train to the fictive platform stronger than the assignment of a train to a real platform. Note that on the fictive platform track, multiple trains can be present at the same time. Thanks to the fictive platform, we will also be able to report which trains could not be assigned to a platform, if any, rather than just having to report the problem to be infeasible.

Our main research question of this paper is whether adding automated platforming to the process of designing a new timetable, can work in practice, and if so, what its benefits are. Can more trains be placed by the automatic platformer than is the case for the human planners? Are any problems still present in the manual plans and can these be resolved in the automatically generated platform plans? Does the computation of the platform plans of all stations take more or less time than the time spent manually platforming all essential stations? Is robustness of the solutions of the automatic platformer better or worse than the solutions for manual planners? Possible problems preventing the construction of a platforming solution could be that not all input data is present to compute such solutions. It could also be that tool users feel they lose too much control over platform and route assignment. These research questions will be answered after a detailed exposition of our method and the results achieved.

1.5. *Paper Overview*

In this paper, we demonstrate that our platforming model is generally applicable on any station in Belgium and that it both indicates problems in manually made platforming plans, if they already exist, as well as automatically and quickly produces platforming plans without these problems. Section 2 discusses the train platforming problem in some more detail. Section 3 describes the needed data input and user parameter input and shows the graphical output for an example station. Section 4 shows key performance indicators for application of our expert system on the hardest 18 stations. It also shows averages and totals for the indicators over all 530 stations. The Sections 5 and 6 answer the research questions, draw conclusions and indicate possible further work.

2. The Train Platforming Problem

Essentially, a platforming model has to map train traffic onto station infrastructure, taking care that no trains collide. We specify more in detail which input data is required, how it is obtained or derived and combined.

2.1. *Infrastructure Input*

Figure 1 schematically pictures a station and the infrastructure elements that have to be modelled to solve a platforming problem. These elements will now be discussed one by one. Figure 1 clearly shows that the TPP is to be distinguished from the train routing problem in shunting yards. Indeed, shunting yards are tree shaped bundles of tracks often connected to station areas. The shunting problem consists of recombining cars of freight trains into other sequences in the most efficient manner. This comes down to a shortest path problem with additional multiple domain-specific constraints (Adlbrecht et al., 2015).

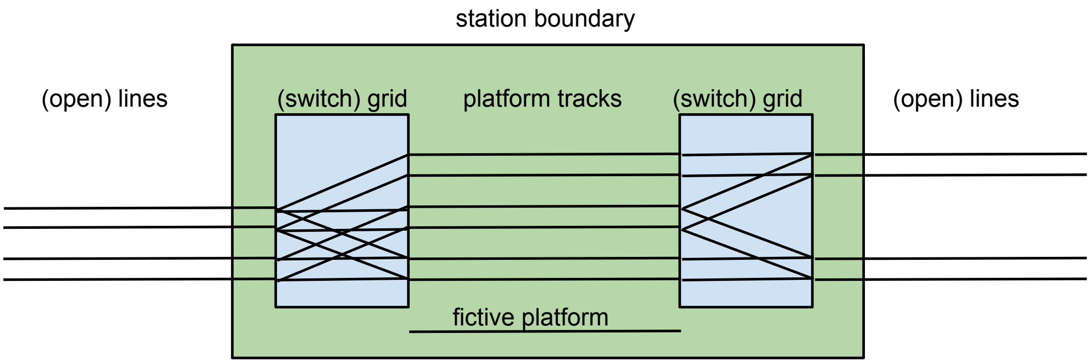


Figure 1: Platforming on Infrastructure

Lines, sometimes called *open lines*, are the railway tracks that trains ride on from station to station. To define a macroscopic timetabling problem, for each train type and composition, a minimum ride time has to be known. For the platforming problem in a station, the lines serve as entry or exit point of a station.

Platform tracks are the tracks that passengers have access to in a station. Passengers can board or alight a train when the train stops at a platform track.

An *in-route* is a path that connects a line to a platform track. An *out-route* is a path that connects a platform track to a line. Routes are located in *grids*, sometimes called *switch grids*. Since it is practical for platforms to be laid out parallel to each other, stations most commonly have two sides only and this typically leads to two switch grids. In Belgium, all stations can be captured by this abstract layout. For 'head stations', like Oostende, which have lines connected to it on only one side due to the presence of the North Sea on the other side, one of the switch grids degenerates to the empty switch grid. So these stations can be represented by our model as well. Note that in Figure 1 one has to additionally imagine that from each open line to the fictive platform track, there exists a *fictive route* as well. Similarly, to each open line, from the fictive platform track a fictive route exists as well. These have been left out to not overload the figure. Railway companies define routes on top of the infrastructure hardware. These routes are implemented in embedded software and define the set of paths along which trains can ride from (to) a certain line to (from) a given platform track.

To avoid train collisions on routes, it will be important to know if a pair of routes is totally separated or not. We call a pair of routes *dependent* if they share at least one part of the infrastructure. One example is a pair of crossing routes. A second example is a pair of identical routes. This dependency or independency of all pairs of routes will have to be derived by the software. How this is done is described in Section 2.3.

2.2. *Train Traffic Input: Movements and Occupations*

Train traffic is known in terms of train *movements* coming into the station and going out of the station. Each in-movement specifies a train number (ID), an input line and a station arrival time. Each out-movement specifies a train number (ID), an output line and a station departure time. For an in-movement, the platformer has to deduce a platform track and a compatible route between the in-line and platform-track. For an out-movement, the platformer has to deduce a platform track and a compatible route between platform track and the out-line.

Sets of in and out-movements are grouped into *occupations*. Each occupation implicitly specifies that all its movements have to occur on the same platform track. As such, a train pass and a train stop are occupations with each one in-movement and one out-movement. However, also train splits and train merges can be specified as occupations. Indeed, a train split could have one in-movement and two out-movements and a train merge could have two in-movements and one out-movement. Leopard does not pose any limits on grouping of movements into occupations, so even an occupation with say 4 in-movements and say 6 out-movements would be possible.

2.3. Platforming is Mapping Traffic on Infrastructure

One can now summarise the platforming problem as the challenge to choose, for each occupation, a certain platform track and for each movement in each occupation, a certain route that is connected to that chosen platform track. At any time, no two trains can be present on *equal* platform tracks. At any time, no two trains can be present on *dependent* routes.

A platforming system needs to know or derive the time it takes for each train to take each route. However, the Infrabel standard databases only contain information on speed restrictions and route lengths for some stations. Extra information, entered by Infrabel in the simulator LUKS, was used in Leopard to derive this information as well as derive when route variants are dependent.

2.4. Our Platforming Model

We derived and discussed an Integer Linear Programming (ILP) model in Sels et al. (2014). We refer the reader to Sels et al. (2014) for all details and a comparison with other platforming models. This model is also used in the current paper. The topic of the current paper is the application of this model on all stations in Belgium. The type of input needed for this tool and the type of output produced by it are described in Section 3 while the results of the application on all Belgian stations are given in Section 4.

3. Platformer Input and Output

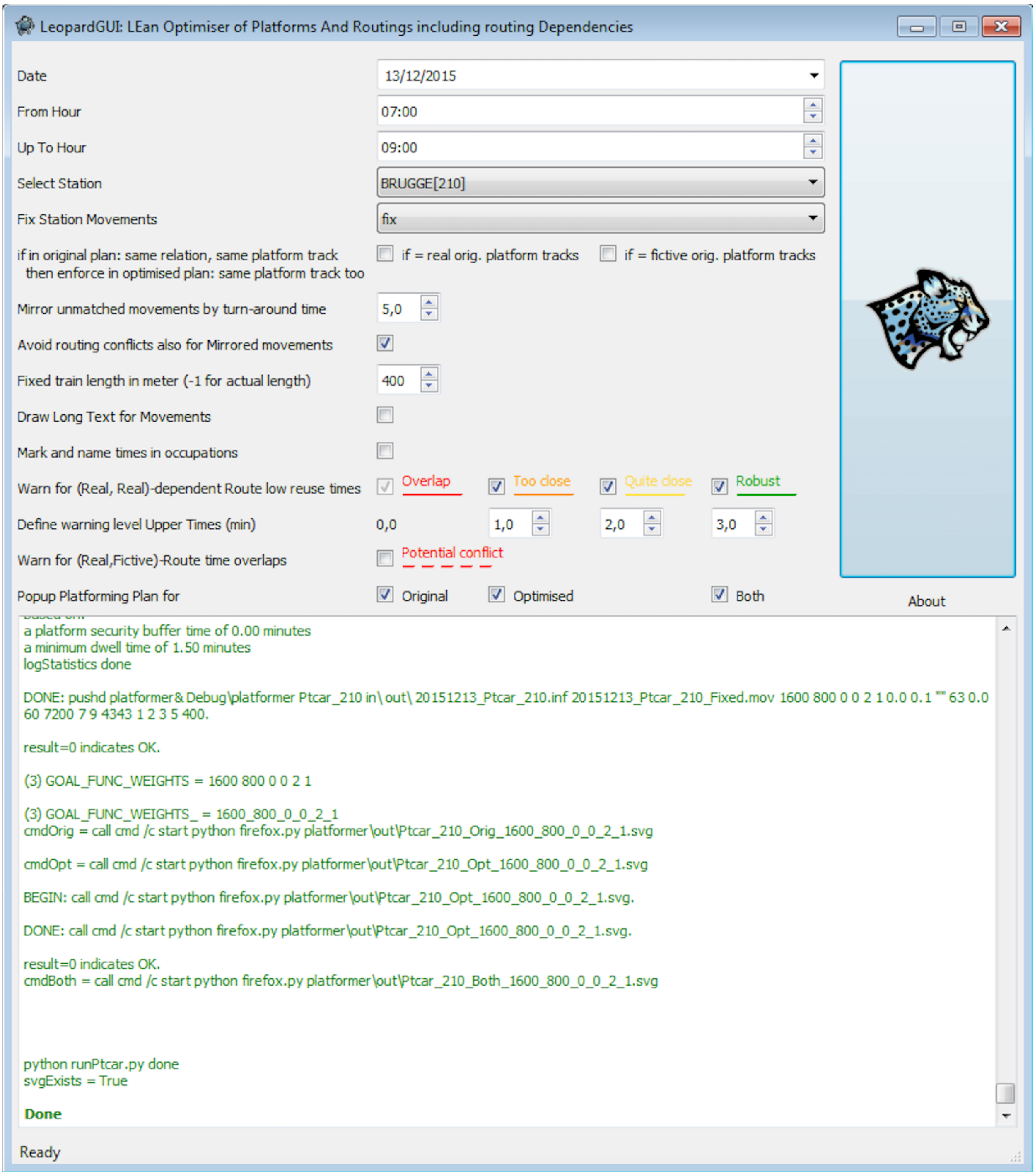
We describe our expert system Leopard from a user's perspective. So we discuss the necessary input required, also from the user, and the output Leopard produces.

3.1. Input from the User

Figure 2 shows the input part of the graphical user interface (GUI) of Leopard. The platformer user needs to specify a date and a begin and end hour. Only train traffic that is planned during this time window in the Infrabel databases will be platformed and routed. A station has to be selected as well from a listbox. These are the mandatory input parameters. The further input fields are optional.

The 'Fix Station Movements' field can be set to three values: 'fix', 'fixed' and 'unfix'. This is relevant for stations where for some movements an in-line or out-line is unknown. In those cases, when the user selected 'fix' here, the user will receive a pop-up window that asks the user to supply this missing input. This is done by selecting the line from a listbox containing all possible lines. For an in-movement, only in-lines can be selected and for an out-movement, only out-lines can be selected. The user's selected lines are stored. This allows, via the 'fixed' mode, that in subsequent runs of Leopard, the user does not need to reselect everything again. If the user wants to change any lines again, he will have to select the 'fix'-mode again. The third mode, 'unfix', is used to run Leopard in autonomous mode. No user input is then required and Leopard will then select a random in-line or out-line for movements lacking this information. This is only acceptable for temporary platform plans or in cases where relatively few movements are missing this line information. Final platform plans should ideally be generated with the modes 'fix' or 'fixed'.

The two checkboxes marked 'if = real orig. platform tracks' and 'if = fictive orig. platform tracks' refer to the conditions under which occupations of the same train relation are to be planned on the same platform track every hour. Leopard looks at the original, human-constructed platform plan (called the 'original platform plan'), if available, and assumes that if in this platform plan, a pair of occupations of the same train relation are planned on the same platform track, this should also be planned on the same platform track in the Leopard constructed platform plan (called the 'optimised platform plan'). The idea is that, if human planners have reasons to plan periodically, Leopard can be forced to do so as well. The left checkbox in Figure 2 requires this for two occupations of a train relation if they are



LeopardGUI: LEan Optimiser of Platforms And Routings including routing Dependencies

Date: 13/12/2015

From Hour: 07:00

Up To Hour: 09:00

Select Station: BRUGGE[210]

Fix Station Movements: fix

if in original plan: same relation, same platform track then enforce in optimised plan: same platform track too: □ if = real orig. platform tracks □ if = fictive orig. platform tracks

Mirror unmatched movements by turn-around time: 5,0

Avoid routing conflicts also for Mirrored movements: ☑

Fixed train length in meter (-1 for actual length): 400

Draw Long Text for Movements: □

Mark and name times in occupations: □

Warn for (Real, Real)-dependent Route low reuse times: ☑ Overlap ☑ Too close ☑ Quite close ☑ Robust

Define warning level Upper Times (min): 0,0 1,0 2,0 3,0

Warn for (Real,Fictive)-Route time overlaps: □ Potential conflict

Popup Platforming Plan for: ☑ Original ☑ Optimised ☑ Both

```
a platform security buffer time of 0.00 minutes
a minimum dwell time of 1.50 minutes
logStatistics done

DONE: pushd platformer& Debug\platformer Ptcar_210 in\ out\ 20151213_Ptcar_210.inf 20151213_Ptcar_210_Fixed.mov 1600 800 0 0 2 1 0.0 0.1 "" 63 0.0
60 7200 7 9 4343 1 2 3 5 400.

result=0 indicates OK.

(3) GOAL_FUNC_WEIGHTS = 1600 800 0 0 2 1

(3) GOAL_FUNC_WEIGHTS_ = 1600_800_0_0_2_1
cmdOrig = call cmd /c start python firefox.py platformer\out\Ptcar_210_Orig_1600_800_0_0_2_1.svg

cmdOpt = call cmd /c start python firefox.py platformer\out\Ptcar_210_Opt_1600_800_0_0_2_1.svg

BEGIN: call cmd /c start python firefox.py platformer\out\Ptcar_210_Opt_1600_800_0_0_2_1.svg.

DONE: call cmd /c start python firefox.py platformer\out\Ptcar_210_Opt_1600_800_0_0_2_1.svg.

result=0 indicates OK.
cmdBoth = call cmd /c start python firefox.py platformer\out\Ptcar_210_Both_1600_800_0_0_2_1.svg



python runPtcar.py done
svgExists = True

Done
```

Ready

Figure 2: Leopard input GUI

planned on the same real platform track. The right checkbox requires this for two occupations of a train relation if they are planned on the (same) fictive platform track. Naturally, requiring periodicity of platform planning will sometimes lead to optimised solutions that have fewer trains planned on real platform tracks.

The ‘Mirror unmatched movements by turn-around time’ field allows the user to indicate a number of minutes. This number is needed for the following cases. When a time window, say from 7am to 9am is specified, some occupations, say one from 6:58am to 7:02am will only partially fall within this time window. The in-movement at 6:58am will not be selected from the database. This means that the beginning in-movement of the occupation is unknown. We then need to construct a placeholder in-movement. This in-movement is generated from the known out-movement by supposing that it happens ‘x’ minutes before the out-movement. ‘x’ is the value that the user supplies in this GUI field and was set to 5.0 in Figure 2. Due to lack of any information about the line the train comes from, we suppose that the train is coming in from the same line as the one mentioned in the out-movement. The treatment for missing out-movements is entirely symmetrical.

Since mirrored movements are somehow invented ones, they can cause unrealistic conflicts with other movements. If one wants to ignore these conflicts, one can check the checkbox at ‘Avoid routing conflicts also for Mirrored movements’. If one want to also see the real movements and conflicts around the time window boundaries, another option is of course to specify a larger time window that fully includes all movements the user is interested in and possibly some more.

Leopard generates a visual representation of both the original platform plan and the optimised platform plan. It also generates a picture of both platform plans interleaved together. This last representation allows to easily compare what has changed between original and optimised platforming plans.

Checking the ‘Draw Long Text for Movements’ checkbox in the Leopard GUI results in the generated visual platform plan showing additional information (station side: 1/2, movement direction IN/OUT, track A/B) in the text describing the movements. For example, without this checkbox checked, a movement may be labeled as: ‘E7803’, while with this output checked, the same in-movement may be labeled as ‘E7803_S:1_D:I_T:A’, additionally indicating to the user that this movement occurs at station side 1, has the in direction (I) and occurs on track A. This is more informative but may generate quite some overlapping text with labels of other movements. In any case, this longer text is still dynamically displayed as a tooltip when the user hovers over this text.

‘Mark and name times in occupations’ demarcates and names the specific times and their abbreviations as defined in Sels et al. (2014) and reproduced in Table 1. Note that two subsequent rows in this table marked in the same colour indicate the same moment in time. For example, for an in-movement, *trho* = *tphi*, because the time the train head gets out of the IN route is equal to the time the train head heads into the connected platform. Similarly, *tpho* = *trhi* for an out-movement. Visualising this extra information will allow the user to understand the meaning of the time intervals better at first use. Since this can also generate text overlap, once the meaning is understood, we recommend to switch off this checkbox.

Table 1: Movement time abbreviations and definitions.

abbreviation	meaning
trhi	train enters (in) routing head in
trho	train leaves (in) routing head out
tphi	train enters platform head in
trto	train leaves (in) routing tail out
tparr	train platform arrival time, the time the middle of the train arrives at the middle of the station as planned in the timetable.
tpdep	train platform departure time, the time the middle of the train leaves the middle of the station as planned in the timetable.
tpho	train leaves platform head out
trhi	train enters (out) routing head in
tpto	train leaves platform tail out
trho	train leaves (out) routing head out
trto	train leaves (out) routing tail out

The four checkboxes for ‘Warn for (Real, Real)-dependent Route low reuse times’ allow the user to specify for which cases of subsequent reuse time levels, the user wishes Leopard to generate visual indications in the output platform plans. Leopard draws a line from the end time of a movement to the begin time of the next movement, if these movements are using dependent routes. It does so in a colour that depends on the closeness in time of these two events. If there is overlap, meaning the time difference between these events is zero or negative, the colour is red. If the time difference is between 0 and 1 minute, dark orange is chosen. For a difference between 1 and 2 minutes, light orange is chosen. For any time difference larger than 2 minutes and up to 3 minutes a green line is drawn. Of course, red indicates an error, dark and light orange indicate warnings that can be seen as lack-of-robustness issues. Green lines indicates dependent routes that are used by two subsequent trains with enough time in between to not cause frequent delay propagation. Note that all these overlap and robustness checks are performed on both the original and the optimised platform plans. In the optimised platform plans, the red lines do not occur at all, since the constraint that no two trains may use dependent routes at the same time is present in the mathematical model. However, dark and light orange lines may occur in the optimised plan since low robustness is not forbidden by hard constraints nor is it discouraged in the objective function.

In the GUI, in the minute entry boxes in the field 'Define warning level Upper Times (min)', the user can define the specific minute boundaries at which he considers reuse times should be shown in dark orange, light orange and green. These levels could be different in different stations or countries.

The field 'Warn for (Real, Fictive)-Route time overlap' checkbox has the following meaning. If Leopard generated an optimised platform plan where some occupations had to be put on the fictive platform track, this partial solution does not by itself tell the user what is the cause or are the combined causes of not being able to supply a total solution with all occupations on real platforms. We only visualise the 'simultaneity aspect' of these problems. So we checked the times of movements of fictive occupations and movements of real occupations and connected them with a red dashed line if they occur simultaneously. Any of these lines indicates a *possible* cause of the fictive occupation being on the fictive track.

Via the three checkboxes at 'Popup Platforming plan for', the user can indicate if he wants to see the original platforming plan, the optimised one and/or both together in one output graph. Examples of each of these output graphs are shown in Section 3.2.

3.2. Output to the User

As mentioned, Leopard generates up to three SVG files for the original, optimised and combined platform plans. We give an example of the first two types in Figure 3 and 4 respectively. These are both platforming plans for the station of Namur. Each figure shows the time axis as the horizontal axis. The platforms are enumerated from I to XI along the vertical axis. Train occupations are shown as yellow rectangles. A basic train occupation shows one in-movement at the start (left) of the rectangle and one out-movement at the end (right) of the rectangle. In-movements are shown in blue and out-movements in brown.

Figure 3 shows the manually created platform plan, also called the *original* plan. Around 6:26 on platform track IV, two red line segments indicate that train [ME4956 - E4956] and train [E904 - E928] are both present on this platform track at the same time. The red text '-6.00' indicates that this problem of overlap of time windows has a duration of 6 minutes. This overlap is a planning mistake in the original platform plan. Figure 3 shows dark and light orange lines as well, indicating dependent route reuse robustness issues, but these are warnings rather than planning errors. Around 6:37, a dark orange line connects the end of the outgoing train movement E9406, leaving platform track III, to the beginning of the outgoing train movement E928, leaving platform track IV. The text '0.42' indicates that 0.42 minutes pass between the time that the tail of the train E9406 leaves the route and the time the head of the train E928 enters a dependent route. This means that the route assignment of these two movements is not very robust. Similarly, around 6:52, only 0.61 minutes are available between the IN-movement E3805 towards platform IV and the OUT-movement E4578 leaving from platform track V. With the default user settings in Leopard, if the time between movements on dependent routes is lower than 1, 2 respectively 5 minutes, the line colour used is dark orange, light orange and green. If the time between movements on dependent routes is greater than 5 minutes, no lines are drawn anymore since these movements are considered far enough apart to not give rise to frequent delay propagation problems. In the manually produced platforming plans for other stations than Namur, for some movements on dependent routes, cases with time overlap were detected. These cases were indicated with red lines between the overlapping movements and a negative number then indicates the overlap time.

Figure 4 shows the optimised platforming plan that Leopard generated for the station of Namur. Clearly, there are no red lines, so there are no two trains using the same platform at the same time and neither are there dependent routes used at the same time by two different trains. There are still orange lines, but these were not explicitly forbidden nor discouraged by the model. Leopard was able to assign these four extra occupations to real platform tracks without conflicts by changing the platform and route assignment of some other train occupations. For stations like Namur, for which it is not possible to platform all trains, a solution with the maximum platformable set of trains, without generating any conflict, is reported.

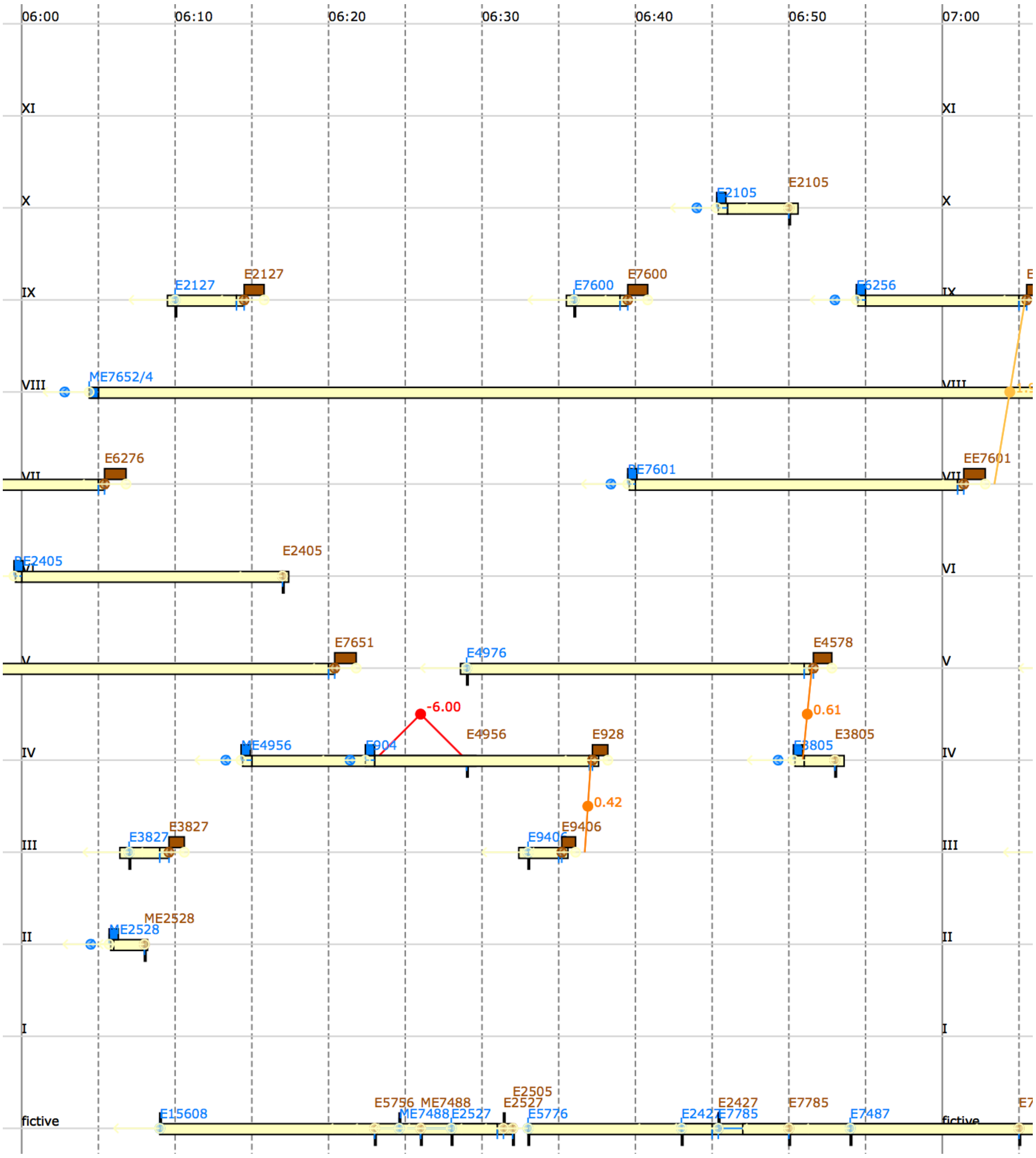


Figure 3: Namur original platform plan, created by human planners and drawn by Leopard

Our tool can also combine Figures 3 and 4 by interleaving them into one figure. These figures are intended to clearly show for which trains a different platform track and for which trains the same platform track has been chosen for the original and for the optimised assignment.

4. Results of Application on all Belgian Stations

In the previous section, we have shown that Leopard can be started from a GUI and have shown the graphical output that Leopard generates for one station, Namur. Leopard can also be run from the command line including

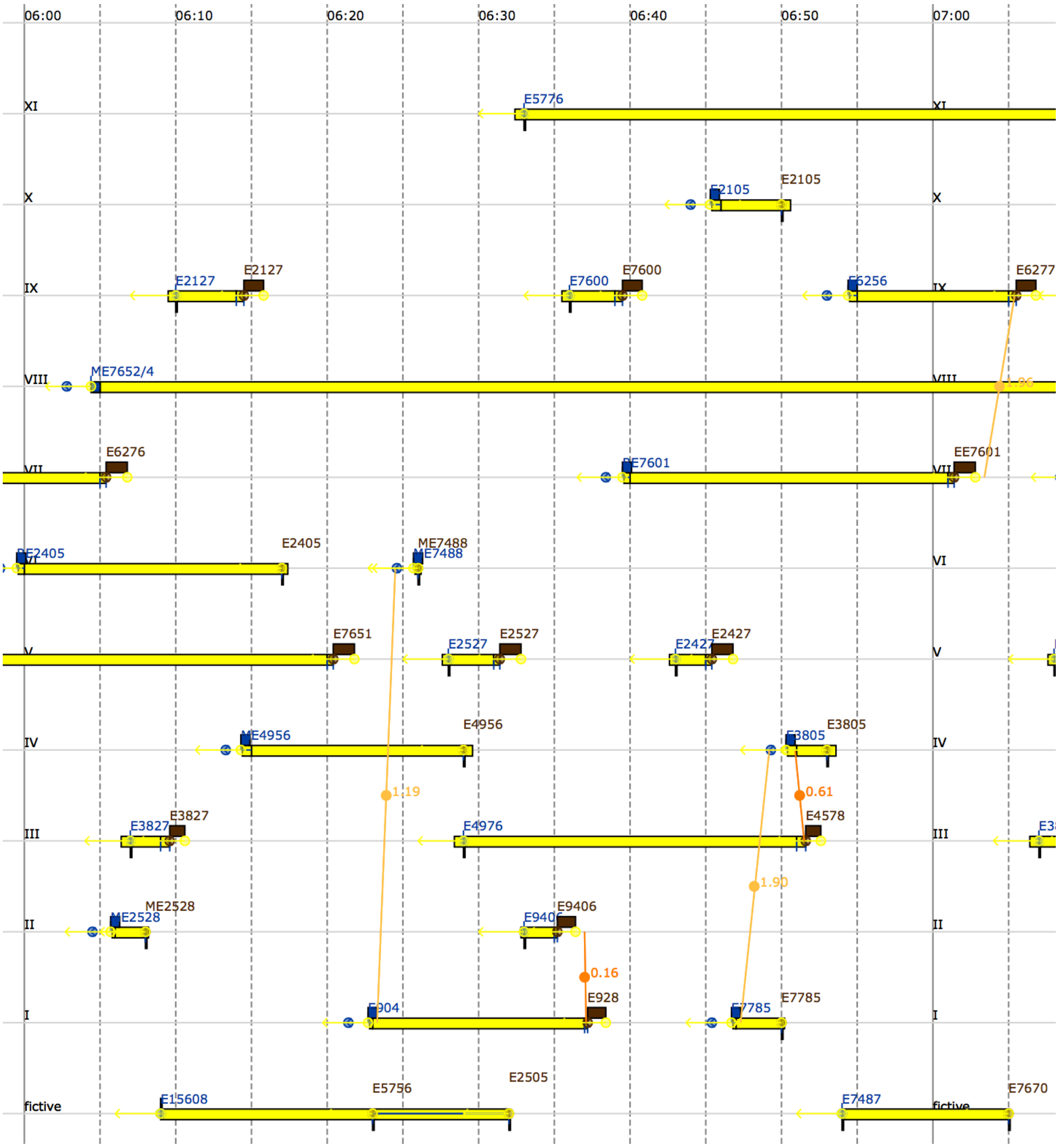


Figure 4: Namur optimised platform plan, generated and drawn by Leopard

the specification of all settings for parameters mentioned in the GUI. In that case, no user interaction is required. This allows creation of a script that runs Leopard in batch for all stations in Belgium. Such a script was run for all 530 stations in Belgium. Tables 2 and 3 show some parameters of original and optimised platform plans for the 16 stations that required the highest computation time for the first objective function (as shown in columns 3 and 4). The objective function for the optimised platform plan has been set to four different functions. First it was set to ‘progressive non-periodic’. ‘Progressive’ means that in the optimisation, deviation from the original platform assignments is not penalised. ‘Periodic’ means that occupations assigned to the same real platform tracks in the original platform plan and belonging to the same train relation are not forced to occur on the same platform track in the optimised platform plan. In the column headings of Tables 2 and 3, the mention of ‘Real:P’ signifies ‘Real Periodical’ and means that when two train movements belonging to the same train relation and occurring on the same platform track in the original platforming plan, these are also forced to occur on the same platform in the optimised platform plan. This happens via addition of extra hard constraints to the model. The ‘Real:P’ corresponds to the case where the GUI setting ‘if = real orig. platform tracks’ is checked. This was described in Section 3.1. The same holds for occupations of the same train relation that are both assigned to the fictive track in the original platform plan. So ‘Fict:P’ means Fictive Periodical. ‘Real:NP’ stands for ‘Real Non-Periodical’ and corresponds to the case where the GUI setting ‘if = real orig. platform tracks’ is unchecked and no corresponding periodicity constraints are added to the model.

The second objective function adds the deviation in number of platform tracks from the original to the optimised platform plan as extra terms. Since the objective function is always minimised, pressure is on the solver to produce solutions that contain platform choices not too far removed from what was chosen in the original platform plan. Since the goal of trying to platform as many trains on real platform tracks has a higher priority than staying close to the originally assigned platform track, a ‘big M’-coefficient was used to realise both sub-goals with a single objective function. This compound objective function was then once used with (Real:NP, Fict:NP), once with (Real:P, Fict:NP) and once with (Real:P, Fict:P).

For each objective function, for each station, an optimisation was run with Gurobi v6.0.0 (G columns) and one with FICO XPRESS v7.6 (BCL v4.6.1) (X columns). Gurobi was run on a 64 core server Intel Xeon CPU E5-4650 at 2.693 GHz with 268 GB of RAM running Windows Server 2008 R2 Enterprise while XPRESS was run on a 4 core desktop Intel Xeon CPU E31240 3.30GHz with 16 GB of RAM running Windows 7 Enterprise.

We now describe the results for all these different objective functions for the station of Aalst which is the first station given in Table 2. ‘unplatformed occupations/31’ in the first column and ‘4’ in the column of the original, human constructed platform plan (column 2) means that in Aalst, there are 4 occupations that were not assigned to real platform tracks in the original platform plan out of a total of 31 occupations in the window from 6am to 9am. The c(0,6) in column 2 means that there were 0 platform conflicts and 6 route-conflicts in the original platform plan of Aalst. Columns 3 up to 10 are about the optimised, Leopard generated platform plans and do not need to indicate conflicts since no conflict ever occurs in these optimised plans. Indeed, our model has hard constraints that forbid any of those conflicts. However, columns 3 up to 10 do indicate the number of occupations that are assigned to the fictive platform track, so are unplatformed. We see that columns 3 and 4 indicate that for Gurobi and for XPRESS, the same number of occupations are put on the fictive track. In the case of Aalst, there are 4 of these. Since for both solvers, (i) we solve the same model, (ii) up to optimality (MIP gap = 0%) and (iii) because of our safe ‘big M’-value, the same number of unplatformed occupations will always result per station for both solvers. So for two different solvers, the same platforming problem can result in two different solutions but these solutions will necessarily show the same number of train occupations on the fictive platform. This can indeed be verified in Table 2 and 3.

Columns 5 and 6 show that for the objective function that additionally also tries to reduce the deviation from platform tracks that were preferred in the original platform plan, the number of unplatformed occupations remains the same; 4. This is logical, since no extra hard constraints have been added from columns 3 and 4 to columns 5 and 6. However, if we add hard periodicity constraints to our model requiring that some occupations pairs have to occur on identical platform tracks, the number of unplatformed occupations can increase. This is illustrated in columns 7 to 10 which all show that 5 occupations could not be platformed for Aalst. This incidates that requiring periodicity can sometimes reduce the effective capacity of a station. The first row of the report on Aalst also shows the computation time in seconds which is between 0.0911 seconds and 0.153 seconds. This is very quick indeed. The rows ‘dark orange indications’, ‘light orange indications’ and ‘green indications’ show the number of lines between dependent routes that are drawn in the output pictures. The meaning of these lines as indicating robustness of the solution was described in detail in Section 3.2. How to control the thresholds via the GUI to change major and minor warning levels was described in Section 3.1. Tables 2 and 3 consistently show results for thresholds levels set to 1, 2 and 3 minutes as also indicated in Figure 2. We represent the number of red, dark orange, light orange and green lines with the tuple (r, do, lo, g). For Aalst, we see that the original platform plan in column 2 shows $(r, do, lo, g) = (6, 12, 9, 12)$ while this tuple is lowered to $(0, 9, 5, 10)$ for the Gurobi solution in column 3. This means that fewer pairs of movements on dependent routes are close together, which means that robustness of this Gurobi solution is better than for the original platform plan. We give an, admittedly somewhat artificial measure of robustness, by the always negative expression $R = -3^2 \cdot r - 2^2 \cdot do - 1^2 \cdot lo$. The green lines do not play a role in the

Table 2: Optimisation time in seconds. Stations 1 to 8 for train traffic from 6am to 9am. P = Periodical, NP = Non Periodical.

	6am - 9am	6am - 9am		6am - 9am		6am - 9am		6am - 9am	
	Original	Optimised		Optimised		Optimised		Optimised	
		Progressive		Min. Distance		Min. Distance		Min. Distance	
		Real:NP, Fict.:NP		Real:NP, Fict.:NP		Real:P, Fict.:NP		Real:P, Fict.:P	
1	2	3	4	5	6	7	8	9	10
		G	X	G	X	G	X	G	X
Aalst (#conflicts) resp. (s)	c(0.6) ($\leq 0'$)	0.0911	0.082	0.119	0.128	0.153	0.149	0.153	0.151
unplatformed occupations/31 ($\leq 0'$)	4	4	4	4	4	5	5	5	5
dark orange indications ($\leq 1'$)	12	9	12	9	9	9	9	9	9
light orange indications ($\leq 2'$)	9	5	4	5	5	6	6	5	6
green indications ($\leq 3'$)	12	10	9	7	7	8	8	10	8
robustness score	-111	-41	-52	-41	-41	-42	-42	-41	-42
Antw.-Centraal (#conflicts) resp. (s)	c(12,0) ($\leq 0'$)	3.53	6.909	2.63	2.943	5.84	8.690	7.31	8.641
unplatformed occupations/82 ($\leq 0'$)	0	6	6	6	6	9	9	9	9
dark orange indications ($\leq 1'$)	3	1	1	1	1	1	1	1	1
light orange indications ($\leq 2'$)	24	18	19	16	16	16	16	13	16
green indications ($\leq 3'$)	6	6	5	4	4	8	8	7	8
robustness score	-228	-22	-23	-20	-20	-20	-20	-17	-20
Antw.-Berchem (#conflicts) resp. (s)	c(0,0) ($\leq 0'$)	0.253	0.300	0.234	0.294	0.288	0.290	0.288	0.292
unplatformed occupations/140 ($\leq 0'$)	0	15	15	15	15	18	18	18	18
dark orange indications ($\leq 1'$)	0	0	0	0	0	0	0	0	0
light orange indications ($\leq 2'$)	0	0	0	0	0	0	0	0	0
green indications ($\leq 3'$)	7	0	0	0	0	0	0	0	0
robustness score	0	0	0	0	0	0	0	0	0
Braine-le-Compte (#conflicts) resp. (s)	c(0,1) ($\leq 0'$)	0.124	0.229	0.134	0.222	0.242	0.349	0.161	0.372
unplatformed occupations/37 ($\leq 0'$)	0	0	0	0	0	0	0	0	0
dark orange indications ($\leq 1'$)	0	6	9	1	1	0	0	6	0
light orange indications ($\leq 2'$)	9	5	3	5	5	6	6	3	6
green indications ($\leq 3'$)	1	3	4	3	3	3	3	1	3
robustness score	-18	-29	-39	-9	-9	-6	-6	-27	-6
Brugge (#conflicts) resp. (s)	c(1,1) ($\leq 0'$)	0.316	0.659	0.377	0.690	0.569	0.991	0.569	1.036
unplatformed occupations/50 ($\leq 0'$)	0	3	0	3	3	3	3	3	3
dark orange indications ($\leq 1'$)	4	3	2	2	2	2	2	2	2
light orange indications ($\leq 2'$)	10	13	11	9	9	9	9	12	9
green indications ($\leq 3'$)	7	3	7	4	4	4	4	7	4
robustness score	-51	-25	-19	-17	-17	-17	-17	-20	-17
Brussel-Lux. (#conflicts) resp. (s)	c(4,0) ($\leq 0'$)	0.916	0.504	0.415	0.551	0.432	1.197	0.432	1.226
unplatformed occupations/54 ($\leq 0'$)	0	0	0	0	0	0	0	0	0
dark orange indications ($\leq 1'$)	5	0	1	0	0	0	0	0	0
light orange indications ($\leq 2'$)	3	8	9	3	3	3	3	6	3
green indications ($\leq 3'$)	28	26	30	35	35	27	27	29	27
robustness score	-87	-8	-10	-3	-3	-3	-3	-6	-3
Brussel-Zuid (#conflicts) resp. (s)	c(12,12) ($\leq 0'$)	94.64	7200’	104.42	7199.82	862.86	7200’	2905.2	7200
unplatformed occupations/223 ($\leq 0'$)	0	8	8	8	8	13	13	13	13
dark orange indications ($\leq 1'$)	36	25	26	27	28	26	22	23	22
light orange indications ($\leq 2'$)	91	84	89	83	82	80	78	80	78
green indications ($\leq 3'$)	55	51	58	54	56	44	51	49	51
robustness score	-535	-184	-193	-191	-194	-184	-166	-172	-166
Brussel-Noord (#conflicts) resp. (s)	c(0,4) ($\leq 0'$)	18.92	17.56	20.96	34.60	135.52	619.42	224.51	649.03
unplatformed occupations/219 ($\leq 0'$)	0	39	39	39	39	61	61	61	61
dark orange indications ($\leq 1'$)	21	3	4	2	2	1	1	1	1
light orange indications ($\leq 2'$)	121	66	67	67	65	49	49	51	49
green indications ($\leq 3'$)	63	37	41	29	28	23	24	21	24
robustness score	-241	-78	-83	-75	-73	-53	-53	-55	-53

robustness score since the movements they connect are considered to be separated by enough time to be robust against delays. The lower this expression is, the worse the robustness is. For the original platform plan, this robustness score becomes $R_{orig} = -3^2 \cdot r - 2^2 \cdot do - 1^2 \cdot lo = -9 \cdot 6 - 4 \cdot 12 - 1 \cdot 9 = -111$. For the Gurobi optimised schedule this becomes $R_{opt} = -9 \cdot 0 - 4 \cdot 9 - 1 \cdot 5 = -41$. and for the XPRESS delivered solution, this gives $R_{opt} = -9 \cdot 0 - 4 \cdot 12 - 1 \cdot 4 = -52$. Robustness is better for both optimised platform plans than for the original one. Of course, the fact that 4 occupations

Table 3: Optimisation time in seconds. Stations 9 to 16 for train traffic from 6am to 9am. P = Periodical, NP = Non Periodical.

	6am - 9am Original	6am - 9am Optimised Progressive Real:NP, Fict.:NP		6am - 9am Optimised Min. Distance Real:NP, Fict.:NP		6am - 9am Optimised Min. Distance Real:P, Fict.:NP		6am - 9am Optimised Min. Distance Real:P, Fict.:P	
1	2	3	4	5	6	7	8	9	10
		G	X	G	X	G	X	G	X
Charleroi-Sud (#conflicts) resp. (s)	c(4,5) (≤ 0′)	0.909	2.94	3.58	1.962	2.79	4.532	3.06	4.472
unplatformed occupations/63 (≤ 0′)	2	5	5	5	5	6	6	6	6
dark orange indications (≤ 1′)	9	9	9	8	8	5	6	7	6
light orange indications (≤ 2′)	11	17	13	14	16	15	14	15	14
green indications (≤ 3′)	15	11	12	7	5	9	9	10	9
robustness score	-156	-53	-49	-46	-48	-35	-38	-43	-38
Denderleeuw (#conflicts) resp. (s)	c(0,31) (≤ 0′)	0.163	7200	1.64	217.24	4.37	16.01	3.42	16.20
unplatformed occupations/64 (≤ 0′)	0	19	19	19	19	25	25	25	25
dark orange indications (≤ 1′)	18	9	9	11	11	3	3	2	3
light orange indications (≤ 2′)	24	10	13	7	9	11	11	14	11
green indications (≤ 3′)	21	12	15	9	9	8	8	9	8
robustness score	-375	-46	-50	-51	-53	-23	-23	-22	-23
Dendermonde (#conflicts) resp. (s)	c(0,3) (≤ 0′)	0.174	0.257	0.288	0.477	0.579	0.951	0.579	0.955
unplatformed occupations/42 (≤ 0′)	0	3	3	3	3	3	3	3	3
dark orange indications (≤ 1′)	4	0	1	1	1	1	1	0	1
light orange indications (≤ 2′)	9	10	10	9	9	7	7	8	7
green indications (≤ 3′)	7	7	6	3	3	4	4	4	4
robustness score	-52	-10	-14	-13	-13	-11	-11	-8	-11
Flémale-Haute (#conflicts) resp. (s)	c(0,25) (≤ 0′)	1.110	3.963	2.20	2.067	3.06	2.377	1.044	2.41
unplatformed occupations/21 (≤ 0′)	0	5	5	5	5	6	6	6	6
dark orange indications (≤ 1′)	9	7	7	7	7	7	7	7	7
light orange indications (≤ 2′)	8	5	3	5	5	5	5	5	5
green indications (≤ 3′)	12	9	13	8	8	8	8	9	8
robustness score	-269	-33	-31	-33	-33	-33	-33	-33	-33
Gent St.-Pieters (#conflicts) resp. (s)	c(0,12) (≤ 0′)	1.100	1.932	1.94	1.564	2.92	4.292	3.64	. 4.31
unplatformed occupations/97 (≤ 0′)	1	8	8	8	8	12	12	12	12
dark orange indications (≤ 1′)	23	19	18	14	14	10	10	11	10
light orange indications (≤ 2′)	23	17	16	22	22	17	17	17	17
green indications (≤ 3′)	28	26	20	16	16	19	19	12	19
robustness score	-223	-93	-88	-78	-78	-57	-57	-61	-57
Hasselt (#conflicts) resp. (s)	c(0,5) (≤ 0′)	0.310	0.754	0.357	0.526	0.374	0.694	0.769	0.677
unplatformed occupations/47 (≤ 0′)	0	0	0	0	0	0	0	0	0
dark orange indications (≤ 1′)	7	11	9	8	8	8	9	10	9
light orange indications (≤ 2′)	11	12	11	10	10	9	9	11	9
green indications (≤ 3′)	12	9	14	13	13	13	13	9	12
robustness score	-84	-59	-47	-42	-42	-41	-45	-51	-45
Kortrijk (#conflicts) resp. (s)	c(0,4)(≤ 0′)	0.298	0.371	0.723	0.530	0.556	0.592	0.306	0.584
unplatformed occupations/47 (≤ 0′)	0	3	3	3	3	7	7	7	7
dark orange indications (≤ 1′)	4	2	3	2	2	1	1	1	1
light orange indications (≤ 2′)	11	13	12	11	11	9	9	9	9
green indications (≤ 3′)	17	10	11	10	10	7	7	11	7
robustness score	-63	-21	-24	-19	-19	-13	-13	-13	-13
Leuven (#conflicts) resp. (s)	c(0,1) (≤ 0′)	1.052	3.532	1.76	1.250	2.92	4.386	1.610	4.36
unplatformed occupations/84 (≤ 0′)	0	0	0	0	0	6	6	6	6
dark orange indications (≤ 1′)	5	8	7	7	7	5	5	7	5
light orange indications (≤ 2′)	9	7	12	9	9	9	9	11	9
green indications (≤ 3′)	17	14	12	17	17	15	15	12	15
robustness score	-38	-39	-40	-37	-37	-29	-29	-39	-29

are not planned in the optimised platform plans could play a role in this. In the end, the manually planned platform plans, like the optimised platform plans, should also become void of platform and route conflicts and then robustness comparison between them will be fair.

The other stations in Tables 2 and 3 show that optimisation times are mostly less than a few seconds. Brussel-Zuid and Brussel-Noord are the only exceptions to this. They are the largest stations both in terms of platform tracks and train traffic. Antwerp-Centraal is large as well in both respects but is composed of three physical levels that have little interaction with each other, which simplifies the platforming problem. Solving Brussel-Zuid, depending on the chosen objective function, takes from 95 up to 2905 seconds for Gurobi and always 2 hours for XPRESS. We chose two hours to be the maximum allowed optimisation time limit and for XPRESS it means that no optimal solution is reported. Even then, we see that the same amount of unplatformable occupations is reported as for the Gurobi reported solutions. Solving Brussel-Noord takes between 19 and 225 seconds for Gurobi and between 18 and 649 seconds for XPRESS. When looking at the number of platform conflicts and route conflicts reported in the original platforming plans in column 2 in the c(.,.) notation, one remarks that there are usually only very few platform overlap problems. In fact, the only stations with platform conflicts are Antwerp-Centraal which has 12, Brugge (1), Brussel-Luxembourg (4), Brussel-Zuid (12) and Charleroi-Sud (4). When platforming manually, the used drawings directly show these overlaps, so this must mean that these manual plannings are work in progress.

In the original, manually produced platform plans, the number of route conflicts is not zero for a number of stations: Aalst has 6, Brain-le-Compte has 1, Brugge has 1, Brussel-Zuid has 12, Brussel-Noord has 4, Charleroi-Sud has 5, Denderleeuw has 31, Dendermonde has 3, Flémale-Haute has 25, Gent Sint-Pieters has 12, Hasselt has 5, Kortrijk has 4 and Leuven has 1. These conflicts can sometimes be caused by mirrored movements at the boundaries in which case they are not actual conflicts. In the other cases they still have to be inspected and corrected by the timetablers. In the current practice of planning without Leopard, these route conflicts are not shown in any visualisation, so this makes Leopard a useful addition to the platforming tools. The objective of the final timetable and associated platforming plans for all stations should be to have platforming plans that have no single platforming conflict and no single route conflict. Leopard is the ideal tool to automatically check that. In any case, whether the original platforming solution contains conflicts or not, an optimised solution is also calculated. This solution has to be seen as the best possible solution without conflicts. It can be used as a suggestion for improvement in case of conflicts in the original solution.

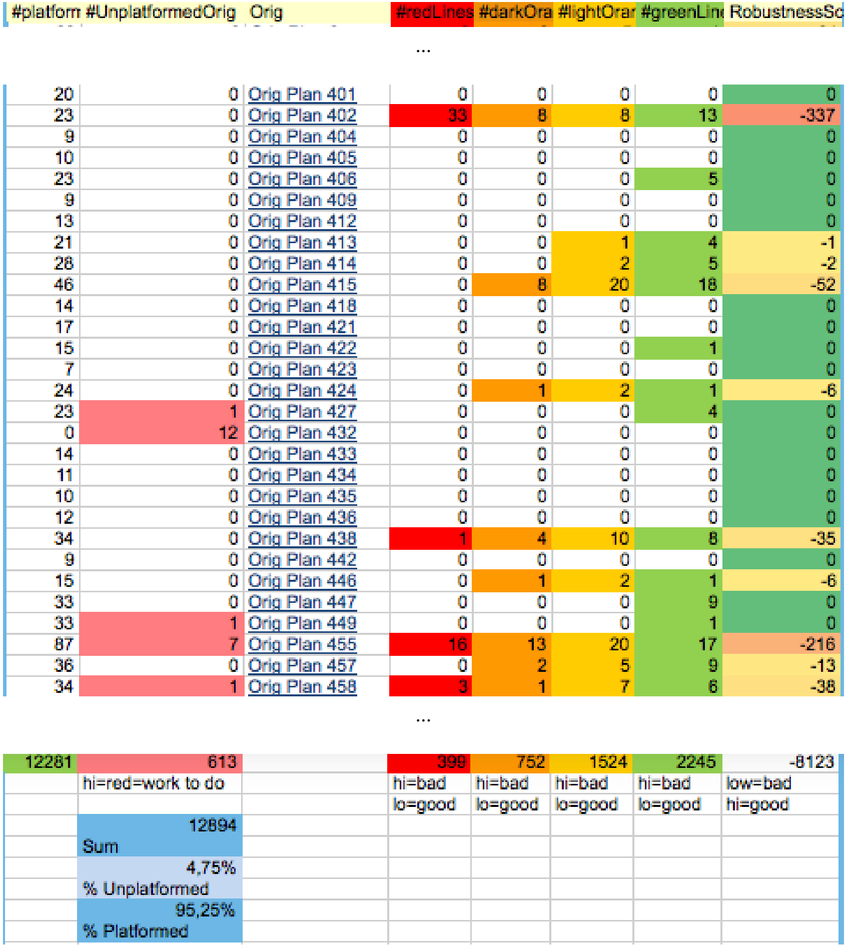
To allow timetabling and platforming practitioners at Infrabel to evaluate the platformability of a new timetable and also inspect some more specifics of the automatically generated platforming plans in all stations easily, Leopard can automatically generate an overview spreadsheet of this information for all stations. A partial extract of such a spreadsheet is shown in Table 4 for the original platform plans and Table 5 for the optimised platform plans. These figures show the same properties as given in Tables 2 and 3, but now every row represents one station. The cells showing the number of red, dark orange, light orange and green lines are now also coloured in the corresponding colour. The blue underlined text Orig_Plan_401, Opt_Plan_401 and Both_Plan_401 are clickable links that will popup respectively the original, optimised and combined platform plans for station number 401 in a browser. These platform plans look like the figures 3, 4 and the combination of both. So, say if a user sees a number of platform or route conflicts for a station he wants to inspect in more detail, he can just click the corresponding link.

The bottom section of Tables 4 and 5 show the totals and averages of all KPIs. For a different application date, a one hour window and a different objective function from Tables 2 and 3 it is shown that in the original platforming plan, 4.75% of all occupations across all stations remained unplatformed while in the optimised platform plans, this was only 1.96%.

Table 6 gives some total and average values of properties for all 530 Belgian stations together. The optimisation time limit per station was set to 7200 seconds. If no solution is found in that time, the station is counted as needing 7200s in calculation of totals or averages. Note that XPRESS can end after 7200 seconds, reporting a solution with a gap higher than the required gap, while in the same situation, Gurobi considers this not as a valid solution and returns no solution. These cases for XPRESS are marked with 'no', which stands for not optimal. For Gurobi these cases are marked with '>7200', but no such case occurs.

The first row of Table 6 shows the number of stations that could be solved for each of the four objective functions. For Gurobi, this is always 530, while for XPRESS this is always 528. XPRESS does return the model to be invalid if no train traffic is present in the specified time window and this is the case for 2 small stations. As mentioned, for the XPRESS solver, for some instances the solutions are not optimal down to gap 0% since at the time limit of 7200 seconds a non-optimal solution is allowed to be returned. Gurobi solves all 530 stations in less than 7200 seconds optimally. The total optimisation time over all stations together for Gurobi is significantly smaller than the time needed for XPRESS, but this is mainly due to just a few stations on which XPRESS spends 7200 seconds. Over the four objective functions, Gurobi spends an average computation time of 0.25 up to 5.97 seconds, while XPRESS spends from 14.16 to 27.42 seconds on the same. Averaged over all 530 stations, XPRESS seems to find the

Table 4: Overview spreadsheet for comparison of original and optimised platform plans and their KPIs: original part

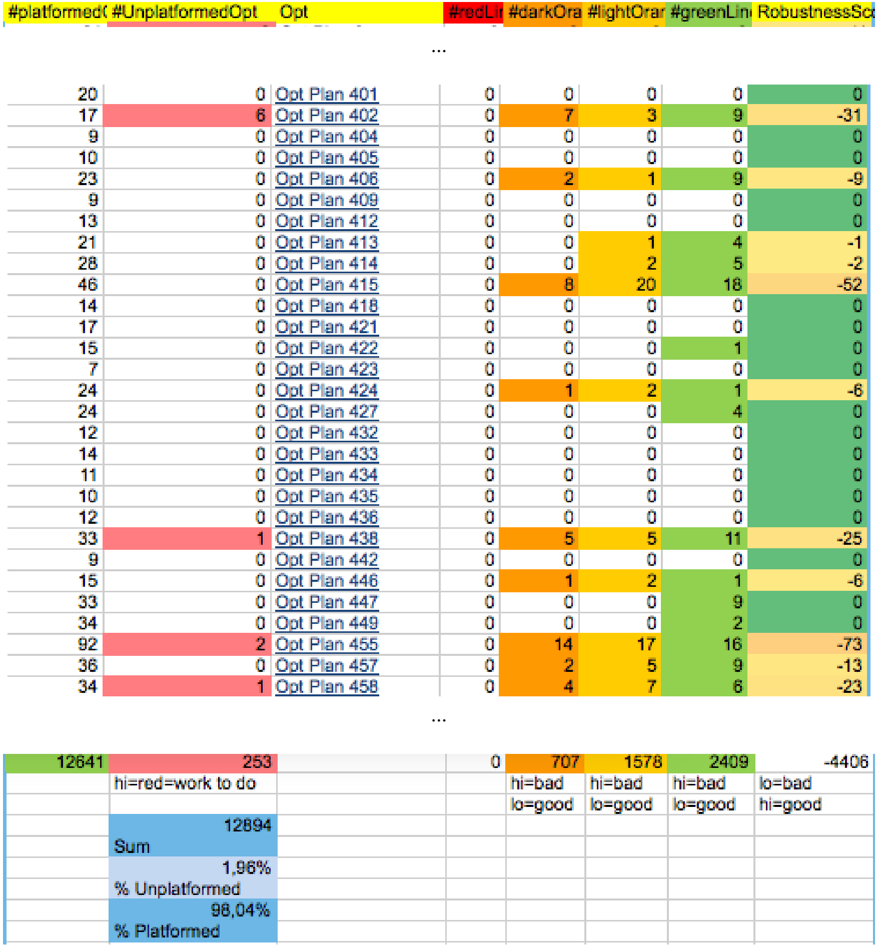


#platform	#UnplatformedOrig	Orig	#redLines	#darkOra	#lightOrar	#greenLin	RobustnessSc
...							
20	0	Orig Plan 401	0	0	0	0	0
23	0	Orig Plan 402	33	8	8	13	-337
9	0	Orig Plan 404	0	0	0	0	0
10	0	Orig Plan 405	0	0	0	0	0
23	0	Orig Plan 406	0	0	0	5	0
9	0	Orig Plan 409	0	0	0	0	0
13	0	Orig Plan 412	0	0	0	0	0
21	0	Orig Plan 413	0	0	1	4	-1
28	0	Orig Plan 414	0	0	2	5	-2
46	0	Orig Plan 415	0	8	20	18	-52
14	0	Orig Plan 418	0	0	0	0	0
17	0	Orig Plan 421	0	0	0	0	0
15	0	Orig Plan 422	0	0	0	1	0
7	0	Orig Plan 423	0	0	0	0	0
24	0	Orig Plan 424	0	1	2	1	-6
23	1	Orig Plan 427	0	0	0	4	0
0	12	Orig Plan 432	0	0	0	0	0
14	0	Orig Plan 433	0	0	0	0	0
11	0	Orig Plan 434	0	0	0	0	0
10	0	Orig Plan 435	0	0	0	0	0
12	0	Orig Plan 436	0	0	0	0	0
34	0	Orig Plan 438	1	4	10	8	-35
9	0	Orig Plan 442	0	0	0	0	0
15	0	Orig Plan 446	0	1	2	1	-6
33	0	Orig Plan 447	0	0	0	9	0
33	1	Orig Plan 449	0	0	0	1	0
87	7	Orig Plan 455	16	13	20	17	-216
36	0	Orig Plan 457	0	2	5	9	-13
34	1	Orig Plan 458	3	1	7	6	-38
...							
12281	613		399	752	1524	2245	-8123
	hi=red=work to do		hi=bad	hi=bad	hi=bad	hi=bad	low=bad
			lo=good	lo=good	lo=good	lo=good	hi=good
	12894						
	Sum						
	4,75%						
	% Unplatformed						
	95,25%						
	% Platformed						

progressive mode of optimisation the hardest. It spends 27.42 seconds on it on average per station. Gurobi only spends 0.25 seconds on this. XPRESS spends relatively less time on objective functions that contain terms to minimise the deviation of the original platform tracks. Gurobi finds the addition of the periodic platforming constraints and the minimisation of the deviations from the original platform tracks relatively harder since it spends more computation time if these are added to the model.

In the second block of Table 6, the number of platformed, unplatformed and percentage of unplatformed occupations are given. Columns 3 to 6 indicate that there are 793 unplatformed occupations for both solvers. When adding periodicity constraints to the platforming mode in columns 7 to 10, this number rises to 1045. The original platform plans have 4.98% of occupations platformed and the optimised non-periodical ones about 6.36% and the optimised periodical ones about 8.38%. Note again that the original platform plans can more easily platform more occupations since they still contain quite some conflicts while the optimised platforming plans are not allowed to contain any conflict. This is indeed shown in the last horizontal block of Table 6, where column 1 shows that there are 158 platform conflicts and 187 routing conflicts in the original platform plans of all stations together. The optimised platform plans are verified to contain no conflicts as can be seen by the zeroes in columns 3 to 10. As for the collected numbers of dark orange indications between movements on dependent routes, the fourth last row of Table 6 shows that these are significantly lower for both Gurobi and XPRESS columns than the numbers for the original platform plans in column 1. The situation is entirely similar for the light orange and green indications. This means that robustness should be improved from original to optimised platform plans and the last row of Table 6 confirms this indeed by its improved (higher) robustness scores.

Table 5: Overview spreadsheet for comparison of original and optimised platform plans and their KPIs: optimised part



#platformed(	#UnplatformedOpt	Opt	#redLi	#darkOra	#lightOrar	#greenLin	RobustnessSc

...

20	0	Opt Plan 401	0	0	0	0	0
17	6	Opt Plan 402	0	7	3	9	-31
9	0	Opt Plan 404	0	0	0	0	0
10	0	Opt Plan 405	0	0	0	0	0
23	0	Opt Plan 406	0	2	1	9	-9
9	0	Opt Plan 409	0	0	0	0	0
13	0	Opt Plan 412	0	0	0	0	0
21	0	Opt Plan 413	0	0	1	4	-1
28	0	Opt Plan 414	0	0	2	5	-2
46	0	Opt Plan 415	0	8	20	18	-52
14	0	Opt Plan 418	0	0	0	0	0
17	0	Opt Plan 421	0	0	0	0	0
15	0	Opt Plan 422	0	0	0	1	0
7	0	Opt Plan 423	0	0	0	0	0
24	0	Opt Plan 424	0	1	2	1	-6
24	0	Opt Plan 427	0	0	0	4	0
12	0	Opt Plan 432	0	0	0	0	0
14	0	Opt Plan 433	0	0	0	0	0
11	0	Opt Plan 434	0	0	0	0	0
10	0	Opt Plan 435	0	0	0	0	0
12	0	Opt Plan 436	0	0	0	0	0
33	1	Opt Plan 438	0	5	5	11	-25
9	0	Opt Plan 442	0	0	0	0	0
15	0	Opt Plan 446	0	1	2	1	-6
33	0	Opt Plan 447	0	0	0	9	0
34	0	Opt Plan 449	0	0	0	2	0
92	2	Opt Plan 455	0	14	17	16	-73
36	0	Opt Plan 457	0	2	5	9	-13
34	1	Opt Plan 458	0	4	7	6	-23

...

12641	253		0	707	1578	2409	-4406
	hi=red=work to do			hi=bad	hi=bad	hi=bad	lo=bad
				lo=good	lo=good	lo=good	hi=good
	12894						
	Sum						
	1,96%						
	% Unplatformed						
	98,04%						
	% Platformed						

Table 6: Optimisation time in seconds. Sum and average range over all 530 stations in Belgium. P = Periodical, NP = Non Periodical.

	6am - 9am Original	6am - 9am Progressive Real:NP, Fict.:NP		6am - 9am Min. Distance Real:NP, Fict.:NP		6am - 9am Min. Distance Real:P, Fict.:NP		6am - 9am Min. Distance Real:P, Fict.:P	
1	2	3	4	5	6	7	8	9	10
		G	X	G	X	G	X	G	X
# stations		530	528	530	528	530	528	530	528
sum optimisation time only (s)		132.87	14448.67	151.41	7474.51	1034.7	7877.09	3163.2	7906.70
average optimisation time only (s)		0.25	27.36	0.29	14.16	1.95	14.92	5.97	14.97
sum all time (s)		7620	18780	9780	11460	10200	18840	9480	14580
average all time (s)		14.38	35.57	18.45	21.70	19.25	35.68	17.89	27.61
# platformed occupations	11845	11673	11669	11673	11667	11421	11421	11421	11421
# unplatformed occupations	621	793	793	793	793	1045	1045	1045	1045
relative # unplatform occupations (%)	4.98	6.36	6.36	6.36	6.36	8.38	8.38	8.38	8.38
# lines = # conflicts:									
red platform conflicts (≤ 0′)	158	0	0	0	0	0	0	0	0
red route conflicts (≤ 0′)	187	0	0	0	0	0	0	0	0
dark orange indications (≤ 1′)	534	175	178	151	151	131	126	147	126
light orange indications (≤ 2′)	1453	608	627	578	579	548	544	575	544
green indications (≤ 3′)	2206	1180	1243	1112	1110	1020	1025	1058	1025
robustness score	-7800	-1308	-1339	-1182	-1183	-1072	-1048	-1163	-1048

In our previous paper on platforming, Sels et al. (2014), Leopard was also used as platformer and router of trains, but then in a context where station capacity was being estimated. This was done by adding supplementary trains that are expected in the future and trying to platform and route them together with the currently planned trains, until no supplementary train could be placed anymore. At large over-saturation of a station, the model is relatively hard to solve optimally compared to models with an amount of trains that can all or can almost all be assigned to real platform tracks. The computation times for these saturated and over-saturated situations were in the order of minutes. In the current paper, we do not saturate the station with as much platformable traffic as possible, rather, we try to platform only the traffic implied by the macroscopic timetable, which is of course a more realistic situation. It turns out that this means that our computation times are much smaller. Indeed, an average 0.25s per station is shown here.

Table 7: Optimisation for a 24h window. Sum and average range over all 530 stations in Belgium.

	0am - 24pm Original	0am - 24pm Optimised Min. Distance Real:NP, Fict.:NP (NP = Non Periodical)	
	c(platform-conflicts, route-conflicts)	G	X
Aalst (#conflicts) resp. (s)	c(0,0)	1.672	7.048
Antwerpen-Centraal (#conflicts) resp. (s)	c(270, 0)	67.37	34.555
Antwerpen-Berchem (#conflicts) resp. (s)	c(1,0)	1.504	1.350
Braine-le-Compte (#conflicts) resp. (s)	c(6,0)	2.510	4.765
Brugge (#conflicts) resp. (s)	c(3,8)	3.692	3.322
Brussel-Lux. (#conflicts) resp. (s)	c(8,0)	5.096	7.534
Brussel-Zuid (#conflicts) resp. (s)	c(212, 1098)	>7200	>23067
Brussel-Noord (#conflicts) resp. (s)	c(0,14)	1065.3	7200no
Charleroi-Sud (#conflicts) resp. (s)	c(42,15)	14.99	19.08
Denderleeuw (#conflicts) resp. (s)	c(0,93)	10.00	7200
Dendermonde (#conflicts) resp. (s)	c(0,3)	2.632	1.785
Flémale-Haute (#conflicts) resp. (s)	c(0,183)	11.46	7200no
Gent Sint-Pieters (#conflicts) resp. (s)	c(0,112)	93.68	348.08
Hasselt (#conflicts) resp. (s)	c(0,18)	3.53	3.269
Kortrijk (#conflicts) resp. (s)	c(0,6)	4.32	2.560
Leuven (#conflicts) resp. (s)	c(0,9)	13.34	10.43
# stations		530	528
sum optimisation time only (s)		8559	45175
average optimisation time only (s)		16.15	85.56
sum all time (s)		60720	104514
average all time (s)		115	198
# platformed occupations	64382	64275	64722
# unplatformed occupations	3425	3532	3572
# unplatformed occupations (%)	5.05	5.21	5.23
#lines = # conflicts:			
red platform conflicts ($\leq 0'$)	1158	0	0
red route conflicts ($\leq 0'$)	946	0	0
dark orange indications ($\leq 1'$)	2339	628	725
light orange indications ($\leq 2'$)	6012	2274	2591
green routes ($\leq 3'$)	9822	5128	5283
robustness score	-42410	-4786	-5491

Table 7 gives totals and averages of the same properties as Table 6, but now for a window of train traffic from 0h to 24h, so for a full day, instead of just the window from 6am to 9am. Also, Table 7 only shows results for the objective function 'Minimim Distance, Real: NP, Fict: NP'. Naturally, with more train traffic to be platformed, computation times and numbers of conflicts are larger. For Gurobi, only for one station, for Brussel-Zuid, 7200 seconds are not enough to solve the problem. XPRESS can solve that station suboptimally in 23067 seconds. XPRESS has two more stations that it cannot solve to optimality in two hours, Brussel-Noord and Flémalle-Haute, but it solves them suboptimally in that time.

Note that the number of unplatformed occupations for all stations together is in all cases about 5%. This is remarkable, since even with the requirement of the platforming plans having to be free of conflicts, the same percentage of total occupations platformed nationwide can still be achieved. Table 7 shows that the number of platform conflicts and route conflicts are indeed zero for all optimised platform plans while the original platform plans have 1158 platform conflicts and 946 route conflicts over all stations together. It also shows that route robustness issues, for each of the 3 categories, dark orange, light orange and green, occur much less than in the original platform plans as well. Consequently, we find that the national robustness score for the original platforming plans together is a low -42410 while the ones for the optimised platforms together are much higher. Indeed, for the Gurobi solutions this totals to -4786 and for the XPRESS solutions to -5491. So Gurobi performs about 13% better than XPRESS here. Note however that no robustness was present in the objective function. However, a better robustness score is also quite an achievement given that we platform the same amount of occupations as in the original platform plans, especially since we do this without conflicts while the original platform plans still contain both platform and route conflicts. It should be noted here that in a final timetable and its associated platform plans, all planned trains should be assignable to real platforms.

Since automatic platforming spends at most a few hours on the largest stations while, according to De Luca Cardillo (1998), it can take an expert human planner up to 15 days, it is clear that significant time can be saved by adopting an automatic platforming tool. The quality of platforming has to be separated into (i) how many conflicts are left in the plans, (ii) how many robustness issues are still contained in the plans and finally (iii) how many train are platformed on real tracks. The automatic platformer certainly performs better on (i) and (ii). Given that it is strict about (i), that no conflicts can remain, it sometimes platforms fewer trains than the manual platforms, but even so, we think it is better to have a realistic plan than an over-optimistic one, since the latter will lead to train delays in practice. In total, over all stations together, both original and optimised platform plans assigned about 95% of all trains to real platforms. Robustness of the optimised platform plans is significantly better than for the original plans, even though robustness was not part of the objective of our platforming model (yet). So, overall our research questions could be answered positively.

The use of our tool also gave rise to some further planner and managerial insights. For three stations, no routes were defined, so no platforming plans could be generated. For some other stations, data concerning the line which a train originates from or goes to was absent from the databases and still requires human interaction. Both these cases of lacking data input could be amended if this data was added to the Infrabel databases directly. When taking into account that optimised solutions should preferably select platform tracks that stay close to the ones also selected in the manually generated solution, naturally the average platform deviation was lower than for the experiment not taking this into account. Human planners indicated that they prefer this mode of operation of Leopard since then solutions stay ‘close’ to what they are used to plan. This gives them more ‘control’ over the tool.

5. Conclusions

We developed a tool called Leopard for Infrabel, that can now be used to check station platforming and routing plans that were created by human planners. Leopard will produce a visual, detailed error whenever two trains are present on the same platform or route simultaneously and produce a visual, detailed warning whenever these train occupations are considered as being rather close together in time. Leopard can also automatically generate an optimal station platform and routing plan, without platform or routing conflicts and with as many trains as possible assigned to real platform tracks. Our tool produces graphical output and gives a robustness score for each analysed solution, be it a manual or an automatically created one. This paper shows that for a three hour window of train traffic, on average, the platforming plan for a Belgian station is calculated in 0.25 seconds with Gurobi. Only for a few stations it takes more than a second. Leopard can also be automatically run for all stations in Belgium. As such, it can be used to check the in-station feasibility - according to the rules used by Infrabel planners - for all stations for a given macroscopic timetable.

Leopard automatically generates an overview spreadsheet with key performance indicators for each station. These performance indicators are totalised and averaged so that comparison of platformability, conflict freeness and robustness can be readily compared per station but also globally for all stations together. From this sheet, for each station, all graphical platform plans, containing all information, are easily accessible with one click.

Our comparison of original and optimised platforming plans shows that in total, the amount of unplatformed train occupations is in both cases about 5%. Bear in mind that even with this similar performance, our optimised platform plans are conflictless and the original platform plans are not. Additionally, the robustness score for the optimised platforms is also significantly better for the optimised platform plans than for the original ones.

6. Further Work

Currently, database interfacing and writing out large graphical files consume more computer time than model resolution. This could be improved upon by more caching of data, some database query reorganisation and use of flash memory instead of traditional spinning hard disks which have higher access times.

Connection of our timetabling (Sels et al., 2016) and platforming (Sels et al., 2014) tools by automatically running the platforming on the optimised timetable would open perspectives for further automation of the total planning process. Of course, the issue on how to automatically feedback intra-station problems of platforming or routing to the inter-station macroscopic timetabling level and resolve them is still not automatically solved.

Acknowledgements

We want to thank Infrabel for their cooperation in this project and specifically Eric Vercauteren for indicating what to extract from which database and Stéphanie Godart and colleagues for the many iterations in completely specifying and iteratively adapting all the route variants in stations in the application LUKS.

References

Adlbrecht, J.-A., Hüttler, B., Ilo, N., Gronalt, M., May 2015. Train routing in shunting yards using Answer Set Programming. Expert Systems With Applications 42, 7292–7302.

Barbucha, D., 2012. Agent-based guided local search. Expert Systems With Applications 39 (15), 12032–12045.

Bešinović, N., Roberti, R., Quaglietta, E., Cacchiani, V., Toth, P., Goverde, R., 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.

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

Caprara, A., Galli, L., Toth, P., 2011. 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.

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) (1), 195–224.

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

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) (1), 160–164.

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

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2013. Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming. Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark. Young Railway Operations Research Award 2013 of IAROR.

Goverde, R., Hansen, I., 2013. Performance indicators for railway timetables. Proceedings of IEEE International Conference on Intelligent Rail Transportation: ICIRT2013, August 30-September 1, 2013, Beijing, China., 301–306.

Kroon, L., Romeijn, H., Zwaneveld, P. J., 1997. Routing Trains Through Railway Stations: Complexity Issues. European Journal of Operational Research 98(3) (1), 485–498.

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

Narayanaswami, S., Rangaraj, N., 2014. A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation. Expert Systems With Applications 42, 2638–2656.

OpenTrack Railway Technology Ltd., 2012. OpenTrack. Tech. rep., OpenTrack Railway Technology Ltd.

Rail Management Consultants GmbH (RMCon), 2016. RailSys Users. Tech. rep., RMCon GmbH.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 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.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2016. Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable. Transportation Research Part B 84, 124–156.

Sels, P., Vansteenwegen, P., Dewilde, T., Cattrysse, D., Waquet, B., Joubert, A., 2014. The train platforming problem: The infrastructure management company perspective. Transportation Research Part B 61, 55–72.

ViaCon GmbH, 2008a. LUKS Basics. Tech. rep., ViaCon GmbH.

ViaCon GmbH, 2008b. LUKS Users. Tech. rep., ViaCon GmbH.

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

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.