KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Large-Scale, Passenger Oriented, Cyclic Railway Timetabling and Station Platforming and Routing

Peter Sels

Supervisor:
Prof. dr. ir. P. Vansteenwegen
Prof. dr. ir. D. Cattrysse, co-supervisor

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

May 2016

Large-Scale, Passenger Oriented, Cyclic Railway Timetabling and Station Platforming and Routing

Peter SELS

Examination committee:
Prof. dr. ir. O. Van der Biest, chair
Prof. dr. ir. P. Vansteenwegen, supervisor
Prof. dr. ir. D. Cattrysse, co-supervisor
Prof. dr. ir. C. Tampère
Prof. dr. P. De Causmaecker
Prof. dr. R.M.P. Goverde
(Delft University of Technology)
Prof. dr. D. Huisman
(Erasmus University Rotterdam)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

May 2016

Uitgegeven in eigen beheer, Peter Sels, Celestijnenlaan 300 box 2442, B-3001 Leuven (Belgium)

Preface

“If at first, an idea is not absurd then there is no hope for it.”

– Albert Einstein

As a consultant to Infrabel - the Belgian railway infrastructure manager - and being assigned the task of automatically timetabling all Belgian passenger trains with the objective that passengers will experience the lowest possible travel time in practice, I was faced with a major challenge. It had never been done before for an entire country. So I had to learn what the state of the art in railway timetabling research was. We used many ideas present there. This does not mean that going for a full fledged PhD was a strict necessity. Indeed, the research publication requirement of a PhD means that there is a lot of extra work in the discussing, writing, reviewing and making publishable of papers that report ones results. However, this extra effort creates the upside of reaching a larger as well as broader audience. This was a major motivation. Indeed, drive is all about objectives. We reached these objectives now; we had a tough yet fun puzzle time but also produced timetables which promise reduced passenger travel time. Published papers are available to the railway and research community explaining how this can be done. I could not have done so alone. So I want to thank all people who contributed, knowingly or unknowingly.

I am very grateful to Professor Pieter Vansteenwegen for seeing the value of the idea of expected passenger time in practice as the objective function for a timetable and initiating research in this area [44, 45]. It is a powerful concept and has been and still is a motivating goal. I am also indebted to Pieter's serious revision of and insightful comments on my paper drafts. This often triggered quite some technical, presentation and restructuring discussions and

invariably led to better final versions. I also thank my eagle eyed colleague and co-researcher Dr. Thijs Dewilde for the reading of my writings and his useful comments, both on high level issues and on the very smallest details. He spotted everything. I am also grateful to Professor Dirk Cattrysse for his paper comments and his calming demeanour, humour and wisdom.

We wish to thank Infrabel for the fruitful cooperation and trust during these 6 years. Our thanks go especially to Eric Vercauteren who defined what Infrabel wanted and did not want from a timetable and was always willing to answer my countless questions. I also think back with pleasure to the discussions I had with Alain Wenmaekers about the best ways to code something and about the occasional C++ or Qt issue as well as to the various entertaining pseudo-religious debates. It was fun. Thanks also go to Stéphanie Godart for the work on defining route variants in all Belgian stations, to Anneleen Van de Mergel and TrafIT GmbH and VIA Consulting & Development GmbH for defining the Infrabel set of parameters for LUKS and OnTime simulations. I thank Katrien Pauwels for sharing her knowledge about and helpfulness with Infrabel databases and to Bertrand Waquet for the preparation of train traffic scenarios for our first platforming paper when this part was not automated yet. Thanks also go to Mathieu Senelle for proofreading our papers as Infrabel representative.

We are indebted to Tove Møller of Banedanmark for reacting on our claim, made at the IAROR conference in Copenhagen, that we are able to automatically timetable a country like Belgium and asking us to try this for Denmark. Katrine Meisch was instrumental in the challenging task of preparing the input data for this project. She managed to combine this with her other obligations to Banedenmark. Our cooperation could not have been better. So was the result; a timetable for all Denmark's passenger trains that would save passengers 2.9% of journey time on average. Thanks got to Jens Parbo for proofreading our paper about these results for Denmark and for his likeminded spirit in supporting the cause for passenger oriented planning in his country.

We also wish to thank Professor Rob Goverde and Nikola Bešinović for the many discussions on timetable quality criteria and quality levels. This certainly cleared up things but also convinced us further that expected passenger time as an objective function covers most of these criteria and is a good objective to go for. Thank you, Jonas Harbering, for the enjoyable NP-complete discussions and the checking of our always-feasible proof. Thanks to all of them for the fun nights out on various conferences from Barcelona to Tokyo.

One other essential factor of our ability to produce useable timetables is the unique performance of the Gurobi solver, as well as the excellent and forthcoming support of Gurobi personnel: Greg Glockner, Sonja Mars, Renan Garcia,

Cynthia Sosa, Tonya Peek, Mark Steidel, Tracey Pesanelli and Christopher Riche. They are a truly exceptional team, increasing standards of performance, efficiency and support in the industry.

I find the following also a fascinating thought. Imagine how much of this work was only possible thanks to the centuries of development of science and technology continuously building upon and improving itself. From Newton's laws of motion and differential and integral calculus to Moore's descriptive law of computer calculation speed, to computer network and multicore development, we needed and used all of it. So thanks to these giants and the support of their shoulders.

I am grateful to my parents, my family, my friends and my teachers for triggering my curiosity, for showing interest, for listening and encouraging, for getting me to know the challenge and the hardship of, the needed persistence for and the ultimate satisfaction of creative intellectual labour; for the agony and the ecstasy.

Abstract

The two topics of this thesis are macroscopic railway timetabling and station platforming and routing. These problems are challenges that are faced by all railway infrastructure companies. This thesis was carried out in cooperation with Infrabel, the Belgian railway infrastructure manager.

Macroscopic Cyclic Timetabling of Entire Countries

In the macroscopic *Train Timetabling Problem* (TTP), the goal is to obtain, for each train and for each station it visits, an arrival and a departure time. Sufficient time should be provided for trains to ride from one station to the next. In stations, trains that stop should be allowed a minimal stop time, so that passengers can board, transfer or leave. Of course trains cannot collide on the same track and should even be separated by typically 3 minutes for safety reasons. No commercial software tools exist yet that automatically solve this problem. Over the last decennia, research at universities has led to automatically constructed 'draft' timetables for the Netherlands, but these timetables still lack robustness against small recurrent delays. Subsequent optimisation to improve robustness was carried out, but leaves trains in the same order as in the input timetable, which constitutes a suboptimal approach compared to simultaneous optimisation for train order and robustness. A timetable for the 37 trains of the Berlin underground was also constructed automatically, but also lacks robustness. For Belgium, about 200 hourly trains have to be planned, but Infrabel additionally requires that the timetable delivers minimal passenger travel time in practice. This implies it will be robust. This was a new modelling and computational challenge. This time, which has to be minimised, includes journey time and small recurrent delays. In our latest paper, we also include excess journey time, which is the sum of waiting time at departure and at arrival.

We are able to generate a timetable for all 196 hourly Belgian passenger trains from the 2013 timetable in a computation time of about 2 hours. Assuming common delays, the reduction of expected passenger time is 3.8% and the average missed transfer probability goes down from more than 10% in the original timetable to less than 3% in our timetable. Additionally, we also optimised the Danish timetable with 88 trains in about one hour, with a reduction of 2.9% of expected passenger time and reduced percentages for the probability of missing a transfer that are similar to the Belgian case.

Station Train Platforming and Routing

For the problem of station platforming and routing, the arrival and departure times of trains are considered as given by the timetable. The sequential planning of the timetable and then station platforming and routing corresponds to the process currently used at Infrabel and almost all other railway companies. The challenge is then, to assign, per station, a platform track to each train such that no two trains are using the same infrastructure element at any time. This *Train Platforming Problem* (TPP) has been intensively studied and solved during the last decennia. Computation times are of the order of seconds per station. However, realisation of these automated methods at railway companies or into commercial tools for the railway industry has been limited. In most research, in the TPP formulation, train arrival and departure times are still allowed to be varied within limits. In practice at Infrabel, contrary to most research, the TPP formulation is one whereby train arrival and departure times are fixed. This is the case because this allows easy integration into their process flow. The preceding timetabling phase is supposed to decide on all train arrival and departure times. As a consequence, it is of course possible that after platforming, if not all trains can be assigned platforms and routes, the timetabling has to be adapted so as to make the blocking platforming problem easier.

We solve the platforming problem as defined by Infrabel, by developing and integrating a platforming tool in the Infrabel tool flow. So one can consider our platforming efforts as applied research. This tool produces a picture showing the current - manually made - platforming plan and indicates any conflict between any pair of trains in the station area.

We then generate an assignment without conflicts, of as many trains to platforms as possible. Sometimes, not all trains can be assigned to platform tracks and only a partial solution is given. This means that in practice, some trains cannot be assigned or otherwise the arrival and/or departure times have to be adapted manually. In the last case the macroscopic timetable also will have to be changed

again. We also produce a picture of our optimised assignment which consistently shows that indeed, in our solution, there is no single conflict left. We are able to automatically generate train to platforming assignments for all 530 Belgian stations in only 10 minutes of optimisation time for all 530 models together.

Both our macroscopic timetabling tool and platforming and routing tool are integrated and useable at Infrabel.

Beknopte Samenvatting

De twee hoofdonderwerpen in deze thesis zijn het opstellen van een dienstregeling en, voor elk station, het opstellen van een perronspoortoewijzing inclusief keuze van routes. Deze problemen zijn uitdagingen voor alle beheerders van spoorweginfrastructuur. Deze thesis werd uitgevoerd in samenwerking met Infrabel, de beheerder van de Belgische spoorweginfrastructuur.

Een Macroscopische Cyclische Dienstregeling

In de macroscopische treindienstregeling is de doelstelling om, voor elke trein en voor elk station dat de trein aandoet, een aankomst- en vertrektijd te bepalen. Voldoende tijd moet voorzien zijn om van een station naar het daaropvolgende te rijden. In elk station waar een trein stopt, moet voor die trein ook voldoende stoptijd voorzien zijn om reizigers te laten in-, over- of afstappen. Uiteraard mogen treinen op hetzelfde infrastructuurelement ook niet met elkaar botsen en moeten ze om veiligheidsredenen zelfs gescheiden worden door typisch 3 minuten. Er bestaan nog geen commericiële softwarepakketten die dit probleem automatisch oplossen. In de laatste decennia leidde onderzoek aan universiteiten tot het resultaat van een automatisch geconstrueerde dienstregeling voor de Nederland, maar deze dienstregeling is niet robuust tegen veelvoorkomende kleine vertragingen. Een dienstregeling voor de 37 treinen van de Berlijnse metro werd ook automatisch berekend, maar daarin ontbreekt ook robuustheid. Voor België moeten ongeveer 200 uurlijkse treinen gepland worden, maar Infrabel eist ook dat de dienstregeling een minimale verwachte reizigerstijd in praktijk oplevert. Dit impliceert robuustheid. Dit was een nieuwe uitdaging zowel qua modellering als rekenkundig. In deze te minimaliseren tijd zit de totale trajecttijd, maar ook de tijd van vaak voorkomende vertragingen. In onze laatste paper is in het objectief ook de wachttijd bij vertrek en de wachttijd bij aankomst vervat.

We zijn in staat om een dienstregeling te genereren voor alle 196 uurlijkse treinen uit de Belgische dienstregeling van 2013 in een rekentijd van ongeveer 2 uur. Wanneer we standaard vertragingen veronderstellen, kunnen we de reizigerstijd reduceren met 3.8%. De kans op een gemiste overstap daalt van meer dan 10% in de oorspronkelijke dienstregeling naar minder dan 3% in de geoptimaliseerde dienstregeling. Ook de Deense dienstregeling met zijn 88 uurlijkse treinen hebben we geoptimaliseerd. Dit gaf een reductie in verwachte reizigerstijd van 2.9% in een rekentijd van 1 uur. De kans op gemiste overstappen werd gereduceerd met gelijkaardige percentages als voor België.

Treinen Toewijzen aan Perronspoor en Route

Voor het probleem van perronspoor- en routetoewijzing in elk station beschouwen we de aankomst- en vertrektijden van treinen als gegeven door de macroscopische dienstregeling. De sequentiële aanpak van de dienstregeling en daarna de perronspoor- en routetoewijzingen per station is in overeenstemming met de huidige praktijk bij Infrabel en waarschijnlijk alle andere spoorwegondernemingen. De uitdaging is dan om per station, aan elke trein een perronspoor toe te kennen, zo dat op geen enkel moment twee treinen hetzelfde infrastructuurelement gebruiken. De laatste decennia is er veel onderzoek naar dit *'Train Platforming Problem'* (TPP) gebeurd en het probleem is essentieel opgelost. Rekentijden zijn van de orde van seconden per station. Echter, de implementatie van deze automatische methodes werd nog niet doorgedreven tot een tool dat in gebruik is bij spoorwegondernemingen. We lossen dit op door de ontwikkeling en het integreren van een tool bij Infrabel.

Deze tool genereert een grafische voorstelling van de huidige - manueel aangemaakte - perronspoortoewijzing en duidt daarop elk conflict tussen treinen aan. We genereren dan een geoptimaliseerde perronspoortoewijzing die conflictloos is en zo veel mogelijk treinen op perronsporen plaatst. Soms kunnen niet alle treinen geplaatst worden en dan geven we een partiële oplossing. Dit betekent dat in praktijk soms niet alle treinen geplaatst kunnen worden of dat de aankomst- en/of vertrektijden manueel moeten gewijzigd worden. In het laatste geval moet ook de macroscopische dienstregeling worden gewijzigd. We produceren ook een grafische voorstelling van de geoptimaliseerde perronspoortoewijzing die consistent toont dat er inderdaad geen enkel conflict overblijft. We zijn in staat om automatisch een perronspoortoewijzing te genereren voor alle 530 Belgische stations samen in slechts 10 minuten rekentijd.

De programmas voor het genereren van de macroscopische dienstregeling en perronspoortoewijzing zijn beide geïntegreerd en bruikbaar bij Infrabel.

Contents

Abstract **v**

Contents **xi**

1 Overview **1**

1.1 Planning . . . 2

1.1.1 Traditional Planning Stages . . . 2

1.1.2 Reasons for Planning . . . 2

1.1.3 Two Phased Planning in this Thesis . . . 4

1.2 Macroscopic Timetabling . . . 4

1.2.1 A Timetable Objective is Required . . . 4

1.2.2 The Control Domain: Thousands of Time Supplements 5

1.2.3 Previous Research . . . 5

1.2.4 What Our Research Solves . . . 7

1.2.5 Benefits to Infrabel and Passengers . . . 9

1.3 Station Platforming . . . 10

1.3.1 Station Platforming After Macroscopic Timetabling . . 10

1.3.2 Our Research Model . . . 11

1.3.3 What Our Research Solves . . . 11

1.3.4 Benefits to Infrabel and Passengers 12
1.4 Chapter Overview 13

2 Cyclic Timetabling for Entire Countries 15

2.1 Deriving all Passenger Flows in a Railway Network from Ticket Sales Data 16
2.2 Expected Passenger Travel Time for Train Schedule Evaluation & Optimisation 39
2.3 A Passenger Knock-On Delay Model for Time-table Optimisation 63
2.4 Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable 76
2.5 Towards a Better Train Timetable for Denmark Reducing Total Expected Passenger Time 115
2.6 Optimal Temporal Spreading of Alternative Trains in order to Minimise Passenger Travel Time in Practice 135
2.7 Practical Macroscopic Evaluation and Comparison of Railway Timetables 160

3 Platforming and Routing in Stations 173

3.1 The Train Platforming Problem: the Infrastructure Management Company Perspective 174
3.2 Automated Platforming & Routing of Trains in All Belgian Railway Stations 193

4 Other Publications as First Author 219

4.1 Deriving all Passenger Flows in a Railway Network from Ticket Sales Data 219
4.2 Calculation of Realistic Railway Station Capacity by Platforming Feasibility Checks 220
4.3 Automated, Passenger Time Optimal, Robust Timetabling, Using Integer Programming 220
4.4 Automatically and Quickly Planning Platform and Route of Trains in Railway Stations with a Case Study for Mechelen Station 221

4.5 Light Rail, Tram and Bus co-Timetabling Minimising Passenger Travel Time in Practice 221

5 Co-Authored Publications 223

5.1 Defining Robustness of a Railway Timetable 223

5.2 Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming (IAROR) 224

5.3 Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming (JRTPM) 224

5.4 Improving the robustness in railway station areas 225

5.5 Unification of Research 225

6 Conclusions and Further Work 227

6.1 Conclusions 227

6.1.1 Results 228

6.1.2 Recommendations to Railway Companies 229

6.2 Useful Further Work 231

6.2.1 Improvement of Input Data 231

6.2.2 Further Reduction of the MIP Gap 231

6.2.3 Addition of Boundary Conditions 232

6.2.4 Scalable Optimisation of Temporal Spreading of Alternative Trains 233

6.2.5 Further Reduction of the Expected Transfer Time Component 233

6.2.6 Execute Multiple Reflowing and Retiming Iterations . . 233

6.2.7 Coupling of Timetabling and Platforming 234

6.2.8 Further Cooperation on Timetabling 235

Bibliography 237

Curriculum Vitae **243**

List of Publications **245**

Chapter 1

Overview

"Some puzzles are too good to share."

– Doctor Gregory House

"Some solutions are too good to hide."

– Peter Sels

This PhD thesis is presented based on the published papers during the time of PhD research. The papers bundled in this document were written in the past 5 years by Peter Sels as first author. These were all revised by Dr. Thijs Dewilde, Professor Pieter Vansteenwegen and Professor Dirk Cattrysse before they were submitted to conferences and journals for review. In this thesis, papers are bundled, structured and every time introduced by a short, non-specialist description of its contents. These introductions do not describe the literature in detail because this is done in the papers themselves. The bibliography at the end of this thesis only mentions the publications that were cited in the thesis introduction, the introductory text to all papers and the thesis conclusion but not the publications that were cited in each of the reproduced papers.

The timetabling tool RhinoCeros and the platforming and routing tool Leopard were entirely coded by Peter Sels in his role as Logically Yours consultant to

Infrabel, the Belgian railway infrastructure manager.

This first section discusses the relation between the different papers and brings these together. It also provides an overview of what has been realised. Chapter 2 presents the main papers published on timetabling while chapter 3 presents the papers dealing with platfoming and routing in station areas. These chapters form the body of work performed for the PhD. Chapter 4 mentions other abstracts or papers that were published as first author, while chapter 5 does this for papers published as co-author. Chapter 6 concludes and hints at possible further useful work.

1.1 Planning

1.1.1 Traditional Planning Stages

In the railway industry, it is common practice to separate the planning of operations into four stages [14, 4, 21]. First, it is decided from where to where trains will run and also at which stations they will stop and which stations they will just pass. This is called *line planning*. When these paths are decided upon, the question arises at what times trains should arrive at and depart from each of the visited stations. Planning this is called *timetabling*. Often a timetable with a fixed period is preferred. Any train service is then repeated after this period, usually an hour, in the same way. In that case this is called *cyclic timetabling*. Once it is clear what ride and dwell operations trains will be occupied with at which time, specific rolling stock, like locomotives and train cars (wagons) need to be assigned to these operations. This is called *rolling stock planning*. Once it is known where rolling stock is needed at which time, one has to plan the operations that train employees, like drivers, controllers and maintenance crews, have to carry out to keep this rolling stock running. This is called *personnel planning* or *crew scheduling and rostering*.

1.1.2 Reasons for Planning

So much for the tradition of planning, but the question may arise why we need to plan trains at all. We do not do this for cars. One sufficient reason is that train passengers need to know when they can expect a train to depart at a station so that they know when to arrive there, preferably just some minutes in advance of the departure time. Similarly, they want to know when they will arrive so that they can plan to take a train that arrives in advance of the time their meeting starts at the arrival place. So departure and arrival times of all trains need to be known to potential passengers.

But knowing these times is also important for coordination between trains in the network itself. True, since the arrival time of a first train and the departure time of a second train need to be planned in a coordinated way if it is expected that a lot of people will transfer from the first to the second train. Indeed, in that case we do not want the time difference to be so big that people will have to stand waiting on the platform too long during this transfer. On the other hand, it cannot be so small that the least delay of the first train causes passengers to miss the transfer to the second train. So there will be an optimal time difference between the arrival time of the first and the departure time of the second train. This can be calculated during timetabling.

So, the above two reasons for planning, for departing and arriving passengers and for transfers, have to do with coordination of train planning with the external planning of passengers respectively with self-coordination of trains that both belong within the system. Coordination is also necessary at points where resources are too restricted to allow just any amount of train traffic to come in, or/and go out, at any time. This is the case, for example, on an open line track between two stations. There, only one train can ride at a time. This is why tracks are divided into sections and signals are added: to guarantee that only one train is allowed per section at a time. This is the most restrictive and demanding type of constraint on *macroscopic train timetabling*.

Another restriction is that, inside a station which has a limited amount of platform tracks as well as connectivity between open lines between stations and these platform tracks, we cannot allow trains to just arrive and depart at random. We have to time these actions precisely to allow for use without train collisions. The planning inside train stations at this lower level is called station planning, the train platforming problem (TPP) or the platforming and routing problem. Note that both macroscopic as well as microscopic planning approaches exist aiming at solving this problem. Microscopic models take into account all signals, block sections and liberation points, while macroscopic models do not model these to the full extent. In manual practice, the platforming problem is often only solved after timetabling has been carried out. To date, even in automated planning research, this is still the case, because simultaneous optimisation of timetabling and platforming for entire countries at once is still considered to be too hard.

1.1.3 Two Phased Planning in this Thesis

This thesis' first focus is on macroscopic cyclic timetabling. A macroscopic model typically considers large scale networks but does not model the infrastructure in detail, like switches or number of platforms. A microscopic model does consider these infrastructure details but typically models smaller network areas like one or a few station areas only. In Belgium, at Infrabel, timetabling is done at a macroscopic level and its feasibility inside stations is later verified at the microscopic level. This thesis also deals with this second problem of feasibility checking in stations. It does this by platforming and routing inside stations of as many as possible of the macroscopically scheduled trains.

1.2 Macroscopic Timetabling

1.2.1 A Timetable Objective is Required

As mentioned, macroscopic planning is about deciding on *good* train arrival and departure times at every station. The question is what it means to call a timetable *good*. Many objectives can be chosen for a timetable. Objectives depend on the stakeholders. The train operator will prefer a timetable that is cheap to operate, for example with a few less train sets, but still offering a good service to its customers, the passengers. The passengers will prefer a timetable that brings them quickly from their origin station to their destination station. They will also want that this time varies little from day to day. The railway infrastructure company, Infrabel, will benefit from a timetable where many trains use its infrastructure lines or sections for which it makes a good profit. This means that such lines or sections are used intensively yet without wearing these out too quickly. In this thesis, because we think passengers are the ultimate and most important stakeholders, we put ourselves in their position and aim to generate a timetable that has minimal expected passenger travel time in practice [33, 32, 37, 35]. This objective includes ride time, dwell time, transfer time as well as statistical expected delays and their knock on delays these cause for other trains. The objective contains this total expected journey time for all passengers together. During mathematical optimisation of the timetable, sometimes a journey time for one big stream of passengers could be shortened at the expense of increasing the expected journey time for another, smaller stream of passengers. Making this kind of trade-offs is the essence of good timetabling, whether it is done manually or automatically.

1.2.2 The Control Domain: Thousands of Time Supplements

We know now what we want to achieve, the objective defined in the previous section. But how can we achieve this? What parameters can we change that affect it, for better or worse? For each train, we can calculate minimal ride times from a station to the next station. When these trains stop in a station, we can preset minimal stop times that allow passengers to exit or enter these trains. When passengers have to transfer between trains, we can calculate their walking time between those trains and take this as minimum time between the arriving and departing train. All these minima have to be respected by a timetable. One can only add supplementary time to it, on top of these minima. We call these extra times 'time supplements' or simply 'supplements'. It will seem intuitively logical that very large supplements on ride, dwell or transfer activities are not ideal. Indeed, passengers then have to wait very long and lose time. But also if these supplements are very small, a negative effect occurs. Indeed, the slightest primary delay on that activity then makes that the timetable cannot be realised as planned. This would mean that all planning was unrealistic and missing the point. In railway timetabling, one also needs to separate trains on common tracks by a minimum headway time. In a macroscopic approach, this time is usually about 3 minutes. These times can be seen as an extra set of minimum times that have to be respected. In cyclic timetabling, on top of this, we have to make sure that for every planned train in the system, one hour later, a similar train with the same timing will occur. This train can again potentially interfere with all the other trains. This imposes again a new set of constraints to be respected. So the delicate art of cyclic railway timetabling on the mathematical level comes down to the following. Choose a set of time supplements (about 10000 for Belgium) so that all ride, dwell, transfer, headway minimal time constraints as well as hourly periodicity constraints (about 100000 for Belgium) are respected. Respecting these constraints is the minimal requirement. A timetable respecting these constraints is called a *feasible solution*. A timetable that also achieves a good value for the objective is called a *good solution* and that is what we want to achieve. Preferably we want one that is better than the current timetable in terms of the objective. Can we achieve that? It is something I would hate to do without a computer but love to do with one. So let's go.

1.2.3 Previous Research

Since there are no tools available to timetable practitioners, one may wonder why this is the case. Is this problem impossible to solve automatically? What is the state of the art in research? Infrabel wants to arrive at a schedule where every hour, the same set of trains is repeated with the same timing. Previous research into this kind of optimisation approach resulted in the Periodic Event Scheduling Problem (PESP) formulation [41].

This model defines the constraints that need to be fulfilled to result in a *feasible* timetable. Note that the term *'feasible'* has been defined in the literature on the microscopic level [11]. According to this definition, a feasible timetable has to be *realisable*, which means that it has sufficient time provided for all processes (run and dwell activities) and must also be *conflict free*, which means that it cannot contain any conflict between any pair of trains on the same physical resource (track, route). Contrary to this definition, our definition of 'feasible' stems from the rules the human planners at Infrabel use in their macroscopic timetable construction. This entails two aspects that can be different from the microscopic definition of feasibility. (i) Infrabel computes minimal process times for run activities and adds a 5% margin on top of this. The resulting times are taken as minimal times in our software. Because of the 5% addition, this could be more conservative than the strict definition of 'realisable'. (ii) Human planners use a rule of thumb of 3 minutes minimal headway time between subsequent trains as sufficient to guarantee 'feasibility'. This could be either more conservative or less conservative than the strict definition of 'conflict free'. Indeed, when analysed on the microscopic level, depending on trains length, speed and other factors, 3 minutes could be either too much or too little minimum headway time to avoid conflicts. Even though, at Infrabel, human planners aim to stick to their two rules, it is difficult to do so due to the complex interdependencies of a timetable. Additionally, the fact that multiple human planners work on one large problem and will not always notify each other of modifications they carried out, can complicate the job. In practice, we do indeed notice that current timetables do not respect these two rules in all places for all trains. However, with our timetabling tool, RhinoCeros, we guarantee that any timetable it produces respects both Infrabel rules in all places for all trains. In this sense we guarantee 'feasibility' on the macroscopic level, as the human planners interpret it. Generating timetables that are 'feasible' according to the definition in the literature would require accurate calculation of minimal headway times between all pairs of train that could be subsequent in the timetable being produced. Calculation methods of these headways are known in the literature, but the job of collecting all the input data needed for this is huge and to the best of our knowledge has not been carried out before on a national scale.

The PESP model in itself, does not guarantee to generate *good* timetables in any sense, just feasible ones. So in the timetable solutions produced by a bare PESP model, there could be very large time supplements planned at ride or dwell activities of trains or very large or small time supplements planned at transfer activities of passengers. Note that supplements that are really small make that the timetable is not robust against even very small primary or secondary delays.

An example of a research toolset that implements this PESP model is the CADANS and DONS system [27, 24]. CADANS and DONS were developed at the Nederlandse Spoorwegen (NS), the Dutch main train operating company. A somewhat similar initiative at the German railways resulted in the timetabling tool TAKT [22, 13, 23, 18]. The timetable generating tools CADANS and TAKT are known to have generated feasible national timetables for The Netherlands [24, 16] and Germany respectively. However, for both these processes, no timetable quality criterium or objective function was defined nor optimised. For the CADANS produced timetables, it is known that post optimisation towards more robustness is needed [15]. This is the case because CADANS adds a fixed amount of supplements according to a percentage of $x\%$ of the minimum times spread out over the activities (ride, dwell) of each train. The value of x can be specified in the CADANS input. A typical value is $x = 5\%$. This post-optimisation for robustness adapts time supplements but does not allow swapping orders of trains. This means that not the full search space is searched and that the resulting timetable can be suboptimal in this respect. TAKT uses a satisfiability (SAT) solver which also does not create robust timetables since it minimises the sum of total slack. This will introduce time supplements equal to zero on critical paths which is essentially non-robust against delays. Timetables that are not robust are basically not useable yet in practice. Indeed, even the slightest delays will make that the real arrival and departure times of trains will deviate from the planned timetable.

This effect will even increase in time if the timetable is also not *stable* [12, 42]. Essentially, a stable timetable can have 'negative' buffer times, but the delays they inject into the timetable should then be absorbed by sufficiently large time buffers elsewhere so that delay propagation remains limited, instead of rising continuously. So stability follows from feasibility, but not the other way around. Since CADANS and TAKT have no objective function, they don't strive for stability either, but their hard constraints guarantee feasibility which in its turn guarantees stability.

1.2.4 What Our Research Solves

What is needed to avoid these problems of undesirable supplement values is an objective function that penalises time supplements that are too large or too small. [19, 20] demonstrate that these problems can be partially solved by adding an objective function to the bare PESP model. Their case study is the timetable of the Berlin underground. This metro network has 37 train lines. They report that only some transfers were still assigned a very large duration.

Not surprisingly, these transfer durations were not present in the objective function. We conclude that (i) *all* durations that matter for the objective of a timetable should be somehow present in the objective function and (ii) to the degree that they matter for the objective. Since our objective is to minimise the expected passengers time in practice, this means that, for a certain duration, the number of passengers experiencing that duration will be the weight factor. This also means that each *expected* duration should be formulated as a function of the *planned* durations, which are the decision variables of the timetabling model. Therefore, the expected durations that should be present in the objective function and not the planned durations, because the expected durations are what matters in practice.

This is the first major part of what our timetabling research in this doctorate essentially adds: the objective of expected passenger travel time in practice [31, 33, 32, 36]. Minimising this objective avoids that all too large or all too small time supplements are chosen. This results in timetables that are effectively useable [37, 35]. Other major work has been carried out on adding our own sets of supplementary constraints to the PESP model [37] so that it solves more quickly. These are cycle sets as discovered by [24] and further elaborated on by [25] and [19].

For Infrabel, we developed the tool RhinoCeros that implements this PESP based model, extended with this objective function and extra cycle set constraints. We are successful in applying RhinoCeros on the network of all 196 hourly passenger trains in Belgium. The result is that in about 2 hours of computation time, we can generate a timetable that is feasible on a macroscopic level, but that is also good for passengers. Our optimised timetable is even better for passengers than the manually constructed 2013 timetable that served as starting point, since according to our model, it requires 3.8% less passenger time in practice. Our calculations also predict that the average missed transfer probability goes down from 13.9% in the original timetable to only 2.6% in our optimised timetable. We also applied RhinoCeros on the Danish network of all 88 passenger trains and in about 1 hour we could construct a timetable with similar beneficial properties. Expected passenger travel time is reduced by 2.9% and the probability of a missed transfer goes down from 11.34% in the original timetable to 2.45% in the optimised timetable.

To the best of our knowledge, it is the first time that a timetable for an entire country has been generated with a system that also tries to optimise an explicit and complete objective. In both cases, for Belgium and Denmark, our work resulted in a timetable that is significantly better than the current one in terms of our chosen objective. These timetables should be directly beneficial to passengers since they save them several percentages of passenger travel time in practice. Note that these timetables also contain a degree of robustness against delays, more specifically, the degree of robustness that is ideal for passengers, not too little, not too much.

1.2.5 Benefits to Infrabel and Passengers

Currently, at Infrabel, about 20 human timetablers need to keep shifting train arrival and departure times until (i) all minimal ride times from station to station are respected, (ii) all minimal dwell times in stations required by the train operating company are respected and (iii) all subsequent pairs of trains are separated by the minimum headway time of three minutes on open line tracks. Adapting a train arrival or departure time with the aim to satisfy a constraint of type (iii) often again breaks a previously satisfied constraint of type (i) or (ii) and vice versa. In practice, this takes many months and even after that, the manually constructed macroscopic timetables still contain violations of constraints of types (i), (ii) and/or (iii). Every year, Infrabel uses the same process of planning the timetable for the next year. This means that, at some point, planning time is simply over and typically some violations remain present in the timetable being planned.

Our software, RhinoCeros, developed for Infrabel and totally integrated there, guarantees that, for any solution it produces, all constraints of all three types (i), (ii) and (iii) are satisfied. No single violation occurs. RhinoCeros respects the line planning given by the original timetable and also guarantees perfect periodicity with one hour for all train relations. For the Belgian case, the resulting timetable is usually produced after about 2 hours of calculation time.

So increased speed of the timetabling process is the first advantage and a resulting timetable without a single constraint violation is the second. From our optimisation experiments, we conclude that the third advantage is that our resulting timetables typically also imply less expected travel time in practice to the passengers. A typical figure is a 3.8% reduction. Rolling stock planning and crew scheduling were not part of the scope of this thesis. However, a rough preliminary analysis of the needed rolling stock based on the total number of expected train hours in practice showed that no more such hours occurred in the optimised than in the original planning. This would mean that in practice, no more train sets would be needed to run the optimised timetable than the original one.

A faster timetabling process allows Infrabel to make timetables for more versions of line planning. Indeed, with RhinoCeros, Infrabel can now automatically evaluate the total expected passenger time associated with the automatically constructed timetable, given a fixed amount of computation time, for each of these line plannings. It can then select the combination of line planning and associated optimised timetable that results in the lowest total expected passenger travel time.

1.3 Station Platforming

1.3.1 Station Platforming After Macroscopic Timetabling

When a macroscopic timetable has been defined, the next question is if it can be realised in every station microscopically. It could indeed be the case that too many trains arrive in a certain time window in a certain station and that consequently, not all these trains can directly ride through to their platform tracks. Similarly, not all trains may be able to leave from platform tracks to open lines leading to the next stations because they use crossing or identical tracks at the same time. The combination of simultaneously incoming and outgoing trains makes matters even more complex. This depends both on their timing, their chosen platform tracks and the available routing variants connecting the open lines the trains come from or go to and the platform tracks. So for some station, given the train arrival and departure times dictated by the timetable, some platforming and routing choices would make the timetable feasible in that station, while with other such choices, it could become infeasible. Note that we consider solving the macroscopic timetable first and then all the platforming and routing problems in all stations. If in a station, none of all the possible platforming and routing choices allows the timetable to be implemented, the timetable will have to be changed cleverly. Often shifting a few train's arrival or/and departure time by a half or a full minute or so can solve this. However, in our method, this work still has to be performed manually.

Note that we can not claim that it is guaranteed that a 100% placement of trains on tracks is possible. This depends on whether the number of trains and their arrival and departure times can be mapped feasibly on the station's infrastructure. If too many trains are planned or too many trains arrive or depart simultaneously this will be impossible. What we do claim though, is that if it can be done, according to the rules for feasibility that Infrabel uses, our tool Leopard will find a solution that platforms all 100% of trains. It may find a solution the planners found as well or a different one. It will usually find a solution much quicker.

Some recent research has shown that it is possible to solve timetabling and routing and platforming together. However, we could only demonstrate this for areas around stations [9, 10, 6]. These solutions are feasible according to the Infrabel criteria. [1] demonstrate this at the signalling level, generating microscopically feasible solutions, for a case study for a corridor of 15 macroscopic points with 40 trains. These approaches consistently use some form of iteration. In general it has not been proven that this leads to the optimal solution. For [1], for the minimum headways in the last iteration, the solution is optimal according to their objective. Note that these approaches did not use a passenger weighted objective function yet.

1.3.2 Our Research Model

We start from a given macroscopic timetable which defines all arrival and departure times for all trains visiting a station. These times are not changed in our model. We set up a Mixed Integer Linear Programming (MILP) Model that platforms and routes as many trains as possible. This is the objective Infrabel wanted to use. So in this problem, there is no modulation for the number of people per train and we do not minimise passenger time. All trains are considered to have the same importance. We indicate which trains cannot be placed and display placed and unplaced trains graphically so that the planner can consider what times to change and resolve that manually. If there is a manually constructed platforming solution already present in the Infrabel databases, this solution is also checked for possible train conflicts; double use of platforms and double use of routings. The manual platforming plan can be used as reference assignment in the optimisation, so that deviation from the chosen platform tracks, counted in number of tracks deviated, can be used as penalty and as such be discouraged.

1.3.3 What Our Research Solves

In our platforming model, we generate constraints that avoid that train pairs use the same infrastructure resource at the same time. We avoid generating such constraints when we can be sure that they will never overlap in time. This speeds up model construction. It also speeds up model resolution significantly compared to other research [2]. Contrary to some other research that allows train arrival and departure times to vary slightly [46, 17, 3], we, like [5] and [2], adhere to the fixed arrival and departure times of trains. For us, this has the advantage that our model and tool can be embedded in Infrabel's process flow.

Note that in our platforming model we do not require 3 minutes separation time between trains. Macroscopic timetabling is supposed to avoid some of the conflicts in stations indirectly in some cases. Firstly, for trains that enter or leave the station on the same open line track, a separation of trains of 3 minutes is imposed at the macroscopic timetabling level. Of course the 3 minutes separation on that level can be enough there but can potentially be too little inside a station. If this is detected in the microscopic analysis at the signal and block level, and say a 3.5 minutes separation would be required, the macroscopic timetable has to be adapted. In case the macroscopic timetabling is constructed manually, this timetable has to be manually adapted by shifting apart the trains by an extra half minute. In case our timetabling tool RhinoCeros is used, this could be done by incrementing the minimal headway time between these trains to 3.5 minutes and rerun the tool. Note that we did not yet implement the possibility of specifying different minimum headways per train couple of subsequent trains. Secondly, some train couples can be on different open lines, and so not need the 3 minute separation outside the station, but can still generate a conflict inside the station. Leopard will try to choose a platform and connecting routes to avoid any conflict. If this is not possible, shifting arrival and departure times and adapting the macroscopic timetable accordingly still has to be performed manually. At Infrabel, human timetablers are used to having control over these times.

1.3.4 Benefits to Infrabel and Passengers

We successfully apply our platforming model [39] and tool, called Leopard, to train traffic in all Belgian stations [28] and report that a similar amount of trains can be platformed as in the manual platform plans present in Infrabel databases. We obtain a higher robustness, even though the robustness against delays was not explicitly present in the objective for platforming. Per station the platforming and routing optimisation time takes about 0.25 seconds on average. Reading databases and writing out graphical files typically takes some minutes per station. A typical large station takes a human planner about two weeks to plan. So the platforming process is sped up considerably. In contrast to the current slower manual platforming method, this gain in speed allows platforming feasibility checks to be carried out for multiple alternative timetables of which the best one can then be selected. Leopard can also try to add additional train traffic compared to what is already present today. In so doing, it can check the feasibility of future traffic scenarios. When this train traffic saturates the station, this becomes equivalent to making practical station capacity estimates. Up to now no tooling was available at Infrabel to make such station capacity estimates.

Since Leopard produces platforming plans that avoid the use of the same platform by two trains at any time and avoid the use of dependent routing resources by two trains at any time, when its plans for actual train traffic are put into practice, less secondary train delay is expected and passengers will also benefit from this.

1.4 Chapter Overview

Chapter 2 compiles my main papers about macroscopic timetabling for entire countries and gives a non-specialist introduction to each. Chapter 3 does the same for my platforming and routing papers. Chapter 4 lists all other papers.and abstracts and extended abstracts that I wrote as first author during my PhD time. The publications where I was co-author are mentioned in chapter 5. Finally, chapter 6 gives the main conclusions of this thesis and also mentions useful further work that could further improve our method and the results.

Chapter 2

Cyclic Timetabling for Entire Countries

"Failure comes only when we forget our ideals and objectives and principles."

– Jawaharlal Nehru

To be able to compute a timetable that is ideal for passengers, we split up our process in two steps. The first is the *reflowing* phase, where we flow train passenger over the train network. This is discussed in section 2.1. The second step is the *retiming* phase, which computes ideal train arrival and departure times for the timetable. What we consider ideal is discussed in sections 2.2 and 2.3 and amounts to the *'objective'* of a timetable. We choose to minimise statistically expected passenger travel time in practice for all passengers together. We then apply this two step process to the case of all Belgian passenger trains, as described in section 2.4 and to the case of all Danish passenger trains as described in section 2.5. In section 2.6, we add the *'excess journey time'* to the journey time already in the objective function. The excess journey time corresponds to the time passengers wait for their train departure at their first station and wait on the next mode of transport at their station of arrival. This can be minimised as well by temporally spreading alternative trains between each couple of origin and destination stations.

2.1 Deriving all Passenger Flows in a Railway Network from Ticket Sales Data

Full paper presented at International Association of Railway Operations Research (IAROR), Rome, Italy, 16-18 February 2011

Our final aim is to automatically generate a cyclic train timetable for Belgian passenger trains that takes all passengers as quickly as possible from their origin station to their destination station, averaged over all statistically expected primary delays, say, over a year. The total expected passenger journey time depends on the chosen departure and arrival times for all trains in all stations. To be able to exactly calculate this total time, we need to know for each train ride activity from each station to each next station and for each train dwell activity inside each station, how many passengers participate in this activity. This information is not directly available at Infrabel. However, in the following paper, we describe how this information can be calculated by mimicking how passengers choose the trains that will take them from their origin to their destination in reality.

There is a lot of research about how such 'route choice models' or 'passenger assignment models' can best be constructed. Since this is not the core part of our thesis, we opt for a basic version. This comes down to setting up an event activity graph representing the train service offering as a network. Passengers know their origin and destination station and software can just as well find the quickest route and some almost as quick routes for them. This is what router software on websites also does, but we need to do it for all passengers together, efficiently. When all passengers are routed or *flown* over the network, per individual activity, ride, dwell and transfer, all passenger streams passing that activity must be added up to obtain the total number of passengers participating in that activity. Since there are about 20000 combinations of origin with destination stations ('OD-pairs') present in the ticket sales data from our main operator NMBS, we need to be efficient to perform this routing process. A modified Dijkstra algorithm with priority queue is chosen. The algorithm was also parallellised so that all cores within one computer can execute this Dijkstra algorithm in parallel with other cores. A list of computers on the network can also be specified and these computations will then also be distributed evenly and executed over these machines. The subsequent addition of stream sizes per activity is done serially since it takes relatively little time. Thanks to the priority queue and high parallellisation over 3 machines with a total of 12 cores, we now manage to calculate the routings for the whole set of Belgian OD-pairs in just a few minutes.

Our passenger assignment method potentially assigns passengers that want to go from an origin O to a destination D over more than one train route only. It first selects the quickest route, then eliminates this route from the graph and searches the quickest route again. It repeats this procedure until the found route takes more than $x\%$ more time than the very first route found. ($x\%$ is settable parameter. We usually set it to 20%.) At this point it discards the last route found and spreads passengers over the collection of found routes. More passengers are assigned to the quicker than to the less quick routes. Note that the following paper mentions that only one route is calculated, but we later adapted our software so that it quickly finds multiple routes according to the algorithm just described.

We can route over the graph representing the current timetable. In that case we can take into account the assigned ride, dwell and transfer supplements. To avoid any bias towards the probably suboptimally distributed supplements in the current timetable, we can also route over a graph that represents only the ride, dwell and transfer minima. A transfer is then supposed to be of medium quality and is supposed to take 15 minutes. As such the routing algorithm is somewhat discouraged from taking routes that use too many transfers. Note that in either case, we do not take into account train departure or arrival times or their spreading over the timetable period. This corresponds to assuming that the alternative routes are reasonably well spread over the timetable period. More advanced passenger assignment methods may use some more refined values for the transfer penalty and/or take into account some assumptions about how many passengers are informed about the timetable and/or about the spreading of alternative routes in the timetable period even if this spreading is not totally known in advance. They may also be modal split models that also decide if some passengers choose to go by car or other alternative mode of transport instead. Of course, the more factors are considered in these models, the more realistic the resulting passenger assignment will be.

The local passenger numbers that result from our passenger flow phase will be used later as constants in the timetable optimisation objective function. This will be described in detail in sections 2.2, 2.3 and 2.6.

We checked some routes we found with our software with the routes produced by the online routeplanner (now at `http://www.belgianrail.be/nl`). Most routes were clearly the same, but some smaller differences occurred as well. For some cases we investigated the differences and found that they were caused by our software extracting the slowest train per train series. [43] constructed an identical graph type and very similar routing algorithms and could verify with actual data derived from routes taken in practice and found that their algorithms predicted the correct routes for more than 95% of the time.

The following paper gives some recommendations to the train operator company concerning record keeping of OD data at the station level and performing regular passenger counting initiatives in practice. It should be added that regarding the question where to count, [26] present some useful results. More papers can be found by searching for NSLP (Network Sensor Location Procedure).

Deriving all Passenger Flows in a Railway Network from Ticket Sales Data

Peter Sels [1,2,3], Thijs Dewilde[1], Dirk Cattrysse[1], Pieter Vansteenwegen[4]
[1]Katholieke Universiteit Leuven,
Centre for Industrial Management/Traffic & Infrastructure,
Celestijnenlaan 300a, 3001 Heverlee, Belgium

[2]Logically Yours BVBA,
Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium
e-mail: sels.peter@gmail.com, corresponding author

[3]Infrabel, Department of Network Access,
Frankrijkstraat 91, 1070 Brussels, Belgium

[4]Ghent University, Department of Industrial Management,
Technologiepark 903, 9052 Zwijnaarde, Belgium

Abstract
In optimizing the national railway timetable for the Belgian infrastructure company Infrabel, we use the quality criterium of minimizing expected total passenger travel time. Optimizing the timing of all trains fairly over all passengers requires passenger flow information. Focus is usually on determining the *train service offer side*, being the train timing, without considering the *demand side* data, being the passenger flow information. In this paper, we focus on passenger flow derivation.

We start from train ticket sales data and use Dijkstra's modified shortest path algorithm on the Belgian passenger train graph to determine a route per passenger origin destination station pair. We sum route passenger flows over all these routes. We also show how, in optimization of a timetable, timing and flows should be treated as interdependent. This means that we can use a process of iteration over these two phases, which we then call *reflowing* and *retiming*.

The infrastructure company did not have this passenger flow information available for the current schedule, but they need it to optimize many decisions. We determined these passenger flows for the current timetable.

We also give some recommendations regarding the improvement of passenger flow related data recording procedures, which would significantly improve passenger service.

Keywords
Railway timetabling, Passenger service, Passenger flows, Shortest path routing, Dijkstra

1 Introduction

In the context of setting up a *good* national railway timetable for the fixedly treated lines of Belgian passenger trains, the question arises as to what *quality criterium or criteria* should be used. For a client driven organization, asking the passengers is the logical answer. In questionnaires, this - not surprisingly - consistently points to lower total travel times as number one on their wish list. Travel times can be seen as average times or worst case times, but the variation in times is also required to be low. Low variation is a property of a robust schedule. To objectively evaluate any of these criteria, total passenger time is to be summed over all passengers, each one having the same weight. For this to be possible, we need to know passenger numbers on any unit piece of their trips. Rather surprisingly, it turns out, that this data is not available. A chip card system would give this information but is not yet implemented in Belgium. Yearly, partial counts are performed, but this generates too little information to be usable. This means that up to now, no such objective quality criterium was used in optimizing a railway schedule in Belgium. We need these passenger numbers, so we set up a procedure to derive them from ticket sales and some other data. Later, we give some recommendations for data collection that could greatly improve results, and are easy to perform.

In Section 2, we introduce the train network graph, how the reflowing and retime phase work on this graph, and how iteration over them works and decreases the total expected passenger time. Section 3 explains the hourly asymmetry inherent to real passenger flows, the lack thereof in the available data and how data collection can be improved. Section 4 describes how zone to zone based ticket sales numbers can be converted to approximate station to station based data and again how data collection can even improve its results. Section 5 describes how, from these station to station passenger numbers, passenger routes and the resulting flows can be derived. It also shows how this procedure can be used on an existing as well as a still to be optimized train timetable. We give a practical algorithm, verify its output data where we can, and show what is new and useful about this data. Section 6 sketches how the problem scope should be extended when the effect of a change in Origin-Destination-matrix has to be included. Section 7 summarizes the main contributions of this paper while Section 8 mentions our plans for related further work.

2 The FAPESP Problem

In formalizing the problem and the graph to model it, we are inspired by the formulation of the timetable problem as a Periodic Event Schedule Problem (PESP) [7, 8, 10, 11, 12, 15, 16]. This is the problem of determining good or optimal values for periodic event times or equivalently the durations between these event times. Since PESP does not consider (passenger) flow variables, we need to extend it. We propose to call the extended problem FAPESP, for Flow Allocation and Periodic Event Scheduling Problem. Comparable research has been performed by Bouma et. al. [1], Hofker et al. [9] and Exel et. al. [6] and led to the tools PROLOP and TRANS, developed by QQQ Delft, for passenger flow derivation on the Dutch train network. The graph we set up is similar to the PESP graph, but apart from a duration variable for each edge, it also has a flow variable for each edge. We describe and construct all objects in detail here, which will lead us to the final FAPESP graph.

2.1 The FAPESP Graph

Train and Passenger Actions

To be able to let passengers choose their preferred routes present in the train network, we need to model this network. This network models the *train* service. All trains intermittently ride and dwell between and at stations. Also, we have *passengers* realizing transfers between trains. This results in three types of actions. The train actions *ride* and *dwell* and the passenger *transfer* action. Every action has one passenger flow. So these are the values to be derived.

For the further optimization, it is noted that either the flow numbers on these separate actions will be used, or otherwise, flows per route will be used, where we need to be able to follow passenger flows per route, through their whole sequence of ride, dwell and transfer edges. This could be useful for example, if we need to evaluate the probability that a person following a certain route will arrive on time [13]. So we keep data-structures for both separate action flows and origin to destination routes with their associated flows.

Vertices and Edges

The set of actions can be practically represented by a directed graph $G(V, E)$ with vertex set V and edge set E. Refer to Figure 1 for construction of this graph.

Vertices come in two kinds for trains and two kinds for passengers.

- a train arriving at a station, set $V_{arr} \subset V$
- a train departing from a station, set $V_{dep} \subset V$
- a set of in passengers going to depart from and now entering a station, set $V_i \subset V$
- a set of out passengers having arrived at and now leaving a station, set $V_o \subset V$

where, $V_{arr} \cup V_{dep} \cup V_i \cup V_o = V$ and all pairwise sets are distinct.

Edges model the activities for which we want to determine flows and durations. They are all directed ones and come in two kinds for trains and three kinds for passengers:

- a train riding between its departing time at the previous station and its arrival time at the next station, set E_r
- a train dwelling between its arriving time at the current station and its departing time from the current station, set E_d
- a train turning around at its end station to go the other direction, set E_{ta}. The set E_{ta} may be considered a subset of the set E_d but in fact other constraints apply. For example passengers will not be able to enter a train when it is performing a turn around action. Also, a minimal service time for a turn around action will usually be larger than a minimal dwell time for a standard dwell action.
- a set of incoming passengers entering our system graph at a certain departing outbound train, set E_i
- a set of passengers transferring between a feeder train arriving in a station and a different, outbound train departing from that same station, set E_{tr}. Note that the transferring passenger can start his transfer at the beginning of the dwell action of

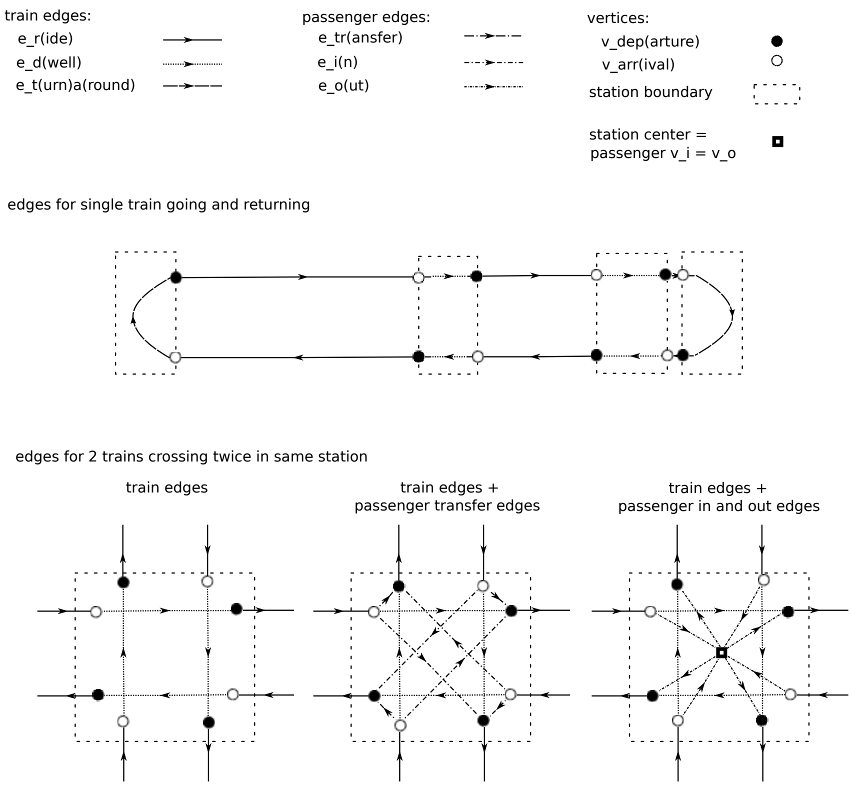


Figure 1: FAPESP EdgeTypes

the feeder train and should end transferring before the end of the dwell action of the departing train. Because an edge represents the maximum allowed time for its action, we model the transfer edge between these points.

- a set of outgoing passengers leaving our system graph at a certain departing outbound train, set E_o

where again, $E_r \cup E_d \cup E_{ta} \cup E_i \cup E_{tr} \cup E_o = E$ and all pairwise subsets sets are distinct. Note that the in and out edge types are necessary if we want to distinguish passengers flows entering (or leaving) different trains at the same station.

2.2 The Two-Phased Approach

Reflow Phase

The task of the reflow phase will be to assign a passenger flow f_e to every edge $e \in E$. Note that the reflow phase does not do any assignment to any vertex variables. The input data here is the graph structure, including the edge duration d_e of each edge e, together with the number of passengers per origin station to destination station pair, called *OD-pair*. A shortest route will be chosen for each OD-pair. This is done here with Dijkstra's shortest path algorithm. Per edge e, flows from all routes that pass through e, can then be accumulated to obtain the total flow f_e of e.

Retime Phase

Similarly, the task of the retime phase will be to assign a duration d_e for every edge $e \in E$, but this time, also satisfying all infrastructure constraints. Currently, we consider the duration for all in and out edges from E_i and E_o equal to 0. The input data here, is again the graph structure, this time annotated with the flows f_e obtained in the previous phase: the reflow phase. These f_e are treated as fixed now. The variables are d_e for which we determine an ideal value here. There is of course a minimum to each d_e per action, which we represent by m_e. The technique we use to do this is Mixed Integer Linear Programming (MILP).

On the vertex level, the retime phase results in event times t_v per vertex $v \in V$ that are separated by the durations of the edges between them. But in fact this is only a derived property. Also, if we add a constant time to each vertex time t_v, the result is still a valid solution of exactly the same quality. With quality, we mean goal function value, which is discussed next.

Same Goal Function for Both Phases

The goal of both phases: reflowing and retiming, will be to choose f_e and d_e values respectively so that a *single* goal function is minimized. In this function the assignments f_e and d_e are the complete set of variables.

We can write this as

$$G : E^{|E|} \to \mathbb{R} : e \mapsto g(f_e, d_e), \tag{1}$$

where the goal function G does a mapping from all edges e, using its f_e and d_e values of the latest reflow and retime phases respectively to a real goal function value.

So, for the reflow phase, we derive the set f_e, supposing that the d_e set is constant while for the retime phase, oppositely, we derive the set d_e while supposing the set f_e to be constant.

Each phase only has an effect on one of the variable sets. But what is important towards global optimization, is that they share the *same* goal. This is taken care of by using the same goal function for both. We can then even iterate over these two consecutive phases. Note that an identical goal function, in general, does not guarantee yet that iteration of both phases will converge to a fixed point solution. Oscillation between different good or not so good yet solutions could still be possible in steady state.

The retime phase corresponds to the aforementioned PESP problem. Because, for retiming, our goal function has fixed f_e values, it depends only on d_e. Integer linear programming techniques are often used to solve PESP. This can be done if (1) is linear in d_e. As a more specific example of a linear goal function (in this case to be minimized) we propose to use the total expected passenger time

$$g(f_e, d_e) = \sum_{e \in E} f_e d_e. \tag{2}$$

Other functions that use the assignments f_e and d_e can be used though, without breaking the workings of the iterative system.

Using a stochastic formulation of the total expected passenger time, we can even include a balance between a speedy service and robustness and still remain linear. Constraints are linear too. This will be the topic of a future paper on retiming. Concerning this approach to robustness, we refer to the papers of Vansteenwegen et al. [17, 18] and Dewilde et al. [4].

2.3 Self Sufficient Iterations

We already mentioned that the reflow and retime phases can be executed iteratively in a sequential loop. Figure 2 shows this idea graphically. The rectangular boxes represent operations, while the rounded boxes show data in- and outputs or decision points. zod_{z_o,z_d} stands for the zone origin destination matrix which specifies for every origin zone to destination zone how many passengers we have. The Mapper in Section 4 will do a mapping from this to a station to station OD-matrix, called sod_{s_o,s_d}. FA stands for Flow Allocation, or the reflow phase, which results in initial values for f_e in the first iteration or more appropriate values for f_e adapted to the changed d_e values, in all subsequent iterations. In all iterations, the retime phase PESP, will produce better values for the edge durations d_e, adapted to these changed f_e flow values. Cordone and Radaelli [2] as well as Schmidt and Schöbel [14] also came to an approach of iterations of similar two steps. We call this FAPESP iteration a self sufficient one, because the reflow phase, by being able to model peoples behavior in selecting the shortest route, does not depend on external data.

What remains to be done now is to describe in detail what we mean by the data and operations in the rounded and rectangular boxes. This is the focus of the remainder of this paper. We start with the meaning of zod_{z_o,z_d} in theory versus practice.

3 Morning versus Evening Traffic Separation, Hourly Asymmetry and Hourly Sum

3.1 Hourly Asymmetry

In fact the data for zo_{z_o,z_d} we had available is entirely and exactly symmetric in their OD-pairs.

FAPESP

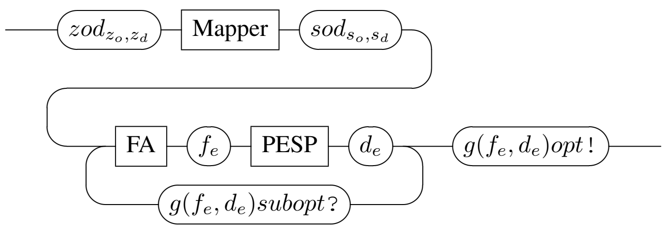


Figure 2: FAPESP Two Phase Flow Chart
offer and demand iteratively adapting to each other

This means that the number of passengers between zones A and B is always equal to the number of passengers between B and A. This was at first surprising, but is explained by the fact that it are sale figures of seasonal tickets, always valid in two directions between two zones.

However, this is unfortunate, since here information gets lost about the direction of travel. For instance, no procedure could possibly be able to separate out the passenger flows in the morning from the ones in the evening.

One could argue that this information is not needed, since timetables should be symmetric anyway. One of the reasons given is that *people who travel one way in the morning usually come back in the evening*. This is true, but not the correct reason. Symmetry is a property of a timetable, even within one period, usually one hour [11].

To know

$$\forall e \in E : d_e = d_{s(e)}, \tag{3}$$

where $s(e)$ is the symmetric edge of e. The symmetric edge for a train riding from station A to subsequent B is the edge for the opposite train going from station B to A. This implies that every train has another train in the opposite direction. But e can be any train or passenger edge type, ride dwell, transfer, in, out. What is important here is that both edges e and $s(e)$ make part of the same hourly instance of a the full days schedule.

However, within one hour, commuting flows are certainly not symmetric. Typically morning flows bring people from smaller places to bigger cities and oppositely the evening flow brings people from big cities to these smaller places again. So separately treated, morning passenger flows within each morning hour is not symmetric nor is the evening passenger flow. This means that, morning and evening separately treated again, we would need fewer trains per hour in one direction than in the other.

The main reason that still, per hour symmetric schedules are common, are rolling stock constraints.[8, 11] Indeed, if the frequency of trains from A to B is higher than backwards from B to A, material availability will increase in B and decrease in A. In fact this would be good to support the backwards flow in the evening, but this requires storage capacity and also leaves material sitting idle, which all have a cost too. These costs should in fact be weighted against the extra cost of requiring a symmetric schedule. For a clear description of benefits and costs of symmetry, we refer to Liebchen [11].

For now, missing data, working for short term and also considering that material handling cannot be immediately totally redesigned from scratch, we will keep using the symmetric simplification of the OD-matrix. But we will remember that using symmetric flow data misses a chance on a passenger service improvement of which the importance is yet unknown.

3.2 Separation Recommendation

As conclusion of the previous paragraph we can recommend an improvement to the data collection procedures for the train operator company.

At the source, let the train passengers specify their morning source and morning destination at ticket purchase time and keep accurate record of this. This would show the asymmetric nature of passenger flow data, which is important to keep, to be able to later, correctly balance passenger service benefits with material handling costs.

4 The Mapper: Mapping Zone Pair to Station Pair Numbers

We mentioned in 2.3 and showed in Figure 2, that we start from seasonal subscription ticket sales data to derive local flows on edges. Note that, this information represents the current customer OD-matrix and not the desired one, independent of the current train service offer. This corresponds in choice modelling with *preferred* versus *stated* preference. Also, we will see that this information is not yet in a directly usable format to start calculating routes for them in our graph, and how we can convert it first. This is the task of the mapper.

4.1 Zones instead of Stations

Tickets, in Belgium, are issued and sold, from one *origin zone* to a *destination zone*. A zone is a collection of geographically clustered stations. This means that from the ticket sales data, only the origin zone and destination zone are known, and not the specific respective stations. This is unfortunate. We will have no other choice than to somehow proportionally map zone to zone pairs to station to station pairs.

4.2 Zone to Station Matrix Conversion

From an Infrabel study, we had available the absolute numbers poM_s of departing passengers per origin station per Morning, based on actual physical countings. In these countings, passengers at the origin station were not asked what destination they were going to. So these countings only give a single number per origin which is the total for all the destinations corresponding with this origin.

To obtain the (departing) importance of a station in a zone, we derived all ratios of each stations' poM_s over the sum of all poM_s over the stations in their zone. Formally, if $zodM$ is the origin destination matrix expressed in zones for the Morning, and S_z the set of stations in the zone z, then the ratios roM_s standing for ratios of origin in the Morning are defined and calculated as

$$\forall (z_o, z_d) \in dom(zodM) : \forall s \in S_{z_o} : roM_s \equiv \frac{poM_s}{\sum_{s' \in S_{z_o}} poM_{s'}}, \tag{4}$$

where $dom(zod)$ represents the *domain* of the zone origin destination matrix zod, which represents all the origin-destination zone pairs. (The range, $ran(zod)$ would be their respective passenger numbers.)

Similarly, for the arriving importance of a station in its zone, on the destination side we have

$$\forall (z_o, z_d) \in dom(zodM) : \forall s \in S_{z_d} : rdM_s \equiv \frac{pdM_s}{\sum_{s' \in S_{z_d}} pdM_{s'}}, \tag{5}$$

where pdM_s are the numbers of passengers arriving per destination station in the Morning, and rdM_s their correspondingly derived ratios. In fact the pdM_s were not counted and are not available, but are necessary for the explanation that follows. We will show later how we solve this problem.

With poM_s and pdM_s values, we could derive the station to station Morning OD-matrix $sodM$ from the zone to zone Morning OD-matrix $zodM$ as

$$\forall (z_o, z_d) \in zodM : \forall (s_o, s_d) \in (S_{z_o} \times S_{z_d}) : sodM_{s_o,s_d} \approx roM_{s_o} \times zod_{z_o,z_d} \times rdM_{s_d}. \tag{6}$$

As an example, suppose we have 100 passengers, going from one origin zone to a destination zone. We suppose now that the departing respectively arriving passengers are counted on origin and destination zone stations and give the poM_{s_o} and pdM_{s_d} numbers given in Table 1. This table also shows the sod_{s_o,s_d} numbers derived by (6).

Table 1: Derived Origin Stations to Destinations Station Matrix Example

		pdM_{s_d}	10	20	40	30
		rdM_{s_d}	0.1	0.2	0.4	0.3
poM_{s_o}	roM_{s_o}	$sodM_{s_o,s_d}$				
40	0.4		4	8	16	12
10	0.1		1	2	4	3
50	0.5		5	10	20	15

In fact, the situation is more complex. In practice, the roM_{s_o} ratios can be different across origin stations, resulting in different ratios per destination station for each origin station. This exactly represents the unrecoverable data loss resulting from being supplied a zone to zone matrix instead of a station to station matrix. Hence also the approximation sign in (6).

4.3 Lack of Destination Countings

As we mentioned in 4.2, only departing passengers were counted, and not the arriving ones, the pdM_{s_d} values. So we had to approximate rdM_{s_d} by roM_{s_d} which further approximates (6) to

$$\forall (z_o, z_d) \in zodM : \forall (s_o, s_d) \in (S_{z_o} \times S_{z_d}) : sodM_{s_o,s_d} \approx roM_{s_o} \times zod_{z_o,z_d} \times roM_{s_d}. \tag{7}$$

In conclusion, using (7), our double approximative Mapper procedure derives a station to station OD-matrix from the zone to zone one. Certainly passenger flows between stations that are important within their respective zones will still be assigned an important part of the traffic between these respective zones.

4.4 Results

In Belgium, we define 497 zones, but only 21 contain more than one station. The number of stations per zone varies from 1 to 7, with the exception of Brussels zone, which contains 31 stations. The zone OD-matrix zod has 19087 zone to zone OD-pairs. Our Mapper, programmed in C++, expands these to 61725 station OD-pairs. We say that the matrix is more *diffused* now. This process takes only 0.97 seconds on a 2.88 GHz dual core Apple machine. Other processes in FAPESP will be much slower than this, so it is not necessary to further speed this up.

4.5 Station Resolution versus Station Countings Recommendations

We have shown that due to the absence of accurate station to station flow data, even a procedure trying to reconstruct it with other data available, can only be an approximation of the original. So, as conclusion of Section 4 we can recommend a more ideal, yet low cost improvement to the data collection procedures for the train operator company.

At ticket purchase time, let the train passengers specify his or her morning source station and morning destination station instead of just recording zones, and keep accurate record of this. This would represent more correctly, the resolution of passenger flow data, which is important to avoid any station within zone uncertainty for the optimization stage. This then, would make the Mapper totally redundant, which is a good thing.

If the OD-matrix is only available on the zone level, we need to rely on countings to obtain the OD-matrix on the station level to do the mapper transformation. The countings should be performed well, which means that:

- Departing passengers should be counted at origin station and arriving passengers at destination station. As demonstrated these countings poM_{s_o} and pdM_{s_d} are both necessary.
- Better still, would be to also ask the departing passenger what is his destination station, as such obtaining $sodM_{s_o,s_d}$ directly. This requires talking to instead of just counting passengers which is more involved. Note that to be entirely accurate even transfer passengers should be excluded in these countings, which makes countings even more complex.
- This is another reason why directly defining tickets and subscriptions in terms of stations instead of zones is largely preferable to these compensating counting methods.

5 The Router: Routing Station OD-Pairs to Determine Flows

The Router, or FA module, has two main tasks. It needs to derive a graph from existing trains and add transfers. Then it needs to perform the actual routing over it, for all station OD-pairs, meanwhile accumulating their flows over the route edges.

To realize this, we have to find the routes that each passenger going from origin station s_o to destination station s_d will choose. Passengers will usually choose the shortest or almost shortest route in time. Some bad experience along a slightly shorter route with varying time, for example due to a transfer often missed, may drive the passenger towards a slightly longer but more duration stable route. For simplicity, we currently supposed that we can select the shortest route. Before doing the actual routing, we start with constructing the graph that routes will be chosen from.

5.1 Graph Derivation

The graph is the FAPESP graph whose elements are defined in Section 2.1. There are two actors in our graph, trains and passengers. We start with trains.

Dwell And Ride Edges From Existing Train Lines

To produce results relevant to todays operations we started from the existing passenger train set. We focus on the morning peak. We want to end up with a train set that can be copied from hour to hour and as such must be representative for every morning peak hour. We first had to make a decision on what trains to select for inclusion in the graph. We chose the trains that run during the hours 6 to 10 as the ones to start from. Then, per couple of opposite trains occurring every hour, we took one representative direction and hour. We noticed that symmetry is not perfectly maintained within each train couple, but we consistently chose the first train in the couple as representative. There are also deviations from periodicity, which meant that for a series of hourly repetitions, we had to choose a certain representative hour. We took the criterium of selecting the hour of the train that used the most capacity. This means it is riding for the longest time. The rationale here is that, if we are able to plan this train, we will also be able to plan a train that uses less capacity. Among the quasi-symmetrically reversed trains, the same capacity-based selection process is used.

This way, we obtained 205 trains over different categories as they exist in Belgium. These trains stop and pass through 1114 points of which 550 are stations and 564 are other reference points. The combinations of these trains with these points gives 11771 vertices and service edges in our graph. Graph edge numbers are given in Table 2. The train categories are given in order of having few to many stops per distance travelled. Note that the P trains, which are *peak trains*, do not have an opposite train during the same hour, indicating the apparent benefit of asymmetry during peak hours.

Table 2: Belgian Passenger Train Graph Main Figures

Train Type	Lines	Service Edges		Potential Transfer Edges to					Total
		Ride	Dwell	**IC**	**IR**	**L**	**CR**	**P**	
IC	50	2294	2244	2897	2205	1338	989	38	7467
IR	41	1390	1349	2159	1431	1181	682	36	5489
L	92	1723	1631	1319	1184	1542	238	47	4330
CR	20	528	508	989	701	237	850	54	2831
P	2	53	51	35	34	45	50	0	164
Total	205	5988	5783	7399	5555	4343	2809	175	20281

Many Potential Transfer Edges

Apart from ride and dwell edges, Table 2 also shows a lot of *potential* transfer edges, 20281 in total. We explain here what they mean and why there are so many.

The railway companies have defined where they want to guarantee a transfer. They also have minimal times for these, which they consider in planning. There are however, a lot of *other, non-planned, potential* transfers, whose duration corresponds to the difference between planned arrival and departure times of a pair of trains in the same station. If this time difference is practical, say around 2 to 20 minutes, they will be *exploited* by some passengers. We should not ignore these other transfers.

When we are in the process of scheduling, we do not know the relative order of trains yet. We only know minimal durations of ride and dwell actions m_e per edge e. Suppose a train A arrives at time $t_{arr,A,S}$ in station S and a train B departs at time $t_{dep,B,S}$ in the same station. We model a vertex $v_{arr,A,S}$ for the arrival of train A in S and $v_{dep,B,S}$ for the departure of B in S. We want to model a transfer edge e_{tr} from $v_{arr,A,S}$ to $v_{dep,B,S}$. Suppose as well that a walking time between the platforms of train A and B takes time $m_{e_{tr}}$ The condition that the transfer can be realized on time, is that

$$t_{arr,A,S} + m_{e_{tr}} \leq t_{B,dep,S}. \tag{8}$$

The minimum time $m_{e_{tr}}$ is a known constant but $t_{arr,A,S}$ and $t_{dep,B,S}$ are still unknown. So how can we avoid mapping passengers on a transfer that may go back in time? The answer is that the transfer is always possible, since due to periodicity, from A, there is always a next-hour-B to transfer to, however, sometimes with a high transfer time.

This means that we can and should define potential transfers between all train pairs stopping at a common station, even in both transfer directions. Refer to Figure 1 for the 8 transfers edges generated by the crossing of two train pairs in a station. We have modeled 169 morning trains of which quite some pairs can run along and stop at the same stations, and the number of transfer edges is quadratic in the number of trains per common stopping station. So it is not surprising that 10879 potential transfers can result. Table 2 shows these 10879 potential transfers, divided up between each pair of train categories. Their importance shows the potential interactivity between different categories.

Some reduction in this high potential transfer number could be made here. In order of increasing crudeness, we could disallow transfers

- from a train to other trains that simply go back to where a passenger just came from,
- between trains with the same direction on a common line in every common station. (Selecting a first or last transfer over their common trajects is sensible.)
- in very small stations which are supposedly only end stations or departing stations for people.

For now we try to reduce development time, and still leave all transfer edges in the graph. We prefer to spend our efforts on tuning the retiming MILP model, such that the solver can cope with the full graph in a reasonable time and full optimality can be reached.

5.2 Flow Derivation on a Timed versus Untimed Schedule

To derive how flows are distributed over a graph, we need to distinguish three cases. First, we can use this procedure for an already planned schedule, that has already a duration for each of its edges. We call this a *timed* schedule. Second, in the case of optimization from scratch, no edge durations are known for the initial schedule. We call this an *untimed* schedule. Third, in all next iterations retiming has at least been performed once, so these are called *retimed* schedules. We examine the differences in treatment here.

Flow Derivation on a (Re)Timed Schedule

In both a timed and a retimed schedule, we make use of the planned edge durations d_e. This is in general the minimal time required for the action e, called m_e, plus a certain supplement, called s_e, which is defined in the planning or retime phase respectively. Indeed the task of the planning as well as retime phase, can be said to be defining, for every e a supplement $s_e \geq 0$, on top of the minimal action times m_e. The result is

$$\forall e \in (E \setminus E_{tr}) : d_e = m_e + s_e \geq m_e. \tag{9}$$

So in both contexts, we can use already chosen d_e values. We excluded the transfer edges from 9 because they require special treatment. Their durations can be derived from the other edges as

$$\forall e \in E_{tr} : d_e = (t_{e_+} - t_{e_-} + T) \bmod T, \tag{10}$$

where e_+ is the end vertex of the edge e and e_- its beginning vertex and T is the period of our graph. Since we suppose periodicity T, say one hour, of all trains in our graph, also the transfer between every train pair can occur every hour. Consequently, these two trains can also never be more than one hour apart, so $\forall e \in T_{tr} : d_e < T$. Note that we could add a penalty to (10), being the time for the inconvenience of having to take a transfer instead of spending an equal total time on a slower train without a transfer. A subjective time value is possible here. This has been described by Vansteenwegen and Dewilde et al. [4, 17, 18].

Flow Derivation on an Untimed Schedule

We mentioned before that we can also use this procedure on a schedule, which is still being optimized. This means that in the FAPESP context, it has not passed any retime phase yet. We should take care that, in this case, no prejudice is present in our graphs edge durations. We believe, this means we can choose the values $d_e = m_e$, since this is equivalent with supplements $s_e = 0$, so

$$\forall e \in (E \setminus E_{tr}) : d_e = m_e. \tag{11}$$

This, at least a priori, gives no preference for any edge to get extra slack over other edges.

We again excluded transfer edges from (11). We could, here as well, use (10) to derive transfer edges durations. However, since current edge durations from (11) are not really to be trusted yet, we do not want to derive them from this still uncertain information.

We want to select initial times for the transfers that are not given preference to any of them, so they all need to get the same initial value, say C_{tr}.

$$\forall e \in E_{tr} : d_e = C_{tr} < T. \tag{12}$$

On the other hand, since we need to model peoples choices in choosing train routes, we want to avoid choosing too many transfers. Hence we choose a value for C_{tr} that is equal to the alternative cost in time of being on riding or dwell edges. We took

$$C_{tr} = 15min \tag{13}$$

as an estimate for this penalty. So 15 minutes is considered a balance between a transfer short enough to be considered interesting and taken by passengers and long enough to not result in people changing forth and back at every station between parallel trains in wanting to skip dwell times longer than this transfer time.

It may seem strange that (12) breaks the usual edge constraints in (10) which seem to have universal validity in any graph where vertices represent absolute times and edges durations between them. In fact, it should, but consider that this is only a postcondition of the retime phase and not a precondition of the reflow phase. Dijkstra and other routing algorithms are certainly able to cope with a schedule satisfying (12) instead of (10), since they only use edge duration variables and no vertex variables.

In conclusion, in the FAPESP iterative context, (11) and (12) are the ones to use for *bootstrapping* the procedure in the initial iteration. All next iterations will be able to use (9) and (10).

5.3 Router Algorithms

The main part of our reflow phase is the router algorithm. There are many algorithms described in literature. We start with a classic shortest path algorithm.

Dijkstra Shortest Path Algorithm

The single pair Dijkstra shortest path algorithm [3, 5] finds a sequence of edges, called a *route*, in a directed graph between two specified vertices v_o and v_d, that has minimal length. The route length is defined as the sum of the length of all edges belonging to this route. In our case, we use as edge length for edge e, the time d_e, so this results in a route with minimum duration. The Dijkstra algorithm only works for non-negative edge lengths. Since our durations are non-negative, this is fine. It uses a greedy breadth first search and a process of relaxation that allows it to make shortcuts. The total running time of this standard algorithm is $O(|V|^2 + |E|)$ which is $O(|V|^2)$ [3]. Our average Dijkstra execution time was 1.28 seconds per pair, on a 2.88 GHz dual core Apple machine, which still seems quite high.

Modified Dijkstra Shortest Path Algorithm

Since for our graph $|V| = 11771$ and $|E| = 11771 + 20281 = 32052$, it is sparse. For sparse graphs, it is recommended [3] to implement a priority queue, with a binary heap, which then results in the *modified* Dijkstra algorithm. The total running time then becomes $O(|V| + |E| lg|V|)$ which is $O(|E| lg|V|)$ [3]. This predicts an asymptotic time saving factor for going to the modified version, of $O(|V|^2/|E| lg|V|) = 40107$. Running the serial Dijkstra shortest path algorithm sequentially over the 61725 station to station pairs, on the same machine, now takes 23 minutes 45 seconds, so 1425s. This is an average of $1425s/61725 = 23ms$ per OD-pair. So the binary heap implementation of the priority queue gained us a factor $1.28s/23ms = 55.6$ which is quite a bit lower than the predicted magnitude for the asymptotic case but still satisfying.

Further Possible Improvements

Implementing the priority queue as a Fibonacci heap instead of a binary heap could still lower execution time to $O(|V| lg|V| + |E|)$ [3].

On multicore systems a parallel version which can execute multiple route findings in separate threads simultaneously could also be implemented.

More essentially, instead of considering all the $O(|V|^2)$ OD-pairs separately, we could cache results of sub-routes already calculated in executions of previous OD-pairs and reuse them. This idea occurs also in the approach in the *all pairs* Dijkstra Algorithm [3]. Johnson's algorithm has the best asymptotic performance from the ones mentioned by Cormen et al. [3]. Instead of executing singe pair algorithm $|V|^2$ times sequentially which would give a performance of $O(|V|^3+|V|^2|E|lg|V|)$, Johnsons algorithm gives $O(|V|^2lg|V|+|V||E|)$ when using the Fibonacci heap. This looks promising, but since we expect retiming to take much more execution time than the current 23 minutes for reflowing, we, for now, prefer to spend our efforts there rather than here.

We have now described the reflow phase. We now want to execute the reflow phase on the current schedule to know current flows. Then we want to see if we can do better by running the reflow phase on the untimed schedule.

5.4 Reflowing the Current Timed Belgian Schedule

The current schedule is a timed one as defined in 5.2.

Output Data Consistency Check

The first post condition we check is that flows add up in every vertex. We have the *flow law*

$$\forall v \in V \setminus (V_i \cup V_o) : \sum_{e \in v_+} f_e - \sum_{e \in v_-} f_e = 0, \tag{14}$$

where v_+ is the set of in edges of vertex v and v_- its set of out edges. This check passes, which means that flows for each route have been properly accumulated over their edges.

Output Data versus Expectation Checks

We listed all edges sorted by decreasing flows and see if we get results that we recognize in reality. The 10 edges with the highest calculated flows in the country are all around Brussels, which is known as the bottleneck in Belgium. Indeed, a lot of people commute by train to Brussels every day. Edge flows on them vary around 10000 people per day. These are all ride and dwell edges, some consecutive.

The transfer edge with the highest flow is number 866 in the list with 3029 transfer passengers per day, also in Brussels, between a fast IC train and a more local CR train. This means people come from far and have to take a second local train in Brussels to get to their jobs.

We can also look at stations. A first check is to compare the in and out numbers with the ones we derived in the mapper. This is a close match. Where we see differences, they can partly be attributed to the approximations described in the Mapper Section but also to the fact that the absolute numbers we have are averages over the years 2003 to 2007 while our OD matrix dates back from 2002. We see a growth over the years of in and out flows, which varies a bit over different cities.

We checked the transfer flow numbers, per station. From an Infrabel expert, we know what are stations with many transfers in reality. For the current schedule, to see if we obtain the same list, we sum all transfer flows per station and order the stations by total transfer flow in Figure 3. The stations in the list are confirmed to be the *transfer stations*.

For validation of our results for separate flows per train or train pair, we tried to obtain data about passenger counts per ride or transfer edge, but could not get this data.

Further removed from our input data, and largely dependent on the Dijkstra algorithm are the dwell edges. Dwell edges are never counted, because it is not practical because of

Table 3: Biggest Stations: Current, Robust Schedule Flows and Average Edge Durations

Station Name	In= Out p/day	Ride F p/day	Ride $\overline{D}$ min	Dwell F p/day	Dwell $\overline{D}$ min	Transfer F p/day	Transfer $\overline{D}$ min
BRUSSEL-NOORD	20202	304820	1.6	158844	0.8	18001	6.9
BRUSSEL-ZUID	24403	321084	1.6	174662	0.8	13715	5.4
GENT-SINT-PIETERS	18201	73259	2.7	10252	2.2	8211	8.2
OTTIGNIES	2929	52873	1.8	17343	0.9	7766	8.1
MECHELEN	8294	95020	2.0	52031	0.4	6949	6.4
DENDERLEEUW	2900	68479	2.5	23959	1.8	7381	6.7
BRUSSEL-CENTRAAL	36983	203596	1.3	59419	1.0	5419	6.3
LEUVEN	10879	80074	2.5	31865	1.2	4313	7.4
LIEGE	6554	31750	2.6	12478	0.6	3521	7.4
NAMUR	9083	43395	2.9	15861	0.6	2950	10.4
ANTWERPEN-BERCHEM	6351	52855	0.8	17751	0.8	2327	6.7
BRUGGE	8590	84921	2.0	38947	0.2	1722	10.2
CHARLEROI	5149	17968	2.3	2601	1.4	1683	10.4
ANTWERPEN-CENTRAAL	8186	24523	1.8	2430	1.0	1647	6.1
KORTRIJK	4500	25185	1.6	7020	0.2	1078	17.0
MONS	4646	18828	3.6	4050	1.4	718	14.0

moving people, so this data is entirely new. For Brussels Central we come to 58257 daily dwell passengers, dwelling 1 minute on average.

We consider our output data starting from the current schedule, validated enough to have confidence in our reflow procedure. We now want to improve on this schedule, by starting from a clean slate, meaning on the untimed, unbiased schedule as defined in 5.2.

5.5 Reflowing the Untimed Belgian Schedule

We also ran our Router on the untimed Belgian schedule with the same trains, but with edge durations $d_e = m_e$ and transfers equal to 15 minutes. This gives the result in Table 4. Of course as for the current schedule reflowing in 5.4 we checked and passed the flow law. However, since the untimed schedule is an abstract one, contrary to the current schedule reflow results from 5.4, we cannot compare with any data from reality.

We see that the untimed schedule numbers in Table 4 are different from timed schedule numbers in Table 3. Comparing both is treacherous though. We have to realize that the untimed schedule does not meet the requirements (9) and (10) for a final, timed schedule yet. The requirement (11) it does meet, results in lower edge durations and averaged per station durations $\overline{D}$ than when it has to meet (9) with $s_e > 0$ for some edges e.

The requirement (12) with $C_{tr} = 15min$ it meets, allows it to optimize flows more transfer time unbiased and hopefully better than when it has to meet (10) directly. Indeed for the timed schedule a couple of opposite transfers will sum to the period T, one hour here. For the untimed one, each transfer takes 15 minutes, so the sum of each opposite transfer couple is only 30 minutes. To be able to compare the two tables, we need to pass the untimed schedule through the retime phase first, so that it will meet (9) and (10). Apart from this retiming, we need to realize that the current schedule is planned to be robust too, while our untimed schedule, even if passed through retiming, will still be *tight* [8], meaning that it has only the minimal amount of supplements $s_e \geq 0$ to make the schedule feasible. So, in general, more than these supplements will have to be included to make it also robust, which is amongst others, needed for *reliable connections* [8]. This can be done by including

Table 4: Biggest Stations: Untimed, Tight Schedule Flows and Average Edge Durations

Station Name	In=	Ride		Dwell		Transfer	
	Out	F	$\overline{D}$	F	$\overline{D}$	F	$\overline{D}$
	p/day	p/day	min	p/day	min	p/day	min
BRUSSEL-NOORD	20202	287067	1.6	153069	0.8	11253	15.0
BRUSSEL-ZUID	24403	312473	1.6	169762	0.8	12660	15.0
GENT-SINT-PIETERS	18201	71557	2.7	10480	2.1	7133	15.0
OTTIGNIES	2929	52399	1.8	19399	0.9	5450	15.0
MECHELEN	8294	96507	2.0	54846	0.4	5364	15.0
DENDERLEEUW	2900	64645	2.5	21538	1.4	7885	15.0
BRUSSEL-CENTRAAL	36983	198001	1.3	61262	1.0	779	15.0
LEUVEN	10879	78962	2.5	32845	1.4	2659	15.0
LIEGE	6554	30965	2.6	12370	0.6	2951	15.0
NAMUR	9083	43755	2.9	15663	0.5	3323	15.0
ANTWERPEN-BERCHEM	6351	50715	0.8	16269	0.8	2739	15.0
BRUGGE	8590	82429	2.0	36973	0.2	1511	15.0
CHARLEROI	5149	17541	2.3	1619	1.1	2445	15.0
ANTWERPEN-CENTRAAL	8186	22454	1.8	2573	1.0	470	15.0
KORTRIJK	4500	25976	1.6	7180	0.2	1313	15.0
MONS	4646	18481	3.6	3838	1.5	757	15.0

robustness elements in the goal function [4, 17, 18]. So, under the conditions that retiming is performed and robustness measures are taken, direct comparison with the current schedule will be fair.

5.6 Route Registration versus Edge Countings Recommendation

In Section 5, we tried to reconstruct the intentional route of all train passengers. This is something we have to do, because we lack this initial information.

- But, we recommend that, at ticket purchase time the train operator company asks the traveller to supply them with the route he would normally choose himself, including specific trains and possible transfers between them. To ease this question, some routes could be suggested and the traveller could get these as a multiple choice list, including an *other route* choice which he then mentions himself. These recorded routes could be used in our system as initial routes, determining initial flows. This is the best possible initial route information we could get. This does not make the reflow phase redundant. Reflow will still be necessary for the generation of routes at ticket purchase time, and for the mentioned reflow iterations.

- As mentioned in 5.4, we would have liked to be able to check our output data with passenger numbers counted in real life, but could not obtain this data. So we can recommend to the railway companies, that, in addition to the low cost and essential questions to be asked at ticket purchase time, counting passenger flows in trains helps to complete or update the changing passenger flow picture. Since at dwell time, passengers are moving through a train a lot, it is easier to count passengers during riding and even counting transfer passengers between two trains. By making the differences, using the flow law (14) at arrival or/and departure time, the number of dwell passengers can then be derived.

6 FAPESP Extension: CODFAPESP

Dependent Iterations
In flow allocation in FAPESP, we model the expected reaction of passengers potentially changing train routes between their constant origin destination pairs, when the timing of their schedule is changed by the railway company.

What we have not modeled yet, is that, when the railway company changed the timing of the schedule, people can also have the following reactions

- People who are not currently using the train, but other means of transport like their car or a bus, are suddenly attracted by a new shorter and robust train route from home to their daily job location and back. They could decide to buy a subscription and become new members of the zone to zone origin destination matrix.
- Other people may be on a route that due to low f_e values is less important to the rail company. As a consequence, service on this route may go down, reflected in higher d_e values along this route. Some of these people may give up using the train and switch to other means of transport.

We call these people-reactions reflected in a Change of the OD-matrix the COD phase or reOD phase. Obviously, decisions about buying or giving up a railway subscription happen over a longer time, probably months or even years, than the usually, daily decisions about which route to take to and from work. We have *little hope* that modeling the reOD phase can be usefully accurate. Indeed it is much harder to model since more factors play a role in this human decision than purely calculating the shortest duration path to work. Luckily, there is also *little need* of modeling this process, precisely because changes in od_{z_o,z_d} values happen much slower than changes in f_e values. We take the simple wait-and-see approach, where we let the passengers decide for themselves and seasonly, use the result of adapted ticket sales in our model.

This rationale leads to an adapted process with an extra longer term loop for people-decided OD-matrix adaptations. We call this problem the CODFAPESP, for Flow Allocation and Periodic Event Scheduling under Changing OD matrix Problem. The process flow chart to tackle this problem is given in Figure 3.

Note that the CODFAPESP process has two nested loops with different execution frequencies as well as contexts. The outer loop describes that every season, we will need to propagate the changes in ticket sales in our model. The inner loop needs to be executed at planning time, until converged to a satisfying solution. Only then can the new planning be put in place in reality. We do not want passengers and the railway companies having to perform this inner loop in reality, implying too slowly or even never converging three-month iterations, which is what happens now.

7 Conclusions

This paper presents three major contributions. Firstly, it builds up the theoretical extension of the PESP retime phase with the passenger reflow phase to the FAPESP. Secondly a practical implementation of the FAPESP is demonstrated, which is not so straightforward as the theory suggests. Thirdly, we give some recommendations to the railway companies about data they can and should gather when they want to focus more on passenger service and optimize the timetable from a passengers point of view.

CODFAPESP

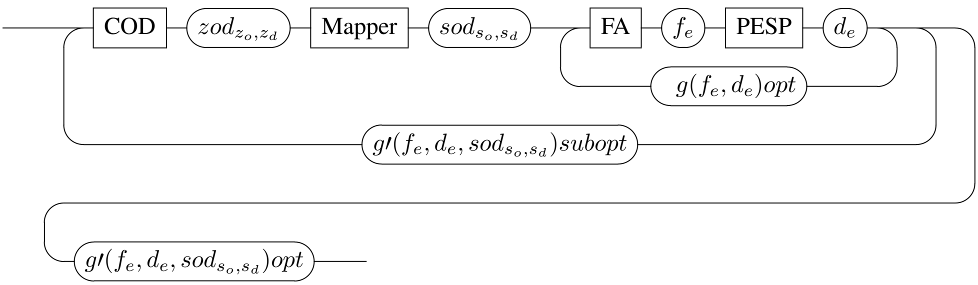


Figure 3: CODFAPESP Two Leveled, Two Phase Flow Chart
Showing effect of long term change in demand

In order to optimize the timetable later, the procedure described in this paper allows to determine passenger flows for a railway network. The passenger flows, together with the action durations, are the variable components in our minimized goal function of total passenger travel time. This initiates a practical process for railway timetable optimization.

Quite apart from the goal of timetable optimization purposes, the reflow phase can also be used separately to derive passengers flows for a complete schedule. This has been done for the Belgian network for the schedule in operation and resulted in accurate passenger flow numbers. This information has never been available before and can now be used by the Belgian railway infrastructure company, Infrabel, to weigh choices and take decisions with.

8 Further Work

We identified some further work. We need to further verify the result of flow allocation phase with future, partial passenger counting data as it becomes available from the railway company. We can still use more refined routing algorithms that account for passengers choice spread amongst all the best routes instead of only an arbitrary best route now. We can also use more balanced routing algorithms that account for passengers choice spread amongst good routes instead of only the best route or best routes. We want to complete the retime phase so that the reflow and retime phase can be iterated over and convergence can be checked. We also would like to perform FA and PESP in a single phase [14], for its own right and to be able to compare its speed and results to our two phase iterative method.

References

[1] Bouma, A., Oltrogge, C., "Linienplanung und Simulation für öffentliche Verkehrswege in Praxis und Theorie", *ETR*, Nr 43, H. 6, pp. 369-378, 1994.

[2] Cordone, R., Redaelli, F. "Optimizing the demand captured by a railway system with regular timetable", *Transportation Research Part B: Methodological*, 2010.

[3] Cormen. T. H., Leiserson, C. E., Rivest, R. L., "Introduction to Algorithms", *The MIT Press, Mc Graw Hill*, 1989

[4] Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., "Defining Robustness of a Railway Timetable", *Proceedings of RailRome*, under review, 2011.

[5] Dijkstra, E. W., "A Note on Two Problems in Connection with Graphs." *Numerical Mathematics*, Vol. 1, pp. 269-271, 1959.

[6] Exel, M., Velzeboer, J., Warmerdam, J., Colloquium Vervoersplanologisch Speurwerk 2001: "Tariefdifferentiatie in treintoedelingsmodel TRANS" *CVS 2001*, deel 3, pp. 1483-1498, 2001.

[7] Goverde, R.M.P., "Synchronization Control of Scheduled Train Services to Minimize Passenger Waiting Times", *Proceedings of the 4th TRAIL Annual Congress*, part 2, TRAIL Research School, Delft, 1998.

[8] Goverde, R.M.P., "Improving Punctuality and Transfer Reliability by Railway Timetable Optimization", *Proceedings of the 5th TRAIL Annual Congress*, TRAIL Research School, Delft, 1999.

[9] Hofker, H., Suurland, M., Warmerdam, J., Colloquium Vervoersplanologisch Speurwerk 2000: "Wie betaalt, bepaalt!" *CVS 2000*, part 3, pp. 1547-1560, 2000.

[10] Kroon, L., Dekker, R., Vromans, M.J.C.M., "Cyclic Railway Timetabling: A stochastic Optimization Approach", Geraets, F., Kroon, L., Schöbel, A., Wagner, D., Zaroliagis, C.D. (eds), *Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science*, pp. 41-66, 2007.

[11] Liebchen, C., "Symmetry for Periodic Railway Timetables", *Electronic Notes in Theoretical Computer Science* Vol. 92, pp. 34-51, 2004.

[12] Nachtigall, K., "Periodic network optimization with different arc frequencies", *Discrete Applied Mathematics*, Vol. 69, pp. 1-17, 1996.

[13] Schöbel, A., Knödl, H., "Trendanalyse von Verspätungsentwickelungen im Personenfernverkehr", *ETR*, Nr 11, November 2006, pp. 806-809, 2006.

[14] Schmidt, M., Schöbel, A., "The Complexity of Integrating Routing Decisions in Public Transportation Models", *10th Workshop on Algorithmic Approaches for Transportation Modeling Optimization and Systems, ATMOS'10*, pp. 156-169, 2010.

[15] Schrijver, A., Steenbeek, A., "Spoorwegdienstregelingontwikkeling (Timetable Construction)" *Technical Report, CWI, Amsterdam*, (in Dutch), 1993.

[16] Serafini, P., Ukovich, W., "A Mathematical Model for Periodic Scheduling Problems", *SIAM Journal on Discrete Mathematics*, Vol. 2, pp. 550-581, 1989.

[17] Vansteenwegen, P., Van Oudheusden, D., "Developing railway timetables which guarantee a better service", *European Journal of Operational Research (EJOR)"*, Vol. 173, pp. 337-350, 2006.

[18] Vansteenwegen, P., Van Oudheusden, D., "Decreasing the passenger waiting time for an intercity rail network", *Transportation Research Part B: Methodological*, Vol. 41, pp. 478-492, 2007.

2.2 Expected Passenger Travel Time for Train Schedule Evaluation & Optimisation

Full paper presented at International Association of Railway Operations Research (IAROR), Copenhagen, Denmark, 13-15 May 2013

In the previous section 2.1, describing *reflowing*, we derived all the local passenger numbers for each train ride, train dwell and passenger transfer activity in the train service network graph. In this first section about *retiming*, we start to set up a mathematical formula for the objective of a timetable. The passenger numbers calculated in section 2.1 will all appear in this function. Essentially they will be multiplied with the durations of the associated activity, ride, dwell or transfer. In this retiming step, passenger numbers are constants here, while the activity durations are the variables to be optimised. This is the actual timetabling step.

For this cyclic timetabling problem, we base ourselves on the well known Periodic Event Scheduling Problem (PESP) formulation [41]. [41] formulate a model of linear constraints, all being either linear equalities or linear inequalities. This model belongs to the class of mixed integer linear programming (MILP) models. [41] mention no objective function and also in further research the objective has been mainly absent from the timetabling models. As a consequence, thanks to the present constraints, the resulting timetable is feasible (on a macroscopic level), but due to the absence of an objective, the resulting timetable still contains time supplements of an undesirable size, either too large or too small. (These time supplements are either time reserves on ride or dwell or transfer activities or are time buffers between couples of trains that follow each other on the same railway track.) The reason is that there is nothing present in these bare PESP models that penalises for this undesirable sizes of supplements.

As mentioned before, it will seem intuitively logical that very large supplements on ride, dwell or transfer activities are not ideal. Indeed, passenger then have to wait very long and lose time. But also if these supplements are very small, a negative effect occurs. Indeed, the slightest primary delay on that activity then makes that the timetable cannot be realised as planned. This would mean that all planning was unrealistic and missing the point.

Especially in the case of a transfer, a very low supplement makes the transfer time too short and causes passengers to miss the transfer. They will then have to wait for the next train that takes them in the right direction. The optimal supplement will be the result of a perfect trade-off between the expected time costs of being in-time and of being over-time, each weighted with the integrated respective probabilities that these in-time and over-time costs occur. For each activity, these probabilities and costs can be calculated mathematically, based on the numbers of passengers in the participating activities, the type of these activities and the relevant expected delays.

To obtain a practically usable timetable, it is essential to make this trade-off. In a mixed integer linear programming (MILP) model, this can be done by adding an objective function to the bare PESP model that only contains constraints. This paper sets up the objective function of total expected passenger journey time in practice, which makes this trade-off in all locations over the whole network.

Our paper uses negative exponentials as stochastic distributions of primary delays on ride, dwell and transfer activities. Another common choice would have been a Weibull distribution. The essential difference is that the Weibull distribution has a zero on time probability and then quickly peaks to maximal probability for some very low delays to then drop down quite quickly to larger delays again. Since we already took into consideration a minimum time that contains 5% on top of the bare calculated minimum, it is seen in practice that the probability of having a 0s delay is not zero. This means that the negative exponential distribution is more suitable in our case.

In our model, we have chosen a transfer penalty of an hour because we know this is the worst case situation for an hourly timetable. A penalty of half an hour could be closer to the average value in practice. However, we prefer to have a known worst case over a possibly underestimated average. An improved modelling approach over the hour upper bound would be to model the actual waiting time until the *departure time* of next train in the *same end station or same direction*. First, the ‘ same direction’ is not a clear concept. For how many stations should it be the same direction? Even supposing that one would require the next train to be one that stops at the end station the passenger was going to, one does not know its departure time yet, so let alone the time to wait until that departure time. Even if one would manage all that, it would still have to be done for all passenger streams, split up between different destinations separately. This would be a large effort. Anyway, the fact that one does not know the train departure times yet makes it either totally impossible or close to that.

Since the following paper concentrates on the objective function, the constraints of the model are given in another publication in the journal Transportation Research, Part B, reproduced in section 2.4.

The following paper mentions that our large upper bounds on supplements solve the infeasibilities that other researches with lower artificial upper-bounds sometimes struggle with. In our paper in section 2.4, we give mathematical proof that our upper-bounds can never give rise to our model becoming infeasible. When too many trains are specified in the input, of course our model can still become infeasible, but this is a matter of lower bounds on headways and cannot be avoided.

Expected Passenger Travel Time for Train Schedule Evaluation and Optimization

Peter Sels [1,2,3], Thijs Dewilde[1], Dirk Cattrysse[1], Pieter Vansteenwegen[1]
[1]Katholieke Universiteit Leuven,
Centre for Industrial Management/Traffic & Infrastructure,
Celestijnenlaan 300a, 3001 Heverlee, Belgium

[2]Logically Yours BVBA,
Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium
e-mail: sels.peter@gmail.com, corresponding author

[3]Infrabel, Department of Rail Access,
Fonsnylaan 13, 1060 Brussels, Belgium

Abstract
When evaluating and optimizing the national railway timetable for all passenger trains in Belgium, we use the criterium of expected total passenger travel time in practice. During evaluation, we measure this, during optimization, we try to minimize this. The focus in this paper is to analytically derive the total passenger time as a stochastic function of the schedule. In a previous paper [9], we already derived the necessary local passenger flows from ticket sales data. These flows will serve as fixed weights in this new objective function.

We start from measured delays in the current schedule and assume that they can be described well by a negative exponential distribution. In the case of constructing an optimized schedule, we assume delays will also occur in the new schedule according to the same distribution. We believe this is a conservative assumption, since the optimized schedule will be more robust against delays.

We then decompose the global schedule of actions into local actions or pairs of subsequent actions, and derive probabilities and objective functions on the corresponding one or two local actions only. We sum these local functions to obtain a global objective function. This then becomes the schedules' global quality criterium. We linearize this and use the result as the objective function of a linear programming based optimizer. With this objective function, we optimized the train schedule for the Belgian railways, which proves the method is scalable to large and complex real-life problems.

Keywords
Railway Timetabling, Expected Passenger Time, Objective Function, Stochastic Optimization

1 Introduction

We consider total passenger travel time in practice as the most important criterium for judging the quality of a train schedule [1]. In such a objective function, we need to use passenger numbers [9] instead of trains as weight factors. In evaluating an existing schedule we should check that this objective function stays low and that few or no planned transfers are missed frequently, especially if many people are transferring. In optimizing a schedule, we should guarantee these properties. Not only transfers, but also ride and dwell actions should be planned with supplements that are neither too small, such that they are unable to absorb any practical delay, nor too big, such that they create unnecessary idle time in practice. In this paper, we analytically derive the stochastic expected passenger time on each location throughout the network, so that it can be used for evaluating if a specific supplement is well chosen or not. The global function of total expected passenger time is derived from this, which directly gives a practical quality criterium for the whole schedule. This global function (or some convex, linear approximation of it) will also be used as the objective function for a (linear) solver when optimizing.

2 Related Research Overview and Comparison

Concerning linear modeling of a train service, some theoretical research has been carried out already [3, 4, 5, 6, 8, 10, 11]. The first schedule that has been mathematically calculated, be it only for feasibility and not for optimality, was the one for the Netherlands, and is due to Kroon et al. [5]. Whether one of these schedules was actually put in practice is unclear. Christian Liebchen states [6] to have produced the first schedule that was mathematically calculated and then also put into practice. This included both schedule feasibility as well as optimization of some selected transfers and concerned the Berlin Underground, containing 37 train relations. We go further here, by handling a much larger network - all periodic passenger trains in Belgium - and more importantly, using an all-embracing optimization objective function: passenger time in practice, which automatically considers all *potential transfers* [9]. Since we minimize the objective function, the resulting timetable will be robust, and this, in a natural, consistent and balanced way, across all ride, dwell and transfer actions. The idea of using expected passenger time as a objective function to minimize is based on research from Vansteenwegen et al.[12, 13] and Dewilde et al. [1]. The great power of this objective function is that it gives an automatic and sensible trade-off between a speedy yet robust service, where buffers and supplements are neither too large nor too small. It also weighs and balances all minutes spent in the network fairly across all passengers. For optimization of a schedule, apart from a objective function, of course, a large number of hard constraints is necessary. We implemented and use all of these in practice, but they are not discussed in this paper. The sole focus of this paper is the objective function.

3 Decomposing the Train Service Network into Actions with Local Passenger Flows

We model each train trip as consisting of five types of actions. Figure 1 shows that each action can be represented by a rectangle with a width, indicating the planned duration of this activity and the height, indicating the number of people participating in that action. A train trip can be seen as a sequence of mainly blue *train ride* and yellow *train dwell actions*. Additionally, there are three types of *passenger actions*: green colored *source actions*, which represent people getting on the train, orange colored *transfer actions* representing passengers walking from one train to another, and red *sink actions*, representing people arriving at their destination.

Like Meng [7], Goverde [2], and Vansteenwegen et al. [12, 13], we suppose a general negative exponential delay distribution for the duration of each of the actions: ride, dwell and transfer. Source and sink actions can be considered as having a zero expected delay. The duration of source and sink actions does not influence the planning. They are only present in Figure 1 to indicate per train, the number of passengers entering or exiting the station at each of its stops. The ride, dwell and transfer actions all have a given *minimal duration*. After each of these three actions, a time supplement is added. Figure 1 shows these in a lighter color shade of blue, yellow and orange for the respective actions. These supplements contain the time to buffer possible delays of the previous action(s), but can also be much bigger than that in case the subsequent action has to start much later to allow departing or transferring passengers to enter the train considered. The durations of these buffers or filler actions are called the *supplement times*, and are the only relevant time parameters present in our objective function: the total expected passenger travel time. The optimization problem we consider here, in the so called *retiming phase* [9], is to determine a supplement time for each action, so that the total *expected* passenger travel time is minimal. Note that Figure 1 shows the *planned* schedule and that actual delays on actions can be smaller than their shown supplements, in which case there is slack time, but that when actual delays are bigger than the supplements, delay will propagate, yet hopefully be absorbed by supplements later.

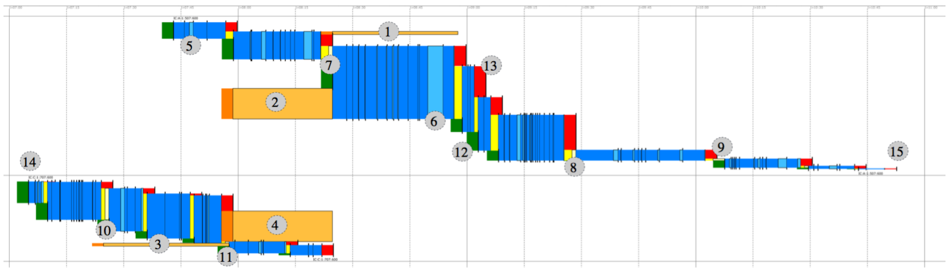
The top couple of trains in Figure 1 are taken from the current Belgian planning, and the bottom couple of trains show the time optimized version according to the objective function we will derive here. Some corresponding actions between both schedules are indicated with the same numbers in grey circles. Note that there are two passenger transfers between the two trains (Numbers 1=3 and 2=4) and that this influences the relative time alignment between the two trains. We also suppose that each of the trains occurs every hour. It must be noted that the current expected delay distributions on each action currently are realistic, but fictive. In the end, these will be replaced by distributions based on actual measurements.

Whether the bottom couple of trains takes more or less *planned* passenger time is not so relevant. However, in operation, taking all statistical occurrences of these train trips together, for all the actual numbers of passengers, the *expected* passenger time in practice will be lower than for the original top train couple. This is the case, because the bottom couple of trains will be more robust against operational delays on its actions. So, this global objective function allows us to evaluate as well as optimize the timing for trains. To derive this function, we will now start with introducing the level of action pair types.

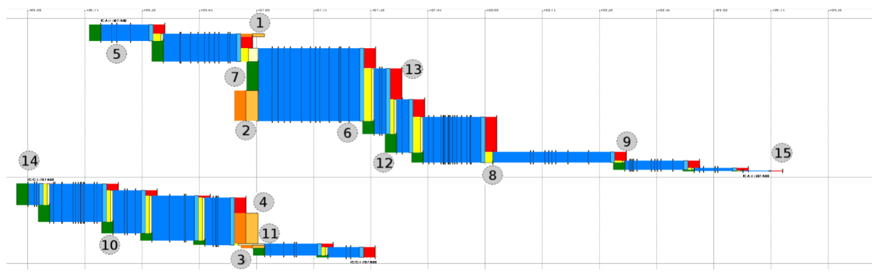
4 Regrouping per Action Pair Type

Using a bottom up approach, we now cut up a train schedule at ride departure times. We are then left with separated parts, consisting of pairs of subsequent actions.

Similar to Vansteenwegen et al. [12, 13] and Dewilde et al. [1], we call passengers who experience the successive actions: ride and dwell, *through passengers*. *Transfer passengers* are the ones taking a ride and then a transfer action to the next ride action. *Departing passengers* follow a source action to a ride action. *Arriving passengers* come from a ride action and go to a sink action. We call these four kinds of passenger types, the *passenger flow*



(a) Original Schedule



(b) Optimized Version

Figure 1: Two timetables displaying *planned* time for a pair of trains with transfers between them: Original and Optimized. Horizontal dimensions are action durations. Vertical dimensions are passenger numbers participating in these actions. Ride actions are blue, dwell actions yellow, transfer actions orange, departing actions green and arriving actions red. Supplements or slack time are indicated in the lighter shade of the preceding action they belong to. Circled numbers correspond between schedules. In the original schedule (a) , the supplement of transfer 1 (08:25 to 8:57) = transfer 3 (7:25 to 7u57) takes 33 minutes, while in the optimized schedule (b), the supplement of transfer 1 (06:59 to 7:02) = transfer 3 (06:59 to 7:02) takes only 3 minutes which is more efficient and still robust. For the opposite transfers, in the original schedule (a), transfer 2 (7:58 to 8u25) = transfer 4 (7:58 to 8u25) has 27 minutes supplement, while in the optimzed schedule (b) transfer 2 (6:58 to 7:01) = transfer 4 (6:58 to 7:01) only takes 3 minutes supplement. Again , this is more efficient and still robust. Note that since all trains are repeated periodically every hour, also the transfers between them occur every hour. For blocks of consecutive ride-dwell action pairs that have no interaction with other trains, nor entering, transferring or exiting passengers, the light blue colored supplement occurs at the end, which is indeed the best place to buffer against delays.

types. We summarize these and their corresponding action types in table 1. In conclusion, there are 5 types of actions which only occur in 4 types of pairs of subsequent actions.

Table 1: Four Passenger Flow Types and Corresponding Action Sequences

Passenger Flow Type	Action a	Action b
Departing	(Source)	Ride
Through	Ride	Dwell
Transfer	Ride	Transfer
Arriving	Ride	(Sink)

It must be clear that during their journey, a single departing passenger, becomes a through passenger, can become a transfer passenger, a through passenger again and finally always becomes an arriving passenger. Each passengers' flow type is only locally valid, for each two successive actions.

So we decomposed the total passenger flows on all trains into actions, to then recompose them into pairs of subsequent actions, corresponding to a particular passenger flow type. Each type of passenger is interested in having a minimal *expected* travel time for all the action pairs he undergoes.

For *through* and *transfer passengers*, the exact trade-off will be dictated by minimization of the expected passenger travel time. This depends on two actions and their delays, so for this, convolution of the delay distributions of both actions will be necessary.

For *arriving passengers*, only the last ride action matters for the delay on this last part of their journey.

For *departing passengers*, we will reasonably suppose their travel time starts counting at the planned departure time. Indeed, their choice to potentially arrive earlier to avoid missing their departure train, is independent of any particular time we will assign to this departure event. So their relevant departure delay is the delay of the previous ride and dwell action pair of their train, when such a pair exists. If such a pair does not exist, the departing passengers are catching a train at its very first ride action, and we suppose there is no possible delay nor associated passenger time cost.

For each of the four passenger flow type cases, a corresponding pair of action types, makes up the time in between two subsequent ride departure times. They correspond to the times we want to evaluate or optimize, weighted with the net corresponding passenger flow number on this action pair.

We know that, in reality, delays can be accumulated over a full train journey from beginning to end. This is often the case when at places where delays often occur, no time supplements are inserted to absorb these delays. In this research, we try to do exactly this: calculate supplements that absorb enough delay locally, not globally, yet not too much, since we still want to maintain a speedy service. In this way, we avoid that delays propagate in most cases. On top of this, for each train, we will accumulate sequences of *undisturbed blocks* of ride-dwell-pairs up to the point where either the passenger numbers change, or another train enters or exits the same track. For each such block, the measured delay is accumulated and the (accumulated) supplements are then correctly placed at the end of the block only, as can be seen in Figure 1(b) by the position of the light blue rectangles in the optimized schedule.

5 Local Objective Function Derivation

We now analytically derive the expected passenger time in practice of one action and of an action pair, based on their delay distributions.

5.1 One Single Action Only

Fitting Model Delay Distribution to Delays

We first consider an action a or process with a minimum time m and a probabilistic delay variable x as described by Vansteenwegen et al. [12, 13]. We can measure the occurrences of delays for this process and *fit* a normalized negative exponential distribution on these measurements. The probability $p_a(x)$ on a delay x can then be expressed as

$$p_a(x) = ae^{-ax}. \tag{1}$$

Note that (1) is *normalized* which is required for a probability distribution. Indeed

$$\int_0^\infty p_a(x)dx = \int_0^\infty ae^{-ax} = -e^{-ax}\big|_0^1 = (-0) - (-1) = 1. \tag{2}$$

The fitting is done by choosing the single negative exponential distribution that has the same expected value $\overline{d_a}$ as the originally measured discrete delay sample histogram. For action a, for the measured discrete histogram, considering samples collected in N groups, per value x_i at index i and respective number of occurrences h_i, this gives

$$\overline{d_a} = \frac{\sum_0^{N-1} h_i x_i}{\sum_0^{N-1} h_i}. \tag{3}$$

For the continuous distribution the equivalent expected value is [1]

$$\overline{c_a} = \int_0^\infty xae^{-ax} = \frac{1}{a}, \tag{4}$$

Our fitting criterium is that the expected value of the fitted distribution must equal the expected value of the original distribution. This corresponds to

$$\overline{c_a} = \overline{d_a}, \tag{5}$$

which, considering (3) and (4) gives

$$a = \frac{\sum_0^{N-1} h_i}{\sum_0^{N-1} h_i x_i}. \tag{6}$$

In Time Probability

For this section, to ease the derivation, we ignore the part of the time cost that is related to the minimum time m for an activity. This can be done because this part does not depend on any supplement choice and as such will not influence comparative evaluation nor optimization.

For now, we suppose our activity a is to be followed by another activity b that, according to the planning, starts at time D_0 later than the end of the current activity a. So we suppose

[1] Whenever a calculus step is not trivial, the PDF version of this paper allows the reader to click right of the respective equal sign, which will send the formula in Mathematica notation to www.wolframalpha.com and his invoked browser will then display the result.

that D_0 is the time supplement we will decide to add between activities a and b. Activities a and b are the pairs of activities discussed above, D_0 is the supplement between these.

We have a possibility that we end our current activity within time supplement D_0. We calculate the *in time probability* that this happens as

$$p_{a,x\leq D_0}(D_0) = \int_0^{D_0} p_a(x)dx = \int_0^{D_0} ae^{-ax}dx = 1 - e^{-aD_0}. \tag{7}$$

In Time Cost
Since all time units are supposed to be equally weighted, the corresponding unit time cost of this supplement D_0 is

$$UC_{a,x\leq D_0}(D_0) = \int_0^{D_0} D_0 p_a(x)dx = p_{a,x\leq D_0}(D_0) * D_0 = (1 - e^{-aD_0})D_0. \tag{8}$$

We call $UC_{a,x\leq D_0}$in (8) the *in time cost* of a supplement D_0. This cost is called *unit cost* because this is the cost for one passenger and for each time unit experienced with a unity annoyance weight or level.

Over Time Probability
Now we suppose that we missed the start of activity b because we were delayed in activity a by more than the introduced supplement D_0. We now also suppose that there is a second chance (for instance the next train) to catch b in time D_1 i.o. D_0, where $D_1 > D_0$. The *over time probability* that we miss b at D_0 but can catch b at D_1 is

$$p_{a,x\geq D_0}(D_0) = \int_{D_0}^{D_1} p_a(x)dx = \int_{D_0}^{D_1} ae^{-ax}dx = -e^{-ax}\big|_{D_0}^{D_1} = e^{-aD_0} - e^{-aD_1}. \tag{9}$$

D_1 could be taken to be ∞ if no further b alternatives are considered.

Over Time Cost
When the delay of the activity a somehow takes longer than D_0 we cannot catch the next activity b that starts at that time. We suppose we must then catch the alternative activity, which we suppose starts at time $D_1 > D_0$. When trying to catch the ride of another train after a transfer for example, this will be the ride action of the next train in initially the same direction. We will often find times like about one hour or half hour or so in a practical, regular train service. This can be enforced by requiring that n trains per hour leave at regular time intervals of about $1/n$-th of an hour. The parameter D_1 is supposed to be known, while D_0, in the optimization context, is the variable we want to find an optimum for. We can again, similarly, calculate the probability that we can catch the activity at time D_1 and the corresponding time cost as

$$\left\{\begin{array}{lcl} p_{a,D_0\leq x\leq D_1}(D_0, D1) & = & \int_{D_0}^{D_1} p_a(x)dx = \int_{D_0}^{D_1} ae^{-ax}dx = e^{-aD_0} - e^{-aD_1} \\ UC_{a,D_0\leq x<D_1}(D_0, D1) & = & \int_{D_0}^{D_1} p_a(x)D_1dx = p_{D_0\leq x\leq D_1}D_1 \\ & = & (e^{-aD_0} - e^{-aD_1})D_1. \end{array}\right. \tag{10}$$

Note that probabilities and unit costs are positive.

Going on one more step, from D_1 to D_2 we have similar formulas, by simply substituting D_2 for D_1 and D_1 for D_0 in (10) or in general

$$\left\{\begin{array}{lcl} p_{a,D_{n-1}\leq x\leq D_n}(D_{n-1},D_n) & = & \int_{D_{n-1}}^{D_n} p_a(x)dx = \int_{D_{n-1}}^{D_n} ae^{-ax}dx \\ & = & e^{-aD_{n-1}} - e^{-aD_n} \\ UC_{a,D_{n-1}\leq x<D_n}(D_{n-1},D_n) & = & p_{D_{n-1}\leq x\leq D_n} D_n \\ & = & (e^{-aD_{n-1}} - e^{-aD_n})D_n. \end{array}\right. \tag{11}$$

All Time Units are Weighted the Same

If all time units are considered to have the same subjective annoyance, the total probability and unit time cost over all subsequent time intervals is

$$\left\{\begin{array}{lcl} p_{a,D_0\leq x<D_n}(D_0,D_1,D_2,...,D_N) & = & \sum_{n=0}^{N-1} p_{a,D_{n-1}\leq x\leq D_n}(D_{n-1},D_n) \\ UC_{a,D_0\leq x<D_n}(D_0,D_1,D_2,...,D_N) & = & \sum_{n=0}^{N-1} UC_{a,D_{n-1}\leq x\leq D_n}(D_{n-1},D_n), \end{array}\right. \tag{12}$$

where N is the number of terms we want to consider and $D_{-1} = 0$ to simplify notation. The higher n gets, the lower the contribution to the result becomes, due to the decreasing probabilities of the integration interval between D_{n-1} and D_n. Figure 2(c), shows a plot of these probability functions. We call this the one dimensional case because only a delay on one action plays a role here. $D1 = 39$ and $D_2 = 54$, so 15 time units after the first repetition, a new repetition of the next action creates a possibility to catch it again.

Figure 2(e) shows an example of the first three terms of the unit cost in (12) for chosen values 39 and 54 for D_1 and D_2 respectively. The expected delay duration of action a is $d_a = 6$. Clearly the total time cost gets minimal when $D_0 = 10$ time units. Note that the third term, $UC_{D_1\leq x<D_2}$ is constant for all values of D_0 and as such does not influence the value of the optimal D_0 supplement. Also, this third term is much less important in absolute value, compared to the first two terms.

In practice, for this case with constant annoyance level for all types of time units, we will only use the first two terms to calculate the total cost.

Time Units Can Have Different Weights

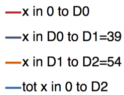
For passengers, some time units are sometimes considered more annoying than others. This depends on the activity. For example, riding feels more useful and pleasant than dwelling. Also, when train connections are missed, annoyance grows.

Like [12, 13], we model these annoyances by multiplying each time unit by a weight. We introduce the weight $w_{a,i}$ for the delay x of an activity a where i stands for the number of next activity instances that has been missed already. This holds for a passenger on a specific switching from one activity to the next. Of course, annoyance grows when missing more next activity instances, so it will hold that

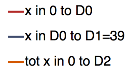
$$\forall a \in A : \forall i \in \mathbb{N} : w_{a,i} \leq w_{a,i+1}. \tag{13}$$

We further introduce $w_{c,i}$ as the weight for the left over idle time $D_0 - x$. The probability calculation remains the same as in (8) and (10) in the previous paragraph, but the corresponding unit cost integrals become somewhat more involved. Indeed,

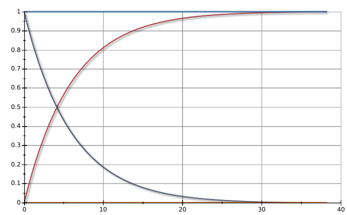
$$\begin{array}{lcl} wUC_{a,x\leq D_0}(D_0) & = & \int_0^{D_0}(w_{a,0}x + w_{c,0}(D_0 - x))p_a(x)dx \\ & = & w_{c,0}D_0\int_0^{D_0} p_a(x)dx + (w_{a,0} - w_{c,0})\int_0^{D_0} xp_a(x)dx \\ & = & w_{c,0}D_0 p_{a,x\leq D_0}(D_0) + (w_{a,0} - w_{c,0})UC_{a,x\leq D_0}(D_0). \end{array} \tag{14}$$



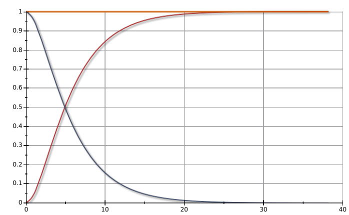
(a)



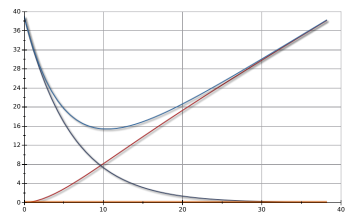
(b)



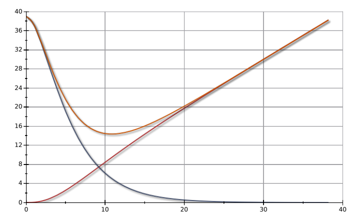
(c) Delay Probabilities for 1 Action, $\overline{d_a} = 6.0$



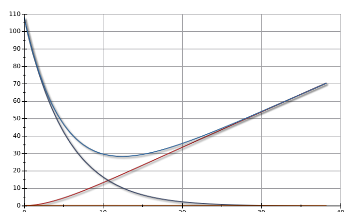
(d) Delay-Prob. for 2 Actions, $\overline{d_a} = 4.0$, $\overline{d_b} = 2.0$



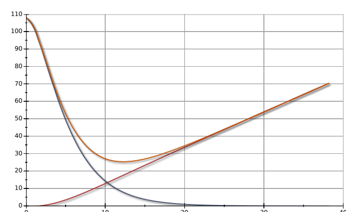
(e) Objective Unit Costs for 1 Action, $\overline{d_a} = 6.0$



(f) Obj. Unit Costs for 2 Actions, $\overline{d_a} = 4.0$, $\overline{d_b} = 2.0$



(g) Subjective Unit Costs for 1 Action, $\overline{d_a} = 6.0$



(h) Subj. Unit Costs for 2 Actions, $\overline{d_a} = 4.0$, $\overline{d_b} = 2.0$

Figure 2: Probablities (p), Expected Passenger Time (UC) and Weighted Expected Passenger Time (wUC) as functions of the (sum of) the chosen supplements (x). Different curves for supplements being less than D0 (in-time), between D0 and D1 (over-time), and their sum (total-time), on the vertical axis. x on the horizontal axis. All units are in minutes. (a) is the legend for (c), (e) and (g). (b) is the legend for (d), (f) and (h).

Note that $UC_{a,x\leq D_0}(D_0)$, in this weighted case, has been redefined as $\int_0^{D_0} xp_a(x)dx$ instead of $\int_0^{D_0} D_0 p_a(x)dx$, in the unweighted case. From the context it will be clear what we mean with it.

We derive the needed expressions for probabilities and unit costs as

$$\begin{array}{lcl} p_{a,0\leq x\leq D_0}(D_0) & = & \int_0^{D_0} ae^{-ax}dx = 1 - e^{-aD_0} \\ UC_{a,0\leq x\leq D_0}(D_0) & = & \int_0^{D_0} xae^{-ax}dx = \frac{1-(aD_0+1)e^{-aD_0}}{a}. \end{array} \tag{15}$$

Note that some terms in (14) may become negative, for example when $w_{a,0} - w_{c,0} < 0$, but the total value of $wUC_{a,x\leq D_0}(D_0)$ will always be positive. The same holds for the next interval, (D_0, D_1) where,

$$\begin{array}{ll} & wUC_{a,D_0\leq x<D_1}(D_0, D1) \\ = & \int_{D_0}^{D_1}(w_{a,0}D_0 + w_{a,1}(x - D_0) + w_{c,1}(D_1 - x))p_a(x)dx \\ = & ((w_{a,0} - w_{a,1})D_0 + w_{c,1}D_1)\int_{D_0}^{D_1} p_a(x)dx + (w_{a,1} - w_{c,1})\int_{D_0}^{D_1} xp_a(x)dx \\ = & ((w_{a,0} - w_{a,1})D_0 + w_{c,1}D_1)p_{a,D_0\leq x\leq D_1}(D_0, D_1) + (w_{a,1} - w_{c,1})UC_{a,D_0\leq x\leq D_1}(D_0, D_1), \end{array} \tag{16}$$

where we derive the expressions for probabilities and unit costs as

$$\begin{array}{lcl} p_{a,D_0\leq x\leq D_1}(D_0, D_1) & = & \int_{D_0}^{D_1} ae^{-ax}dx = \; = e^{-aD_0} - e^{-aD_1} \\ UC_{a,D_0\leq x\leq D_1}(D_0, D_1) & = & \int_{D_0}^{D_1} xae^{-ax}dx = \frac{(aD_0+1)e^{-aD_0}-(aD_1+1)e^{-aD_1}}{a}. \end{array} \tag{17}$$

Based on

$$\begin{array}{ll} & wUC_{a,D_{n-1}\leq x<D_n}(D_0, D_1, ..., D_{n-1}, D_n) \\ = & (\sum_{i=0}^{n-1}(w_{a,i} - w_{a,i+1})D_i + w_{c,n}D_n)p_{a,D_{n-1}\leq x\leq D_n}(D_{n-1}, D_n) \\ + & (w_{a,n} - w_{c,n})UC_{a,D_{n-1}\leq x\leq D_n}(D_{n-1}, D_n), \end{array} \tag{18}$$

one can prove that the formula (16), generalized for N subsequent chances on the next action, becomes

$$\begin{array}{ll} & wUC_{a,0\leq x<D_N}(D_0, D_1, D_2, ..., D_N) \\ = & \sum_{n=0}^{N}[(\sum_{i=0}^{n-1}(w_{a,i} - w_{a,i+1})D_i + w_{c,n}D_n)p_{a,D_{n-1}\leq x\leq D_n} \\ + & (w_{a,n} - w_{c,n})UC_{a,D_{n-1}\leq x\leq D_n}] \\ = & \sum_{n=0}^{N-1}((w_{a,n} - w_{a,n+1})D_n p_{a,D_n\leq x\leq D_N}) \\ + & \sum_{n=0}^{N}(w_{c,n}D_n p_{a,D_{n-1}\leq x\leq D_n} + (w_{a,n} - w_{c,n})UC_{a,D_{n-1}\leq x\leq D_n}), \end{array} \tag{19}$$

where the probabilities p and unit costs UC take the form as in equation (17), with the respective indices.

As in the previous case without the weights, we demonstrate that this weighted unit objective function, considered as a function of D_0, with fixed $D_i, \forall i > 0$, has a minimum for a certain D_0. Figure 2(g) shows this for the subjective time unit weights $w_{a,0} = 1.0, w_{a,1} = 1.5, w_{a,2} = 2.0$ and $w_{c,0} = 2.0, w_{c,1} = 3.0$ and $w_{c,2} = 4.0$. If the weights would all be 1 we get the same cost curves as in Figure 2(e).

The total cost has now increased since for every time unit a weight factor equal or bigger than 1.0 is used. We also see that the minimum cost is now shifted to about $D_0 = 12$. That the optimal supplement is bigger here stems from the fact that the higher annoyance $w_{c,i}$ of the time units after having missed the consecutive next action instances leads to a higher cost for supplements D_0 that are so low that they often cause such a miss. So the subjective appreciation of each type of time unit can have an effect on the optimum supplement value.

Note that the formula (18) for wUC_{D_{n-1},D_n} with subjective weights, simplifies to the formula for UC_{D_{n-1},D_n} for the case where all weights are equal to 1 case in (11). Also, the formulas (18) and (19) remain even valid if another delay probability distribution than the negative exponential is chosen. Of course the corresponding $p_{a,D_{n-1}\leq x\leq D_n}$ and $UC_{a,D_{n-1}\leq x\leq D_n}$ have to be recalculated then.

By simply taking the minimum of the objective function, we are now able to calculate the optimal supplement that should be inserted in the case where only one probabilistic delay plays a role. Sometimes however, two consecutive activities with their own delay distribution will occur. We treat this case in the next section.

5.2 Pair of Two Subsequent Actions

Sum of Delays

We consider two subsequent actions a and b. Their probabilistic delay variables are called x and y respectively. Again, we can measure the occurrences of delays for both processes and fit a normalized negative exponential through these measurements. This gives

$$\left\{ \begin{array}{lcl} p_a(x) & = & ae^{-ax} \\ p_b(y) & = & be^{-by}. \end{array} \right. \tag{20}$$

The fitting is done similarly to the one dimensional case and gives for actions a and b, for their respective expected delays, also called a and b:

$$\left\{ \begin{array}{lcl} a & = & 1/\overline{d_a} \\ b & = & 1/\overline{d_b}. \end{array} \right. \tag{21}$$

We suppose the variables x and y and their distributions are independent, which could be a simplification of reality.

In train planning, the most important times are the ones where a train leaves a station after a stop. This is the latest time a passenger can catch the train, are the times to be published and respecting them in real time is what matters most to passengers. This is the promise to the customer that has to be kept as good as possible. Traditionally arrival times are also published. This is practical for a passenger to be able to plan the activity after his train arrival. But consider that it is generally worse to miss your train at departure than being a few time units later at arrival.

We consider a to be a ride action. b can then be a subsequent dwell, transfer or sink action. Dwell means waiting in a station inside the train. Transfer means a transfer to another train in the same station. Sink means the passenger exits the station and the train system. Dwell and transfer actions have endings that can be delayed. For a sink, a delay is not useful to model since the time for sinking is not part of the train service model.

To mathematically formulate expected passenger time, we are first interested in the probability that the summed delay $x+y$ of both actions together is smaller or bigger than a certain supplement D, that is supposed to statistically, mostly absorb this summed delay. We also will calculate the associated time costs of introducing this supplement. Balancing probabilities and respective costs, we will obtain closed formulas for the total cost of introducing a supplement D on a subsequent pair of actions. Weighting these costs with the number of passengers on the second action of each pair, we arrive at a formula for the total cost of each *fork* structure. A fork is the union of a ride edge and its directly subsequent, unique dwell edge, unique sink edge and potentially multiple transfer edges to different other trains. In

Figure 1, a fork can be identified as a collection of a blue (ride) rectangle and its right direct neighbours: the yellow (dwell), orange (transfer) and red (exiting) rectangles.

Our graph, inherently present in the representation in Figure 1, is set up so that a dwell edge combines all passenger flows dwelling in one edge, irrespective of their origin and destination. Similarly a source edge combines all passengers in one edge, irrespective of their destination and a sink edge all passengers in one edge, irrespective of their origin.

In Time Cost

Contrary to the one dimensional case in the previous paragraph, we will immediately consider the most general case of weighted, subjective time units. If all time units are considered to have the same weight, the w values all equal 1.

The cost of suffering a delay x on action a (e.g.: ride) as well as a delay y on action b (e.g.: dwell) while we are still in time for the next ride start is

$$w_{a,0}x + w_{b,0}y, \tag{22}$$

where $w_{a,0}$ is the possibly subjective weight for an action a in these conditions. Similarly for $w_{b,0}$.

If we are still in time, meaning the supplement D_0 against delay, is not exceeded yet, some time until D_0 is left over. Passengers will wait after actions a and b have completed, during this time $D_0 - x - y$. This time also has to be accounted for and has an annoyance weight of $w_{c,0}$. So this cost is

$$w_{c,0}(D_0 - x - y). \tag{23}$$

Together, the total in time cost of experiencing a delay $x + y \leq D$ on these two actions is the sum of (22) and (23), so

$$w_{a,0}x + w_{b,0}y + w_{c,0}(D_0 - x - y). \tag{24}$$

Note that for $w_{a,0} = w_{b,0} = w_{c,0} = 1$, the total in time cost is D_0.

For every combination of delays x and y, we now have to see how often they occur. This is given by the combined probabilities

$$p_a(x)p_b(y). \tag{25}$$

Integrating over $x + y \leq D_0$ which forms a rectangular triangle gives us the in time cost

$$\begin{array}{ll} & wUC_{0\leq x\leq D_0} \\ = & \int_0^{D_0}\int_0^{D_0-x}\left(w_{a,0}x + w_{b,0}y + w_{c,0}(D_0 - x - y)\right)p_a(x)p_b(y)dydx \\ = & w_{c,0}D_0\int_0^{D_0}\int_0^{D_0-x}p_a(x)p_b(y)dydx \\ + & (w_{a,0} - w_{c,0})\int_0^{D_0-x}xp_a(x)p_b(y)dydx + (w_{b,0} - w_{c,0})\int_0^{D_0-x}yp_a(x)p_b(y)dydx \\ = & w_{c,0}D_0 p_{a\frown b,0\leq x+y\leq D_0} \\ + & (w_{a,0} - w_{c,0})UC_{a\frown b,x,0\leq x+y\leq D_0} + (w_{b,0} - w_{c,0})UC_{a\frown b,y,0\leq x+y\leq D_0}. \end{array} \tag{26}$$

For $a \neq b$ we get

$$\begin{aligned}
& p_{a\neq b,0\leq x+y\leq D_0} = \int_0^{D_0}\int_0^{D_0-x}(ae^{-ax}be^{-by})dydx = \tfrac{a(1-e^{-bD_0})-b(1-e^{-aD_0})}{a-b} \\
& UC_{a\neq b,x,0\leq x+y\leq D_0} = \int_0^{D_0}\int_0^{D_0-x} x(ae^{-ax}be^{-by})dydx \\
= \; & \tfrac{1-(aD_0+1)e^{-aD_0}}{a} - [e^{-aD_0}((b-a)D_0-1)+e^{-bD_0}]\tfrac{a}{(a-b)^2} \\
& UC_{a\neq b,y,0\leq x+y\leq D_0} = \int_0^{D_0}\int_0^{D_0-y} y(ae^{-ax}be^{-by})dxdy \\
= \; & \tfrac{1-(bD_0+1)e^{-bD_0}}{b} - [e^{-bD_0}((a-b)D_0-1)+e^{-aD_0}]\tfrac{b}{(a-b)^2}.
\end{aligned} \tag{27}$$

Care is taken that the last version of each expression in (27) only has negative exponents, so that numerical stability is guaranteed and overflow is avoided.

When $a = b$ we would get a division by zero for each of the expressions in (27), so we recalculate for this special case. This gives

$$\begin{aligned}
& p_{a=b,0\leq x+y\leq D_0} = \int_0^{D_0}\int_0^{D_0-x}(ae^{-ax}ae^{-ay})dydx = 1-(aD_0+1)e^{-aD_0} \\
& UC_{a=b,x,0\leq x+y\leq D_0} = \int_0^{D_0}\int_0^{D_0-x} x(ae^{-ax}ae^{-ay})dydx = \tfrac{e^{-aD_0}(-aD_0(aD_0+2)-2)+2}{2a} \\
& UC_{a=b,y,0\leq x+y\leq D_0} = \int_0^{D_0}\int_0^{D_0-y} y(ae^{-ax}ae^{-ay})dxdy = \tfrac{e^{-aD_0}(-aD_0(aD_0+2)-2)+2}{2a}.
\end{aligned} \tag{28}$$

Over Time Cost

The integral in time delay cost $wUC_{0\leq x\leq D_0}$ for delays below D_0 in (26) gives us a cost supposing $x+y\leq D_0$. Since $x+y > D_0$ can also hold, we calculate this corresponding *over time cost* next.

When $x+y = D_0$, the ride action after a and b takes place immediately and can still just be caught by the passengers. If $x+y > D_0$, they will miss this next action. For action b being a transfer, this means they will have to catch the next train in the same direction.

Suppose the next leaving train in the desired direction leaves at time distance $D_1 > D_0$. Then we get for the cost part for $x+y$ between D_0 and D_1, which forms a trapezium which fits snuggly against the triangle area, from (26) as

$$\begin{aligned}
& wUC_{D_0\leq x\leq D_1} \\
= \; & \int_0^{D_0}\int_{D_0-x}^{D_1-x}[(w_{a,0}+w_{b,0})/2\cdot D_0 + (w_{a,1}+w_{b,1})/2\cdot(x+y-D_0) \\
+ \; & w_{c,1}(D_1-x-y)]\,p_a(x)p_b(y)dydx \\
+ \; & \int_{D_0}^{D_1}\int_0^{D_1-x}[(w_{a,0}+w_{b,0})/2\cdot D_0 + (w_{a,1}+w_{b,1})/2\cdot(x+y-D_0) \\
+ \; & w_{c,1}(D_1-x-y)]\,p_a(x)p_b(y)dydx \\
= \; & ((w_{a,0}+w_{b,0})/2-(w_{a,1}+w_{b,1})/2)\cdot D_0 + w_{c,1}D_1) \\
& (\int_0^{D_0}\int_{D_0-x}^{D_1-x} p_a(x)p_b(y)dydx + \int_{D_0}^{D_1}\int_0^{D_1-x} p_a(x)p_b(y)dydx) \\
+ \; & ((w_{a,1}+w_{b,1})/2-w_{c,1}) \\
& (\int_0^{D_0}\int_{D_0-x}^{D_1-x} xp_a(x)p_b(y)dydx + \int_{D_0}^{D_1}\int_0^{D_1-x} xp_a(x)p_b(y)dydx) \\
+ \; & ((w_{a,1}+w_{b,1})/2-w_{c,1}) \\
& (\int_0^{D_0}\int_{D_0-y}^{D_1-y} yp_a(x)yp_b(y)dydx + \int_{D_0}^{D_1}\int_0^{D_1-y} yp_a(x)p_b(y)dxdy) \\
= \; & ((w_{a,0}+w_{b,0})/2-(w_{a,1}+w_{b,1})/2)\cdot D_0 + w_{c,1}D_1) \\
& (p_{a\backsim b,D_0\leq x+y\leq D_1,p1} + p_{a\backsim b,D_0\leq x+y\leq D_1,p_2}) \\
+ \; & ((w_{a,1}+w_{b,1})/2-w_{c,1})(UC_{a\backsim b,x,D_0\leq x+y\leq D_1,p_1} + UC_{a\backsim b,x,D_0\leq x+y\leq D_1,p_2}) \\
+ \; & ((w_{a,1}+w_{b,1})/2-w_{c,1})(UC_{a\backsim b,y,D_0\leq x+y\leq D_1,p_1} + UC_{a\backsim b,y,D_0\leq x+y\leq D_1,p_2}),
\end{aligned} \tag{29}$$

where $a \backsim b$ stands for the relation between a and b: equality or non-equality and p_1 stands for part 1, the parallellogram and p_2 for part 2, the triangle, that together make up the trapezium over which we integrate here. Note that the swapping of integration order of x and y is only done to make the temporary resulting formulas as simple as possible. For each line in (29), the sum of two double integrals over the full trapezium is the same when integrated in both integration orders.

The averaging of weights in $(w_{a,i} + w_{b,i})/2$ appears because for a time $x + y$ where $D_0 \leq x + y \leq D_1$, it is not known which part of the time was spent for activity a versus b.

The formulas for the unit probabilities and unit cost in this D_0 to D_1 case for $a \neq b$ are

$$
\begin{aligned}
& p_{a\neq b,D_0\leq x+y\leq D_1,p1} = \int_0^{D_0}\int_{D_0-x}^{D_1-x} ae^{-ax}be^{-by}dydx \\
= \quad & (e^{-bD_0} - e^{-aD_0} - e^{-bD_1} + e^{-aD_0-b(D_1-D_0)})\tfrac{a}{a-b} \\
& p_{a\neq b,D_0\leq x+y\leq D_1,p2} = \int_{D_0}^{D_1}\int_0^{D_1-x} ae^{-ax}be^{-by}dydx \\
= \quad & -e^{-b(D_1-D_0)-aD_0}\tfrac{a}{a-b} + e^{-aD_1}\tfrac{b}{a-b} + e^{-aD_0} \\
& UC_{a\neq b,x,D_0\leq x+y\leq D_1,p1} = \int_0^{D_0}\int_{D_0-x}^{D_1-x} xae^{-ax}be^{-by}dydx \\
= \quad & ((-aD_0 + bD_0 - 1)e^{-aD_0} + e^{-bD_0})(1 - e^{-b(D_1-D_0)})\tfrac{a}{(a-b)^2} \\
& UC_{a\neq b,x,D_0\leq x+y\leq D_1,p2} = \int_{D_0}^{D_1}\int_0^{D_1-x} xae^{-ax}be^{-by}dydx \\
= \quad & [(D_0(b-a) - 1)e^{-aD_0-b(D_1-D_0)} + (D_1(a-b) + 1)e^{-aD_1}]\tfrac{a}{(a-b)^2} \\
+ \quad & [(aD_0 + 1)e^{-aD_0} - (aD_1 + 1)e^{-aD_1}]/a \\
& UC_{a\neq b,y,D_0\leq x+y\leq D_1,p1} = \int_0^{D_0}\int_{D_0-y}^{D_1-y} yae^{-ax}be^{-by}dxdy \\
= \quad & [(-bD_0 + aD_0 - 1)e^{-bD_0} + e^{-aD_0})(1 - e^{-a(D_1-D_0)}]\tfrac{b}{(a-b)^2} \\
& UC_{a\neq b,y,D_0\leq x+y\leq D_1,p2} = \int_{D_0}^{D_1}\int_0^{D_1-y} yae^{-ax}be^{-by}dxdy \\
= \quad & [(D_0(a-b) - 1)e^{-bD_0-a(D_1-D_0)} + (D_1(b-a) + 1)e^{-bD_1}]\tfrac{b}{(a-b)^2} \\
+ \quad & [(bD_0 + 1)e^{-bD_0} - (bD_1 + 1)e^{-bD_1}]/b.
\end{aligned}
\tag{30}
$$

For the $a = b$ case the equivalent expressions are

$$
\begin{aligned}
& p_{a=b,D_0\leq x+y\leq D_1,p1} = \int_0^{D_0}\int_{D_0-x}^{D_1-x} ae^{-ax}be^{-by}dydx = aD_0(e^{-aD_0} - e^{-aD_1}) \\
& p_{a=b,D_0\leq x+y\leq D_1,p2} = \int_{D_0}^{D_1}\int_0^{D_1-x} ae^{-ax}be^{-by}dydx = e^{-aD_1}(a(D_0 - D_1) - 1) + e^{-aD_0} \\
& UC_{a=b,x,D_0\leq x+y\leq D_1,p1} = \int_0^{D_0}\int_{D_0-x}^{D_1-x} xae^{-ax}be^{-by}dydx = aD_0{}^2(e^{-aD_0} - e^{-aD_1})/2 \\
& UC_{a=b,x,D_0\leq x+y\leq D_1,p2} = \int_{D_0}^{D_1}\int_0^{D_1-x} xae^{-ax}be^{-by}dydx \\
= \quad & [e^{-aD_1}(a^2(D_0{}^2 - D_1{}^2) - 2aD_1 - 2) + 2(aD_0 + 1)e^{-aD_0}]/(2a) \\
& UC_{a=b,y,D_0\leq x+y\leq D_1,p1} = \int_0^{D_0}\int_{D_0-y}^{D_1-y} yae^{-ax}be^{-by}dxdy \\
= \quad & aD_0{}^2(e^{-aD_0} - e^{-aD_1})/2 \\
& UC_{a=b,y,D_0\leq x+y\leq D_1,p2} = \int_{D_0}^{D_1}\int_0^{D_1-y} yae^{-ax}be^{-by}dxdy \\
= \quad & [e^{-aD_1}(a^2(D_0{}^2 - D_1{}^2) - 2aD_1 - 2) + 2(aD_0 + 1)e^{-aD_0}]/(2a).
\end{aligned}
\tag{31}
$$

As in the previous paragraph, we could calculate a next cost term for $wUC_{D_1 \leq x \leq D_2}$ but from Figures 2(e) and 2(g) we know that this will be a low cost and we can safely ignore it. So the total local cost we consider is $wUC_{loc} = wUC_{0 \leq x+y \leq D_1} = wUC_{0 \leq x+y \leq D_0} + wUC_{D_0 \leq x+y \leq D_1}$ which can be calculated from (30) or (31).

Figure 2(d) shows a plot of the probability functions for the two subsequent action cases. It differs from Figure 2(c) in that now both actions have delay distributions. Figure 2(d) shows probability curves of which the second derivative changes its sign. This is due to the convolution of these two distributions.

Figure 2(f) represents the total costs and its component costs from 0 to D_0 and from D_0 to D_1 for objective time unit costs, all equal to 1. The minimum cost is obtained for about $D_0 = 11$ time units.

Figure 2(h) represents the same cost for the subjective time unit costs, $w_{a,0} = 1.0$, $w_{a,1} = 1.5$, $w_{b,0} = 1.0$, $w_{b,1} = 1.5$, $w_{c,0} = 2.0$ and $w_{c,1} = 3.0$. We see that the minimum in Figure 2(h) is not the same as in Figure 2(f). Now the minimum cost is obtained for $D_0 = 12$ time units. Similarly to the one dimensional case higher annoyance for time units spent after having missed a next action results in introduction of a bigger supplement D_0 to help avoid this.

5.3 Optimal Sum of a Range of Supplements for (Ride,Transfer)-Action Pair

We now have derived the objective function for the (ride, transfer) action pair. We show in Figure 3 that we are able to calculate the predicted optimum for the sum of two supplements associated with two consecutive actions a and b. We show the results obtained for different combinations of values of expected delays for action a $(0, 1, 4, 10, 20)$ and b $(0, 1, 2, 4)$. All units can be seen as minutes. D_1 is supposed to always be 60 here. The case $(0, 0)$, meaning no expected delay on either action, results in no supplements at all. One can also see that an increase in any of the two expected delays results in an increased optimal supplement sum, which is only logical.

Note that these optima are only valid for one action pair. A single supplement can occur in multiple action pairs and their respective local objective functions. Each can have a different optimum value. The global optimization will find a trade-off between these local optima. Section 6 describes how this is done.

5.4 Specialization to All Flow Types: Different Combinations of In Time and/or Over Time Costs

We have derived the objective function for a specific single action and for the (ride, transfer) action pair. These are the most general cases. The other three cases, as mentioned in Table 1 can be derived from it. Departing passengers only have an over-time cost and no in-time cost. Through and arriving passengers only have an in-time cost and no over-time cost. The resulting functions are given in Figures 4(a) to 4(d).

5.5 Linearization

To be able to use the derived objective function in an optimization for a train service network for an entire country, we want to use linear programming solvers. This means that our objective function has to be linearized.

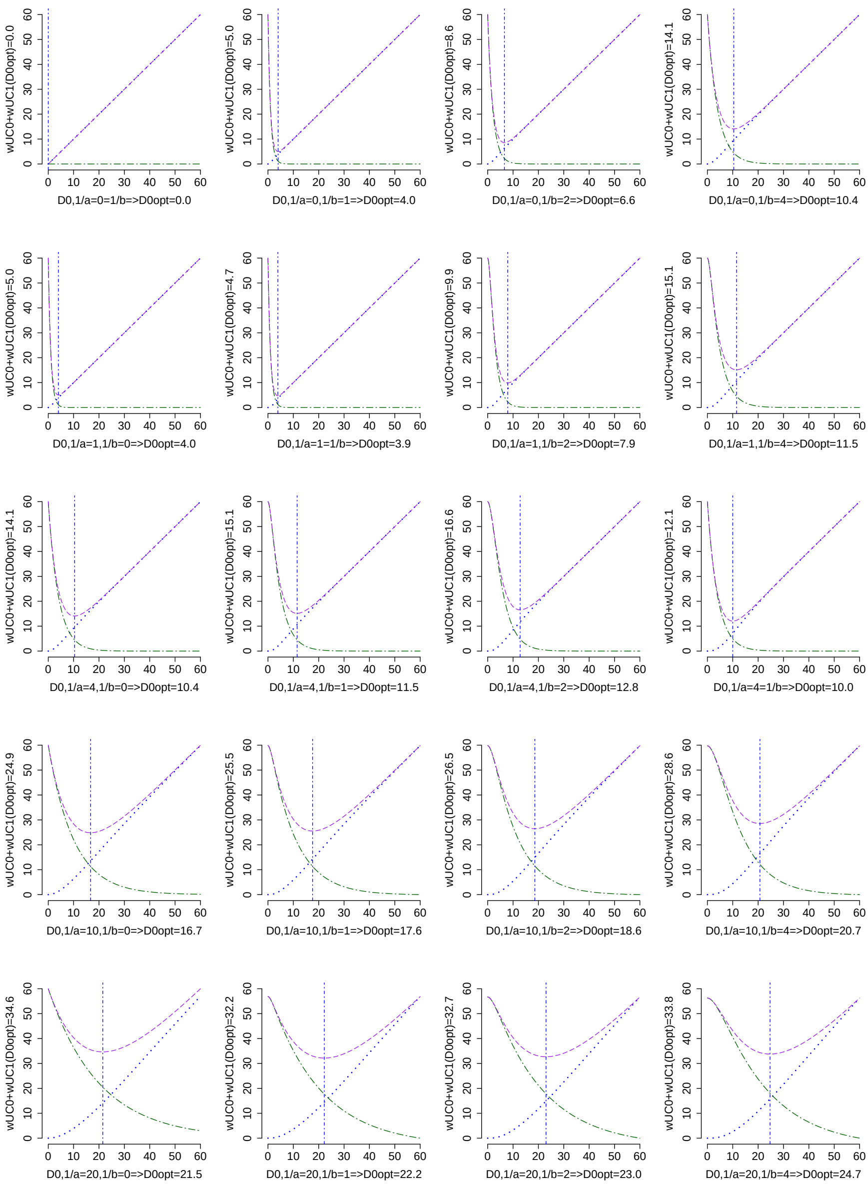
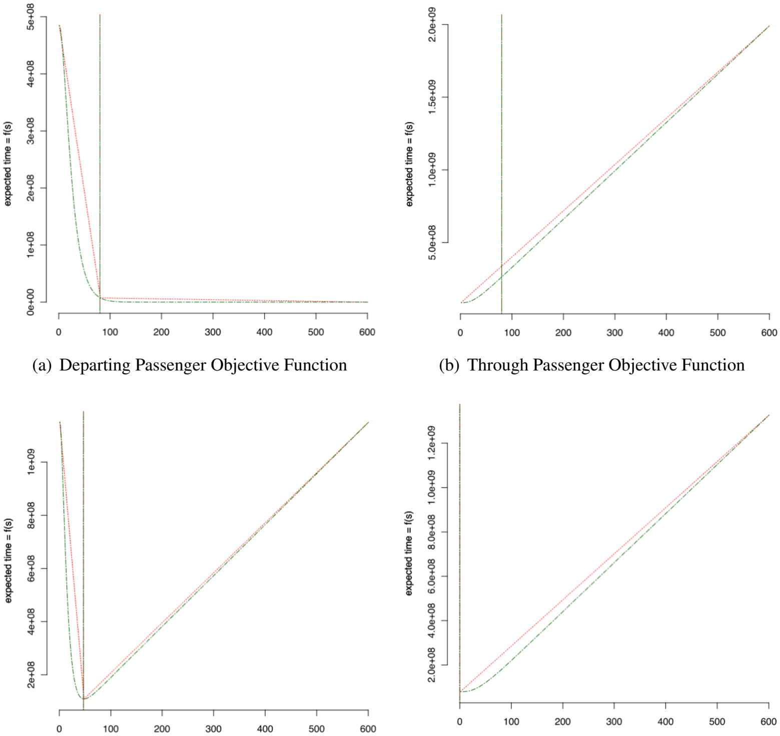


Figure 3: Supplement Sum with Minimal Cost for a Range of Action Pair Expected Delays



(a) Departing Passenger Objective Function

(b) Through Passenger Objective Function

(c) Transfer Passenger Objective Function

(d) Arriving Passenger Objective Function

Figure 4: Expected Passenger Time as Objective Functions of sum of supplements chosen for the four Passenger Flow Types.

There are two options here to linearize the curves of the calculated objective functions in Figures 2(e), 2(f), 2(g) and 2(h). One is to discretize the independent variable to some fixed resolution, like a minute or a tenth of a minute. The other is to approximate the curves with a number of line segments, which allows for the independent variables to remain continuous. When inspecting these objective functions, one can see that they are quite smooth. There are no sudden peaks or dips. This pleads for the second option. With only two line segments the minimum resulting from the intersection of the two line segments is already very close to the minimum of the original curves. For the 2-segment-approximation we use the least squares method. We also tried the first option with a fixed resolution of a tenth of a minute, but this resulted in many variables and an optimal schedule calculation times of a few hours for just 20 trains. This also convinced us that the segment approximation method should result in a faster yet accurate tool. The result of the linearizations are shown in the Figures 4(a) to 4(d) by the approximation of the green curves by the red segments.

A black vertical line indicates the supplement that was chosen. In the cases 4(c) and 4(d) this results in the real local minimum. In the cases 4(a) and 4(b) the results are close to the local minimum. This can happen when constraints or local objective functions in other locations of the schedule play a more important role. We now explain how the local objective functions are composed to form the global objective function.

6 Recomposing the Global Objective Function: Simply Summing

To evaluate the expected passenger time of a schedule, we need to look at all action pairs present in the planned schedule and see how these contribute to the total passenger time.

In the previous paragraphs, we convoluted distributions of pairs of subsequent actions, of which the first is always a ride action. In so doing, we formed forks of actions for which we obtained a (weighted) unit time objective function. By multiplication with the correct number of passengers present on these actions, we obtain the total cost of each action pair.

The weighted global objective function for the whole schedule is just the sum of all local weighted objective functions. This is the case because, (1) we consider all passenger minutes equal and (2) we don't convolute the delay distributions across actions of different (subsequent) forks. The second reason implies that we will actually determine a schedule that is more robust than the real minimal expected passenger time would require. Our schedule will have the advantage though that ride to next ride departure times are also more dependable.

So our *expected passenger time*, which is the objective function to be minimized for schedule optimization, becomes

$$\begin{aligned} wGC(D_0(E)) &= \textstyle\sum_{e\in E_{src}} f_e \cdot wUC_{predpredride(e),predwell(e)}(D_{0,e,succride(e)}) \\ &+ \textstyle\sum_{e\in E_d\cup E_{tr}\cup E_{snk}} f_e \cdot wUC_{predride(e),e}(D_{0,predride(e),e}), \end{aligned} \tag{32}$$

where wGC is the weighted global cost, E is the graph of all actions in the train service network, $D_0(E)$ is a shorthand notation for all D_0 that are present in the graph E, E_d is the set of all dwell edges, E_{tr} is the set of all transfer edges, E_{snk} is the set of all sink edges, $succride(e)$ is the ride successor edge associated with edge $e \in E_{src}$, which is also the successor edge of edge e. $preddwell(e)$ is the ride predecessor edge associated with edge $e \in E_{snk}$, which is also the predecessor dwell edge of the successor ride edge $succride(e)$ of e. $predpredride(e)$ is the ride predecessor edge associated with edge $preddwell(e)$ of $e \in E_{snk}$. $predride(e)$ is the ride edge associated with edge e, which is also always the predecessor edge of edge e. f_e is the local passenger flow over that action or edge e. Note that only flows over non ride edges are directly present. The ride flow is simply the sum of all flows on actions leaving from that ride action, and since flows for all these next actions are present, the ride flows are present implicitly in the sum. D_{0,e_a,e_b} is the D_0 parameter for that specific edge sequence pair (e_a, e_b) and wUC_{e_a,e_b} its weighted local objective function.

The last three columns in Table 2 structurally summarize what quantities are used in (32) for each of the four passenger flow types. The column *Flow of Action* indicates the action that determines the number of passengers on each action pair. Note that the number of ride passengers is not present directly in any of the four cases, but as Figure 1 shows, it equals the sum of dwell, transfer and sink passenger numbers subsequent to the ride action considered. If we consider all terms in this sum, we account for every passenger correctly.

Table 2: Four Passenger Flow Types, their Corresponding Action Sequences, the Action Edges Used and their Time Cost Types

Passenger Flow Type	Action a	Action b	Time Cost	Flow of Action	Edge a	Edge b
Departing	(Source)	Ride	in	Source	PredPredRide	PredDwell
Through	Ride	Dwell	over	Dwell	Ride	Dwell
Transfer	Ride	Transfer	in+over	Transfer	Ride	Transfer
Arriving	Ride	(Sink)	over	Sink	Ride	

The quality of a given schedule with given $D_{0,e}, \forall e$ can now be directly calculated from (32). For the sink edges, the two action formulas may be used with an expected delay of $\overline{d_b} = 0$ or very low. This essentially results in using the one action formula on the associated ride edges, which can of course be used directly as well.

7 Knock-On Delays

In a similar way as the derivations made previously, we also derived a *knock-on objective function* for each couple of trains using the same piece of track. This is a function of the heading time between these two trains. Since in our schedule optimization, the order of trains on a track unit is unknown in advance, a knock-on cost between each of the $N*(N-1)$ couples of the N trains is added. When train a is scheduled very close before train b two factors increase the total knocked-on passenger delay on b. One is a high probability of delay on train a. The second is a high number of passengers on train b, since they are the passengers experiencing the knock-on delay. These factors are present in our knock-on objective function. We have used these knock-on costs in train schedule optimizations of all periodic passenger trains in Belgium of which many pairs have train tracks in common and even though there are many of them, they don't significantly increase the computation time.

8 Results

We have constructed a model with all necessary constraints and with the objective function derived here, for an increasing number of trains. The starting timetable is the one for all trains departing between 7am and 8am on 13 March 2013. We used Gurobi 5.1.0 as the MILP solver and ran it, set to 8 solver threads, on an Apple iMac with a 3.4 GHz Intel i7 processor and with 16GB 1333 MHz DDR3 Memory. The gap desired was set to what was obtained as the gap of the first returned solution in earlier trials. Our results are given in table 3.

The numbers of model variables (#columns) and constraints (#rows) both grow rapidly as trains are added, but the pre-solver is usually able to reduce both numbers by about 50%. For the last case, IC IR L P, we have rescheduled all 186 periodic (IC IR L) trains, as well as the 17 non-periodic peak (P) trains departing between 7am and 8am. For each ride and dwell action we assumed an expected primary delay of 3% on top of its minimum time. All periodic trains are repeated every hour. The P trains occur just once in practice, but due to our cyclic timetable, time slots are automatically reserved for them in every non-peak hour too. Outside the peak period, these empty P slots can possibly be used for freight trains if desirable.

Table 3: Scalability of our Integer Linear Programming Model with necessary Constraints and the Derived Objective Function

train types	trains (#)	model rows (#)	model columns (#)	solver time (s)	MILP gap (%)	passenger time reduction (%)	missed transfer probability (%)
IC	43	18747	13361	50	24.30	10.73	2.95
IC IR	82	48267	33035	449	32.35	12.38	3.11
IC IR L	186	102652	68504	2426	32.80	10.03	2.31
IC IR L P	203	225132	158860	3706	38.27	7.12	2.43

Compared to the IC IR L P-timetable currently in operation, our optimized timetable has some advantages compared to the current one. First, it respects all minimum ride- and dwell-times without exception. Second, it respects all headway time buffers of 3 minutes between all train pairs on the same track section. Third, our calculations show that the average chance of missing a transfer in the current timetable is 34.68% while in our optimized timetable it is only 2.43%. The expected passenger time of our optimized schedule is 7.12% lower than the original schedule. This result is obtained with supplements on ride and dwell actions up to 15 minutes are allowed. Fourth, generating our schedule only takes 3706 seconds, while it takes many human planners many months to generate the current timetable.

9 Conclusions

This paper has four main contributions. Firstly, by making a graphical representation as in Figure 1 it becomes clear that we can decompose a general train network into actions of five different types: ride and dwell train actions and the source, transfer and sink passenger actions and then recombine pairs of subsequent actions into four local passenger flow types: departing, through, transfer and arriving.

Thanks to this, secondly, supposing a general negative exponential delay distribution for each of these actions, we analytically derived the probability and objective functions for each of these local passenger flows, according to what matters for them at that location. For transfers, this includes the probabilistic trade-off for catching or missing a next action. These local functions of one or two supplement variables, indicate the cost tradeoff and can be used to evaluate a given schedule on quality or even optimality of these supplements.

Thirdly, summing these local functions for the whole schedule delivers a global objective function that can be used by an optimizing solver directly. We also show how, for linear solvers, optionally, convex linear approximate functions can be set up.

Lastly, together with the necessary train schedule constraints, we show in Figure 1 that our method produces a schedule for two trains that, at face value, has apparent merits: efficiency as well as robustness. We also used the derived objective function in producing a passenger time optimal timetable for all 203 passenger trains in Belgium. This timetable is quickly generated and is conflictless, efficient and robust by construction.

References

[1] Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., "Defining Robustness of a Railway Timetable", *Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011*,Rome, Italy, 2011.

[2] Goverde, R.M.P., "Optimal Scheduling of Connections in Railway Systems", *Paper presented at the 8th WTCR*, Antwerp, Belgium, 1998.

[3] Goverde, R.M.P., "Synchronization Control of Scheduled Train Services to minimize Passenger Waiting Times", *Proceedings of 4th TRAIL Annual Congress*, part 2, TRAIL Research School, Delft, 1998.

[4] Goverde, R.M.P., "Improving Punctuality and Transfer Reliability by Railway Timetable Optimization", *Proceedings of the 5th TRAIL Annual Congress*, TRAIL Research School, Delft, 1999.

[5] Kroon, L., Dekker, R., Vromans, M.J.C.M., "Cyclic Railway Timetabling: A Stochastic Optimization Approach", Geraets, F., Kroon, L., Schöbel, A., Wagner, D., Zaroliagis, C.D. (eds), *Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science*, pp. 41-66, 2007.

[6] Liebchen, C., "Periodic Timetable Optimization in Public Transport", *Operations Research Proceedings*, 2007, Volume 2006, Part II, 29-36.

[7] Meng, Y., "Bemessung von Pufferzeiten in Anschlüssen von Reisezügen", W. Schwanhäußer, P. Wolf (Hrsg.) *Veröffentlichungen des Verkehrswissenschaflichen Institutes de Rheinischen-Wesfälischen Technischen Hochschule Aachen*, Heft. 46, Aachen, Deutschland. 1991.

[8] Nachtigall, K., "Periodic network optimization with different arc frequencies", *Discrete Applied Mathematics*, Vol. 69, pp. 1-17, 1996.

[9] Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., "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*, Rome, Italy, 2011.

[10] Schrijver, A., Steenbeek, A., "Spoorwegdienstregelingontwikkeling (Timetable Construction)" *Technical Report, CWI Center for Mathematics and Computer Science, Amsterdam*, (in Dutch), 1993.

[11] Serafini, P., Ukovich, W., "A Mathematical Model for Periodic Scheduling Problems", *SIAM Journal on Discrete Mathematics*, Vol. 2, pp. 550-581, 1989.

[12] Vansteenwegen, P., Van Oudheusden, D., "Developing railway timetables which guarantee a better service", *European Journal of Operational Research"*, Vol. 173, pp. 337-350, 2006.

[13] Vansteenwegen, P., Van Oudheusden, D., "Decreasing the passenger waiting time for an intercity rail network", *Transportation Research Part B: Methodological*, Vol. 41, pp. 478-492, 2007.

2.3 A Passenger Knock-On Delay Model for Timetable Optimisation

Paper presented at International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) Dresden, Germany, 2-4 December 2013

The previous paper in section 2.2 derived all terms of the objective function: total expected passenger time in practice, but has not yet described the complete derivation of the expected passenger time due to knock-on delays. In this section, these terms are derived and now added to the objective function. Essentially, the knock-on delays are the delays that a second train experiences due to the train in front of it being delayed. Only the number of passengers of the second train experience this delay. So the total knock-on delay is the multiplication of the number of passengers on the second train with the integrated products for all cases of the probability that the first train will be so much delayed that the second train has to start braking as well and the delay caused on the second train. The probability that this occurs depends on the primary delay distribution of both the first and the second train. Because a first, often delayed train can delay a second, rarely delayed train with few passengers, which can in turn delay a third train with many passengers, we also consider the direct effect of the first on the third train. So for N trains planned on the same section, we then have to consider the full clique of $N \cdot (N-1)$ train pairs. Since the schedule is cyclic, when a train A can delay a train B, train B can also delay the next occurrence of train A in the next hour. This consideration of knock-on delays for the full clique of trains is also necessary since the order of trains is not yet known. Indeed, we are still in the process of timetabling which determines these orders. The following paper derives the expressions for the knock-on terms that can then be added to the objective function of our timetabling model.

We construct the train network for all 203 hourly passenger trains in Belgium, departing between 7 and 8 am in the timetable of June 12th 2013. On this train graph, we carry out an optimisation with these knock-on delays included in the objective and, with certain assumptions about the primary delays, we obtain a timetable with significantly reduced knock-on delays compared to the original timetable. Indeed, in the original timetable, the knock-on delay component amounts to 4.04% of the total expected passenger journey time, while in the optimised timetable it only amounts to 2.60% of the total expected passenger journey time. Note that the total is lower in the optimised than in the original timetable, so the reduction is even stronger in absolute terms. Indeed, it amounts to more than 40% reduction of total time. These results show that our model is able to reorder trains and choose good train inter-times, so that the total expected knock-on delays can be significantly reduced.

Our timetable computation times do not significantly increase due to the addition of the knock-on model. We also notice that increasing the assumed level of primary delays does not increase the computation time for these timetables. Moreover, larger assumed primary delays typically lead to a first feasible timetable with a lower MIP gap.

A Passenger Knock-On Delay Model for Timetable Optimisation

Peter Sels[1 2 3], Thijs Dewilde[1], Dirk Cattrysse[1], Pieter Vansteenwegen[1]
[1]KU Leuven, University of Leuven
[2]Logically Yours BVBA, Antwerp
[3]Infrabel, Department of Rail Access, Brussels

Abstract

In the process of timetable creation, sufficient time should be scheduled between any pair of trains using a common infrastructure section in order to avoid that a delay on the first train will cause a delay on the second train too. However, when this time buffer becomes very high, the positive incremental buffering effect diminishes and other negative effects may appear, like reduced timetable efficiency or lower than optimal remaining time between the other trains on the same infrastructure resource. This means there is a trade-off to make. We make this trade-off by analytically deriving the knock-on delays as passengers experience it in practice and by including these delays in our goal function: total expected passenger journey time in practice.

We use this goal function in our Mixed Integer Linear Programming (MILP) model to optimise from scratch, the timetable of all 203 hourly passenger trains in Belgium. We then also compare our resulting timetable with the original schedule, and conclude that both the knock-on component as well as the total expected passenger time are reduced.

Keywords: Knock-On Delay Model, Expected Passenger Time, Integer Linear Programming, Goal Function

1 Introduction

A railway timetable can be aptly represented by a graph. Graph vertices are train arrival and departure times. The graph's edges are either primary edges representing intra-train actions: ride and dwell, or secondary edges, representing inter-train actions: transfer or turn-around. Other secondary edges represent a required time difference: headway requirements [Kro09]. For all edges, primary or secondary, a minimum time is required and we also add a non-negative supplement. Note that we use the term *supplement* also in the meaning of *buffer* between two trains on a common infrastructure resource. The purpose of the supplements can be twofold. Firstly, they are sometimes needed as slack between two already planned timetable times. Indeed, imagining that one would plan the primary edges first, some slack would result for the inter-train transfer, turn-around and headway edges. Secondly, a larger than slack-only supplement could be needed to make a timetable robust against delay. However, supplements may also not become too large, resulting in trains riding too slow or idling too much and as such resulting in an inefficient planning. So, obviously there is a trade-off, per supplement, between robustness and efficiency. Additionally, when edges are part of a common graph cycle, the sum of minimum process times and supplements over all edges of the cycle have to sum up to a multiple of the timetable period [Gov10]. This means choices of supplements of these edges are related and one also has to be able to properly weigh the costs and benefits of the supplement choices on different edges. We consider one train more important than another when it has more passengers present on it. We could introduce artificial train class priorities, but prefer to directly weigh importance with passenger numbers instead. In [Sel11] we derived passenger numbers on all trains at all locations, starting from ticket sales data. With this information, we can formulate the *total expected passenger time in practice* [Dew11; Sel13a] as a function of the timetable. More specifically, it is a function of 3 parameters sets: (1) the action minima, (2) the assumed primary delays and (3) the planned supplements. Secondary delays also increase this expected passenger time, but are itself a function of the three mentioned parameter sets. The resulting total function is to be minimised to generate an optimal timetable for passengers. The minima are fixed, so in each timetable it will generate the same amount of expected time. The supplements are the decision variables of the timetable, so given the delay assumptions, their values determine any quality criterium of the timetable as expected passenger time, robustness and efficiency.

The total expected passenger time has been analytically derived as a function of minima and supplements in [Sel13a] for departing, through, transfer and arriving passengers. In this paper we add the derivation of the knock-on delay as a function of the minimum and supplement present on a headway edge. Indeed both a headway minimum time as well as a knock-on delay should be modelled whenever two trains on a common resource occur. So a *hard* headway constraint and a *soft* knock-on cost as a term in the goal function are always modelled on the same edge.

Section 2 lays out an analytical derivation of the knock-on delay function. Section 3 presents the results obtained when using these knock-on delay functions as terms in the goal function for a system of all 203 trains currently departing between 7 and 8 am in the cyclic Belgian timetable. Section 4 draws conclusions and hints at some further work.

2 Knock-On Delay Derivation

When train i is riding or dwelling on a track and it gets delayed, it can delay train j which follows it on the same track. We will derive a cost function that gives us the expected delay for all passengers on the second train as a function of the planned time in between the two trains and the expected delays on these trains.

We define the number of passengers on train i as f_i and on train j as f_j. As [Sel13b] explains, for trains riding in the same direction on a common track, headway edges exist between both the vertices representing the beginning of the trains' ride actions in both directions, cyclically and also between the endings of the trains' ride actions again in both directions, cyclically. For trains riding in opposite directions on a common track, a headway edge exists between the end of the first train's ride action and the beginning of the other train's ride action and vice versa, cyclically. In the sequel, when we mention a knock-on edge between train i and j, we more specifically mean the knock-on edge between two vertices v_i and v_j, where these vertices can be a begin or end vertex of a ride edge.

We can suppose the vertices v_i and v_j, which represent event times, to experience primary delays according to (commonly used [Han08]) negative exponential distributions

$$p_i(x) = a_i e^{-a_i x}, p_j(y) = a_j e^{-a_j y}, \tag{1}$$

where x and y are the primary delays of time points v_i and v_j and $p_i(x)$ and $p_j(y)$ their respective probabilities. The *expected* delays of these distributions are calculated to be

$$\overline{c_i} = \int_0^\infty x a_i e^{-a_i x} dx = \frac{1}{a_i}, \overline{c_j} = \int_0^\infty y a_j e^{-a_j y} dy = \frac{1}{a_j}. \tag{2}$$

Say that, on top of the mandatory heading time h between trains i and j, which has to be respected at any time, there is a *planned* supplement time $s_{i,j}$ and similarly a *planned* supplement $s_{j,i}$ between trains j and i. Then, the probability that due to combined delays of trains i and j, one train will delay the other is calculated by adding all cases where the delay difference of both trains exceeds the supplement between them, weighting these cases with the probability that they occur. This is done by integrating over a triangle area where the delay difference $x - y \geq s_{i,j}$ so $x \geq y + s_{i,j}$ and over another where $y \geq x + s_{j,i}$ as in

$$\begin{aligned} p_{x \geq y+s_{i,j}}(a_i, a_j, s_{i,j}) &= \int_0^\infty \int_{y+s_{i,j}}^\infty a_i e^{-a_i x} \cdot a_j e^{-a_j y} dx dy = \tfrac{a_j e^{-a_i s_{i,j}}}{a_i + a_j}, \\ p_{y \geq x+s_{i,j}}(a_i, a_j, s_{j,i}) &= \int_0^\infty \int_{x+s_{j,i}}^\infty a_i e^{-a_i x} \cdot a_j e^{-a_j y} dy dx = \tfrac{a_i e^{-a_j s_{j,i}}}{a_i + a_j}. \end{aligned} \tag{3}$$

In the area where $x < y + s_{i,j}$ and $y < x + s_{j,i}$, $s_{i,j}$ respectively $s_{j,i}$ are large enough to absorb primary delays and avoid knock-on delays. The total *expected* knock-on delay of train i on train j is calculated by multiplying, for each case where a knock-on delay occurs, its probability, with the knock-on delay amount occurring and then integrating these products over the same triangular integration areas as before. Via partial integration, one can prove

$$\begin{aligned} tKO_{i,j}(a_i, a_j, s_{i,j}) &= \int_0^\infty \int_{y+s_{i,j}}^\infty a_i e^{-a_i x} \cdot a_j e^{-a_j y} (x - y - s_{i,j}) dx dy = \tfrac{a_j e^{-a_i s_{i,j}}}{a_i(a_i + a_j)}, \\ tKO_{j,i}(a_i, a_j, s_{j,i}) &= \int_0^\infty \int_{x+s_{j,i}}^\infty a_i e^{-a_i x} \cdot a_j e^{-a_j y} (y - x - s_{j,i}) dy dx = \tfrac{a_i e^{-a_j s_{j,i}}}{a_j(a_i + a_j)}. \end{aligned} \tag{4}$$

From equations (4), two properties can be derived. First, the larger the planned separation time $s_{i,j}$ between the trains, the lower $tKO_{i,j}$, so the lower the expected knock-on delay on train j. Second, the lower the expected primary delay $\overline{c_i} = 1/a_i$ on train i, the higher a_i, the lower $tKO_{i,j}$, so the lower the expected knock-on delay on train j. These tendencies are indeed what we expect in practice as well. Since we are interested in the knock-on delays as *passengers* experience them in practice, we multiply the train knock-on delay with the number of passengers on the knocked-on train and get

$$\begin{aligned} pKO_{i,j}(a_i, a_j, s_{i,j}) &= f_j \cdot tKO_{i,j} = f_j \cdot \frac{a_j e^{-a_i s_{i,j}}}{a_i(a_i+a_j)}, \\ pKO_{j,i}(a_i, a_j, s_{j,i}) &= f_i \cdot tKO_{j,i} = f_i \cdot \frac{a_i e^{-a_j s_{j,i}}}{a_j(a_i+a_j)}. \end{aligned} \tag{5}$$

If only two trains i and j are to be planned on a common resource, in a one hour period, what are the ideal supplement times $s_{i,j}, s_{j,i}$ to be planned in between them? This will depend on their passenger numbers f_i, f_j and their expected delays a_i and a_j. First, note that there is a relation to respect between $s_{i,j}$ and $s_{j,i}$. Indeed, the constraint for the cycle formed by the two headway edges between trains i and j is

$$h + s_{i,j} + h + s_{j,i} = T \text{ or equivalently } s_{j,i} = T - 2h - s_{i,j}. \tag{6}$$

After substitution of $T - 2h - s_{i,j}$ for $s_{j,i}$ in $pKO_{j,i}$, $pKO_{j,i}$ is clearly a function of $s_{i,j}$. Since $pKO_{i,j}$ and $pKO_{j,i}$ are both convex functions of $s_{i,j}$, their sum is a convex function of $s_{i,j}$ as well. This means the optimal spreading of two trains per time period T can be calculated by minimising the total expected delay on all passengers of both trains as

$$\begin{aligned} 0 &= \frac{d}{ds_{i,j}} \left(pKO_{i,j} + pKO_{j,i} \right) \\ \Leftrightarrow 0 &= \frac{d}{ds_{i,j}} \left(f_j \cdot \frac{a_j e^{-a_i s_{i,j}}}{a_i(a_i+a_j)} + f_i \cdot \frac{a_i e^{-a_j(T-2h-s_{i,j})}}{a_j(a_i+a_j)} \right) \\ \Leftrightarrow 0 &= -f_j \cdot \frac{a_j e^{-a_i s_{i,j}}}{a_i+a_j} + f_i \cdot \frac{a_i e^{-a_j(T-2h-s_{i,j})}}{a_i+a_j} \\ \Leftrightarrow f_j \cdot a_j e^{-a_i s_{i,j}} &= f_i \cdot a_i e^{-a_j(T-2h-s_{i,j})} \\ \Leftrightarrow ln\left(\frac{f_j \cdot a_j}{f_i \cdot a_i}\right) &= -a_j(T-2h-s_{i,j}) + a_i(s_{i,j}) \\ \Leftrightarrow s_{i,j} &= \frac{a_j(T-2h) + ln\left(\frac{f_j a_j}{f_i a_i}\right)}{a_i+a_j} \end{aligned} \tag{7}$$

It follows from symmetry that

$$s_{j,i} = \frac{a_i(T-2h) + ln\left(\frac{f_i a_i}{f_j a_j}\right)}{a_i + a_j}. \tag{8}$$

The right hand sides of equations (7) and (8) sum up to $T - 2h$ as equation (6) requires.

As an example, for $T = 60$ minutes and $h = 3$ minutes, a train i with an expected delay of $1/a_i = 3$ minutes and $f_i = 100$ passengers on it and a train j with an expected delay of $1/a_j = 1$ minute and $f_j = 300$ passengers, would be spread according to equations (7) and (8) as $s_{i,j} = \frac{1(60-2\cdot3)+ln(300\cdot1/(100\cdot1/3))}{1/3+1} = 42.15$ minutes and $s_{j,i} = \frac{1/3(60-2\cdot3)+ln(100\cdot1/3/(300\cdot1))}{1/3+1} = 11.85$ minutes and indeed as equation (6) requires $42.15 + 3 + 11.85 + 3 = 60$ minutes.

This kind of balancing of supplements between trains on the same resource will be done by our solver when we add the costs in equation (5) to the goal function. (Note that also choices of supplements on graph edges in common cycles can affect the choice of $s_{i,j}$ and

$s_{j,i}$ and vice versa.) We take the approach of generating all knock-on costs between *all* train pairs using the same infrastructure resource, irrespective of their order. This has two reasons. First, unlike the method where we add only knock-on costs between directly subsequent trains, this method works without relying on the yet unknown order of trains. Second, suppose trains i, j and k follow each other in this order on a resource and train i has a large expected primary delay $1/a_i$, train j has a small $1/a_j$ but has very few people f_j on it while train k has a lot of people f_k on it. Then $pKO_{i,j}$ and $pKO_{j,k}$ can be small for low $s_{i,j}$ and low $s_{j,k}$, allowing the three trains, ordered as i, j, k, to be scheduled close together in time, even though $pKO_{i,k}$ will then be large. The fact that cases where $pKO_{i,k} \gg pKO_{i,j} + pKO_{j,k}$ can occur, shows that $pKO_{i,k}$ has to be added to capture all potential knock-on costs.

For N trains using the same resource during every timetable period T cyclically, this method generates $N \cdot (N-1)$ knock-on terms in the goal function. For each resource R, we define the index set I_R as the set of indices of trains that use R. Then, according to equation (5), the total knock-on cost pKO_R for all trains which use resource R is

$$\forall R : pKO_R = \sum_{\substack{i,j \in I_R \\ i \neq j}} f_j \cdot \frac{a_j e^{-a_i s_{i,j}}}{a_i(a_i + a_j)}. \tag{9}$$

For evaluation of the knock-on cost of a given schedule or for non-linear optimisation, equation (9) can be directly used. For a linear solver though, we need to linearise it first. Since each of the terms in (9) is convex in the variable $s_{i,j}$, we can use a standard linearisation trick for convex cost functions. This entails two steps. First, for each of the terms, we define a helper variable $pKO_{R,i,j}$ and impose on them

$$\forall R : \forall_{\substack{i,j \in I_R \\ i \neq j}} : pKO_{R,i,j} \geq f_j \cdot \frac{a_j e^{-a_i s_{i,j}}}{a_i(a_i + a_j)}. \tag{10}$$

All helper variables $KO_{R,i,j}$ are added to the global goal function of expected passenger time. Units match. Since we *minimise* our global goal function, all $KO_{R,i,j}$ are pushed down so that they will be equal to instead of greater than the right hand side of equation (10). Second, the right hand side of (10) is replaced by a number of line segments approximating it. Here, we use 2 segments. So for each $KO_{R,i,j}$ term, we define three points

$$\forall R : \forall_{\substack{i,j \in I_R \\ i \neq j}} : \begin{cases} (s_{i,j,0}, ko_{i,j,0}) = (0, f_j \cdot \frac{a_j}{a_i(a_i+a_j)}) \\ (s_{i,j,1}, ko_{i,j,1}) = (T/15, f_j \cdot \frac{a_j e^{-a_i T/15}}{a_i(a_i+a_j)}) \\ (s_{i,j,2}, ko_{i,j,2}) = (T, f_j \cdot \frac{a_j e^{-a_i T}}{a_i(a_i+a_j)}). \end{cases} \tag{11}$$

The low and high end of the segments are 0 and T so that the whole supplement range is covered. We use $T/15$, or 4 minutes for T equal to one hour, as the abcis of the middle point, because, in our tests, this resulted in the closest approximation to the curve $KO_{R,i,j}$ for most practical cases. Then, with these known values, equation (10) is linearised to

$$\forall R : \forall_{\substack{i,j \in I_R \\ i \neq j}} : \begin{cases} pKO_{R,i,j} & \geq & ko_{i,j,0} + \frac{ko_{i,j,1} - ko_{i,j,0}}{s_{i,j,1} - s_{i,j,0}} \cdot (s_{i,j} - s_{i,j,0}) \\ pKO_{R,i,j} & \geq & ko_{i,j,1} + \frac{ko_{i,j,2} - ko_{i,j,1}}{s_{i,j,2} - s_{i,j,1}} \cdot (s_{i,j} - s_{i,j,1}) \end{cases} \tag{12}$$

We add all $pKO_{R,i,j}$ as variables to our goal function and add the inequalities (12) with the values calculated as in (11) as hard constraints to our MILP model. As such, we have extended our model with a method that accounts for knock-on delays in a way that is properly balanced with the other goal function terms. Note that the obtained estimation of passenger knock-on delay cost can also be used in other than timetable optimisation models. A linear optimisation model maximising capacity consumption with the goal of capacity estimation, as for example [Mus13], could forbid or penalise scenarios with too much knock-on delay.

3 Optimisation Results

For all 203 hourly passenger trains in Belgium, departing between 7 and 8 am in the timetable of June 12th 2013, visiting 1770 open line track sections and calling at all 550 stations, the macroscopic model of constraints as described in [Sel13b] has been set up. (Overtaking is only allowed in stations with 4 or more platform tracks.) The goal function - expected passenger time in practice - as described in [Sel13a] and now extended with the cost terms for knock-on delays, as derived here in section 2, has been constructed. For each ride and dwell action we assumed varying primary delay distributions with an average of $a\%$ of each action's minimum time. a is given in column 1 of table 1. We compare properties of the original and optimised timetable in the next sections.

3.1 Feasibility: A Solution is Always Returned

Since our model has a goal function that properly penalises the choice of big supplements in a soft yet passenger optimal way, there is no reason for us to add a hard constraint that restricts supplements to any arbitrary value lower than $T - \delta$, where δ is the time resolution of the timetabling model. Other research groups (e.g. Delft [Spa13], e.g. Rotterdam [Kro09]) lack a goal function that automatically restricts *all* supplements and so have to enforce lower more arbitrary upper bounds as a hard constraint on their supplements. As a result they sometimes struggle with infeasibility of their model. We believe we have resolved this issue.

3.2 Quality: The Solution has Lower Expected Passenger Time in Practice

We assume for each action, a primary delay distribution with an average of 2% of the action minimum time. This value of a is Infrabels current best estimate for morning peak hours. Similarly, [Gov07] also uses percentages between 0 to 5%.

Consider figure 1 and its caption. The left half of the figure shows the planned train time total minima and total supplements, both for the *oRiginal* timetable (R) and for the *oPtimised* timetable (P). The right half represents passenger weighted planned time for all

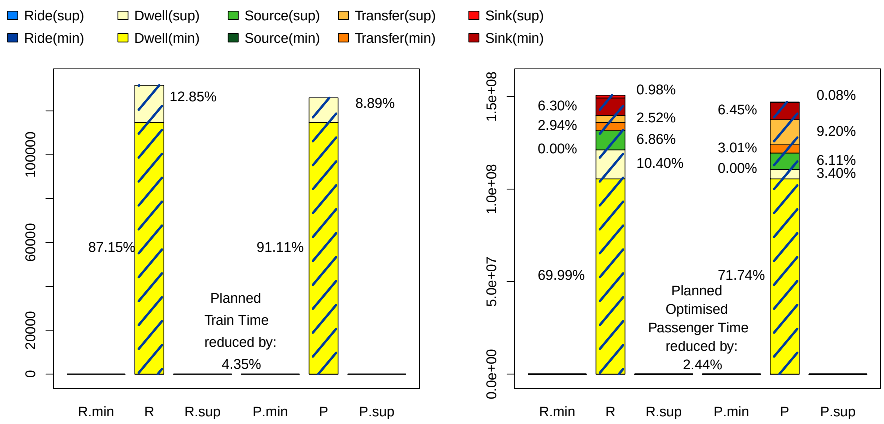


Figure 1: The *planned* time domain. The left half shows total planned train time for all trains. The right half show total passenger time for all passengers. In each box, the left bargraph shows a quantity for the *oRiginal* timetable while the right half shows the same quantity for the *oPtimised* timetable. *min* = sum of all minima, *sup* is sum of all supplements. The sum is not weighted for train time and passenger weighted for passenger time. *Source* corresponds to boarding passengers and *sink* to alighting passengers. In this *planned* domain, the shading with blue lines indicates that these actions were *summed* with ride actions.

origin-destination passenger streams with at least 50 people, again both for original and optimised timetable. There are dark and light versions of some colours (e.g.: yellow, orange). The dark colour indicates the sum of minimum times, while the lighter version indicates the sum of supplement times. The left half of figure 1 shows a decrease of total planned train supplements from 12.85% down to 8.89%. This train time supplement reduction is advantageous for the operator, since, if total train trip time now becomes less than the next lower multiple of hours, the same hourly service can be operated with one less train. [Lie07] also gave an example of this, optimising the Berlin Underground timetable.

The right hand side of figure 1 shows that the planned *passenger* weighted time reduction is a much more pronounced one, from 10.40% down to 3.40% of the same ride+dwell supplements. This is the case because they are now weighted by number of passengers.

In figure 2, instead of planned time, we show expected time, which includes primary delays and their consequences like secondary delays and missed transfers. The left half again represents train time. The right half shows passenger weighted time. The top row is the linear approximation of time as used in the optimisation model. The bottom row shows the actual non-linear time as it is evaluated post-optimisation. The same advantageous stronger supplement reduction in column 2 compared to column 1 is also present in this figure. This is the case for ride+dwell supplements but also for knock-on time. The knock-on compo-

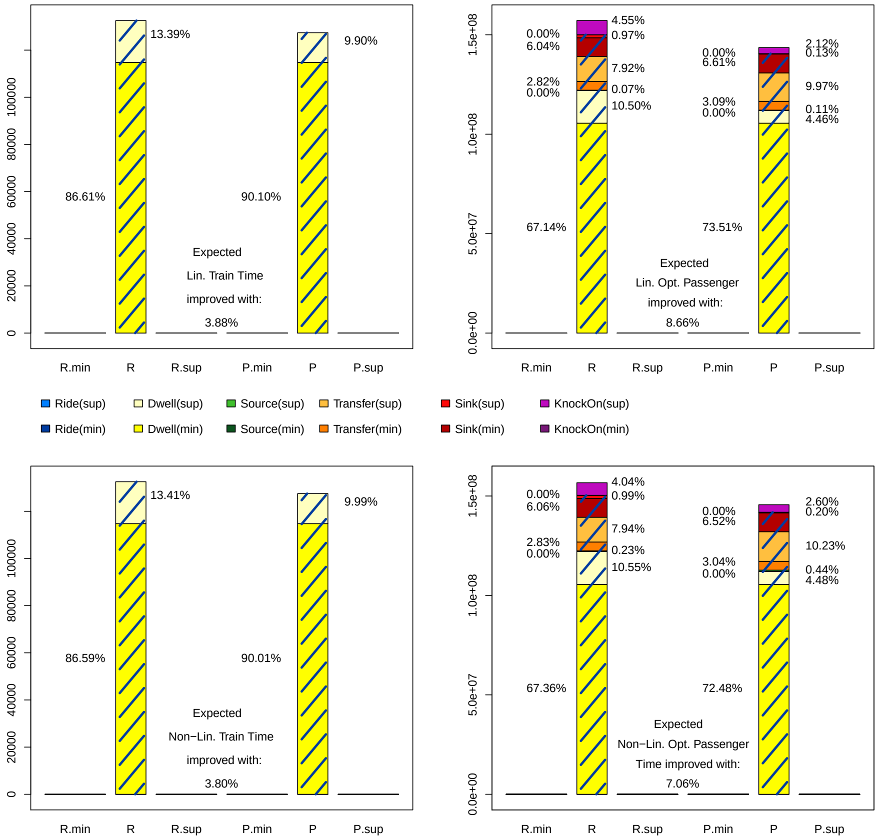


Figure 2: The *expected* time domain. Left and right are train and passenger time as in figure 1. The bottom row shows the non-linear time as used during evaluation. The top row represents the linearised approximation of it as is used during optimisation. So row 1, column 2 shows the totals achieved by optimisation of the goal function. In this *planned* domain, blue line shading indicates these actions were *convoluted* with ride actions. All figures show the case $a = 2\%$ as also reported in table 1.

nent, shown as the top (purple) rectangle of the bar graphs, is reduced in percentage of the total expected passenger time from 4.55% in the original schedule to 2.12% in the optimised schedule. This is for the linearised function as used in optimisation (column 2, row 1). For the non-linearised function (column 2, row 2), post-optimisation evaluation results in a reduction from 4.04% to 2.60%. In both cases, in absolute terms, we more than halve the amount of total expected passenger knock-on delay. The solver achieves this goal by changing orders of trains on common resources and optimally choosing headway supplements, weighing with passenger numbers and also balancing these with other goal function terms. Note that our model assumes the absence of dispatching interventions but with fewer knock-ons happening, the number of necessary dispatching interventions will be lower than in the original timetable as well.

The decrease of the ride+dwell and knock-on times is compensated only slightly by the small increase in expected transfer time. In column 2, representing evaluation on all origin-destination flows of 50 and more passengers, the total time net reduction is 8.66% (row 1, linear) and 7.06% (row 2, non-linear). The fact that the two pictures in column 2 are quite similar, demonstrates that our linearisation, even if only using 2 segments, is effective.

When we evaluate on *all* passenger streams, also the ones with fewer than 50 passengers, the result is a less grand, but still positive 0.42% reduction (non-linear). Plotting distributions of planned passenger journey time versus number of people, we saw that distributions corresponding to the major flows of column 2 are more realistic than the ones corresponding to all passenger streams. None of the major passenger flows, but a minority of the smaller ones have journey times between 2 and 3 hours for a single trip. Some of these are caused by an overenthusiastic diffusion of the zone-OD matrix to the station level [Sel11]. These travellers would most likely prefer other modes of transport. So we consider 7.06% to be our best prediction for reduction of total expected passenger time. Note that an average planned buffer of 8.89% is not enough to totally eliminate all knock-on delays, even though the assumed primary delays have only an average of 2%, seen in train time. The non-zero spread in the primary distribution explains this.

Table 1: Increasing primary delays, characterised by their average of $a\%$ of minimum dwell and ride times. The first column shows $a\%$. Column 2 and 3 show the computation time and the MILP gap achieved. We ran Gurobi 5.5.0 on an Apple MacBook Pro with 2.6GHz Intel i7 processor and 16GB 1.6GHz DDR3 memory. For the first set of result rows, the gap desired was set slightly above what was obtained as the gap of the first returned solution in earlier trials. The results in the last row are obtained by reduction of the desired gap by 1% compared to the first row. Graph size: 203 hourly trains, 5355 ride, 5152 dwell, 17553 major transfer, 31696 knock-on and 166 turn-around edges. Model size: 42609 supplement decision variables, 49415 integer decision variables, 41128 goal function terms for major flows and 58441 evaluation function terms for all flows.

a	solver time	MILP gap	major flows linearised ko-time reduction	major flows linearised time reduction	major flows non-linearised time reduction	all flows linearised time reduction	all flows non-linearised time reduction	missed transfer probability orig.	missed transfer probability opt.
%	min.	%	%	%	%	%	%	%	%
2	95	76.2	57	8.66	7.06	1.71	0.42	14.1	2.2
4	43	71.0	52	6.61	4.42	0.84	-1.41	14.6	4.2
6	75	61.3	63	7.65	5.73	2.07	0.13	15.1	1.8
8	66	61.3	59	5.83	3.86	0.40	-1.61	15.6	4.4
2	112	72.6	66	10.58	9.19	2.54	1.31	14.1	2.6

As table 1 shows, compared to the current timetable, our optimised timetables have quite some advantages. First, they respect all minimum ride- and dwell-times and all headway time buffers of 3 minutes between all train pairs on the same track section. In the original timetable sometimes minimum run times and headway times are not respected. Second, we calculated that, over all primary delay assumptions of table 1, the average chance of missing a transfer in the current timetable is at least 14.1% while in our optimised timetables it is at most 4.4%. Depending on the primary delays assumed, in our timetables the expected passenger times are between 7.06% and 0.42% lower than in the original schedule. This decrease is significant, because, of the total passenger time, the irreducible part of minimal ride and dwell times already consumes 67% in the original and 73% in the optimised timetable.

3.3 Computation Speed: The Solution is Returned Quickly

Using the solver abstraction part of the software library milp-logic [Sel12], which we developed and open sourced, as shown in table 1, Gurobi 5.5.0 was able to return a solution for the whole train set, for any primary delay distributions assumed, within about one hour. This is a big improvement compared to the current manual timetabling process that takes many human planners many months.

4 Conclusions and Further Work

This paper has three main contributions. Firstly, we analytically derived the expected passenger time experienced due to knock-on delays as a function of (i) the headway minima, (ii) the chosen headway supplements in a timetable and (iii) expected train delays and linearised this function, so that it can be used for linear optimisation. Secondly, we used the linearised functions as a method to minimise secondary delays, together with other expected passenger time, in a system containing all hourly trains in Belgium. Our results show that we can more than halve the amount of expected passenger knock-on delay in practice. Also, even with addition of many terms to the goal function, optimisation times for the Belgian timetable are only about one hour. Supposing primary delay distributions with an average of 2% of the minimal time of their corresponding actions, our improved timetable reduced expected passenger time for realistic passenger streams by 7.06% compared to the current one. Finally, although restricting the search space and using curtailed goal functions are the easy way to try to reduce solver time, we show that defining an all-encompassing goal function and searching the full solution space can lead to more desirable results: guaranteed feasibility, optimality and even lower solver times.

As for further work, we want to reduce our MILP gap, refine our minimum transfer time differentiating it by station and calibrate our primary delay distributions with train and location specific delays measured in practice. Also, instead of the *frequency-arc* hard constraint approach [Spa13], we want to add terms to the goal function that are due to *uneven spreading over the timetable period* of alternative trains between the same source and destination.

References

[Dew11] T. Dewilde, P. Sels, D. Cattrysse, and P. Vansteenwegen. "Defining Robustness of a Railway Timetable". In: *Proceedings of 4th Int'l Seminar on Railway Operations Modelling and Analysis (IAROR):February 16-18, Rome, Italy.* (2011).

[Gov07] R. M. Goverde. "Railway timetable stability analysis using max plus algebra". In: *Transportation Research Part B: Methodological* 41 (2007), pp. 179–201.

[Gov10] R. M. Goverde. "A delay propagation algorithm for large-scale railway traffic networks". In: *Transportation Research Part C* 18 (2010), pp. 269–287.

[Han08] I. A. Hansen and J. Pachl. *Railway Timetable and Traffic: Analysis, Modelling, Simulation*. Vol. 1. Eurailpress, Hamburg, Germany, Jan. 2008, pp. 1–288.

[Kro09] L. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, A. Schrijver, and R. Ybema. "The New Dutch Timetable: The OR Revolution". In: *Interfaces* 39 (2009), pp. 6–17.

[Lie07] C. Liebchen. "Periodic Timetable Optimization in Public Transport". In: *Operations Research Proceedings* 2006 (2007), pp. 29–36.

[Mus13] L. Mussone and R. Wolfler Calvo. "An analytical approach to calculate the capacity of a railway system". In: *EJOR* 228 (2013), pp. 11–23.

[Sel11] P. Sels, T. Dewilde, D. Cattrysse, and P. Vansteenwegen. "Deriving all Passenger Flows in a Railway Network from Ticket Sales Data". In: *Proceedings of 4th Int'l Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011, February 16-18, Rome, Italy.* (Feb. 2011).

[Sel12] P. Sels. *milp-logic: a C++ MILP Solver Abstraction Layer with a C++ Boolean Modelling Layer on Top.* Sept. 2012.

[Sel13a] P. Sels, T. Dewilde, D. Cattrysse, and P. Vansteenwegen. "Expected Passenger Travel Time as Objective Function for Train Schedule Optimization". In: *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark.* (May 2013).

[Sel13b] P. Sels, T. Dewilde, D. Cattrysse, and P. Vansteenwegen. "Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable". In: *to be submitted soon* (2013).

[Spa13] D. Sparing, R. M. Goverde, and I. A. Hansen. "An Optimization Model for Simultaneous Periodic Timetable Generation and Stability Analysis". In: *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark.* (2013).

Corresponding author: Peter Sels, KU Leuven, University of Leuven, Centre for Industrial Management/Traffic & Infrastructure, 3001 Leuven, Belgium, phone: +32 486 95 67 97, e-mail: sels.peter@gmail.com

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

Full paper in press for Journal: Transportation Research, Part B: Methodological, 2015

"To arrive on time, one has to depart on time."

– Jo Cornu, CEO of NMBS, the Belgian main train operator

"To depart on time, one has to arrive on time."

– Peter Sels, CEO of Logically Yours

The following paper derives and discusses the constraints of our cyclic timetabling model. It is a model based on the Periodic Event Scheduling Problem (PESP) formulation as first described by [41]. The implementation is a Mixed Integer Linear Programming (MILP) system of equations and inequalities. There are two sets of constraints. The first set is mandatory, in the sense that they are all needed to generate timetable solutions where trains do not collide on the same track and even leave a time of three minutes between each other. The second set of constraints, we call optional, because they are only needed to decrease model resolution time. This set of optional constraints is the result of many experiments we performed and selecting the ones that give results in the shortest time. All constraints are mathematically derived.

The objective function is the total expected passenger travel time in practice as described before in sections 2.2 and 2.3. This objective is of course minimised. The model instance generated is one for all 196 hourly passenger trains in Belgium. After 2 hours of computation, we get the optimised timetable. We then evaluate the original and optimised timetable over all passenger streams and get the following results. Assuming 2% of the minimum times of ride, dwell and transfers for the average of the negative exponential primary delays distributions on these activities, this optimised timetable reduces the expected passenger time by 3.81% compared to the original timetable made manually at Infrabel. Under the same primary delay assumptions, the average probability of a missed transfer per passenger goes down from 13.9% in the original timetable to 2.6% in the optimised timetable. Surprisingly, the total time spent in transfers in practice is smaller in the original timetable, even though less transfers are missed and the total journey time is smaller in the optimised timetable.

Macroscopic simulations for all 196 passenger trains in Belgium, repeated hourly for 5 consecutive hours, for both the original and the optimised timetable are performed with the simulator OnTime. If we let OnTime report the number of stations where at least 50% of trains are on time, we see the the there are more such stations in the optimised timetable than in the original timetable. This remains true if we raise the requirement from 50% to 80% and 90% of trains having to be on time. This confirms that the optimised timetable is more robust against the same primary delay distributions than the original, manually constructed one.

Microscopic simulations of the Belgian bottleneck, the Brussels North-South axis, are also performed in order to verify the feasibility and punctuality. These show that the optimised timetable, also in this critical area, is feasible and, subject to the same primary delay distributions, shows less knock-on delay than the original timetable. Indeed, the ratio of realised versus planned time is 1.6 for the original but only 1.25 for the optimised timetable.

The excess journey time, being the time that passengers wait for their first train at the station of departure plus the time they wait at their station of arrival for the next mode of transport, is not included in the objective function. So one may wonder if bunching of trains occurs in the optimised timetable. Otherwise put, how good or bad is the temporal spreading over the hour of alternative trains between important stations? We evaluate spreading in terms of excess journey time on the 11 corridors between neighbouring main stations in Belgium. This evaluation shows that the sum of journey time and excess journey time for all these corridors together, weighted by their number of passengers is significantly lower in the optimised timetable than in the original one. It is known that human timetablers spend quite some effort on realising spreading of alternative trains. They do this by, for example, spreading 3 alternative trains by 20 minutes plus or minus just a few minutes. However, these alternative trains may not stop in the same amount of stations, or they may even contain material with different performance characteristics. The result is that the train that would ride the fastest if it was optimally planned when considering only itself and not the other trains, is now planned as slow as the other slower trains. So it gets 'over-stretched'. In our optimised timetable, the journey time and excess journey time are both considered together, properly weighted by the applicable amount of passengers and the perfect trade-off can be made. This causes reduction of this total time on these 11 major corridors from 4.2%, if all passengers can or will not adapt their arrival time to the timetable and 15.6% if all passengers arrive just before the train they want to take. It can be concluded that in the original timetable, for these corridors, too much attention goes to temporal spreading, creating a negative over-stretching effect, and a net increase of expected passenger time. Even though we have not added the excess journey time into our objective function, our optimised timetable performs better. This must be the case because journey time is present in the objective function and is relatively much more important than excess journey time. Apart from evaluation, one could also *optimise* for total expected passenger time that *includes the excess journey time.* This is the topic of the paper in section 2.6.

We conclude that our timetabling method generates an optimised timetable that is better for passengers on many levels. So, we think our method and the generated timetables deserves the attention of railway companies.

Since the publication of this paper, we received some further comments and we want to respond to these here.

The paper mentions that our model is a PESP (Periodic Event Scheduling Problem). This is correct in the broad sense in that it models a problem that contains events that will occur periodically. However the strict interpretation - or one could say *traditional formulation* - of PESP is that the main variables represent the process durations and are modelled on edges of the event activity graph and integer variables are either zero or one. This is described in more detail in the paper by equation (1) and the corresponding text in section 1. Our model uses variables on both edges and vertices. The vertex variables represent absolute time instead of time modulo T, the timetable period. In doing so, we only have to model integer variables on primary edges. We provide an integer variable on each secondary edge, connecting vertices belonging to different trains and these integers can take other values besides 0 and 1. Whether this is a more complex or simpler way to model a PESP system can be debated. The fact is that our model can solve very large networks, up to networks of entire countries, even if we include an objective function that contains supplement variables relating to all edges in the graph. This has never been demonstrated with the traditional PESP models before.

For the case of Belgium, Infrabel told us to not take into account any freight trains. (For the case of Denmark, we took into account 4 freight trains, because Banedanmark thought they were important in a specific corridor.) In Belgium, freight trains are currently scheduled in between the planned passenger trains at a later stage. In our research for Belgium, in the original timetable as well, freight trains were removed. So when comparing the original and the optimised timetable, the same train set is compared. However, some unfairness may arise since, for example, in the original timetable two passenger trains may be separated with 6 minutes, allowing for a freight train to be planned in between, whereas the optimised timetable may have the freedom to separate the passenger trains with less than 6 minutes as well, but this would not allow insertion of a freight train in between. However, in Belgium, freight trains do not typically occur during peak times on the busiest passenger train corridors. Also, outside peak hours, ‘peak time passenger trains’ are dropped from the timetable and these time slots then become free for use by freight trains outside peak hours. However, it would be better to also plan freight trains at the same time with passenger trains, if all freight train paths are already known. In our later work for Banedanmark, we model a freight train as a passenger train with zero passengers. Naturally, our passenger reflowing does not assign any passenger to any freight train. During retiming, the weight of each freight train is set to 0 passengers. This makes freight trains to have a lower priority compared to passenger trains, which is also the case in reality.

The paper mentions that the motivation of using integer variables for arrival and departure times is probably that railway companies require these times to be integer minutes. The reason is that traffic control cannot deal with a higher precision.

The paper mentions that we use all constraints of the traditional PESP models. This is true but the upper bounds are larger in our model. This is explained in detail in the paper elsewhere.

The paper mentions that [24] was able to generate timetables for relatively large networks but in fact the networks considered were only small.

The paper sometimes uses the term ‘microscopic’ planning to mean ‘in station’ planning, or the train platforming problem (TPP) handling both platforming and routing. However, both macroscopic and microscopic approaches to the TPP exist in the literature. A microscopic approach models at the level of signals and block sections. A macroscopic approach models at a higher level that may ignore some of these details.

In our model, as described in the paper, the minimal headway time is set to 3 minutes. In fact, for trains on the same track in opposite directions, Infrabel uses a minimum headway of 6 seconds. This means our solutions are still valid but allowing this lower minimum would allow a somewhat bigger search space and possibly an even better solution.

The paper did not mention that time distance graphs for all lines and for both the original and the optimised timetable are automatically generated in PDF format. The graphs for the optimised timetable contain no violation of the macroscopic 3 minute minimum headway rule. The original timetable contained several violations against this rule (not counting the opposite trains on the same track).

Note that the paper implicitly assumes that the minimum ride, dwell and transfer times m_e are smaller than the timetable period T, so $\forall e \in E : m_e < T$. In Belgium this is certainly the case since the maximum value for m_e in the whole network is 13 minutes, compared to a period T of 60 minutes. This means that, since the remaining slack range of the supplement is between 0 and $T - m_e$, there is sufficient margin to deliver high robustness for any activity in the Belgian network.. This supports the validity of equation (4) and the effectiveness of for example equation (11) in the paper. Note that any network can be adapted to satisfy the condition $m_e < T$ by inserting a supplementary 'virtual' timetable point wherever $m_e \geq T$. Also, by taking the same measure for any activity where $m_e > 0.9 \cdot T$, our model has sufficient slack to choose a proper supplement time and can as such effectively strive for robustness on that activity.

As for our main conclusions about the reduction percentage of 3.8% of expected passenger time, the more robust transfers and the higher train punctuality, one can raise the important question if there is any possible unfairness in comparison between the manually constructed timetable and the optimised one. Unfairness can arise when one timetable is *more constrained* that the other. This is only the case for where we do not add freight trains and for where we do not add regularity constraints nor symmetry constraints. However, we explain in the paper that there are few freight trains on busy corridors and that regularity constraints are overly strict and that not enforcing them delivers better results in terms of total expected passenger time, including excess journey time, on as least the 11 main corridors. We als explain that symmetry constraints are not beneficial for the minimisation of expected passenger time.

A related but different question is whether our new timetable is *fair with respect to reality*. Input data, is imperfect and could potentially cause our conclusions to be different when real data is used or equivalently when our timetable would be put into practice. Of course, wherever possible, we used data that reflects reality as good as possible. However, there is uncertainty of primary delays, of the degree of transfer penalisation of passengers or passenger categories, the amount of passengers that adapt their arrival time at their departure station to the departure time of their train. For some, but not all of these a sensitivity analysis has been made.

Input data that only approximates reality can also form a *bias that favours one timetable* over the other. The assumed transfer penalty of 15 minutes in both timetables leads to good transfer times for the original timetable compared to for the optimised timetable. The OD input data and method of diffusing zone OD to the station OD uses uncertain input data and generates many very small passenger streams that are uncertain to also occur in reality. Our automatic timetable suffers from that as seen by its non-ability to also produce low transfer times for these very small streams. If realistic - so fewer and larger - passenger streams would be available it is likely that our optimised timetable would produce even better results.

Decidedly, no timetable for Belgium was constructed with more actual electronic sources of data than the one described here since we considered an OD matrix, all possible passenger transfers and a stochastic description of primary delays. Also, there has of course been no intentional bias to favour the automatically generated timetable.

Transportation Research Part B 84 (2016) 124–156

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Reducing the passenger travel time in practice by the automated construction of a robust railway timetable

P. Sels[a,b,c,*], T. Dewilde[a], D. Cattrysse[a], P. Vansteenwegen[a]

[a] *KU Leuven, Leuven Mobility Research Centre, CIB, Celestijnenlaan 300, Leuven 3001, Belgium*
[b] *Logically Yours BVBA, Plankenbergstraat 112 bus L7, Antwerp 2100, Belgium*
[c] *Infrabel, Department of Traffic Management & Services, Fonsnylaan 13, Brussels 1060, Belgium*

ARTICLE INFO

Article history:
Received 14 September 2014
Revised 11 December 2015
Accepted 13 December 2015

Keywords:
Optimal cyclic timetabling
Mixed integer linear programming
Minimal expected passenger time

ABSTRACT

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

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

1. Introduction and literature comparison

In railway operations, the construction of a train timetable, on the highest level of abstraction, means the determination of the arrival and departure times for each train in the system, in each station it serves or passes. This is a large

* Corresponding author at: KU Leuven, Leuven Mobility Research Centre, CIB, Celestijnenlaan 300, 3001 Leuven, Belgium. Tel.: +32 486956797.
E-mail addresses: sels.peter@gmail.com (P. Sels), thijs.dewilde@kuleuven.be (T. Dewilde), dirk.cattrysse@kuleuven.be (D. Cattrysse), pieter.vansteenwegen@kuleuven.be (P. Vansteenwegen).

http://dx.doi.org/10.1016/j.trb.2015.12.007

combinatorial optimisation problem with, for example, 165,000 constraints and about 200,000 variables for the Belgian network of hourly passenger trains. It requires hours rather than minutes to solve. Many constraints should be respected. For example, minimum ride and dwell times should not be violated, two trains on the same infrastructure resource should be separated by some minimum time and passenger transfers should be assigned neither too much nor too little time. In this section, we introduce our timetabling approach and compare it with the approaches available from the literature. We take the approach of presenting in the order that research expanded. This is usually also the chronological order but some exceptions occur.

1.1. The objective of a timetable

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

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

1.2. Efficiency and robustness of a timetable

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

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

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

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

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

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

1.3. The Periodic Event Scheduling Problem (PESP) approach

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

In PESP, an *event activity graph* $G(V, E)$ consisting of the event set V and the activity set or edge set E is set up. Events are arrival and departure times of trains at stations and represent the vertices of the graph G while edges are train activities (ride, dwell or turn-around) or passenger activities (transfer) or just represent separation constraints between events (headway-times). For a timetable with cyclic period T and model time resolution δ, the general vertex and edge constraints over the graph $G(V, E)$ as adapted from Sparing et al. (2013) are

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

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

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

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

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

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

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

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

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

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

1.4. Other macroscopic approaches

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

Nachtigall (1996), early on, constructed an approach to tackle systems with vehicles that are to be planned with different frequencies. Galli and Stiller (2010) also propose a method for such systems, claiming that traditional PESP methods are underperforming for these. Nachtigall and Voget (1997) introduced an objective function equal to the minimisation of waiting times and obtained schedules with a genetic algorithm. A different approach to macroscopic timetabling other than PESP is taken in Nachtigall and Opitz (2008), by using modulo simplex calculations. This publication presents an example application for 92 trains in Germany with periods of 20, 30, 60 and 120 min and considering 570 of the 1200 transfers that was solved in 20 min. However, in this paper, we show that it is paramount to evaluate over *all* transfers to know the real value of a timetable.

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

1.5. Microscopic and decomposing approaches

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

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

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

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

1.6. Timetabling approach and features comparison

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

1.7. Paper overview

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

2. Assumptions

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

As in many European countries (Austria, Denmark, Germany, Great-Britain, Norway, Switzerland, and The Netherlands) (Peeters, 2003), the Belgian railway operator requires a cyclic timetable with a period of one hour. Larger countries typically also run some longer distance trains with lower frequencies. A reason sometimes given for cyclic timetabling, as in Peeters (2003) and Cacchiani and Toth (2012), is that cyclic timetables are constructed so as to make it easy for passengers to remember the train departure times. A stronger motivation is that a cyclic timetable is more compact and also simplifies the material handling and crew rostering planning problems. Acyclic timetables are described by Ford and Haydock (1992) and are also called market-led timetables because these are more adapted to variation in passenger demand between different hours or days. In cyclic timetables, this flow variation is typically accommodated for by adding some morning peak specific and evening peak specific trains that are essentially each other's opposites. This is also our approach in this paper. The line network, representing the lines along which trains services will be run, also called *train relations*, is fixed by the Belgian railway operator and we considered it given. It is not altered in our optimisation, nor is the halting pattern of each train relation. We currently do not impose temporal spreading of alternative trains. These constraints are usually enforced to limit

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

Publication	Approach	Time res.	n/r	Objective (function)	#trains #stations #lines	Tool name@ company
Macro-timetabling						
Serafini and Ukovich (1989)	PESP	1’	n	Feasibility		
Schrijver and Steenbeek (1993)	PESP, CP	1’	n	Feasibility		DONS&CAD-ANS@NS, NL
Nachtigall (1996)	Multiple freq.	1’	n	Feasibility	40l	
Nachtigall and Voget (1997)	Genetic algorithm	Cont.	r	min(wait t., cost)	28t,42s	
Peeters (2003)	PESP, IP	1’	n,r	Various	37t,74s	
Semet and Schoenauer (2005)	Evolutionary algorithm		r	Min. total delay, resilience		
Lie (2006)				Min. dwell time		
Liebchen (2006) Liebchen (2007)	PESP, ILP	1’	n	Min. wait time some transfers	37t	@Berlin Underground, GE
Kroon et al. (2007)	Stochastic optimisation	Cont.	r	Robustness improvement	56t 40s	
Nachtigall and Opitz (2008)	Modulo simplex calculations	1’	n	Min. weighted sum of slack	92t $\geq$ 104s	
Kroon et al. (2009)	PESP, CP and ILP	1’	n	Feasibility and improved transfers		DONS&CAD-ANS@NS, NL
Schöbel and Kratz (2009) (acyclic)	PESP pareto opt.		r	Eff. and robust.		
Fischetti and Monaci (2009)	Robust optim. and simplified stoch. programming		r	Light robustness	1l	
Cacchiani et al. (2010) (acyclic)	Lagrangian heur.		r	Eff. and robust.	1000t,54s	
Galli and Stiller (2010)	Sharp trees		n	Feasibility		
Großmann et al. (2012) Kümmling et al. (2013)	SAT, binary .. .. search heuristic	1’	n	Min. weighted sum of slack	1336t, 3027s	TAKT @DB Netz AG, GE
Sparing et al. (2013)	PESP ILP	cont.	r	Min. cycle time (feas. & stab.)		
This paper: Sels et al.	PESP ILP	cont.	r	Min. expected total passenger journey time	203t 847s	RhinoCeros @Infrabel, BE
Micro-timetabling: includes platforming and routing						
Odijk (1996)	PESP cut generation	1’	n	Feasibility	18t,1s	
Zwaneveld et al. (1996)	Branch and cut heuristic	1’	n	Feasibility	18t,1s	STATIONS @NS, NL
Caimi et al. (2004)	Fixed point iter. heur. local search optim.		n r	Feasibility delay tolerance	19t 1s	
Caimi et al. (2009) & Caimi (2009)	Decomposition in condensation and comp. zones	Disc.	r	Max. planned train movements	32t 1s	
Burdett and Kozan (2010)	JSP, tabu search	cont.	n	feasibility	54t,	
Caprara et al. (2011a)	Node packing ILP	disc.	r	max. trains planned	237t 1s	
Caimi et al. (2011)	time exp. graph mc flow, ILP	disc.	r	max. planned Train movements	67t 1s	
Liu and Kozan (2011)	JSP, ILP	Cont.	n	Min. makespan	37t,43sc	
De Fabris et al. (2013) (meso)	Heur., mc flow			Meso feas.	,120s	
Sels et al. (2014) (acyclic and cyclic)	ILP, (fixed arr. and dep. times)	cont.	n	max # trains platformed	1028t, 1s	Leopard @Infrabel, BE
Micro–macro timetabling: includes platforming and routing						
Schlechte et al. (2011) (acyclic)	Aggregation	6s...30s	n	Max. planned trains	196t,18s	
Bešinović et al. (2013) Bešinović et al. (2015)	Aggregation & de-aggregation	5s	r	feas. & Robust	40t,26s	

the inter-departure times and reduce the waiting times in the station for some categories of departing passengers. We prefer to impose temporal spreading of alternative trains in a softer way via the objective function in future research.

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

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

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

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

3. Optimality: model objective function

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

We will only give a qualitative description of our objective function, as the main focus of this paper is the derivation of the PESP constraints of our model. As derived formally in detail in Sels et al. (2013a, 2013b), our *objective function* consists of the sum of the expected passenger time for each edge in the event activity graph $G(V, E)$ that corresponds to a passenger activity. So, for each ride, dwell and transfer edge we model an expected passenger time. We express this expected passenger time of an edge as a function of its minimum time and its added *supplement time*. The *shape* of this function mainly depends on the expected primary delay distribution and consequently, so does the supplement that should be ideally added. The *scale* of this function depends on the number of passengers involved. This indicates the relative importance of the expected passenger time of one edge compared to that of another and these are balanced by the objective function.

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

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

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

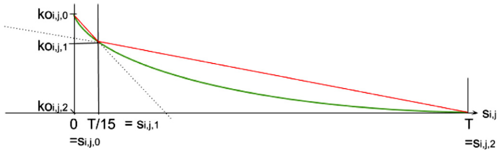


Fig. 1. Knock-on delay cost function.

delay cost function. The horizontal axis shows the supplement between 0 and 57 min and on the vertical axis the green curve is the actual cost function. The red piecewise linear approximation of it is used in our MILP model.

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

4. Feasibility: mandatory model constraints

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

4.1. Definition of graph, vertices and edge types

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

4.2. Function notation

In the following sections, we will need the following notation. The *type* of an edge is either ride, dwell, transfer, headway (=knock-on) or turn-around. We use $\neg$ to mean Boolean negation. We also use the following functions on a graph $G(V, E)$:
i: $E \to V$ so that $i(e)$ equals the unique 'in' vertex of edge e (source) and
o: $E \to V$ so that $o(e)$ equals the unique 'out' vertex of edge e (destination),
$-: E \to E$ so that $-e$ equals the unique inverse edge of edge e, $-e$ is defined by $o(-e) = i(e) \wedge i(-e) = o(e)$. $\forall t \in \{r, d\}$: i_t: $V \to E$ so that $i_t(v)$ equals the unique type-t-edge that ends in vertex v and
$\forall t \in \{r, d\}$: o_t: $V \to E$ so that $o_t(v)$ equals the unique type-t-edge that starts in vertex v,
$\forall t \in \{r, d\}$: i_t: $E \to E$ so that $i_t(e)$ equals the unique type-t-edge that precedes edge e and
$\forall t \in \{r, d\}$: o_t: $E \to E$ so that $o_t(e)$ equals the unique type-t-edge that succeeds edge e.

Note that the word *unique* here, means that no two edges can satisfy the mentioned property and so at most one result can be returned by each of these functions. It is possible that no single edge satisfies the properties of the function definition. For example, $i_r(v_0)$ where v_0 corresponds to any vertex in the station where a train starts is not defined. The set of results of the function is then the empty set. So, for example, in a constraint generation context, no constraint will be generated for the occurrence $i_r(v_0)$. Also remark that, we use the same function name i_t both from the domain V and from the domain E. From the type of the argument to it, it will be clear which function is meant. The same holds for o_t. Lastly, note that for $t \in \{tr, ta, ko\}$, so for secondary edges, i_t and o_t are not defined. Indeed multiple 'in' or 'out' edges could then typically result.
$\bar{\ }$: $E_{hw} \to E_{hw}$ so that $\bar{e}$ equals the unique *opposite* headway edge from train 2 to 1 when e goes from train 1 to 2.

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

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

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

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

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

4.3. Mandatory constraints

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

4.3.1. Edge constraints (intra-edge constraints)

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

$$\forall e \in E : b_e + m_e + s_e = e_e \tag{2}$$

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

4.3.2. Node constraints (inter-edge constraints)

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

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

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

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

4.3.3. General bounds on continuous variables

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

$$\forall_e \in E: \quad 0 \le s_e \le T - max(m_e, \delta) \tag{4}$$

and unlike tighter bounds, these will never cause infeasibilities.

4.3.4. General bounds on integer variables

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

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

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

4.3.5. All trains start in the first hour

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

$$\forall_{\neg\exists i_d(e)} e \in E_r: h_{lo} \cdot T \le b_e < (h_{lo} + 1) \cdot T, \tag{6}$$

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

4.3.6. All passing supplements are zero

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

$$\forall_{\neg s(e)} e \in E_d: \quad s_e = 0. \tag{7}$$

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

4.4. Odijk's rule for integers variables of cycle edges: feasibility and bounds

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

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

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

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

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

Eqs. (8) and (9) together ensure that the total cycle duration is a multiple of T. One can see that this property is a necessary condition for any cycle in the system, if after some multiple n of T, the schedule is to be repeated. The question

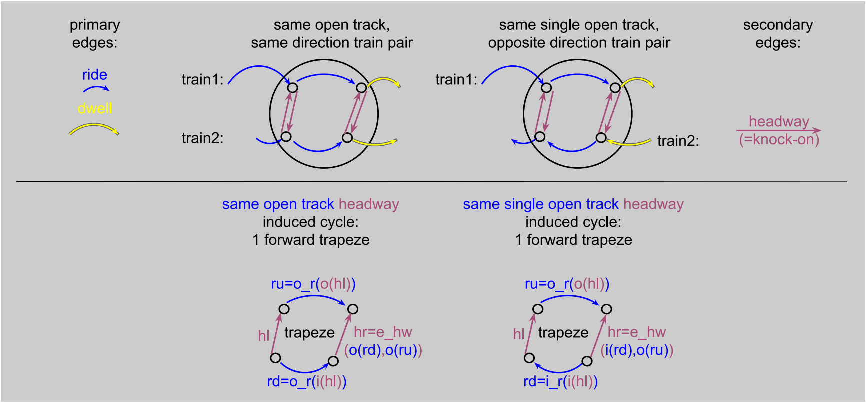


Fig. 2. Mandatory cycles for train pairs riding in the same direction and in opposite direction on the same track.

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

4.4.1. Separate and forbid reordering of same direction trains on the same open track section

Consider the situation shown in the left half of Fig. 2. A pair of trains rides on the same open track section in the *same direction*. The headway time needs to be respected, both between the times that the trains enter and the times that they leave the resource. As described before, in our graph, for these train pairs and at the corresponding ride edges, we added headway edges both between the beginnings of these pairs of corresponding ride edges and also between the endings of these pairs of corresponding ride edges, and this in both directions. Bottom left, we see that a trapeze shaped cycle is formed with two upwards pointing headway edges and two ride edges. The set of all these cycles is defined by a *condition-definition-function* $cd(h_l;\, r_u,\, r_d,\, h_r)$. Here, h_l stands for the left headway time edge (knock-on edge), r_u stands for the upper ride edge, r_d for the down ride edge, and h_r stands for the right headway time edge. In the definition of $cd(h_l;\, r_u,\, r_d,\, h_r)$, we specify how we calculate the edges r_u, r_d and h_r, if they exist. Whenever we can derive all edges r_u, r_d and h_r, we will formulate a constraint over these edges. The condition under which all edges can be derived is

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

Any *condition-definition-function* is implemented in software as a series of nested if-statements. For example, for Eq. (10), 'if $(h_l \prec \overline{h_l})$' will restrict the inner body to cases where $(h_l \prec \overline{h_l})$ holds. Within this first if-statement, the next statement will be 'if $(\exists r_u \in E_r : r_u = o_r(o(h_l)))$' and so forth. The condition $h_l \prec \overline{h_l}$, meaning h_l *comes strictly before* $\overline{h_l}$ in the ordered list of all edges we keep, is added to avoid that we add symmetric constraints, one for $(h_l, \overline{h_l})$ and one for $(\overline{h_l}, h_l)$, which we think would only deliver similar information to the solver. The expression $\exists r_u \in E_r : r_u = o_r(o(h_l))$ selects the ride edge that starts from the end of the headway time edge h_l, if it exists. It does not exist in the case that this headway edge h_l is pointing to the end instead of to the beginning of a ride edge. A similar expression is added to derive the ride edge r_d. Lastly h_r is calculated as the unique headway edge between the end points of the ride edges r_d and r_u. For the cases where $cd(h_l;\, r_u,\, r_d,\, h_r)$ succeeds in defining the three new edges, it defines a cycle with four edges.

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

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

Considering the ranges of terms marked in Eq. (11), applying Odijk's Eq. (9) to these cycles, gives for the ranges of the integers: $\forall_{cd(h_l;r_u,r_d,h_r)} h \in E_{hw} : -1 = \lceil \frac{0+0-(T-\delta)-(T-\delta)}{T} \rceil \le d_{h_l} - d_{h_r} \le \lfloor \frac{(T-\delta)+(T-\delta)-0-0}{T} \rfloor = 1$. Instead of adding these bounds, we first also require that there are no reorderings between trains 1 and 2. This is formally written as

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

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

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

4.4.2. Separate and forbid reordering of opposite direction trains on the same single open track section

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

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

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

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

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

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

4.4.3. Separate and forbid or allow in-station reordering based on infrastructure and halting pattern

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

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

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

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

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

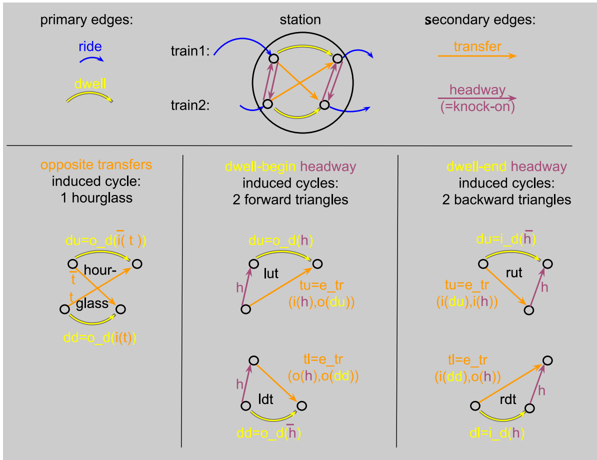


Fig. 3. Optional cycles for train pairs dwelling in a station.

Considering the ranges marked in Eq. (18), applying Odijk's rule here gives $\forall_{cd(h_l;d_u,d_d,h_r)} h \in E_{hw} : -1 = \lceil \frac{(0)+(0)-(T-\delta)-(T-\delta)}{T} \rceil \leq d_{h_l} - d_{h_r} \leq \lfloor \frac{(T-\delta)+(T-\delta)-(0)-(0)}{T} \rfloor = 1$, as the ranges of the integers. The specific cases of reordering or not reordering now will tighten these bounds further, so we first enforce the following narrower intervals:

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

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

5. Computation speed: optional model constraints

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

5.1. Opposite transfer pair induced small cycles: hourglasses

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

$$\begin{aligned} cd(t; d_u, d_d) \quad \equiv \quad & \forall_{t \prec \bar{t}} t \in E_{tr} : \\ & d_u = o_d(i(t)), \\ & d_d = o_d(i(\bar{t})), \end{aligned} \tag{20}$$

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

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

In this equation the minimum dwell times m_{d_u} and m_{d_d} must be strictly positive, since no transfers are possible on trains that do not stop. Applying Odijk's Eq. (9) to this *hourglass* cycle and taking into account the ranges of terms marked in Eq. (21), gives $\forall_{t \prec \bar{t}} t \in E_{tr} : -1 = \lceil \frac{(0)+(0)-(T-\delta)-(T-\delta)}{T} \rceil \le d_t + d_{\bar{t}} \le \lfloor \frac{(T-\delta)+(T-\delta)-(0)-(0)}{T} \rfloor = 1$, for the ranges of the integers. Eq. (21) can be rewritten as

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

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

$$\forall_{t \prec \bar{t}} t \in E_{tr} : \qquad -1 \le d_t + d_{\bar{t}} \le 0. \tag{23}$$

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

5.2. Opposite dwell-begin-headway induced small cycles: forward triangles

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

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

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

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

The constraints corresponding with these forward triangle cycles are

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

Applying Odijk's Eq. (9) to the cycles *lut* and *ldt* and taking into account the ranges of terms marked in Eq. (26) gives for the ranges of the integers

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

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

5.3. Opposite dwell-end-headway induced small cycles: backward triangles

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

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

and as small cycles, the two backward triangles

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

The constraints imposed corresponding with these backward triangle cycles are

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

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

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

5.4. Opposite headway edges integer constraints

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

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

Note that minimal headways m_h and $m_{\overline{h}}$ are strictly positive and s_h and $s_{\overline{h}}$ are zero or positive. This implies the bounds in Eq. (31). Odijk's equation now gives

$$\forall_{h \prec \overline{h}} h \in E_{hw} : -1 = \lceil \frac{(\delta)+(\delta)-(T-\delta)-(T-\delta)}{T} \rceil \le d_h + d_{\overline{h}} \le \lfloor \frac{(T-\delta)+(T-\delta)-(\delta)-(\delta)}{T} \rfloor = 1. \tag{32}$$

In addition to Odijk's rule, we see that the equation equivalent to (31)

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

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

$$\forall_{h \prec \overline{h}} h \in E_{hw} : \qquad d_h + d_{\overline{h}} = -1. \tag{34}$$

5.5. Dijkstra generated cycle constraints

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

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

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

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

5.6. Limiting the set E_{tr}

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

6. Model summary

The notation used is defined in Table 2. In the previous sections, we derived the mandatory constraints for our MILP model as given in Table 3 and the optional, solver speed improving constraints in Table 4. Thanks to the natural, large

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

T	Cyclic timetable period	δ	Timetable time resolution		
h_{lo}	First hour of schedule	h_{hi}	Last hour of schedule		
E'	Set of primary edges	r	Ride	b_e	Begin time of e
E'	$E_r \cup E_d$	d	Dwell	m_e	Minimum time of e
E''	Set of secondary edges	hw	Headway	s_e	Supplement time of e
E''	$E_{tr} \cup E_{hw} \cup E_{ta}$	ta	Turn-around	e_e	End time of e
E	Set of all edges	tr	Transfer	d_e	Integer variable for e
E	$E' \cup E''$	t	Edge type	V	Vertex set
$i(e)$	In vertex of e	$i_t(v)$	Unique type t inbound edge of vertex v		
$o(e)$	Out vertex of e	$o_t(v)$	Unique type t outbound edge of vertex v		
$e_t(v_0, v_1)$	Unique type t edge	$i_t(e)$	Unique type t predecessor edge of edge e		
	from vertex v_0 to vertex v_1	$o_t(e)$	Unique type t successor edge of edge e		
$s(d)$	True iff dwell activity d is a stop	$iar(d_0, d_1)$	True iff station infrastructure allows reordering between trains with dwell activities d_0 and d_1		
$\bar{e}$ is the edge between trains t_2 and t_1, that is the *opposite* of the edge e from train t_1 to t_2					
$cd(...)$ functions are condition defining functions that set the conditions under which rules apply and also define names for edges used in the further expression					

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

Intra-process constraints:	$\forall e \in E : b_e + m_e + s_e = e_e$
Inter-process connection constraints:	$\forall_{o(e_0)=i(e_1)} (e_0, e_1) \in (E', E' \cup E'') : e_{e_0} = b_{e_1}$ $\forall_{o(e_0)=i(e_1)} (e_0, e_1) \in (E'', E') : e_{e_0} + d_{e_0} \cdot T = b_{e_1}$
Continuous variable bounds:	$\forall_e \in E : 0 \leq s_e \leq T - max(m_e, \delta)$.
Integer variable bounds:	$\forall e \in E'' : (h_{lo} - h_{hi}) \leq d_e \leq (h_{hi} - h_{lo})$
All trains start in 1st hour:	$\forall_{\neg \exists i_d(e)} e \in E_r : h_{lo} \cdot T \leq b_e < (h_{lo} + 1) \cdot T$
Passing supplements are 0:	$\forall_{\neg s(e)} e \in E_d : s_e = 0$
Separate same direction trains on the same open track section: and forbid reordering:	$cd(h_l; r_u, r_d, h_r) = h_l \prec \overline{h_l} : \exists r_u \in E_r : r_u = o_r(o(h_l)) :$ $\exists r_d \in E_r : r_d = o_r(i(h_l)) : \exists h_r \in E_{hw} : h_r = e_{hw}(o(r_d), o(r_u))$. $\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : ((m_{h_l} + s_{h_l} + d_{h_l} \cdot T) + (m_{r_u} + s_{r_u})$ $= (m_{h_r} + s_{h_r} + d_{h_r} \cdot T) + (m_{r_d} + s_{r_d}))$ $\wedge (d_{h_l} = d_{h_r})$
Separate opposite direction trains on the same single open track section: and forbid reordering:	$cd(h_l; r_u, r_d, h_r) = h_l \prec \overline{h_l} : \exists r_u \in E_r : r_u = o_r(o(h_l)) :$ $\exists r_d \in E_r : r_d = o_r(i(h_l)) : \exists h_r \in E_{hw} : h_r = e_{hw}(i(r_d), o(r_u))$. $\forall_{cd(h_l; r_u, r_d, h_r)} h_l \in E_{hw} : ((m_{h_l} + s_{h_l} + d_{h_l} \cdot T) + (m_{r_u} + s_{r_u})$ $-(m_{h_r} + s_{h_r} + d_{h_r} \cdot T) + (m_{r_d} + s_{r_d}) = 0))$ $\wedge (d_{h_l} = d_{h_r})$
Forbid or allow reordering within a station depending on infrastructure: and halting patterns:	$cd(h_l; d_u, d_d, h_r) = h_l \prec \overline{h_l} : \exists d_u \in E_d : d_u = o_d(o(h_l)) :$ $\exists d_d \in E_d : d_d = o_d(i(h_l)) : \exists h_r \in E_{hw} : h_r = e_{hw}(o(d_d), o(d_u))$. $\forall_{cd(h_l; d_u, d_d, h_r)} h \in E_{hw} : ((m_{h_l} + s_{h_l} + d_{h_l} \cdot T) + (m_{d_u} + s_{d_u})$ $= (m_{h_r} + s_{h_r} + d_{h_r} \cdot T) + (m_{d_d} + s_{d_d}))$ $\wedge$ (*if* $(\neg iar(d_u, d_d)) : -0 \leq d_{h_l} - d_{h_r} \leq +0$, *else if* $(\neg s(d_u) \wedge \neg s(d_d)) : -0 \leq d_{h_l} - d_{h_r} \leq +0$, *else if* $(s(d_u) \wedge \neg s(d_d)) : -1 \leq d_{h_l} - d_{h_r} \leq +0$, *else if* $(\neg s(d_u) \wedge s(d_d)) : -0 \leq d_{h_l} - d_{h_r} \leq +1$, *else if* $(s(d_u) \wedge s(d_d)) : -1 \leq d_{h_l} - d_{h_r} \leq +1)$

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

Opposite transfers induced small cycles (hourglasses):	$\forall_{t \prec \bar{t}} t \in E_{tr} : ((m_t + s_t + d_t \cdot T) + (m_{\bar{t}} + s_{\bar{t}} + d_{\bar{t}} \cdot T)$ $= (m_{d_u} + s_{d_u}) + (m_{d_d} + s_{d_d})) \wedge (-1 \leq d_t + d_{\bar{t}} \leq 0)$
Opposite dwell-begin-headway induced small cycles (forward triangles):	$cd(h; d_u, t_u, d_d, t_d) = h \prec \bar{h} : \exists d_u \in E_d : d_u = o_d(h) :$ $\exists t_u \in E_{tr} : t_u = e_{tr}(i(h), o(d_u)) :$ $\exists d_d \in E_d : d_d = o_d(\bar{h}) : \exists t_d \in E_{tr} : t_d = e_{tr}(o(h), o(d_d))$. $\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : ((m_h + s_h + d_h \cdot T) + (m_{d_u} + s_{d_u})$ $= (m_{t_u} + s_{t_u} + d_{t_u} \cdot T)) \wedge (0 \leq d_{t_u} - d_h \leq 1)$ $\wedge((m_h + s_h + d_h \cdot T) + (m_{t_d} + s_{t_d} + d_{t_d} \cdot T) = (m_{d_d} + s_{d_d}))$ $\wedge(-1 \leq d_{t_d} + d_h \leq 0)$
Opposite dwell-end-headway induced small cycles (backward triangles):	$cd(h; d_u, t_u, d_d, t_d) = h \prec \bar{h} : \exists d_u \in E_d : d_u = i_d(\bar{h}) :$ $\exists t_u \in E_{tr} : t_u = e_{tr}(i(d_u), i(h)) :$ $\exists d_d \in E_d : d_d = i_d(h) : \exists t_d \in E_{tr} : t_d = e_{tr}(i(d_d), o(h))$. $\forall_{cd(h; d_u, t_u, d_d, t_d)} h \in E_{hw} : ((m_h + s_h + d_h \cdot T) + (m_{t_u} + s_{t_u} + d_{t_u} \cdot T)$ $= (m_{d_u} + s_{d_u})) \wedge (-1 \leq d_{t_u} + d_h \leq 0)$ $\wedge((m_h + s_h + d_h \cdot T) + (m_{d_d} + s_{d_d}) = (m_{t_d} + s_{t_d} + d_{t_d} \cdot T))$ $\wedge(0 \leq d_{t_d} - d_h \leq 1)$
Opposite headway integer constraints:	$\forall_{h \prec \bar{h}} h \in E_{hw} : ((m_h + s_h + d_h \cdot T) = -(m_{\bar{h}} + s_{\bar{h}} + d_{\bar{h}} \cdot T))$ $\wedge(d_h + d_{\bar{h}} = -1)$
Transfer induced Dijkstra cycle constraints:	$\forall t \in E_{tr} : \sum_{e \in (c_t^+ \cap E')} m_e + s_e + \sum_{e \in (c_t^+ \cap E'')} m_e + s_e + d_e \cdot T$ $= \sum_{e \in (c_t^- \cap E')} m_e + s_e + \sum_{e \in (c_t^- \cap E'')} m_e + s_e + d_e \cdot T$

ranges of all supplement variables, we avoid a cause of infeasibilities that do occur in some other models. We give proof of this in Appendix A.

7. Results

Compared to our previous papers Sels et al. (2011b, 2013a); 2013b), that focussed on our goal function, this paper focusses on the constraints. We discuss three main results of our work. Firstly, in Section 7.1, we show that our MILP model avoids a cause of infeasibilities that does occur in other models. Secondly, in Section 7.2, for the case of all passenger trains in Belgium, we show that we significantly reduce expected passenger travel time in practice. Thirdly, in Section 7.5, we show that we can do so in relatively short solver times.

7.1. Feasibility: a solution is always returned in practice

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

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

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

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

7.2. Quality: the solution has lower expected passenger travel time in practice

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

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

Column 2, row 1 of Fig. 4, planned *passenger* time, demonstrates a much larger reduction of the ride+dwell supplements than in planned *train* time, from 10.06% down to 2.76%. These are weighted by number of passengers. This same advantageous larger expected time reduction in column 2 compared to column 1, is also present in the expected time domains. Indeed, in column 2, row 2, representing evaluation of the linearised cost functions, ride-dwell supplements go down from 9.88% in the original timetable to 3.77% in our optimised timetable. In column 2, row 3, representing evaluation on the true,

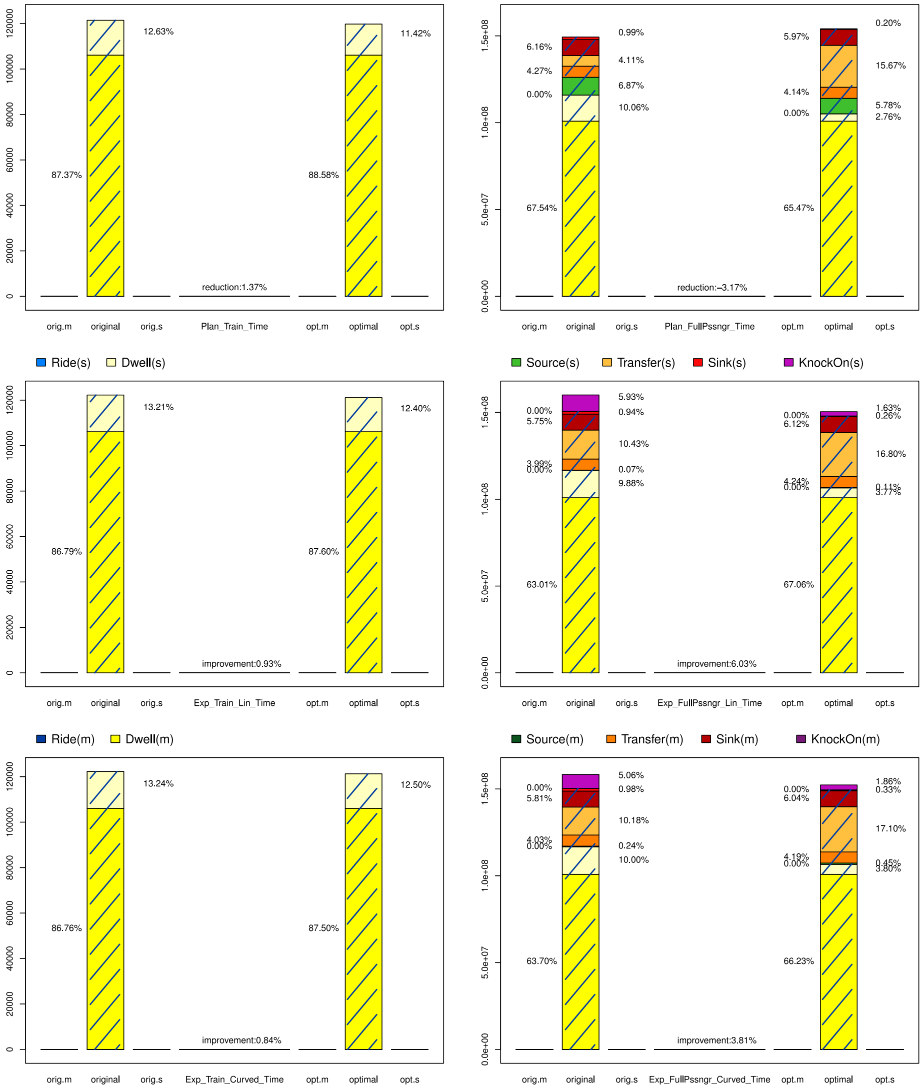


Fig. 4. Reducing train time and expected passenger time from original to optimised timetable as also reported in line 2 of Table 5. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

non-linear cost functions, supplement percentages go down similarly from 10.0% to 3.80%. But also for expected knock-on time we reduce time (row 2 (5.93% down to 1.63%) and row 3 (5.06% down to 1.86%)). The decrease of these expected ride, dwell and knock-on times is compensated only partially by the increase in expected transfer time (row 2 (10.43% up to 16.80%) and row 3 (10.18% up to 17.10%)). Note that this expected transfer time includes both successful and missed transfers, properly weighted with their respective probabilities to occur. Our calculations, both during optimisation and during evaluation, assume a penalty for a missed transfer of 1 h. Since our timetable is cyclic the same train occurs with the same

Table 6
Problem instance statistics.

# ride edges	5078	# (continuous) *b*-variables	52,067
# dwell edges	4882	# (continuous) *s*-variables	52,067
# turn-around edges	168	# (continuous) *e*-expressions	52,067
# knock-on(headway) edges	41,258	# (integer) *d*-variables	42,107
# major transfer edges	154,95	# objective function terms for major flows	50,481
# model rows	165,298	# are functions of 1 *s*-variables	41,258
# model columns	111,576	# are functions of 2 *s*-variables	9223
# model non-zero elements	464,064	# function terms in post-optimisation evaluation	70,909

timing every next hour. In practice, a passenger could also take another train in the same direction without waiting a full hour. We did not model this effect. This means that the expected transfer time in our model is conservatively somewhat overestimated. Since missed transfers are penalised more in our model than in a model that would account for this effect, our model will generate a timetable that has fewer missed transfers. In column 2, row 2, the net total expected passenger travel time reduction is 6.03% for the approximate linear cost function evaluation and in row 3 it is still 3.81% for the actual cost function evaluation. So we conclude that 3.81% is our best prediction for reduction of expected passenger time for all passenger streams together. The average train passenger may expect this reduction amount in practice when the optimised timetable is used.

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

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

Columns 2 and 3 in Table 5 show the solver time and the MILP gap achieved. For all rows, to guarantee a solution in a short time, the maximum desired MILP gap was set slightly above what was obtained as the gap of the first returned solution in earlier trials. We ran Gurobi 5.6.3 on a HP Z210 Workstation Xeon CPU E31240 at 3.3 GHz with 16GB memory running Microsoft Windows 7 Enterprise. In Table 6, we give some numbers that indicate the problem size that hold for all these cases. This allows comparison to other published models. Our graph contains 196 hourly trains, 5078 ride edges, 4882 dwell edges, 15, 495 major transfer edges, 41, 258 knock-on(=headway) edges and 168 turn-around edges. Our model contains 52, 067b and 52, 067s decision variables and 52, 067e expressions. It has 42, 107d integer decision variables, corresponding to one for each secondary edge, and 50, 481 objective function terms for major flows of which 41, 258 terms are functions of 1 s variable and 9223 are functions of 2 s variables. For final evaluation over all streams, 70, 909 function terms are added. After pre-solve, which only takes about 5 s, our model contains 165, 298 rows, 111, 576 columns and 464, 064 non-zero elements.

Compared to the timetable currently in operation, our optimised timetables have quite some advantages. Firstly, they respect all minimum ride- and dwell-times without exception. Secondly, they respect all headway time buffers of 3 min between all train pairs on the same track section. Thirdly, our calculations show that, over all primary delay assumptions of Table 5, the average chance of missing a transfer in the current timetable is at least 14.1% while in our optimised timetable it is at most 3.44%. For the case $a = 2\%$ in Table 5, the expected passenger time is 3.81% lower than in the original schedule. Further reduction of the maximum desired MILP gap below 79.2% did not give any new timetable solution within 12 h. The decrease with 3.81% might seem small, but it should be noted that the fixed minimal ride and dwell times already consume 63.70% of the total passenger time in the original timetable and 66.23% in our optimised timetable.

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

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

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

7.3. Balance: less expected knock-on delay time versus more expected transfer time

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

7.3.1. Less expected knock-on delay

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

In Fig. 5, we show histograms of headway durations for both the original (in the left half) and the optimised timetable (in the right half) and in each case, both for the planned time domain, in terms of edges (top row) and in terms of passengers (bottom row). One can see that, for the optimised timetable, in the right half of Fig. 5, no headways are shorter than the imposed 3 min minimum. Equivalently, none are longer than $T - 3 = 57$ min. In the original timetable, there are a few headways that are smaller than the minimal 3 min and some are larger than the 57 min maximum. There are 2 causes of this. Firstly, between some train pairs, less headway is planned by the planners. Secondly, for a few trains, for a few tracks, the original timetable violates minimum ride or dwell times. This happens precisely when timetablers are trying to fix headway violations. Since in practice, these runtime violations will mean that the train will be delayed, we also adapt the original timetable to reflect this, so arrival and departure times for the part of the train after the violation are increased so that the minimum ride and dwell times are respected. This means that headway times are again decreased. From the top row of Fig. 5, excluding the mentioned violations, it is evident that the average headway, if counted per edge, is always $T/2$. This is the case because every headway edge e of length $m_e + s_e$ also has an opposite headway edge of length $T - (m_e + s_e)$. Small deviations in the average are entirely due to the discretisation in the histogram construction. When weighing with passenger numbers, we get a different situation. Indeed, we see that in our optimised timetable, many people experience a potential knock-on delay associated with a second train of a train-pair that is separated by 7 min and also of $T - 7 = 53$ min. Assigning big headways to trains riding before trains with many people on reduces total expected passenger knock-on time. This is the case because the cost function, as can be seen in Fig. 1, is a decreasing function. Note also that on the second row of Fig. 5, the optimised timetable, in the right column, indicates that fewer people will experience headways of between 3 and 6 min than in the original timetable. The first row, shows a less pronounced difference, and this can again be explained by the optimised timetable paying more attention to trains with many passengers than with few passengers via the objective function. This is consistent with the bar graphs shown in row 3, column 2 of Fig. 4.

7.3.2. More expected transfer time

Our optimisation currently still results in a timetable with more expected transfer time and to improve on this is a topic for further study. So our software is already better at planning headways but humans are currently still better at planning transfers. Important questions now are if the reduction in expected headway time is coupled to the gain in expected transfer

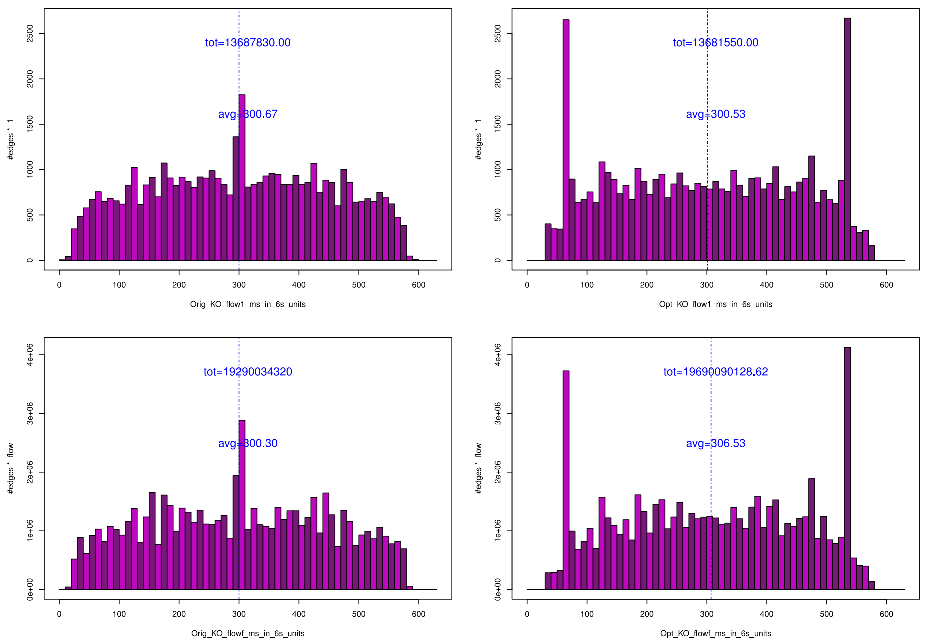


Fig. 5. Planned headway times (minimum of 3 min + supplement) histograms, showing for each headway duration, for how many edges it occurs and how many passengers experience the knock-on time associated with this headway duration.

time or not, if a timetable with lower expected time for both knock-ons and transfers exists and whether it can be found in an automated way and how.

7.4. Balance: less journey time offsets more excess journey time

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

$$E(w) = E(h)/2 \cdot \left(1 + C_v(h)^2\right). \tag{36}$$

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

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

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

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

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

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

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

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

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

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

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

7.5. Computation speed: the solution is returned quickly

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

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

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

(*continued on next page*)

Table 7 (*continued*)

O and D	Orig line 1	Orig line 2	Orig (line 3)	Orig E(.)	Opt line 1	Opt line 2	Opt (line 3)	Opt E(.)	Opt/orig ratio E(.)	Opt/orig Totals
8	IC:G:1825	IC:C:725	IC:A:525		IC:G:1825	IC:C:725	IC:A:525			
Brugge[220]	7:44:00	7:30:00	7:03:00	10.7	7:53:30	7:12:48	7:43:54	11.9	1.11	
Oostende[929]	7:58:00	7:44:00	7:17:00	10.7	8:06:00	7:25:24	7:56:12	11.8	1.11	1.07
E(jt)	14.0	14.0	14.0	14.0	12.5	12.6	12.3	12.5	0.89	0.93
				72.8				74.0		1.02
9	IC:N:4525	IC:I:2025	IC:Q:2625		IC:N:4525	IC:I:2025	IC:Q:2625			654.61
Antwerpen-Centraal[37]	7:47:00	7:17:00	7:42:00	12.9	7:33:48	6:59:54	7:21:36	10.8	0.84	
Mechelen[810]	8:07:00	7:37:00	8:03:00	13.3	7:48:42	7:22:30	7:37:24	12.4	0.94	
E(jt)	20.0	20.0	21.0	20.3	14.9	22.6	15.8	17.8	0.87	
9	IC:N:4500	IC:I:2000	IC:Q:2600		IC:N:4500	IC:I:2000	IC:Q:2600			
Mechelen[810]	7:53:00	7:23:00	7:57:00	13.3	7:12:54	7:39:30	8:22:00	12.5	0.95	
Antwerpen-Centraal[37]	8:13:00	7:43:00	8:18:00	12.9	7:27:42	7:55:18	8:37:48	12.2	0.94	0.92
E(jt)	20.0	20.0	21.0	20.3	14.8	15.8	15.8	15.5	0.76	0.82
				93.0				81.2		0.87
10	IC:R:3425	IC:N:4525	IC:I:2025		IC:R:3425	IC:N:4525	IC:I:2025			525.90
Mechelen[810]	7:57:00	7:10:00	7:40:00	11.3	7:00:00	7:49:42	7:24:30	11.2	0.99	
Brussel-=Noord[221]	8:13:00	7:23:00	7:55:00	12.1	7:14:24	8:01:36	7:35:48	10.7	0.89	
E(jt)	16.0	13.0	15.0	14.7	14.4	11.9	11.3	12.5	0.85	
10	IC:R:3400	IC:N:4500	IC:I:2000		IC:R:3400	IC:N:4500	IC:I:2000			
Brussel-=Noord[221]	7:47:00	7:37:00	7:05:00	12.1	7:15:54	7:58:36	7:25:18	12.5	1.03	
Mechelen[810]	8:03:00	7:50:00	7:20:00	11.3	7:30:42	8:10:48	7:36:06	13.6	1.20	1.03
E(jt)	16.0	13.0	15.0	14.7	14.8	12.2	10.8	12.6	0.86	0.86
				76.1				73.1		0.96
11	IC:G:1800	IC:A:500	IC:E:1500		IC:G:1800	IC:A:500	IC:E:1500			1169.78
Brugge[210]	7:18:00	7:59:00	7:36:00	10.1	6:42:12	7:31:24	7:18:42	13.4	1.33	
Gent Sint-Pieter[455]	7:45:00	8:22:00	8:00:00	10.3	7:05:54	7:50:12	7:37:42	11.8	1.14	
E(jt)	27.0	23.0	24.0	24.7	23.7	18.8	19.0	20.5	0.83	
11	IC:G:1825	IC:A:525	IC:E:1525		IC:G:1825	IC:A:525	IC:E:1525			
Gent Sint-Pieter[455]	7:42:00	7:01:00	7:24:00	10.1	7:29:06	7:17:18	7:48:00	11.3	1.12	
Brugge[210]	8:15:00	7:38:00	8:00:00	10.3	7:52:30	7:36:12	8:06:30	11.2	1.09	1.17
E(jt)	33.0	37.0	36.0	35.3	23.4	18.9	18.5	20.3	0.57	0.68
				100.9				88.5		0.88

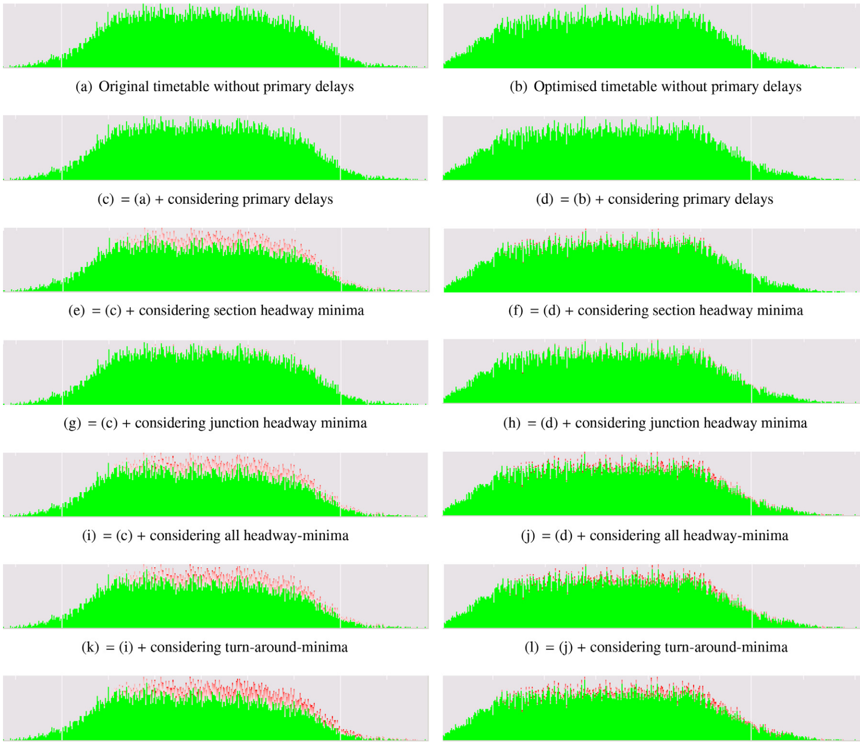
Table 8
Reduction of expected passenger time, including both journey time and excess journey time for varying fractions ($r\%$) of passengers not adapting their arrival time at station of departure to train departure times in the timetable.

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

8. Validation and check of applicability by simulation

8.1. Validation by macroscopic simulation

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



(a) Original timetable without primary delays

(b) Optimised timetable without primary delays

(c) = (a) + considering primary delays

(d) = (b) + considering primary delays

(e) = (c) + considering section headway minima

(f) = (d) + considering section headway minima

(g) = (c) + considering junction headway minima

(h) = (d) + considering junction headway minima

(i) = (c) + considering all headway-minima

(j) = (d) + considering all headway-minima

(k) = (i) + considering turn-around-minima

(l) = (j) + considering turn-around-minima

(m) = (k) + considering transfer-min. and synchronisation-max.

(n) = (l) + considering transfer-min. and synchronisation.-max.

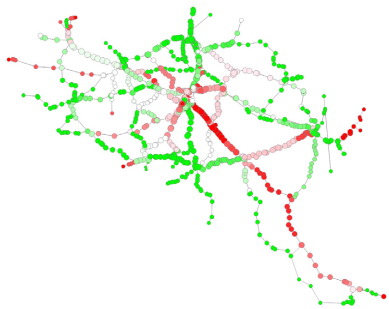
Fig. 6. OnTime event punctuality analysis reports. The horizontal axis is the time of the day. The vertical axis is the amount of train events, where green pixels are on time events, light red pixels are slightly delayed events and dark red pixels are more delayed events. Passenger numbers are not considered in these graphs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Especially the *section* headway minima are causing delays especially in the original timetable. This indicates that on top of these minima, larger buffers exists in the optimised than in the original timetable in places where it matters most. The optimised timetable can 'absorb' these problems better. Fig. 6(k) and(l) shows no significant change in accumulated delays when turn-around minima are imposed. Fig. 6(m) shows that in the original timetable, when also transfers are considered, some more red pixels are added to the graph while for Fig. 6(n) no extra red pixels seem to be generated. Note that punctuality of all transfers are considered without any weighing by passenger numbers.

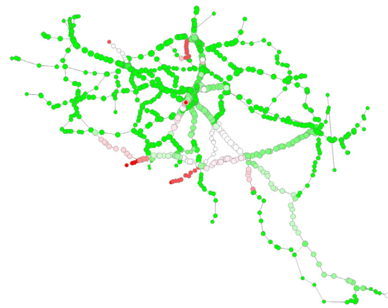
Fig. 7(a)–(f) shows the punctuality analysis for a whole day geographically in terms of the probability of being less than 0 min late in an approaching activity. This includes transfer punctuality. Note that passenger numbers are not considered in these graphs. In these figures, 'red/green x%' means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

(a) Orig. tt, 0':00" late probability, red/green 50%

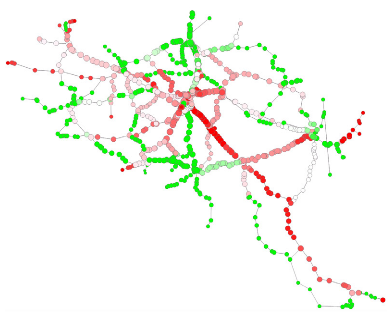
(b) Opt. tt, 0':00" late probability, red/green 50%



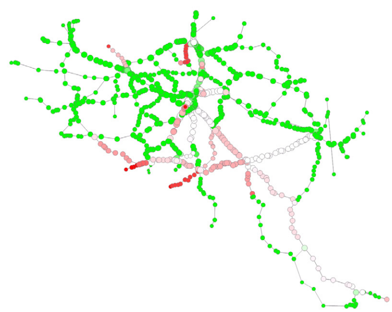
(c) Orig. tt, 0':00" late probability, red/green 80%



(d) Opt. tt, 0':00" late probability, red/green 80%



(e) Orig. tt, 0':00" late probability, red/green 90%



(f) Opt. tt, 0':00" late probability, red/green 90%

Fig. 7. OnTime GeoView 0':00" late probability analysis for a full day, with consideration of transfers. For red/green *x*%, a red dot means that in that location, less than *x*% of all trains arrives in time. Punctuality for the optimised timetable is clearly better than for the original one. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

a buffer against the delays causing this since this may make the timetable unnecessarily inefficient in 99% of the other cases. Our objective function makes this trade-off between robustness and efficiency for passengers. Note that Fig. 7(a)–(f) shows lateness as defined with zero minutes delay, because that is the threshold at which passengers start to experience delays. Train companies usually report punctuality numbers based on lateness definitions of 6 min late (in Belgium and Switzerland) or 3 min late (in The Netherlands).

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

8.2. Check of applicability by microscopic simulation

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

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

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

9. Conclusions and further work

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

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

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

Acknowledgements

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

Appendix A. Proof: our supplement upper bounds never cause infeasibilities.

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

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

- *Step 1: modulo T compression:* For any intra-train constraint, so $\forall e \in E' : b_e + m_e + s = e_e$, we call the solution to $M1$ for a specific $e \in E'$, $(b_{e,1}, s_{e,1}, e_{e,1})$ for which $s_{e,1} < T$ not necessarily holds. Given that $s_{e,1} \geq 0$, we can construct a solution to $M2$ as $(b_{e,2}, s_{e,2}, e_{e,2}) = (b_{e,1}, s_{e,1} - i_e \cdot T, e_{e,1} - i_e \cdot T)$, where the integer i_e is chosen so that $0 \leq s_{e,2} < T$. Said otherwise $s_{e,2} = s_{e,1} \mod T$, so this is always possible. This solution can be applied for every $e \in E'$, performed train per train (independently), and in the chronological order of (ride, dwell) edges of a train. When an edge e is compressed by an amount $i_e \cdot T$, for all succeeding edges e of that same train, $b_{e,1}$ and $e_{e,1}$ are also decremented by $i_e \cdot T$. (Note that this maintains the satisfaction of the constraint: $b_{e,1} + m_e + s_{e,1} = e_{e,1}$.) Afterwards $(b_{e,2}, s_{e,2}, e_{e,2}) = (b_{e,1}, s_{e,1} - i_e \cdot T, e_{e,1} - i_e \cdot T)$ is applied on those succeeding edges. This is always possible since the set E' contains no cycles. Note that also for the minimum scheduling hour $h_{lo} \cdot T$, $\forall e \in E'$: $h_{lo} \cdot T \leq b_{e,2} \leq e_{e,2}$ still holds, since the beginning time $b_{e_{t,0},2} = b_{e_{t,0},1}$ of the first edge $e_{t,0}$ of each train t has not been decreased and all other times $b_{e,2}$, $e_{e,2}$ of that train t have not been decremented below $b_{e_{t,0},2}$. Since no single time b_e or e_e has been increased, it also still holds that $\forall e \in E'$: $b_{e,2} \leq e_{e,2} \leq h_{hi} \cdot T$. So we conclude that constraints for all intra-train edges in M2 as well as the boundary constraints to do with h_{lo} and h_{hi} are satisfied.
- *Step 2: inter-train constraint matching:* For any inter-train constraint, so $\forall e \in E''$: $b_e + m_e + s + d_e \cdot T = e_e$, we call the original solution to $M1$ for a specific $e \in E''$, $(b_{e,1}, s_{e,1}, d_{e,1}, e_{e,1})$ for which $s_1 < T$ not necessarily holds. Since step 1 above was carried out, $b_{e,2}$ and $e_{e,2}$ can now be different from $b_{e,1}$ and $e_{e,1}$. Indeed, after step 1, in general $(b_{e,2}, e_{e,2}) = (b_{e,1} - i \cdot T, e_{e,1}, -j \cdot T)$ holds, with $0 \leq i \leq j$ and $b_{e,2} \leq e_{e,2}$. So the originally satisfied constraint from M1, $(b_{e,1}) + m_e + s_{e,1} + d_{e,1} \cdot T = (e_{e,1})$, can be converted to $(b_{e,2} + i \cdot T) + m_e + s_{e,1} + d_{e,1} \cdot T = (e_{e,2} + j \cdot T)$ or $b_{e,2} + m_e + s_{e,1} + (d_{e,1} + i - j) \cdot T = e_{e,2}$. By setting $s_{e,2} = s_{e,1} - k \cdot T$ with $k \geq 0$ and $0 \leq s_{e,2} < T$, so $s_{e,2} = s_{e,1} \mod T$, we get $b_{e,2} + m_e + (s_{e,2} + k \cdot T) + (d_{e,1} + i - j) \cdot T = e_{e,2}$ or $b_{e,2} + m_e + s_{e,2} + (d_{e,1} + i - j + k) \cdot T = e_{e,2}$. Next, we set $d_{e,2} = (d_{e,1} + i - j + k) \cdot T$ such that $b_{e,2} + m_e + s_{e,2} + d_{e,2} \cdot T = e_{e,2}$ results, which delivers a solution $(b_{e,2}, s_{e,2}, d_{e,2}, e_{e,2})$ for any edge $e \in E''$ for M2. Note that $i - j + k$ can be both negative or positive so $d_{e,2}$ can be smaller, equal or larger than $d_{e,1}$, which initially can raise the concern that a larger range for the d_e variables would be needed for M2 compared to for M1 to be able to guarantee feasibility. However, since step 2 does not change any values $b_{e,2}$ or $e_{e,2}$ compared to step 1, all these values are still in the $[h_{lo} \cdot T, h_{hi} \cdot T]$ window as proven in step 1. Consequently, they are still at most $(h_{hi} - h_{lo}) \cdot T$ apart. This means that the range of d_e can be kept the same for M2 as for M1, more specifically: $\forall e \in E'' : d_e \in [h_{lo} - h_{hi}, h_{hi} - h_{lo}]$. As for the cycle constraints, these are linear combinations of the other constraints, so if the other constraints will not cause infeasibilities, neither will the cycle constraints.

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

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

1. *Lower bounds on ride and dwell times per train:* When the sum of the minima of subsequent ride and dwell actions of a train sum up to S and when the time window $[h_{lo} \cdot T, h_{hi} \cdot T]$ to schedule trains in has length $(h_{hi} - h_{lo}) \cdot T < S$, the model will be infeasible. These problems are caused by *lack of time*.
2. *Lower bounds on headway times per track:* When more than 20 trains are required to be planned on a single infrastructure resource in 60 min and also a lower bound of 3 minheadway time is imposed, one is obviously asking the impossible. The model will then be infeasible. These problems are caused by *lack of capacity* which can also be seen as *lack of space*.

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

References

Bešinović, N., Quaglietta, E., Goverde, R., 2013. A simulation-based optimization approach for the calibration of dynamic train speed profiles. Journal of Rail Transport Planning and Management 3, 126–136. http://www.sciencedirect.com/science/article/pii/S2210970613000243.

Bešinović, N., Roberti, R., Quaglietta, E., Cacchiani, V., Toth, P., Goverde, R., 2015. Micro-macro approach to robust timetabling. In: Proceedings of the 6th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailTokyo2015, March 23–26, Tokyo, Japan, pp. 1–21.

Bourachot, J., 1986. Computer-aided planning of traffic in large stations by means of the AFAIG model . Rail International 2–18.

Burdett, R., Kozan, E., 2010. A disjunctive graph model and framework for constructing new train schedules. European Journal of Operational Research 200, 85–98. http://www.sciencedirect.com/science/article/pii/S0377221708010461.

Cacchiani, V., Caprara, A., Fischetti, M., 2012. A Lagrangian heuristic for robustness, with an application to train timetabling . Transportation Science 46 (1), 124–133. http://pubsonline.informs.org/doi/abs/10.1287/trsc.1110.0378.

Cacchiani, V., Caprara, A., Toth, P., 2010. Scheduling extra freight trains on railway networks. Transportation Research Part B 44 (2), 215–231. http://www.sciencedirect.com/science/article/pii/S0191261509000812.

Cacchiani, V., Toth, P., 2012. Nominal and robust train timetabling problems. European Journal of Operational Research 219, 727–737. http://www.sciencedirect.com/science/article/pii/S0377221711009908.

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

Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., Zenkhusen, R., 2011. A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. Transportation Science 45 (2), 212–227. http://pubsonline.informs.org/doi/abs/10.1287/trsc.1100.0349?journalCode=trsc.

Caimi, G., Hermann, T., Burkolter, D., Chudak, F., Laumanns, M., 2009. Design of a railway scheduling model for dense services. Networks and Spatial Economics 9 (1), 25–46. http://link.springer.com/article/10.1007%2Fs11067-008-9091-6#page-1.

Caimi, G., Herrmann, T., Burkolter, D., 2004. Finding delay-tolerant train routing through stations. In: Fleuren, H., den Hertog, D., Kort, P. (Eds.), Operations Research Proceedings. Springer, Heidelberg, pp. 136–143. http://link.springer.com/chapter/10.1007%2F3-540-27679-3_17.

Caprara, A., Galli, L., Toth, P., 2011. Solution of the train platforming problem. Transportation Science 45 (2), 246–257. http://pubsonline.informs.org/doi/abs/10.1287/trsc.1100.0366?journalCode=trsc.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., Toth, P., 2007. Passenger railway optimization, vol. 14. Handbooks in Operations Research and Management Science. Elsevier, Amsterdam.

Caprara, A., Kroon, L., Toth, P., 2011b. Optimization problems in passenger railway systems. In: Wiley Encyclopedia of Operations Research and Management Science, vol. 6. Wiley, pp. 3896–3905. http://onlinelibrary.wiley.com/doi/10.1002/9780470400531.eorms0647/abstract.

Carey, M., Carville, S., 2003. Scheduling and platforming trains at busy complex stations. Transportation Research Part A 37 (3), 195–224. http://www.sciencedirect.com/science/article/pii/S0965856402000125.

Daamen, W., Goverde, R., Hansen, I., 2009. Non-discriminatory automatic registration of knock-on train delays. Networks and Spatial Economics 9 (1), 47–61. http://link.springer.com/article/10.1007%2Fs11067-008-9087-2.

De Fabris, S., Longo, G., Medeossi, G., Pesenti, R., 2013. Application and validation of a timetabling algorithm to a large italian network. In: Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13–15, Copenhagen, Denmark.

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2011. Defining robustness of a railway timetable. In: Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011, February 16-18, Rome, Italy.

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2014. Improving the robustness in railway station areas.. European Journal of Operational Research 235, 276–286. http://www.sciencedirect.com/science/article/pii/S0377221713008916.

Fischetti, M., Monaci, M., 2009. Light robustness. In: Ahuja, R.K., Möhring, R.H., Zarogialis, C.D. (Eds.), Robust and Online Large-Scale Optimization, Models and Techniques for Transportation Systems. Springer, pp. 61–84. http://link.springer.com/chapter/10.1007%2F978-3-642-05465-5_3.

Ford, R., Haydock, D., 1992. Signalling and timetabling. In: Harris, N., Godward, E. (Eds.), Planning Passenger Railways: A Handbook. Transportation Publishing Company, Glossop, pp. 117–129.Chapter 11

Galli, L., Stiller, S., 2010. Strong formulations for the multi-module PESP and a quadratic algorithm for graphical diophantine equation systems. In: Proceedings of the 18th Annual European Conference on Algorithms: Part I, EAS'10, pp. 338–349. http://link.springer.com/chapter/10.1007%2F978-3-642-15775-2_29.

Goverde, R., 1998. Optimal scheduling of connections in railway systems. In: Paper presented at the 8th WTCR, Antwerp, Belgium. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.1393&rep=rep1&type=pdf.

Goverde, R., 1998b. The max-plus algebra approach to railway timetable design. In: Mellitt, B., Hill, R., Allan, J., Sciutto, G., Brebbia, C. (Eds.), Computers in Railways VI. WIT Press, Ashurst, pp. 339–350. http://www.witpress.com/elibrary/wit-transactions-on-the-built-environment/37/6813.

Goverde, R., 1999. Improving punctuality and transfer reliability by railway timetable optimization. In: Proceedings of the 5th TRAIL Annual Congress, TRAIL Research School, Delft, The Netherlands. http://repository.tudelft.nl/view/ir/uuid:e1728262-c398-499b-b6aa-e5147f28954b/.

Goverde, R., Hansen, I., 2000. TNV-Prepare: analysis of Dutch railway operations based on train detection data. In: Proceedings of International Conference on Computers in Railways, vol. VII, pp. 779–788. http://trid.trb.org/view.aspx?id=669220.

Großmann, P., 2011. Polynomial Reduction from PESP to SAT. Technical Report. TU Dresden. http://www.wv.inf.tu-dresden.de/Publications/2011/report11-05.pdf.

Großmann, P., Hölldobler, S., Manthey, N., Nachtigall, K., Opitz, J., Steinke, P., 2012. Solving periodic event scheduling problems with SAT. In: Jiang, H., Wei, D., Moonis, A., Xindong, W. (Eds.), Advanced Research in Applied Artificial Intelligence. Springer-Verlag, Berlin, Heidelberg, pp. 166–175. http://link.springer.com/chapter/10.1007%2F978-3-642-31087-4_18.

Holroyd, E.M., Scraggs, D.A., 1966. Waiting times for buses in Central London. Traffic Engineering Control 8, 158–160.

Huisman, D., Kroon, L., Lentink, R., Vromans, M., 2005. Operations research in passenger railway transportation. Statistica Neerlandica 59 (4), 467–497. http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9574.2005.00303.x/abstract.

Kroon, L., Dekker, R., Gábor, M., Helmrich, M.R., Vromans, M., 2006. Stochastic Improvement of Cyclic Railway Timetables. Technical Report. Erasmus Research Institute of Management (ERIM). http://www.sciencedirect.com/science/article/pii/S0191261507001324.

Kroon, L., Dekker, R., Vromans, M., 2007. Cyclic railway timetabling: a stochastic optimization approach. Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science. Springer Verlag, pp. 41–66. http://link.springer.com/chapter/10.1007%2F978-3-540-74247-0_2.

Kroon, L., Huisman, D., Abbink, E., Fioole, P.-J., Fischetti, M., Maróti, G., Schrijver, A., Ybema, R., 2009. The New Dutch timetable: the OR revolution. Interfaces 39, 6–17. http://dl.acm.org/citation.cfm?id=1536125.

Kümmling, M., Großmann, P., Nachtigall, K., Opitz, J., Weiß, R., 2013. The state-of-the-art realization of automatic railway timetable computation. In: Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013), December 2-4, Dresden, Germany, pp. 1–10. http://tudpress.de/TUDpressXTC2013/product_info.php?products_id=849.

Labermeier, H., 2013. On the dynamic of primary and secondary delay. In: Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark.

Lie, M., 2006. Optimization of the Timetable of the Dutch Railways. Vrije Universiteit Amsterdam (Master's thesis).

Liebchen, C., 2006. Periodic Timetable Optimization in Public Transport. Technische Universität Berlin (Ph.D. thesis).

Liebchen, C., 2007. Periodic timetable optimization in public transport. Operations Research Proceedings 2006, 29–36. http://link.springer.com/chapter/10.1007%2F978-3-540-69995-8_5.

Liebchen, C., Lübbecke, M., Möhring, R., Stiller, S., 2009. The concept of recoverable robustness, linear programming recovery, and railway applications. In: Ahuja, R.K., Möhring, R.H., Zarogialis, C.D. (Eds.), Robust and Online Large-Scale Optimization, Models and Techniques for Transportation Systems. Springer, pp. 1–27. http://link.springer.com/chapter/10.1007%2F978-3-642-05465-5_1.

Liu, S., Kozan, E., 2011. Scheduling trains with priorities: a no-wait blocking parallel- machine job-shop scheduling model. Transportation Science 45 (2), 175–198. http://pubsonline.informs.org/doi/abs/10.1287/trsc.1100.0332?journalCode=trsc.

Nachtigall, K., 1994. A Branch-and-cut Approach for Periodic Network Programming. Technical Report. Hildesheimer Infromatik-Berichte. http://www.opengrey.eu/item/display/10068/152968.

Nachtigall, K., 1996. Periodic network optimization with different arc frequencies. Discrete Applied Mathematics 69, 1–17. http://www.sciencedirect.com/science/article/pii/0166218X9500073Z.

Nachtigall, K., Opitz, J., 2008. Solving periodic timetable optimisation problems by modulo simplex calculations. In: Fischetti, M., Widmayer, P. (Eds.), In: Proceedings of the 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.509.4752.

Nachtigall, K., Voget, S., 1997. Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks. European Journal of Operational Research 103 (3), 610–627. http://www.sciencedirect.com/science/article/pii/S0377221796002846.

Odijk, M., 1996. A constraint generation algorithm for the construction of periodic railway timetables. Transportation Research Part B 30 (6), 455–464. http://www.sciencedirect.com/science/article/pii/0191261596000057.

Osuna, E.E., Newell, G.F., 1972. Control strategies for an idealized public transportation system. Transportation Science. 6 (1), 57–72.

Peeters, L., 2003. Cyclic Railway Timetable Optimization. Erasmus Resaerch Institute of Management (ERIM), Rotterdam (Ph.D. thesis). http://repub.eur.nl/pub/429/EPS2003022LIS_9058920429_PEETERS.pdf.

Schlechte, T., Borndoerfer, R., Erol, B., Graffagnino, T., Swarat, E., 2011. Micro–macro transformation of railway networks. Journal of Rail Transport Planning and Management 1 (1), 38–48. http://www.sciencedirect.com/science/article/pii/S2210970611000047.

Schöbel, A., Kratz, A., 2009. A bicriteria approach for robust timetabling. In: Ahuja, R.K., Möhring, R.H., Zarogialis, C.D. (Eds.), Robust and Online Large-Scale Optimization, Models and Techniques for Transportation Systems. Springer, pp. 119–144. http://link.springer.com/chapter/10.1007%2F978-3-642-05465-5_5.

Schrijver, A., Steenbeek, A., 1993. Spoorwegdienstregelingontwikkeling (Timetable Construction). Technical Report. CWI Center for Mathematics and Computer Science, Amsterdam. http://homepages.cwi.nl/~lex/files/rapp1a.ps.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2011a. Automated, passenger time optimal, robust timetabling, using mixed integer programming. In: Proceedings of the 1st International Workshop on High-Speed and Intercity Railways (IWHIR), Hong Kong, vol. 2, pp. 87–92. http://link.springer.com/chapter/10.1007%2F978-3-642-27963-8_9.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2011b. Deriving all passenger flows in a railway network from ticket sales data. In: Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011, February 16–18, Rome, Italy. http://www.LogicallyYours.com/RR2011.pdf.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2013a. A passenger knock-on delay model for timetable optimisation. In: Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013), December 2–4, Dresden, Germany, pp. 1–10. http://www.LogicallyYours.com/ITS2013.pdf.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2013b. Expected passenger travel time as objective function for train schedule optimization. In: Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13–15, Copenhagen, Denmark.. http://www.LogicallyYours.com/RC2013.pdf.

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. http://www.sciencedirect.com/science/article/pii/S0191261514000058.

Semet, Y., Schoenauer, M., 2005. An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling. Evolutionary Computation 3, 2752–2759. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1555040.

Serafini, P., Ukovich, W., 1989. A mathematical model for periodic scheduling problems. SIAM Journal on Discrete Mathematics 2, 550–581. http://epubs.siam.org/doi/abs/10.1137/0402049.

Sparing, D., Goverde, R.M., Hansen, I.A., 2013. An optimization model for simultaneous periodic timetable generation and stability analysis. In: Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13–15, Copenhagen, Denmark. http://repository.tudelft.nl/assets/uuid:56a64800-0dde-42fd-a2f1-05ed7c357b0b/293129.pdf.

Sun, L., Jin, J.G., Lee, D.H., Axhausen, K.W., Erath, A., 2014. Demand-driven timetable design for metro services. Transportation Research Part C 46, 284–299. http://www.sciencedirect.com/science/article/pii/S0968090X1400182X.

UIC, 2004. UIC CODE 406: Capacity. Technical Report 1. Union Internationale des Chemins de Fer (UIC). http://www.uic.org/.

Vansteenwegen, P., Van Oudheusden, D., 2006. Developing railway timetables that guarantee a better service. European Journal of Operational Research 173 (1), 337–350. http://www.sciencedirect.com/science/article/pii/S0377221705001360.

Vansteenwegen, P., Van Oudheusden, D., 2007. Decreasing the passenger waiting time for an intercity rail network. Transportation Research Part B 41 (4), 478–492. http://www.sciencedirect.com/science/article/pii/S0191261506001044.

Watson, R., 2008. Train Planning in a Fragmented Railway – A British Perspective. Loughborough University (Ph.D. thesis). https://dspace.lboro.ac.uk/dspace-jspui/bitstream/2134/4485/1/Train+Planning+in+a+Fragmented+Railway+Robert+Watson+PhD+November+2008.pdf.

Welding, P.I., 1957. The instability of close interval service.. Operational Research Society 8 (3), 133–148.
Wendler, E., 2007. The scheduled waiting time on railway lines. Transportation Research Part B 41 (1), 148–158. http://www.sciencedirect.com/science/article/pii/S019126150600018X.
Yuan, J., 2006. Stochastic Modelling of Train Delays and Delay Propagation in Stations. Technical University of Delft (Ph.D. thesis). http://repository.tudelft.nl/assets/uuid:caa72522-26b1-4088-afc0-59c6e5c346f6/trail_yuan_20061018.pdf.
Zhao, J., Frumin, M., Wilson, N., Zhao, Z., 2013. Unified estimator for excess journey time under heterogeneous passenger incidence behaviour using smart-card data. Transportation Research Part C 34, 70–88.
Zwaneveld, P.J., Kroon, L., Romeijn, H., Salomon, M., Dauzère-Perès, S., Van Hoesel, S.P.M., Ambergen, H.W., 1996. Routing trains through railway stations: model formulation and algorithms. Transportation Science 30 (1), 181–194. http://pubsonline.informs.org/doi/abs/10.1287/trsc.30.3.181?journalCode=trsc.

2.5 Towards a Better Train Timetable for Denmark Reducing Total Expected Passenger Time

Full paper presented at Conference on Advanced Systems in Public Transport (CASPT), Rotterdam, The Netherlands, 19-23 July 2015

The method and model as laid out in section 2.4 and applied there on the case of all 196 hourly passenger trains for Belgium is applied in this section to all 88 hourly passenger trains in Denmark. We also use the same passenger reflowing technique as described in section 2.1 and the same objective function as described in sections 2.2 and 2.3. Banedanmark also assumes the same macroscopic 3 minute rule of thumb for minimal headway time.

We only need to schedule half of the trains and the computation time to solve the model is also roughly halved. It goes down to 1 hour. The results also look similar to the Belgian case. We are again assuming 2% of the minimum times of ride, dwell and transfers for the average of the negative exponential primary delays distributions on these activities. Under this assumption, the optimised timetable reduces the expected passenger time by 2.90% compared to the original timetable made manually at Banedanmark. Under the same primary delay assumptions, the average probability of a missed transfer per passenger goes down from 11.34% in the original timetable to 2.45% in the optimised timetable. Again, the total time spent in practice in all transfers is smaller in the original timetable, even though the total expected journey time in practice is smaller in the optimised timetable.

This paper shows that our timetabling methodology is general enough to also be applied to other countries than Belgium, as long as they desire to construct a cyclical timetable with the objective of minimising the passenger travel time in practice.

Note that similarly to our earlier optimisation of the Belgian timetable, the excess journey time, which is related to spreading of alternative trains, was not controlled by the objective function yet. This is the topic of a later paper in section 2.6.

CASPT 2015

Towards a Better Train Timetable for Denmark

Reducing Total Expected Passenger Time

Peter Sels · Katrine Meisch · Tove Møller · Jens Parbo · Thijs Dewilde · Dirk Cattrysse · Pieter Vansteenwegen

Abstract With our Periodic Event Scheduling Problem (PESP) based timetabling method we are able to produce a passenger robust timetable for all 88 hourly passenger trains running on tracks managed by the Danish Infrastructure Manager Banedanmark. The objective function of our model is the total expected passenger journey time in practice and is minimised. The result of this is that the produced timetable reduces the expected journey time of all corresponding train passengers together by 2.9% compared to the original timetable defined by Banedanmark. Our simulations show that the average probability of missing a transfer is also reduced from 11.34% to 2.45%. The computation of this timetable takes only 65 minutes. The major innovations of our approach are the addition of a complete objective function to the PESP model and the addition of a particular cycle constraint set that reduces computation times. In this paper, we demonstrate that these combined innovations result in a method that quickly generates cyclic timetables for a train network spanning an entire country and that these timetables also reduce the expected passenger travel time in practice.

Keywords Expected Passenger Time · Integer Linear Programming · Optimal Cyclic Railway Timetabling · Periodic Event Scheduling Problem

P. Sels
Logically Yours BVBA
Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium
Tel.: +32-486-95 67 97, E-mail: sels.peter@gmail..com
Infrabel, Traffic Management & Services
Fonsnylaan 13, 1060 Brussels, Belgium

K. Meisch, T. Møller
Banedanmark, Amerika Plads 15, 2100 København, Denmark

J. Parbo
DTU Transport, Bygningstorvet 116B, 2800 Kgs. Lyngby, Denmark

P. Sels, T. Dewilde, D. Cattrysse, P. Vansteenwegen
KU Leuven, Leuven Mobility Research Centre, CIB
Celestijnenlaan 300, 3001 Leuven, Belgium

1 Introduction

This paper's topic is the automatic construction of a cyclic, macroscopic railway timetable. The word cyclic means that there is a timetable period, here 1 hour, by which every train repeats itself. The word macroscopic means that a standard value for the minimum headway times of 3 minutes is assumed and inside stations, the microscopic headway constraints that arise from the block sections staircase model are not enforced. We also assume that line planning is fixed including the halting pattern for each line. This means that for each train, for each station, only the arrival and departure time are to be determined. In other words, only ride and dwell supplements are to be chosen. Of course, many solutions exists, but these supplements have to be chosen so that the resulting timetable possesses some desirable proporties. We previously constructed a Periodic Event Scheduling Problem (PESP) based model which has as objective function: the total expected passenger journey time in practice over all passengers (Sels et al, 2015b). In Dewilde et al (2013), the authors conclude that, unlike to what is the case for some alternative definitions of robustness, this objective function is a *practical* method to obtain robustness and that the obtained robustness is ideal for passengers. Our objective function integrates and makes a trade-off between efficiency and robustness. It penalises supplements that are so big that they would lower efficiency too much but also penalise supplements that are so small that robustness would be compromised.

In Sels et al (2015b), this MILP model is generated for the set of all 196 hourly trains in Belgium. The main results were that a timetable, automatically generated in about 2 hours, saves about 3.8% of total expected passenger journey time. This timetable also significantly reduced the percentage of missed transfers from 13.9% to 2.6%. To study how generally applicable this model is to practice, we now also test it on the set of all 88 hourly trains using Banedanmark's infrastructure.

2 Timetabling Methodology and Assumptions

Our timetabling approach consists of the basic constraints of the popular *PESP* model (Serafini and Ukovich, 1989; Schrijver and Steenbeek, 1993; Nachtigall, 1996; Goverde, 1998a,b; Peeters, 2003; Kroon et al, 2007; Liebchen, 2007; Kroon et al, 2009; Caprara et al, 2011; Sparing et al, 2013) using a standard *event activity network*. We impose its classic *constraints* enforcing minimal ride times and minimal dwell times. As described in detail in Sels et al (2011), we automatically construct all *potential transfers*. By this, we mean that if two trains stop in the same station, a transfer edge will be added between the arrival time of the feeder train and the departure time of the target train. Currently, a minimum of 3 minutes is assumed for each transfer. Headway edges and the respective minimum headway time constraints are also automatically constructed between entry times of each pair of trains that enter the same infrastructure resource and similarly also between all pairs of exit times. For single track sections, between each leaving and each entering train, a similar headway time constraint is imposed. The headway minimum time assumed on this macroscopic level is 3 minutes. This summarises all hard constraints in our model. For more details, we refer to Sels et al (2015b), where all these mandatory constraints and some supplementary ones that are merely intended to speed up computation are discussed.

We will also only give a qualitative description of our objective function here, as the main focus of this paper is the application of our timetabling model on the Danish train network. As derived formally in detail in Sels et al (2013b) and Sels et al (2013a), our *objective function* consists of the sum of the expected passenger time for each edge (action) in the event activity graph $G(V, E)$ that corresponds to a passenger activity. So, for each ride, dwell and transfer edge we model an expected passenger time. We express this expected passenger time of an edge as a function of its minimum time and its added *supplement time*. The *shape* of this function mainly depends on the expected primary delay distribution and consequently, so does the value of the supplement that should be ideally added. The *scale* of this function depends on the number of passengers involved. This indicates the relative importance of the expected passenger time of one edge compared to that of another and these are balanced by the objective function.

For the primary delays, as do Schwanhäußer (1974); Meng (1991); Ferreira and Higgins (1996); Goverde (1998a); Vansteenwegen and Van Oudheusden (2006); Kroon et al (2006) and Yuan (2006), we assume negative exponential distributions. These distributions have an average (=expected value) that can be set to a certain fixed percentage 'a' of the minimum time for that action. This average can in theory be determined by inspecting logs of trains as they are running in the current timetable. This has been described by Goverde and Hansen (2000) and Daamen et al (2009) for the Dutch and by Labermeier (2013) for the Swiss infrastructure. So, for example, if the minimum time of a ride action is 5 minutes from one stop to the next, if one sets 'a' to 10%, the average primary delay on that ride action is assumed to be 0.5 minutes. By this one parameter, the negative exponential distribution $p(d)$ of the primary delay d is unambiguously defined, as $p(d) = exp(-d/a)/a$. For now, we assume the same value of 'a' for all ride, dwell and transfer edges, for all trains and for all tracks. The value of 'a' is typically chosen in the range of 1% to 5% (Goverde, 1998a).

Depending on the action type that passengers participate in, the expected passenger time is another type of function of the supplements added to these actions. We now discuss these types of passengers and associate cost functions.

For *through* passengers, experiencing a ride and subsequent dwell action, the expected time, as a function of the added ride and dwell supplements s, as can be seen in the example in figure 1, is almost the function $f(s) = P \cdot s$, with P equal to the number of participating passengers. This is logical, since for whatever supplement is added to a ride or dwell action, the through passengers just have to sit it through. So high values of s are not beneficial to

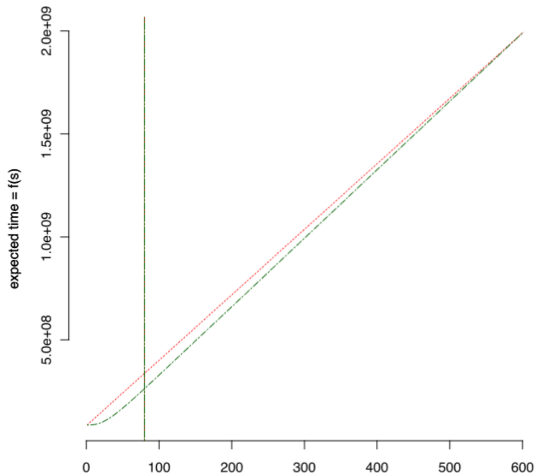


Fig. 1 Through and arriving passenger expected time as a function of the chosen supplement. All time is given in in 6 second multiples.

these passengers. At low values of s, the slope of $f(s)$ is a little flatter because small delays occur more often than large delays and so, waiting for the end of s takes a smaller fraction of time s on average than for larger supplements. The larger the supplement, the smaller the fraction that common delay sizes form compared to it. So for larger supplements this secondary 'curving effect' diminishes. The situation is entirely similar for arriving passengers, experiencing a ride plus sink action, and so the cost function for arriving passengers is also similar to the one shown in figure 1. Note that the green vertical line shows that an 8 minute supplement was chosen by the solver. A supplement equal to 0 minutes would be locally optimal, but other hard constraints like headway constraints may forbid this here.

Note that all cost functions in figures 1, 2, 3 and 4 show an actual expected time cost function in green that is used in evaluation and a piecewise linear approximation of it in red which is used in linear optimisation. The green vertical line indicates an example of an actual chosen supplement. Its associated expected passenger time cost can then also easily be read from the graph. In each case, we see that the linearisation error is relatively small.

To *departing* passengers, experiencing a source plus ride action, it is beneficial when the train they get onto departs as scheduled. This is ensured by providing enough time buffers against primary delay on this train on the sections this train traverses *before* these departing passengers embark on it. The curve in figure 2 shows indeed that the selection of a larger buffer on the previous sections for this train statistically leads to lower expected delay for departing passengers than a lower buffer. However, it also demonstrates that a supplement larger than 10 minutes does not significantly increase the buffer-

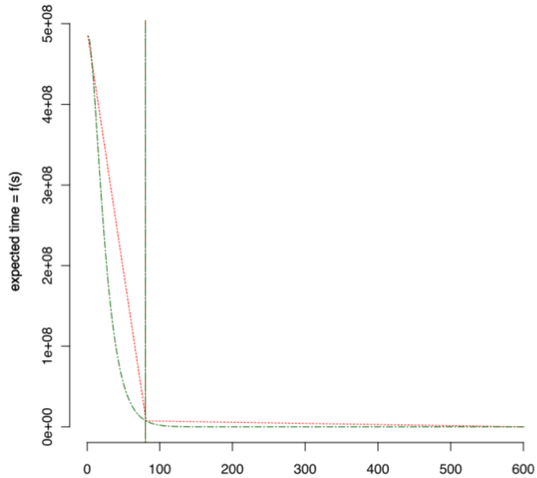


Fig. 2 Departing passenger expected time as a function of the chosen supplement. All time is given in in 6 second multiples

ing effect compared to a 10 minute supplement. The green vertical line shows that the MIP solver decided to set the supplement to 8 minutes. This is not the local minimum, 60 minutes, but due to competition with other terms in the objective function this could be a reasonable choice. The value 8 minutes is the absis of the crossing of the two red segments which meet on the green curve so the linearisation error is 0 here.

For passengers who are *changing* between trains, experiencing a ride plus transfer action, we model an expected transfer time that depends on the chosen supplement for this transfer, on top of the minimum of 3 minutes. If the supplement is low, the probability that the transfer is missed is high. If the transfer is missed, we conservatively assume a penalty waiting time of the timetable period, here 1 hour. If the supplement is high, the probability of missing the transfer is low, but the transfer passenger will always have to wait until the supplement time has elapsed. The above means that the expected passenger time for a transfer is a U-shaped function of the supplement. An example of a transfer cost curve is given in figure 3. So there is a trade-off and a locally optimal value for the transfer supplement somewhere between 0 and 60 minutes. This supplement range is very broad and naturally very large supplements will rarely be added. Exceptionally, like when a transfer is only taken by very few people, and a small supplement on this transfer would mean a large supplement on an action with more people, a very large supplement on this less important transfer can occur though. The allowed range for supplements is defined as 0 to 60 minutes to avoid infeasibility problems. Note that a transferring passenger can be seen as the combination of both an arriving and a departing passenger and this is reflected in the cost function in figure 3 being

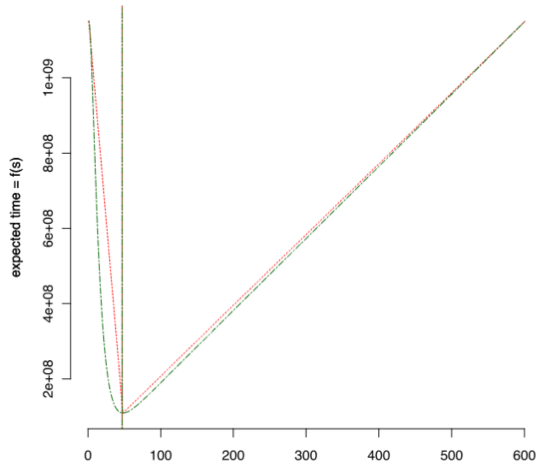


Fig. 3 Transfer passenger expected time as a function of the chosen supplement. All time is given in in 6 second multiples

the addition of the cost functions of figures 1 and 2. The vertical green line in figure 3 shows that the MIP solver was able to select a supplement equal to 4.5 minutes which minimises the local linearised expected transfer time. This also coincides with the minimum of the green curve.

As for secondary delays, or *knock-on delays*, our model already contains the graph edges associated to these. Indeed, they are the same edges as the headway edges, temporally separating pairs of trains that use the same infrastructure resource. So for each headway edge, we also add a term in the objective function that represents the knock-on time or secondary delay that passengers on the second train may experience in case the first train is delayed. In our model, as derived in Sels et al (2013a), this time depends on the delay distributions of both trains and on the number of passengers on the second train. Obviously, the total knock-on time is proportional to the number of passengers on the second train. Also, the expected knock-on passenger time forms a decreasing function of the train separating supplement $s_{i,j}$, since the higher the time separation between two trains i and j, the lower the expected knock-on delay. Figure 4 shows an example of a knock-on delay cost function. The horizontal axis shows the supplement between 0 and 60 minutes and on the vertical axis the expected knock-on time is given. Note that our MIP model optimises over all possible train orders. This means that when N trains use a common resource, for all train pairs, cyclically, $N(N-1)$ knock-on terms are added to the objective function. Knock-on costs are a major determinant for the optimal train orders, but major transfers will also play a role in this.

We could also consider the expected waiting time that passengers experience at their station of departure. This depends on the spreading between alternative trains in the timetable. In this paper we did not add these terms

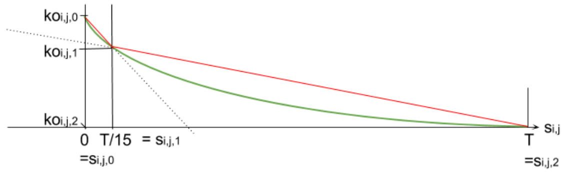


Fig. 4 Shape of expected knock-on delay as a function of the chosen supplement. T is the timetable period, which is 60 minutes here. The vertical axis has no specific scale here.

to the objective function since our model developed to estimate this expected time does not scale well yet to networks with many trains (Sels et al, 2015a).

All types of objective function time terms described are seen as *objective time*. No subjective weights are added. This concludes our qualitative discussion of the objective function of our PESP MILP model representing the timetabling problem. In the next section, we apply our model to the train network of all passenger trains in Denmark and show the results.

3 Application to the Danish Railway System

Our complete method first constructs an event activity graph representing the train service network. Then, we route passengers over this graph to derive local passenger flow numbers for every ride, dwell and transfer action in this graph. We subsequently reschedule trains, deriving ideal arrival and departure times for all trains in all stations. We report results for each of these three phases.

3.1 Constructing the Event Activity Graph

For this project, Banedanmark started from the infrastructure they manage. This is 1956km or 79.5% of the the total of 2636km of railway track in Denmark. These tracks are visualised in figure 5. Subsequently, for an 'average' Wednesday in 2013, all trains running on this infrastructure were collected and slightly adapted, so that the timetable became exactly periodical with one hour. One representative hour for this network contains 84 passenger trains and 4 freight trains. Note that we do not schedule the suburban trains on the infrastructure of S-bane. The S-bane operates in the København area and is completely independent of the rest of the network, so it has no effect on our case. Some private operators run trains that briefly also use the Banedanmark infrastructure in just three places. These trains have not been modelled but are expected to have little influence on our main results. Freight trains were defined in the input only on sections where Banedanmark knows that there is a capacity bottleneck. For other sections, no freight trains were defined. It is assumed that they can be fitted between the scheduled passenger trains later.

We then generated the event activity network that corresponds to this service. This graph contains 88 trains, 264 stations, 3346 vertices and 9918 edges. The number of ride edges is 1541. Table 1 shows more problem instance statistics for this Danish event activity network.

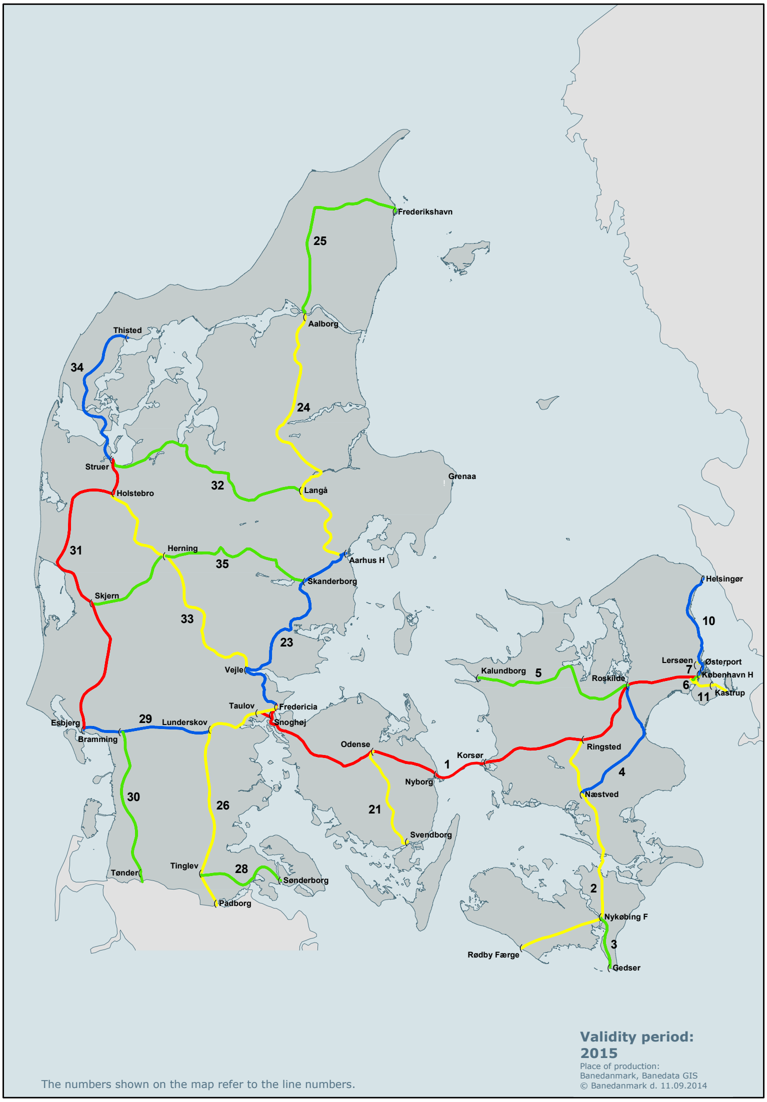


Fig. 5 Danish train infrastructure lines managed by Banedanmark

3.2 Routing: Reflowing

Now that the basic service graph is constructed, we mimmick the process were passengers decide what train to take if they go from an origin station (O) to a destination station (D). The number of commuters per day is 394377.

Table 1 Graph and timetable MIP problem instance statistics

# ride edges =	1533
# dwell edges =	1445
# turn-around edges =	0
# knock-on(headway) edges =	13596
# major transfer edges =	4908
# model rows =	47335
# model columns =	32057
# model non-zero elements =	140516
# objective function terms for major flows =	16652
# objective function terms in post-optimisation evaluation =	21522

The morning peak OD matrix of these commuters is used to route passengers over this train service network, according to the routing algorithm described in Sels et al (2011). This is a modified Dijkstra algorithm implemented in C++. For efficiency, the modified Dijkstra algorithm was parallellised both on the core-level (using openMP, 2013) and the machine-level (using openMPI, 2014). For every OD-pair in the OD matrix, the best routings from O to D are calculated independently. First the modified Dijkstra algorithm is run to find the route with the lowest planned time, based only on the sum of minima for its ride and dwell actions. To avoid too many transfers in a route we penalise the choice of a transfer with 15 minutes. Note that the actual duration of a transfer is not known yet at this point. Next, all edges forming this route are eliminated from the graph and a new route search is performed. This route finding process is repeated until the new found route takes more than 20% more time than the first route found. At this point, it is assumed that no passengers will still opt for such a slower route. Passengers for a specific OD-pair are then distributed over the different OD-routes found, where more are assigned to the shorter routes than to the longer routes. Note that in our method, routing passengers comes before timetabling. This means that arrival and departure times are still unknown and so is their spreading out across one timetable hour. We simplify by assuming that these factors play no role in the passenger distribution over different routes for a given OD-pair (Jolliffe and Hutchingson, 1975). This assumption will be more realistic with good termporal spreading than with bad temporal spreading of alternative trains (Sels et al, 2015a). After the routing phase, which is parallellised for all OD-pairs, a non-parallellised merging phase, for each action (ride, dwell, transfer) on each link of the network is performed. Passenger numbers from the different OD-streams passing along an action are accumulated. We obtain the passenger number for every action (edge) in the event activity graph. Note that the freight trains in our system start in a technical station that passengers do not have access to. The freight trains also do not halt nor stop in passenger stations and so, in our routing algorithm, no passengers can get on or off these trains, as is the case in practice. This means that in our timetabling model, a freight train is treated like a passenger train with no passengers on, so it will be of lower priority during scheduling. If one wants a higher importance, one could assign a virtual number of passengers to each freight train.

The results from the full passenger routing phase, accumulated per track section, are given graphically in figure 6. In this figure, the area of each circle

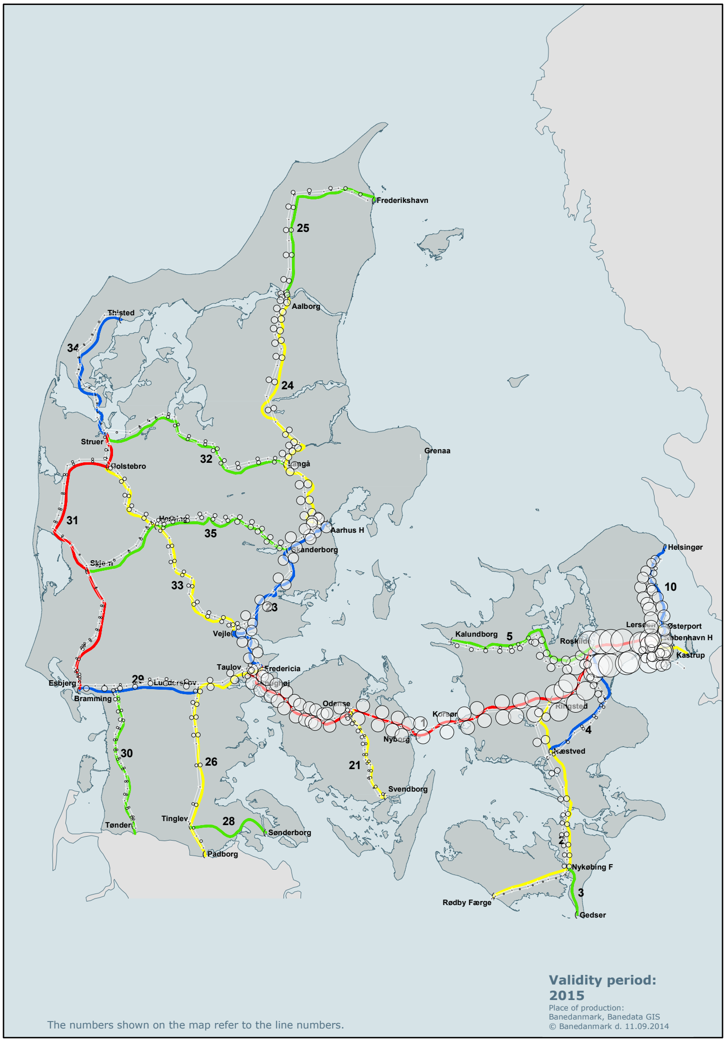


Fig. 6 Passenger flows in Denmark for a typical Wednesday morning peak

incident to a track section is proportional to the number of people traveling on trains that travel along that track section. It is clear that the set of trains in the area around København transport the most passengers. All trains together going from Høje-Taastrup to Hedehusene, carry 29215 passengers in the morning peak. This is the maximum flow present in the graph. The second highest passenger flows occur on the tracks from København westwards to Odense and back and also from Fredericia North to Århus and back. It can be seen that the collected trains for other track sections in the rest of Denmark each transport a lot less passengers.

3.3 Scheduling: Retiming

Now that we know the number of passengers for each ride, dwell, transfer and knock-on action, we perform timetabling, according to the methodology described in section 2. We use the obtained local passenger numbers as fixed weights in the objective function.

4 Results

With different parameter settings, different MILP timetabling models were constructed. With each model, we construct a different timetable. Our software has a solver independent architecture, using the open source library milp-logic (Sels, 2012). This way, a simple solver setting and recompilation allows the software to call any solver supported by milp-logic. Currently, these are CPLEX, Gurobi, XPRESS. In this paper, we restrict ourselves to reporting of results with Gurobi. Each of our timetabling models was tackled by the MILP solver Gurobi version 6.0.0 on an Intel Xeon E31240 3.3GHz processor with 16GB of RAM. When constructing and optimising a MIP model, we noticed that computation times were sensitive to the amount of passenger flows we consider in the objective function. When all streams are considered, computation time becomes excessive so we defined a threshold of number of passengers. Streams with fewer passengers than this threshold are not considered in the objective function. The threshold of 210 passengers per morning peak gave manageable computation times. A further parameter is the required MIP gap. Setting this to 74% resulted in schedules with a lower total expected passenger time than the original schedule. Gap values lower than 74% result in better schedules but computation time also rises. For these parameter values 210 and 74% we get an optimised timetable. This is the timetable we report results for in sections 4.1 and 4.2. Section 4.1 describes that for this optimised timetable, there are no minimum headway time violations. Section 4.2 shows that large time supplements can be and are here assigned to train actions where no passengers are expected. Section 4.3 shows that for various parameter settings, the total passenger time in practice that is expected for the resulting optimised timetables, is always reduced compared to the current timetable.

4.1 No Collisions nor Headway Violations

The current and optimised timetables were verified by Banedanmark by visual inspection of space-time graphs per infrastructure line. Some examples of these graphs are given as figures 7, 8, 9 and 10.

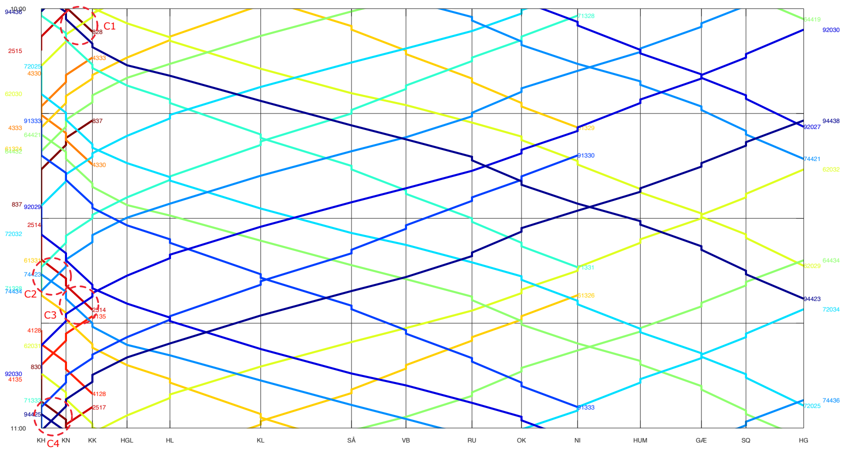


Fig. 7 Space time graph for the original timetable for line 10

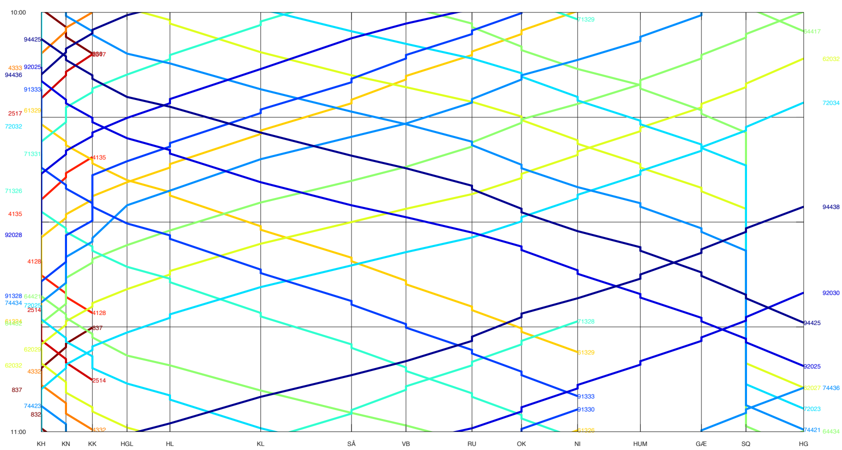


Fig. 8 Space time graph for the optimised timetable for line 10

Figure 7 shows the space-time diagram of trains running on the train infrastructure line 10 between København (KH) and Helsingør (HG) and back for the original timetable. Figure 8 shows the same trains but now for the optimised timetable. In figure 7 it can be seen that the original table generally leaves the required 3 or more minutes between each couple of subsequent trains except for 4 cases between København (KH) and Østerport (KK) and back as indicated by the red dashed circles C1 to C4. Indeed, in circles C1 and C4, train 828 (brown) and train 94423 (dark blue) only have a headway time of 2 instead of 3 minutes between them. The same happens between train 2514 (dark red) and train 74423 (semi-light blue) in circles C2 and C3. In the optimised timetable in figure 8, it can be seen that no such violations of the minimal headway time constraints of 3 minutes occur.

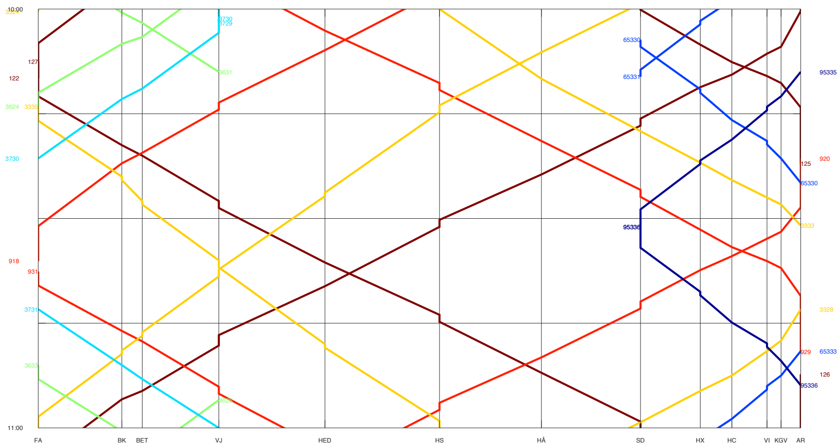


Fig. 9 Space time graph for original timetable for line 23

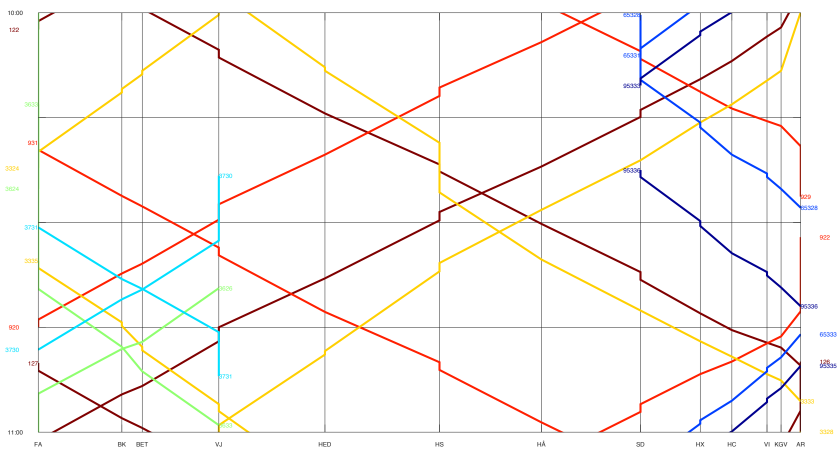


Fig. 10 Space time graph for the optimised timetable for line 23

Figures 9 and 10 show the space-time diagram of trains running on the train infrastructure line 23 between Fredericia (FA) and a Århus (AR) and back, respectively for the original and the optimised timetable. One can verify that on this line, for both timetables, no single train collision nor violation of minimal headway time constraints occurs. For all other infrastructure lines, similar graphs were generated and verified as well and as such Banedanmark declared the optimised timetable as free of headway conflicts.

4.2 Large Dwell Times on Line 10 Explained

Figure 8 shows that, at the station Snekkersten (SQ), 4 trains heading for Helsingør (HG) are assigned large dwell times. These trains are (64421 (medium green), 72025 (light blue), 62029 (yellow-green) and 74423 (semi-light blue)) This is caused by the fact that our routing phase resulted in no passengers between Snekkersten (SQ) and Helsingør (HS). This can be seen in figure 6, where no white circle occurs between Snekkersten and Helsingør. This also means that these dwell times are not penalised in our objective function of our timetabling model. They can become arbitrarily large without having an effect on any passengers indeed.

Furthermore, it should be noted that our current timetable is only ideal for passengers traveling in the morning. Since one usually wants a timetable that is the same for morning and evening, one can express that by supplying an OD matrix that contains both morning and evening OD-pairs together. If then, all ride and dwell actions of all trains will have at least some passengers on them, in both directions, none of these dwell or ride times will stay unaccounted for in the objective function of our timetabling model. As such, all these actions will also have sensible supplements assigned to them.

Also note that in Snekkersten (SQ), in practice, there are not enough platform tracks in the station to allow simultaneous dwelling of 4 trains. Our timetabling model does indeed not take microscopic issues like this into account. Again, when some passengers would be assigned to these dwell actions, shorter dwell times will result and with that the number of simultaneously dwelling trains will most likely be significantly reduced.

4.3 Reduced Expected Passenger Time

By construction, our optimised timetables contain no single violation of hard (minimum run time, minimum dwell time, minimum headway time) constraints. For headway times this was illustrated in the previous sections graphically. In this section, we show that the optimised timetable also results in lower expected passenger time in practice than the original timetable. The relevant results are shown in table 2. Each of our timetabling models was tackled by the MILP solver Gurobi version 6.0.0 on an Intel Xeon E31240 3.3GHz processor with 16GB of RAM. Results for the different optimisations and their respective input parameter values are ordered from less to more demanding from top to bottom. By more demanding, we mean that either the required MILP gap (column 3) is lower or the number of transfers considered in the optimisation is higher or a combination of both. The transfer threshold (column 2) is the number of people that are required as minimum for a transfer

Table 2 Results for different timetable optimisations of all 88 hourly Danish trains. req. = required, obt.=obtained, exp. time = expected *passenger* time, red. = reduction, eval. = evaluation, orig.tt = original timetable, opt.tt = optimised timetable, rd. + dw. t = ride + dwell *train* time.

1	2	3	4	5	6	7	8	9
a	transfer	gap	gap	solver	exp. time	missed transfers:		planned
	threshold	req.	obt.	time	red.eval.	orig.tt	opt.tt	rd.+dw. t
(%)		(%)	(%)	(s)	(%)	(%)	(%)	red.(%)
2	420	75	74.92	19421	1.67	11.34	2.07	-4.88
2	420	74	73.63	62417	1.96	11.34	5.21	-3.95
2	210	79	78.07	1534	0.82	11.34	2.83	-7.77
2	210	77	76.62	2436	1.59	11.34	3.20	-4.89
2	210	75	74.96	2924	2.45	11.34	1.12	-3.08
2	210	74	73.83	**3922**	**2.90**	11.34	**2.45**	-2.53
2	210	73	72.96	20726	3.16	11.34	2.07	-2.05
2	195	76	≥76.8	≥101000				

to be considered in the optimisation. Column 6 shows the reduction in percent from original to optimised timetable of the expected time as evaluated over all streams, also the ones with fewer people than the threshold value. Column 7 shows the missed transfer probability in the original timetable as simulated over all streams and column 8 shows the same for the optimised timetable. Column 9 shows the reduction in percent of the planned ride and dwell supplements from the current to the optimised timetable.

We see that setting the transfer threshold to 420 makes that the solver spends a lot of time (19421 and 62417 seconds) before it finds a solution with an optimality gap below the required one. When the transfer threshold is lowered to 210 transfer passengers, resulting in more transfers considered in the optimisation, the model seems to become easier for Gurobi. When subsequently also lowering the required gap from 79% to 74% (column 3), timetable solutions are found within 1534 to 3922 seconds (column 5) and corresponding savings of total expected passenger time increase from 0.82% to 2.90% (column 6). Lowering the required gap further to 73% still improves the solution with a total reduction of expected passenger time of 3.16%, however, the computation time then increases significantly to 20726 seconds, being 5.76 hours. To test if lowering the transfer threshold further below 210 reduces computation time, we investigate whether a threshold of 195 combined with a not so demanding required gap of 76% gives us a good timetable quickly. The last line of table 2 shows that after 101000 seconds, no acceptable timetable solution was found yet, since the solver is still at a gap of 76.8%. So the value 210 as a transfer threshold somehow seems a good trade-off between giving Gurobi enough information about a good timetable and not too many terms in the objective function.

For the timetable that reduces the passenger time by 2.90% compared to the original one, we show the expected passenger time and its components graphically in figure 11. This figure stacks expected time components on top

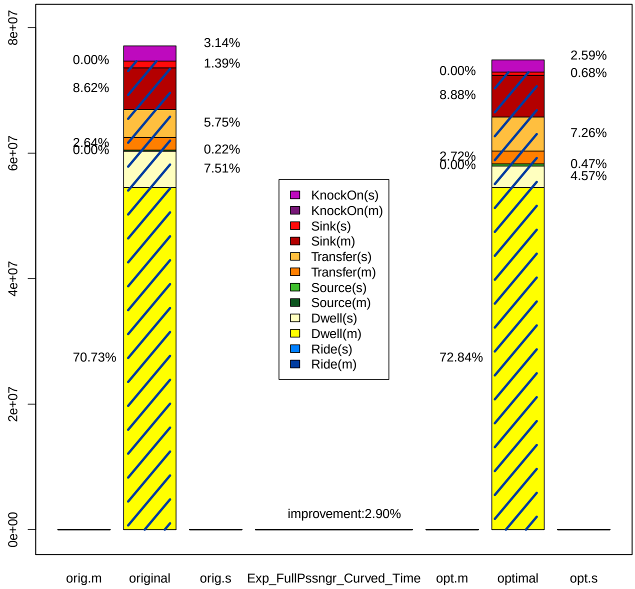


Fig. 11 Reduction of expected passenger time of 2.90% compared to the original timetable.

of each other to reveal the total expected time for all passenger streams, large and small, for this optimised timetable. Expected time components can indeed be added together since all of them are expressed in the same units: (tenths of) passenger minutes. In figure 11, the left bar indicates the original timetable (orig) and the right bar indicates the optimised timetable (opt). The vertical dimension represents expected passenger time, also for its constituent components: ride (blue), dwell (yellow), transfer (orange), knock-on (purple). For dwell and transfer time, all ride time of the ride action preceding it, is convoluted with it, which is what the blue shading refers to. On the left of each bar, the percentages (orig.m and opt.m) indicate the ratio of the total *expected* passenger time part, that can be seen as the consequence of the planned minima (m), to its total bar height. Note that this part is equivalent to the *planned* passenger minimum time. On the right, the percentages (orig.s and opt.s) indicate the ratio of the total *expected* passenger time part, that can be seen as the consequence of the planned supplements (s), to its total bar height. This part is equivalent to the difference of the total *expected* time minus the total *planned* passenger minimum time. For each color, the minima are shown in a darker tone of the color and the supplements in a lighter tone of the same color. Figure 11 shows clearly that the obtained reduction of total expected time of 2.9% is caused by the net effect of three main changes. First, the amount of time spent in supplements on ride and dwell actions is significantly lowered from 7.51% to 4.57%. Second, the expected knock-on delay time is reduced from 3.14% to 2.59% of the total expected time. Third, the expected transfer time is increased from 5.75% to 7.26% of the total expected time. In absolute terms, the transfer time increase is smaller than the sum of decreases in expected time spent in ride and dwell supplements and in knock-on events. This means the net result is a reduction in total expected passenger time.

We go back to table 2. For the best timetables found, its last column mentions that these possess between 3.08% and 2.05% more *train weighted planned* ride and dwell time than the original timetable. Even then, the total passenger time is reduced. This is possible due to a number of factors. Firstly, our method adds supplements to trains but weighs them by passengers. Secondly, supplements can cause extra robustness, so adding planned time can reduce experienced time in practice. Thirdly, classical manual timetabling uses rules of thumb like assigning a certain percentage of supplement *to each train*. To avoid knock-on delays, we expect these rules to perform worse than our rule of assigning supplements *between each couple of trains* sharing an infrastructure resource, even more so since we do this proportionally with the number of passengers on the second train and dependent on the expected delay distributions of both trains.

Table 2 also mentions that the expected missed transfer probability for all passenger streams together, both large and small, is 11.34% (column 7) for the original timetable while not more than 2.45% (1.12%, 2.45% and 2.07%, column 8) for our best three timetables. This is clearly a significant improvement that will be appreciated by the railway passengers. These results were obtained by a post optimisation calculation on the obtained timetables, for all passengers streams, small and large, where expected delays are accumulated and resulting in fractions of missed and non-missed transfers. For the original timetable the percentage is always the same, 11.34%, since the value of the transfer threshold plays no role is the missed transfer calculations. Indeed all passenger streams are considered here and not only streams with more than the number of passengers indicated by the transfer threshold.

4.4 Further Verification

Further verification of realistic parameter settings like the value of 'a' and the value of transfer minima is warranted for fair comparison with the current timetable. Also verification of other timetable quality criteria like the possible preference of some operators to avoid large inserted supplements, even for actions with very few passengers, is required and ongoing.

5 Conclusion

This paper demonstrates that our PESP based method with an objective function representing total expected passenger time in practice, improves the timetable for the whole train network of Banedanmark. Total passenger time in practice can be reduced by 2.9% and the average probability of missing a transfer is reduced from 11.34% to 2.45%. The fact that, after our successful application to the Belgian train network, the application to a second country now delivers satisfying results as well, indicates that our approach is quite generally useful.

Thanks to the addition of a particular set of cycle constraints to the PESP model (Sels et al, 2015b), computation time stays limited to 65 minutes. This could lead to huge time savings in the current timetabling practice which, for the biggest part, is still carried out manually. Alternatively, the time spent on manual timetabling now, can instead be used to create more alternative line planning proposals which can be fed to our timetabling system. The line plan leading to the optimised timetable with the lowest total expected passenger time can then be selected. This would further improve passenger service.

6 Further Work

Even though the total expected passenger time of our optimised timetable is lower than the one for the original timetable, the total expected transfer time component of our optimised timetable increased. It would be interesting to see if our model could be adapted so that this expected transfer component is reduced while still also reducing the total expected passenger time.

Some degree of temporal spreading of alternative trains between origin and destination is beneficial to reduce the inter-departure waiting time for passenger travelling between these points. Also considering this inter-departure waiting time at the origin and inter-arrival-time at the destination would avoid potential bunching of trains and further generalise our method.

We now produce a timetable that respects headway time minima of 3 minutes everywhere in the network, which is the most common headway minimum value for macroscopic railway models. On a microscopic level, the actually needed headways can be derived from the blocking model (Hansen and Pachl, 2014) and depend on parameters like station infrastructure, train speed and train length. Per train pair, per station, the required minimum headway between these train pairs for that station can be calculated and these values can be substituted for the 3 minute macroscopic headway minima. When our method is used with these more accurate headway minimum values as input, a microscopically feasible timetable will result.

Acknowledgements We thank Banedanmark for supplying us with the input data and verifying the output as well as Infrabel for letting us use their software on the Danish data.

References

(2013) OpenMP Application Programmer Interface v4.0. URL `http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf`

(2014) The OpenMPI API specification for MPI. URL `http://www.open-mpi.org/doc/`

Caprara A, Kroon L, Toth P (2011) Optimization Problems in Passenger Railway Systems. Wiley Encyclopedia of Operations Research and Management Science 6:3896–3905

Daamen W, Goverde R, Hansen I (2009) Non-Discriminatory Automatic Registration of Knock-On Train Delays. Networks and Spatial Economics 9(1):47–61

Dewilde T, Sels P, Cattrysse D, Vansteenwegen P (2013) Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming. Journal of Railway Transport Planning & Management (Young Railway Operations Research Award 2013 of IAROR) 3:68–77

Ferreira L, Higgins A (1996) Modeling Reliability of Train Arrival Times. Journal of Transportation Engineering 22(6):414–420

Goverde R (1998a) Optimal Scheduling of Connections in Railway Systems. Paper presented at the 8th WTCR, Antwerp, Belgium

Goverde R (1998b) Synchronization Control of Scheduled Train Services to minimize Passenger Waiting Times. Proceedings of 4th TRAIL Annual Congress, TRAIL Research School, Delft, The Netherlands

Goverde R, Hansen I (2000) TNV-Prepare: Analysis of Dutch Railway Operations Based on Train Detection Data. International conference on computers in railways VII:779–788

Hansen I, Pachl J (eds) (2014) Railway Timetabling and Operations, 2nd edn. Eurailpress

Jolliffe JK, Hutchingson TP (1975) A Behavioral Explanation of the Association between Bus and Passenger Arrivals at a Bus Stop. Transportation Science 9:248–282

Kroon L, Dekker R, Gábor M, Helmrich MR, Vromans M (2006) Stochastic Improvement of Cyclic Railway Timetables. Tech. rep., Erasmus Research Institute of Management (ERIM)

Kroon L, Dekker R, Vromans M (2007) Cyclic Railway Timetabling: A Stochastic Optimization Approach. Algorithmic Methods for Railway Optimization Lecture Notes in Computer Science pp 41–66

Kroon L, Huisman D, Abbink E, Fioole PJ, Fischetti M, Maróti G, Schrijver A, Ybema R (2009) The New Dutch Timetable: The OR Revolution. Interfaces 39:6–17

Labermeier H (2013) On the Dynamic of Primary and Secondary Delay. Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark

Liebchen C (2007) Periodic Timetable Optimization in Public Transport. Operations Research Proceedings 2006:29–36

Meng Y (1991) Bemessung von Pufferzeiten in Anschlüssen von Reisezügen. Veröffentlichungen des Verkehrswissenschaftlichen Institutes der Rheinischen-Westfälischen Technischen Hochschule Aachen 46:29–36

Nachtigall K (1996) Periodic network optimization with different arc frequencies. Discrete Applied Mathematics 69:1–17

2.6 Optimal Temporal Spreading of Alternative Trains in order to Minimise Passenger Travel Time in Practice

Full paper presented at International Association of Railway Operations Research (IAROR), Tokyo, Japan 23-26 March 2015

Our objective function of total expected passenger time, as developped in sections 2.2 and 2.3. and as used in the optimisations for Belgium and Denmark in sections 2.4 and 2.5, contained all passenger travel time from departure to arrival. However, it did not contain the time that passengers spend before departure, waiting for their train and after arrival, waiting for their next mode of transport or before a visit or meeting actually starts. This time, together, is called excess journey time. We now model this time and add it to our objective function, which already contained journey time. As such, our model will now minimise the total travel time, being the sum of journey time and this new excess journey time. This has the beneficial effect of spreading in time of the different trains between an origin and destination. Indeed, this is the case, because this lowers the expected inter-departure time as well as the inter-arrival time. We investigate this way of spreading in time of these alternative trains via the objective function. The function of excess journey time is well published in previous literature. However, we believe it is the first time it has been used in a timetable optimisation context. The challenge in the optimisation context was to model this function of times between train departure or arrival times when the order of the trains is still unknown. We managed to do this.

We also consider alternative ways of spreading via hard constraints on the origin as well as destination sides of OD-pairs, but this leads to 'over-stretching' of these planned journeys containing too many time supplements. We then tried hard spreading on the origin and soft spreading, via the objective function, on the destination side. This method had the strongest total expected time reduction of all three methods. An additional noted and peculiar benefit was that the total expected transfer times component was almost not growing anymore compared to the original timetable. Remember that our optimisations for Belgium and Denmark in sections 2.4 and 2.5 showed that this transfer component grew from original to optimised timetable, even though the total expected time was already shrinking by several percentages. So we conclude that the hard-soft technique for spreading is the preferable technique.

Our experiments work well in terms of computation times up to a set of 26 trains. However, beyond that, model resolution times still become impractically high. So this aspect requires further study. Possibly fixing some orders of alternative trains can reduce this tractability problem without affecting optimality too much since the trains are considered alternatives and so alike in timing and halting pattern. Another possible remedy could be to construct cycle constraints along ride and dwell edges of these alternative trains.

Optimal Temporal Spreading of Alternative Trains in order to Minimise Passenger Travel Time in Practice

Peter Sels [a,b,c,1], Thijs Dewilde [a], Dirk Cattrysse [a], Pieter Vansteenwegen [a]
[a] KU Leuven, Leuven Mobility Research Centre, CIB,
Celestijnenlaan 300, 3001 Leuven, Belgium
[1] E-mail: sels.peter@gmail.com, Phone: +32 (0) 486 95 67 97
[b] Logically Yours BVBA, Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium
[c] Infrabel, Traffic Management & Services, Fonsnylaan 13, 1060 Brussels, Belgium

Abstract
In an earlier paper on timetabling, we derived a stochastic objective function for what we consider an ideal timetable: minimal expected passenger travel time in practice. This includes primary delay distributions and a degree of robustness against these via a knock-on delay model. This timetable is ideal for passengers who know the timetable in advance and plan their departure and/or arrival times accordingly. In general, it is considered as a good service to passengers that the different alternatives to get from an origin to a destination are equally spread in time. This is even more important for passengers who are not informed about the timetable or who cannot adapt their time of arrival in the station, for instance because they want to go home as soon as possible after a meeting. Spreading the trains decreases the average waiting time at departure and/or waiting time at arrival. In this paper, we add these terms to the objective function to also provide the advantage of low waiting times before and after the routes in the resulting timetable. We properly balance benefits of spreading the trains with the other objectives included in the objective function. The model solves quickly for a set of 26 trains and 12 sets of alternative routes but still proves to be difficult to solve for bigger instances.

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

1 Introduction

It is generally understood, also by train passengers, that in a train service network with several alternative trains connecting an origin station with a destination station, spreading of these alternative trains in time is beneficial for train passengers travelling between these stations. Assuming, for a certain portion of all passengers, a random arrival pattern, it decreases the total waiting time at departure and symmetrically, also at arrival. The mathematical expression for these expected waiting times as a function of the planned heading times at departure (or arrival) between these alternative trains is well known and widely published and used in transport practice. In this paper, this expression is added to the objective function of our Periodic Event Scheduling Problem (PESP) based Mixed Integer Linear Programming (MILP) model (Sels et al., 2013b, 2014). When solving this model, the objective function will be minimised. When doing so, the expected waiting time at departure and arrival are minimised together with the other components that were already present in the objective function: expected ride time, dwell time, transfer time and knock-on time. The result is that we now minimise not only the total *journey time*, being in vehicle plus transfer time, but additionally also the *excess journey time*, being inter-departure plus inter-arrival waiting time (Zhao et al., 2013). Their sum, *actual or observed journey time* (Zhao et al., 2013), is now the new total objective function. This results in improved timetables, as the objective function approximates better the real expected time that passengers experience in practice. Indeed, *excess journey time* is a real expected time component for passengers and should not be forgotten.

Section 2 gives an overview of research into waiting time and alternative approaches to realising temporal spreading during timetable construction. We will conclude that our approach is a novel one. Section 3 formally derives the linear constraints and objective function terms that will be added to our MILP model later. In the process, some statistics on the typical number of alternative trains is given for the network of all Belgian passenger trains. Section 3.4 then replaces the constant departure (or arrival) times to variable ones. Indeed, in the PESP model begin (or end) times of activities are unknown still. As a consequence, some terms become quadratic and must be linearised. Section 4 discusses the successful application on a network consisting of 26 Belgian IC trains. We first limit the objective function to excess journey time without transfer time, then add transfers to the model, then add more OD-pairs for temporal spreading to the model and then attempt to tackle train service networks with more trains. The section shows, as is expected, that there is a noticeable competition between the minimisation of the different time components, especially between excess journey time and transfer time. Section 5 concludes and hints at further work that could potentially improve computation times.

2 Related Research Overview and Comparison

Research concerning the advantages of temporal spreading of alternative trains has been dominated by the analysis of how passengers choose their trains, when they choose to arrive at their station of departure and how this affects their expected inter-departure waiting time. The published results are described in section 2.1. Section 2.2 discusses which subcategories of passengers display random arrival behaviour. While the mentioned research is very interesting, it is very noticeable how inter-*arrival* waiting time has been left mostly unmentioned. We discuss this in section 2.3. Also, less attention has been paid to how this waiting time should be incorporated in models to construct timetables with the desirable property of temporal spreading. The traditional method of timetabling that intends to realise temporal spreading and a newer method are mentioned in section 2.4. Section 2.5 gives a quick overview of the rationale of our novel method.

2.1 Models of Expected Inter-Departure Waiting Time

The most simplistic model of passengers arriving in their station of departure is one where it is assumed that a constant number of passengers f arrives during every unit of time. This implicitly assumes that passengers are not adapting their arrival time to the knowledge of the train departure times as planned in the timetable. This model is called the *random arrival* model. This model has been assumed for studying transit reliability by de Pirey (1971), Barnett (1974), Bly and Jackson (1974) and Friedman (1976).

Welding (1957), Holroyd and Scraggs (1966) and Osuna and Newell (1972) derived that, for passengers who are arriving randomly at their station of departure, the expected waiting time until departure $E(w)$ can be expressed as a function of the *average vehicle planned heading time* $E(h)$ (over all heading times H_i in the cyclic timetable period T) and the *variation coefficient* $C_v(h)$ of the real time heading times:

$$E(w) = E(h)/2 \cdot \left(1 + C_v(h)^2\right). \tag{1}$$

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

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

where p_i is the probability of a passenger experiencing heading time H_i in period T. $C_v(h) = \sigma(h)/\mu(h)$ is the ratio of the standard deviation over the mean, both of the heading time distribution in real time. Substitution of the right hand side of equation (2) for $E(h)$ in equation (1) and multiplication by the number of considered randomly arriving passengers f delivers

$$E(f \cdot w) = \frac{f}{2T} \sum_{i=0}^{N-1} H_i^2 \cdot \left(1 + C_v(h)^2\right). \tag{3}$$

The general objective is to minimise the expected waiting time at arrival, $E(w)$, in equation (1). This can be done by either minimising $E(h)$ or $C_v(h)^2$ or a combination of both. Note that, for equal temporal spreading between N vehicles in time T, it holds that $E(h) = H = T/N$ and that otherwise $E(h) > H = T/N$. The second can be understood, from equation (2) and realising that the probability p_i equals H_i/T, so the probability to experience a higher heading time during time T is higher than the probability to experience a lower one. So, when supposing there are no real time delays, $E(h)$ is minimised by equal spreading of headways in the timetable if the amount of vehicles N per period T is fixed or otherwise by increasing the number of vehicles N. The value of $C_v(h)$ is decreased by lowering $\sigma(h)/\mu(h)$, representing the reduction of relative deviations from the timetable headways. This is done by better control of the timing of vehicles. Islam and Vandebona (2010) mention that $C_v(h)$ is often used as a measure for reliability. They study bus systems where *planned* heading times are all equal. In that case $E(h) = H$ and, assuming a fixed amount of vehicles, decreasing $E(w)$ can then only be obtained by reducing $C_v(h)$ by making the timing of the system more reliable. For a train system, often one must, like when having to insert a freight train between a set of N passenger trains, plan unequal heading times H_i. In that case, even in the planning $E(h) > H$ occurs. This summarises the effect of the number of alternative vehicles N, planned vehicle spreading in time $E(w)$ and reliability $C_v(h)$ on passenger waiting time for randomly arriving passengers. In this paper we will try to reduce $E(w)$ by reduction of $E(h)$, with fixed number of trains N, so by the use of optimal spreading only. Reducing $C_v(h)$ is useful but cannot be done via the timetable.

Later, studies appeared where the more complex behaviour of passengers, adapting their arrival time at their station of departure to their chosen vehicle planned departure time is studied (Okrent, 1971; Jolliffe and Hutchingson, 1975; Jackson, 1977; Turnquist, 1978).

Jolliffe and Hutchingson (1975) subdivide passengers in 3 categories. Some fraction q is supposed to manage to arrive coincidental with the vehicle arrival time and are assumed to have no waiting time. The remaining fraction $(1-q)$ of passengers is split into a proportion $(1-q)p$, who arrive so as to minimise expected waiting time and a proportion $(1-q)(1-p)$ arrive randomly.

In this paper, we want to optimise the heading times of all trains a passenger going from a station O to station D can choose, so that the corresponding waiting time stays low. This will be achieved by minimisation of the total journey time of which this waiting time is a component. The effect on passengers arriving randomly is described by equation (1). We suppose they come in a portion r. So their total waiting time is $r \cdot E(w)$. The primary effect on passengers waiting time of passengers that shift their arrival times with time δt when their planned vehicle departure time is shifted in a new timetable with δt is zero. This means that the fixed waiting time of the portion $(1-r)$ of passengers needs not and so will not occur in the objective function waiting time at departure terms of our timetable optimisation model. When evaluating the total waiting time, their fixed waiting time x can possibly be added as fixed amounts, $(1-r) * x$, which is indeed independent of the timetable.

Bowman and Turnquist (1981) studied passenger arrival patterns at bus stops at seven locations around Chicago. They write that expected passenger waiting time at a transit stop is dependent on three things: the distribution of *passenger arrival times* at that stop, the *planned* arrival times of the *vehicles* at that stop and the deviations in practice from the arrival times of these vehicles. They use a *passenger choice model* that is fitted to measurements of passenger arrival times at bus stops. They study the effect of both decreasing headway times (by adding more vehicles) and decreasing the standard deviation on vehicle arrival times (by increasing reliability in operations) on expected waiting times. They conclude that both can decrease expected waiting time both for the random arrival model and for the passenger choice model. Bowman and Turnquist (1981) suggest that higher heading times will lead to fewer random arrival passengers, since they will want to avoid the higher waiting penalty time. This indicates that our r should in fact be dependent on the number of vehicles N from O to D per timetable period. This is indeed what Fan and Mechemehl (2002) also report from their data analysis. They identified a 10 minute vehicle headway as the transition from random to non-random passenger arrivals.

More recently, the terms *excess journey time (EJT)* and *passenger incidence behaviour* have been added to the transport research vocabulary (Furth and Muller, 2006; Frumin and Zhao, 2012; Zhao et al., 2013). EJT is the difference between the journey time as implied by the published timetable and the actual journey time (Zhao et al., 2013). The waiting time before the first vehicle departs from the moment the passenger arrives at station O and the waiting time after the last vehicle arrives before the passenger leaves station D are the two components making up this excess journey time. The second component can also be seen as inter-departure time of the next transportation system the passenger moves to after the one considered. *Passenger incidence behaviour* is the generic name for how passengers interact with the transportation system (Furth and Muller, 2006). Information about the transportation system can influence how they make choices and use the transportation system. The above mentioned random arrival model and passenger choice model are two different passenger incidence behaviours.

The research mentioned above is mainly focussed on accurately measuring in reality and then fitting and modelling passenger and vehicle arrival distributions and the resulting expected passenger waiting time. However, we are not aware of any research also incorporating any of these models in a timetabling method that automatically minimises passenger waiting time at departure or arrival. This is our focus in this paper.

2.2 Random Arrival Passenger Subcategories

Jolliffe and Hutchingson (1975) mention that the fraction of passengers who are *uninformed or unaware* about timetable departure times of their alternative trains, especially during peak hours, is low. That may be true for some networks, but they are not the only passengers that show random arrival behaviour. Consider table 1. We will traverse it from left to right.

<table>
<tr><td rowspan="6">informed</td><td rowspan="5">caring</td><td rowspan="4">adaptable</td><td rowspan="4">adapting</td><td rowspan="2">in time</td><td>for departure</td><td rowspan="2">$(1-r)$</td><td rowspan="2">passenger choice model</td></tr>
<tr><td>for transfer</td></tr>
<tr><td rowspan="2">over time</td><td>for departure</td><td rowspan="5">r</td><td rowspan="5">random arrival model</td></tr>
<tr><td>for transfer</td></tr>
<tr><td>unadaptable</td><td colspan="3" rowspan="3">not adapting</td></tr>
<tr><td colspan="2">not-caring</td></tr>
<tr><td>uninformed</td><td colspan="2"></td></tr>
</table>

Table 1: Sub-categories of passengers: fraction (1-r) showing passenger choice model behaviour and fraction r showing random arrival behaviour.

Amongst the *informed* passengers , there are some that do not care or don't want to be too stressed out about getting the first possible train departure. They are informed but are *not adapting* their arrival time to the train departure times available in the timetable. Amongst the passengers who do care, some *cannot* adapt, because they have time constrained obligations before arriving at the train platform. For example they have to bring their kids to the creche at a certain fixed time. We categorise them as *unadaptable*. Amongst the *caring*, *adaptable* passengers, all will be *adapting* but only some will be successful and *in time* for the aimed train departure (at origin or transfer station) and some will be unsuccessful and *over time* for the train departure (again at origin or transfer station). Only the *in time* passengers, with fraction $(1-r)$ should be considered to behave according to the passenger choice model. For this fraction, temporal spreading of alternative trains will not decrease their expected waiting time at departure. The rest, fraction r, behave according to the random arrival model. For them, spreading of alternative trains will decrease their expected waiting time at departure and so is useful. We do not claim that the vertical dimensions of the rows in table 1 are scaled proportionally to the ratios in reality. However, consider that there are quite some types of passengers making up the ratio r. The table can be used to derive the value of r if one can estimate the splitting fractions, at each step, going from left to right through the table.

2.3 Expected Inter-Arrival Waiting Time

Most research focussing on excess journey time mentions inter-departure waiting time but does not mention inter-arrival waiting time. Temporal spreading at train departure is inspired by the idea of randomly arriving passengers, which is the best assumption one can make if no more detailed information is known about passenger incidence behaviour. Similarly, spreading at the arrival side of the transportation system considered is desirable because one wants to minimise waiting time for the transportation system coming after the system a timetable is being constructed for. So at the arrival side, we suppose a waiting time occurs, according to the same model as for the waiting time occurring at the departure side.

2.4 Approaches to Realise Temporal Spreading of Trains

Within the PESP approach to timetabling (Serafini and Ukovich, 1989; Schrijver, 1993; Nachtigall, 1996; Goverde, 1998a,b), the usual approach to enforce constant heading times between trains along the same path is the use of *regularity constraints* (Peeters, 2003; Kroon et al., 2007; Liebchen, 2006, 2007; Caprara et al., 2007; Kroon et al., 2009; Caprara et al., 2011; Sparing et al., 2013). A regularity arc in the PESP model connects two departure times of alternative trains and imposes a fixed heading difference (modulo the timetable period T). We think that sometimes deviation from this optimal spreading should be allowed since minimisation of other expected time components is also important. Secondly, the enforcement of spreading is intended for the portion r of passengers that arrive randomly at their departure station. The portion $(1 - r)$ of people that adapt their arrival time to a particular train departure time have no primary beneficial effect. Hard (regularity) constraints cannot differentiate between two passenger categories while soft spreading can do this via the objective function. Thirdly, in this approach, these regularity constraints are usually only applied to trains which have exactly the same path. So routes which contain transfers do not occur. However this is not a restriction imposed by PESP. We think what matters is that *all alternative train services* leading from O to D should, together, be imposed upon *some degree* of temporal spreading.

When many groups of alternative trains exist, all trains within a group are repetitions of each other and one ends up with a problem called *multi-module PESP* (Galli and Stiller, 2010). For these systems they developed a specific formulation. They claim that the powerful PESP based methods developed for uniform modules generally fail for the multi-module case and demonstrate that their approach helps reduce computation time.

2.5 Minimising Excess Journey Time to Realise Optimal Temporal Spreading of Trains

We think we can formulate the benefit of temporal spreading of alternative trains in terms of the waiting time it saves for passengers. If we minimise the expected waiting time at departure and arrival, together with journey time, the resulting timetable will possess the spreading that is optimal to passengers, not more, not less.

Setting up an integer linear model that achieves minimisation of waiting time, both at departure and at arrival, in the expected passenger domain and applying it on as many trains as possible are the topics of this paper. To the best of our knowledge, no research has been carried out in this direction before.

3 Cost of Waiting for the Next Alternative Train

In section 3.1 the constraints for imposing optimal spreading are derived. Section 3.2 derives the corresponding objective function terms and proves that these will be minimal at equal spreading. Section 3.3 shows that the number of alternative trains stays low in practice.

3.1 Derivation of Basic Constraints, Solving the Unknown Order Problem

Sometimes, a train passenger can choose between multiple train routes that lead him from his origin station O to his destination station D. Some of the alternative routes will consist of a different train or trains visiting the same intermediate stations and some may be composed of different trains that also visit different intermediate stations. Some connections between O and D may require a transfer and some may not. In the sequel, we will simply talk about *alternative trains* or *alternative routes* when we mean the collection of all possibilities to go from O to D using any combination of trains possible. To be able to optimise and also evaluate any train schedule concerning the cost of waiting for alternative routes, we derive this cost analytically here. This cost can then be added to the objective function described by Sels et al. (2013b).

Let $R_{O,D}$ be the set of all route alternatives from station O to D. Say that from O to D, we have a total of $\#R_{O,D} = N$ routes. We define their index set as $I_N = \{0, 1, ..., N-1\}$. We choose the couple of different routes r_i and r_j, where $i \in I_N$ and $j \in I_N$ and $i \neq j$. Let the variables b_i and b_j be these routes' respective planned begin times modulo T in station O. We now choose to develop our derivation for the begin sides of routes, but the derivation for the end sides is entirely similar. To be able to define the times in between two *subsequent* begin times, we need to know the *cyclic order* of these begin times. Since, during timetable optimisation, this is still to be determined in the model, we define a vector $\bar{b}$, with the same N values as vector b, but then sorted in non-decreasing order. Formally, the non-decreasing order is enforced by

$$\forall(O,D): \begin{cases} \forall i \in I_N \setminus \{N-1\}: & \bar{b}_i \leq \bar{b}_{i+1} \\ & \bar{b}_{N-1} \geq \bar{b}_0. \end{cases} \tag{4}$$

Since we need the same scalar values in the vectors $\bar{b}$ and b, we declare that $\bar{b}$ is a permutation, defined by the permutation matrix $p_{i,j}$,of b by imposing

$$\forall(O,D): \forall i \in I_N: \begin{cases} \bar{b}_i = \sum_{j \in I_N} p_{i,j} \cdot b_j \\ \forall j \in I_N : p_{i,j} \in \{0,1\} \\ \sum_{j \in I_N} p_{i,j} = 1 = \sum_{j \in I_N} p_{j,i}. \end{cases} \tag{5}$$

The expected time of waiting for the next departure from O to D will be a function of the time differences in between subsequent $\bar{b}_i, \bar{b}_{i+1}$ values. We call these delta times *supplements* s_i and define them by

$$\forall(O,D): \begin{cases} \forall i \in I_N \setminus \{N-1\} & : & s_i & = & \bar{b}_{i+1} - \bar{b}_i \\ & & s_{N-1} & = & (\bar{b}_0 + T) - \bar{b}_{N-1} \\ \forall i \in I_N & : & 0 & \leq & s_i \leq T - \delta. \end{cases} \tag{6}$$

δ is the smallest time difference in our model, so our time resolution. We used 6 seconds for δ and 1 hour for T, so $T = 600\delta$. The range of s_i from 0 up to, but not including T, is sufficient to allow equation (6) to always have feasible solutions. Indeed, $\forall(i,j) \in I_n : 0 \leq b_i \leq T - \delta$ and so for the non-decreasingly ordered $\bar{b}_i$ it holds that $\forall i \in I_N \setminus \{N-1\} : 0 \leq \bar{b}_{i+1} - \bar{b}_i \leq T - \delta$. This also allows the case $s_i = 0$. Note that from equation (6), it follows that

$$\forall(O,D): \sum_{i \in I_N} s_i = T. \tag{7}$$

Even though the equations (7) are linearly dependent of the equations (6), for computational reasons, we also impose them in our model.

We now defined all necessary variables to be able to derive the expected waiting time until first departure in O for D. Note that, in the equations (4), (5) and (6), for notation simplicity we left out the indices O, D for $N, \bar{b}_i, p_{i,j}, b_i$ and s_i and we will also do so in the sequel.

3.2 Derivation of Objective Function Terms representing the Expected Excess Journey Time

The accumulative function of the number of randomly arriving passengers, with assumed rate f per hour, wanting to go from O to D and randomly arriving in O is a sawtooth function. From the time $\overline{b}_i$ up to the time $\overline{b}_{i+1} = \overline{b}_i + s_i$, the number of randomly arriving passengers at O has accumulated from 0 up to $f \cdot s_i$, at which time they all take the train departing on route r_j. The total average time they have to wait altogether for their route departure is given by the number of additional people arriving during time dt at origin station O, namely $f \cdot dt$, multiplied by the time they will have to wait $(s_i - t)$, and this integrated over t ranging from 0 to s_i. This gives

$$u_i \quad = \quad \int_0^{s_i} (s_i - t) \cdot f \; dt = s_i f \int_0^{s_i} dt - f \int_0^{s_i} t \; dt = f \frac{s_i^2}{2}. \tag{8}$$

This integration results in the surface of a rectangular triangle from the sawtooth function, with base s_i and height $f \cdot s_i$. The total expected waiting time for a randomly arriving passenger arriving at O and wanting to go to D before he has taken his first train is

$$\forall (O, D) : U = \sum_{i \in I_N} u_i = f/2 \cdot \sum_{i \in I_N} s_i^2. \tag{9}$$

Note the equivalence of equation (9) with the equation (3) from earlier literature. Indeed $s_i = H_i$ and $f = F/T$. One can prove that the function U in equation (9) is minimal when $\forall i \in I_N : s_i = T/N$. Indeed, using a Lagrange multiplier for constraint (7) that has to be satisfied for all solutions, one gets that the following should be minimised

$$\sum_{i \in I_N} s_i^2 - \lambda \cdot \left(T - \sum_{i \in I_N} s_i \right). \tag{10}$$

So the partial derivatives to all s_i and to λ should all be zero. This means

$$\forall (O, D) : \left\{ \begin{array}{lllll} \forall_{i \in I_N} & : & 2 s_i + \lambda & = & 0 \\ & : & T - \sum_{i \in I_N} s_i & = & 0. \end{array} \right. \tag{11}$$

From the top half of equation (11), it follows that all s_i are equal and from the bottom part, that they are all equal to T/N. Each second derivative to s_i equals 2 which is positive, so $s_i = T/N$ gives a minimum and not a maximum. □

The $\overline{b}_i$ variables in equations (6) will, when integrated in our complete model, also be connected (via the $p_{i,j}$ permutation matrix variables) to the b_i variables. In our complete model, the b_i variables and also the supplement variables s_i between them also occur in other constraints and objective function terms. These hard and soft constraints may influence the optimal choice of the s_i here. As a consequence, the optimal solution where all s_i are equal, found when nothing apart from equations (4), (5), (6) constrains $\overline{b}_i$, may not be the one that is also found when integrating it in our complete model.

Equation (9) is useable for *evaluation* of a given schedule, but for *linear* optimisation, a function that is linear in the model variables s_i is needed. We will solve this issue here by linearisation of the u_i terms. Every term $u_i = f \cdot s_i^2/2$, can be approximated by a piecewise linear function composed of 2 segments, by sampling the curve $(s_i, f \cdot s_i^2/2)$ in 3 points. Since the a priori optimal spreading of begin times for N routes in a time T is at equidistant intervals of time $\frac{T}{N}$, as proven, we take this delta time as one of the three sample values for s_i. This will lead to a high accuracy approximation of the real cost function in the part of the range of s_i that delivers the most optimal solutions. The other necessary sample points are its lower bound 0 and upper bound T. So, the resulting 3 points are

$$\forall(O,D) : \forall i \in I_N : \begin{cases} (s_{i,0}, u_{i,0}) = (0,0) \\ (s_{i,1}, u_{i,1}) = \left(\frac{T}{N}, \frac{f}{2}\left(\frac{T}{N}\right)^2\right) \\ (s_{i,2}, u_{i,2}) = \left(T, \frac{f}{2}T^2\right). \end{cases} \tag{12}$$

Equation (12) describes a piecewise linear convex function which can be implemented in a linear programming formulation by addition of a linear inequality for each of the two subsequent segments, forming a convex (s_i, u_i) search space together, as follows

$$\forall(O,D) : \forall i \in I_N : \begin{cases} u_i & \geq & u_{i,0} + \frac{u_{i,1}-u_{i,0}}{s_{i,1}-s_{i,0}} \cdot (s_i - s_{i,0}) \\ & = & 0 + \frac{\frac{fT^2}{2N^2}}{\frac{T}{N}}(s_i - 0) = \frac{fT}{2N}s_i \\ u_i & \geq & u_{i,1} + \frac{u_{i,2}-u_{i,1}}{s_{i,2}-s_{i,1}} \cdot (s_i - s_{i,1}) \\ & = & \frac{fT^2}{2N^2} + \frac{\frac{fT^2}{2} - \frac{fT^2}{2N^2}}{T - \frac{T}{N}}(s_i - \frac{T}{N}) \\ & = & \frac{fT^2}{2N^2} + \frac{N}{(N-1)\cdot T}\left(\frac{fT^2N^2 - fT^2}{2N^2}\right)(s_i - \frac{T}{N}) \\ & = & \frac{fT^2}{2N^2}\left[1 + \frac{N(N+1)}{T}(s_i - \frac{T}{N})\right]. \end{cases} \tag{13}$$

Since the units of f, T, N and s_i are respectively $1/time$, $time$, 1 and $time$, the right hand sides of (13) are indeed two costs in units of $time$. The fact that our total objective function is minimised rather than maximised, guarantees that the (s_i, u_i)-points resulting from the model solution will lie *on*, rather than *above* the two line segment function. It is important to realise that the values f, T, N are constant (manifest) to the model, so the piecewise linear functions in s_i can be calculated as known (linear) functions of only the model variable s_i at model setup time.

We have now converted the cost function of second degree in (9) into the necessary linear inequalities (13). Thanks to minimisation of our objective function which contains the terms u_i for every OD-pair, and the enforcement of equation (5) and (6), a non-decreasing order for $\bar{b}_i$ will be selected and as a consequence, the $\geq$-signs in (13) will turn into $=$-signs. This will deliver us the correct, be it linearly approximated cost, instead of only a - possibly weak - upper bound of it.

3.3 N is mostly Small in Practice

We have derived a model extension that takes care of spreading concerns. This extension generates equations and costs when ranging over $i \in I_{O,D}$. For each (O,D)-pair, this range contains $N(N-1)$ elements. The question then is if this N does not get impractically large.

In practice, passengers choose between different routes leading them from their origin station O to their destination station D. Our route choice algorithm, as described in Sels et al. (2011), tries to mimic this behaviour by considering the fastest route. Suppose that this route has length l. We also consider all routes up to length $l(1+a)$, where $a = 20\%$. A transfer is penalised by attributing a time of $p = 15$ minutes to it. We estimate that these values for a and p are realistic average for passengers. van der Hurk et al. (2014) use exactly the same graph for route generation, which they call *extended network* and for one algorithm, they call STA, also consider trip duration only and also penalise transfers. They could validate routes with a route set that passengers take in reality and report that STA is able to generate 95% of the routes correctly. We can expect that our method in Sels et al. (2011), if validated, would reach similar percentages of realistic routes and so that it uses enough relevant information as input.

In our algorithm, we do not consider departure times nor their current spreading in time, since they are still unknown in the timetable to be computed. We believe that taking this information from the current timetable would lead to too much bias to the current, possibly suboptimal, departure times or spreading. To avoid any bias for passengers to prefer routes with particular departure hours, we suppose an a-priori distribution of departure times as well as passengers that is ideal in that sense. This means we suppose both passengers and departure hours to be uniformly spread in the hour.

Train filling levels which passengers may also consider in practice, play no role in our algorithm. Typically, vehicle assignment and decisions on the numbers of cars per train is indeed done after timetabling.

In practice, when routing all passengers through the graph of all Belgian passenger trains, we experience that the number N of alternative routes per OD-pair is usually low. Figures 1 up to 3 represent histograms indicating per number N of found routes for an OD-pair, the number of times that that N occurs over all 18268 OD-pairs derived from ticket sales data in Belgium. Figure 1 shows this for $a = 10\%$, figure 2 for $a = 20\%$ and figure 3 for $a = 30\%$. Naturally, when a is increased, for a particular OD-pair, the number N of found routes cannot decrease. So statistically, when considering all routes, we expect that a higher a leads to a higher average N. The figures 1 to 3 confirm this.

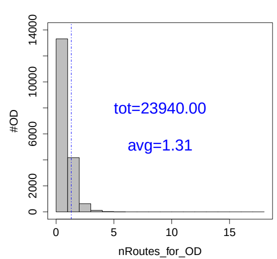


Figure 1: $a = 10\%$

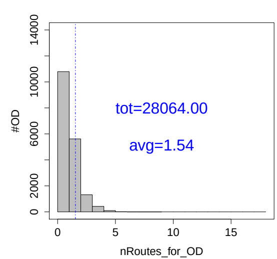


Figure 2: $a = 20\%$

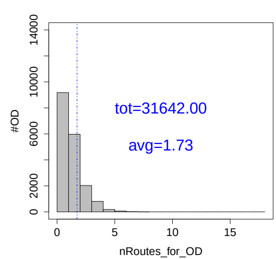


Figure 3: $a = 30\%$

For $a = 10\%$, $a = 20\%$ and $a = 30\%$ respectively only 4, 4 and 6 OD-pairs occur that have a number N between 18 and 32.

The cases with around 32 possibilities for the same OD pair occur between stations around Brussels, where that many trains per hour with equal or similar journey time from O to D do indeed occur. So the number of supplements $N \cdot (N - 1)$, introduced in equations (6), will rarely get big, nor will the same number of boolean variables $p_{i,j}$ in the equations (5) or the number of added inequalities or couples of added inequalities in (13).

3.4 Linearising $p_{i,j} \cdot b_j$

The modelling method described in sections 3.1 and 3.2, has been tested separately with a bunch of random but fixed valued b_i's combined with some values for N. The $p_{i,j}$ indeed give a permutation that makes that $\bar{b}_i$ are sorted in non-decreasing order. With $T = 600\delta$ and $N <= 10$ the model is typically solved in 40 milliseconds. Since N will typically be smaller than 6, this will not be a problem. However, solving such a problem corresponds to determining and calculating the waiting time costs of alternative routes for 1 OD pair only. When considering all trains in Belgium, about 19000 of OD pairs occur simultaneously.

The model should also work when the vector b contains model variables instead of fixed constants. Since the b_j are variables, (since they are also subject to other PESP constraints), the terms $p_{i,j} \cdot b_j$ in equation (5) are quadratic instead of linear. When a linear model is required, this issue is easily solved by introduction of the helper variables $h_{i,j} \in [0, T - \delta]$ where

$$\forall (O, D) : \forall i, j \in I_N : h_{i,j} = p_{i,j} \cdot b_j. \tag{14}$$

Equations (14) are then linearised, using the big-M method (Williams, 1994), to the inequalities

$$\forall (O, D) : \forall i, j \in I_N : \begin{cases} (bl - bu)(1 - p_{i,j}) & \leq h_{i,j} - b_j & \leq (bu - bl)(1 - p_{i,j}) \\ bl \cdot p_{i,j} & \leq h_{i,j} & \leq bu \cdot p_{i,j}, \end{cases} \tag{15}$$

where bl and bu are the lower respectively upper bounds on all b_j. In our case, $bl = 0$ and $bu = T - \delta$. The top inequality imposes $h_{i,j} = b_j$ when $p_{i,j} = 1$ and imposes nothing new when $p_{i,j} = 0$, while the bottom inequality imposes $h_{i,j} = 0$ when $p_{i,j} = 0$ and imposes nothing new when $p_{i,j} = 1$. This is indeed equivalent to what equations (14) impose. The top line of equation (5) can now be replaced with its linear equivalent

$$\forall (O, D) : \forall i \in I_N : \bar{b}_i = \sum_{j \in I_N} h_{i,j}. \tag{16}$$

The PESP timetabling model is now extended with linear objective function terms and linear constraints to include excess journey time. We wil now apply this model to subsets of the passenger train network in Belgium.

4 Results

This section reports on the results of the application of the model developped in the previous sections on sets of trains extracted from the Belgian timetable as it is planned for 16/12/2014. All experiments in this paper ran Gurobi v5.6.3 on an Intel Xeon CPU E31240 3.3GHz processor with 16GB RAM.

In section 4.1, in order to study the effect of OD spreading separately, we start with a proof of concept where no passenger transfers are considered. A network of 26 trains is used as the test network. Section 4.2 makes a comparison of results obtained with three approaches: (i) our soft spreading technique via the objective function, (ii) the traditional hard spreading technique via hard constraints and (iii) a mixed hard-soft method. In section 4.3, we extend the optimisation by including transfers and verify whether total expected passenger time, and its components (expected ride time, dwell time, transfer time, knock-on time and excess journey time related to temporal spreading) can all still be reduced and if so, to what degree. Section 4.4 does the same but for adding more OD-pairs for spreading optimisation. In section 4.5 the scalability of our method is tested by increasing the network size input to our model.

4.1 Proof of Concept

Table 2 shows the results of optimisations run with a computation time limit of 600 seconds. The OD-pairs which are to be optimised for spreading, are selected by requiring at least 1000 passengers per morning peak for them. This results in 12 OD-pairs. The largest OD-flow occurs between O=Leuven and D=Brussels-Centraal and has F=1751.6 passengers per morning peak. In the case of r=100%, per time unit of time δ of 6 seconds for each of all morning peak hours together, $f = F/T = F/600 = 2.9193$ passengers going to D are assumed to arrive at the origin station O. The 3 Inter City train alternatives for (O,D)=(Leuven, Brussels-Centraal) are IC:A:2, IC:E:2 and IC:F:2. All three connect O and D directly without a transfer. In fact, in these 12 OD-pairs, no single train journey with a transfer occurs.

In the optimisations, the fraction of assumed randomly arriving (and departing) passengers is varied over $r = 0\%, 1\%, 5\%, 10\%, 50\%, 100\%$. For each value of r, a separate optimisation was carried out and a different optimised timetable was computed. Of course, the original timetable, used as reference is the same for all values of r. For each of the 12 OD-pairs that was selected, on the rows s_0, s_1, s_2, the inter-departure times (heading times) at O and inter-arrival times (heading times) at D are given, both for the original timetable (columns *orig O* and *orig D*, italic) and for the optimised one (columns *opt O* and *opt D* * (0,1,5,10,50,100), normal). This gives an idea of the improvement in spreading that was reached. For example, for the case Oostende-Brugge, in Oostende, the original timetable has a spreading of $(s_0, s_1) = (310, 290)$ while the optimised timetables, for many but not all cases of r, deliver the perfect spreading of $(s_0, s_1) = (300, 300)$. For three trains per hour, for example, from Brussel-Zuid to Leuven, we get in the original timetable, at O, $(s_0, s_1, s_2) = (264, 180, 156)$ and at D, $(s_0, s_1, s_2) = (182, 154, 264)$ while in the optimised timetable for $r = 100\%$, we get at O, $(s_0, s_1, s_2) = (195, 154, 264)$ and at D, $(s_0, s_1, s_2) = (200, 200, 200)$, a perfect spreading.

The corresponding excess journey time at O and at D are also mentioned, again both for the original timetable (on the rows u_{orig}) and for the optimised timetable (on the rows u_{opt}). Note that the values on the rows u_{orig}, irrespective of their value for r in their column heading, always refer to the same original timetable. Contrary to this, the values on the rows u_{opt}, refer to a timetable specially optimised for the r value in their column heading. These values indicate how much reduction in excess journey time has been accomplished for a specific OD-pair, for a specific value of r. On the u_{opt} rows a value that is lower than the value above it, in row u_{orig}, is underlined, indicating that a reduction in excess journey time has been achieved. It is marked in bold otherwise. The spreading reached amongst the s_i must be seen as the consequence of the minimisation of the corresponding excess journey time as terms u_i in the objective function according to their relation given by equation (9). We see, as we can expect, that roughly spoken, the higher the value for r and f, the more close the s_i values get to each other, and to T/N.

The values in the rows u_{orig} are proportional to their r value. The columns where $r = 0\%$ do in fact result in $u_{orig} = 0 = u_{opt}$. However, we calculated the incidence wait costs for the timetable reached for $r = 0\%$ as if $r = 1\%$ to show what an underestimation

r (%)	*orig*	0(1)	1	5	10	50	100	*orig*	0(1)	1	5	10	50	100
	O			opt. O, Begin Spread				*D*			opt D, End Spread			
Leuven to Brussel-Centraal, F=1751.6, f=2.9193, [0: IC:A:2:5xx, 1: IC:E:2:15xx, IC:F:2:17xx]														
s_0	*152*	63	86	287	287	156	326	*271*	68	118	119	119	198	104
s_1	*175*	64	111	202	202	200	162	*121*	473	401	283	283	248	333
s_2	*273*	473	403	111	111	244	112	*208*	59	81	198	198	154	163
u_{orig}		*1872*	*1872*	*9361*	*18721*	*93607*	*187215*		*1917*	*1917*	*9586*	*19172*	*95861*	*191722*
u_{opt}		**3383**	**2658**	**9889**	**19778**	90406	**211746**		**3384**	**2646**	**9740**	**19480**	90809	**216432**
Brussel-Centraal to Leuven, F=1751.6, f=2.9193, [0:IC:F:1:17xx, 1:IC:A:1:5xx, 2:IC:E:1:15xx]														
s_0	*154*	81	73	151	197	102	196	*182*	74	200	295	179	200	200
s_1	*263*	73	323	149	181	299	206	*154*	450	74	151	221	102	200
s_2	*183*	446	204	300	222	199	198	*264*	76	326	154	200	298	200
u_{orig}		*1845*	*1845*	*9223*	*18446*	*92232*	*184465*		*1847*	*1847*	*9235*	*18470*	*92351*	*184701*
u_{opt}		**3077**	**2208**	**9853**	17641	**101744**	175243		**3120**	**2215**	**9746**	17645	**101599**	175161
Gent-Sint-Pieters to Brussel-Centraal, F=1648.58, f=2.7476, [0:IC:E:1:15xx, 1:IC:A:1:5xx]														
s_0	*156*	451	205	300	178	300	391	*154*	154	396	300	181	299	206
s_1	*444*	149	395	300	422	300	209	*446*	446	204	300	419	301	394
u_{orig}		*3043*	*3043*	*15213*	*30426*	*152131*	*304262*		*3059*	*3059*	*15293*	*30586*	*152928*	*305856*
u_{opt}		**3099**	2721	12364	28818	123644	270040		3059	2726	12364	28620	123645	271565
Brussel-Centraal to Gent-Sint-Pieters, F=1648.58, f=2.7476, [o:IC:E:2:15xx, 1:IC:A:2:5xx]														
s_0	*121*	541	519	198	402	248	333	*151*	86	101	219	227	372	286
s_1	*479*	59	81	402	198	352	267	*449*	514	499	381	373	228	324
u_{orig}		*3353*	*3353*	*16766*	*33532*	*167662*	*335324*		*3083*	*3083*	*15414*	*30829*	*154144*	*308287*
u_{opt}		**4069**	**3791**	13794	27587	127358	250279		**3731**	**3561**	13266	26193	130765	256590
Oostende to Brugge,F=1177.78,f=1.9629, [0:IC:G:1:18xx, 1:IC:A:1:5xx]														
s_0	*310*	66	300	360	300	300	300	*310*	64	300	360	300	300	300
s_1	*290*	534	300	240	300	300	300	*290*	536	300	240	300	300	300
u_{orig}		*1769*	*1769*	*8843*	*17686*	*88432*	*176863*		*1769*	*1769*	*8843*	*17686*	*88432*	*176863*
u_{opt}		**2842**	1767	**9187**	17667	88334	176667		**2860**	1767	**9187**	17667	88334	176667
Brugge to Oostende, F=1177.78, f=1.9629, [0:IC:G:2:18xx, 1:IC:A:2:5xx]														
s_0	*324*	64	457	313	300	312	300	*300*	63	143	289	300	291	300
s_1	*276*	536	143	287	300	288	300	*300*	537	457	311	300	309	300
u_{orig}		*1778*	*1778*	*8890*	*17780*	*88899*	*177798*		*1767*	*1767*	*8833*	*17667*	*88334*	*176667*
u_{opt}		**2860**	**2251**	8850	17667	88475	176667		**2869**	**2251**	**8845**	17667	**88413**	176667
Brugge to Gent-Sint-Pieters, F=1169.78, f=1.9496, [0:IC:A:1:5xx, 1: IC:G:1:18xx]														
s_0	*310*	64	300	240	301	395	300	*300*	68	296	233	300	234	300
s_1	*290*	536	300	360	299	241	300	*300*	532	304	367	300	366	300
u_{orig}		*1757*	*1757*	*8783*	*17566*	*87831*	*175662*		*1755*	*1755*	*8773*	*17547*	*87734*	*175467*
u_{opt}		**2841**	1755	**9124**	17547	**104357**	175467		**2804**	1755	**9211**	17547	**91980**	175467
Gent-Sint-Pieters to Brugge, F=1169.78, f=1.9496, [0: IC:A:2:5xx, 1:IC:G:2:18xx]														
s_0	*307*	51	183	300	313	300	365	*304*	536	436	313	300	313	253
s_1	*293*	549	417	300	287	300	235	*296*	64	164	287	300	287	347
u_{orig}		*1756*	*1756*	*8778*	*17556*	*87781*	*175563*		*1755*	*1755*	*8775*	*17550*	*87749*	*175498*
u_{opt}		**2963**	**2022**	8773	**17580**	87734	**183704**		**2841**	**2115**	**8790**	17547	**87898**	179774
Leuven to Brussel-Zuid, F=1154.47, f=1.9241, [0: IC:A:2:5xx, 1: IC:E:2:15xx, IC:F:2:17xx]														
s_0	*152*	63	86	287	287	156	326	*208*	116	116	200	117	200	162
s_1	*175*	64	111	202	202	200	162	*271*	401	401	117	283	246	106
s_2	273	473	403	111	111	244	112	*121*	83	83	283	200	154	332
u_{orig}		*1234*	*1234*	*6170*	*12339*	*61696*	*123392*		*1264*	*1264*	*6318*	*12636*	*63181*	*126363*
u_{opt}		**2230**	**1752**	**6518**	**13035**	59586	**139560**		**1743**	**1743**	**6435**	**12870**	59759	**142100**
Brussel-Zuid to Leuven, F=1154.47 , f=1.9241, [0:IC:F:1:17xx, 1:IC:A:1:5xx, 2: IC:E:1:15xx]														
s_0	*264*	82	75	153	196	300	195	*182*	74	200	295	179	200	200
s_1	*180*	71	322	148	180	200	205	*154*	450	74	151	221	102	200
s_2	156	447	203	299	224	100	200	*264*	76	326	154	200	298	200
u_{orig}		*1216*	*1216*	*6082*	*12163*	*60817*	*121635*		*1217*	*1217*	*6087*	*12174*	*60868*	*121735*
u_{opt}		**2035**	**1448**	**6480**	11640	**67344**	115495		**2056**	**1460**	**6424**	11630	**66963**	115447
Gent-Sint-Pieters to Brussel-Zuid, F=1086.56, f=1.8109, [0:IC:E:1:15xx, 1:IC:a:1:5xx]														
Brussel-Zuid to Gent-Sint-Pieters, F=1086.56, f=1.8109, [0:IC:E:2:15xx, 1:IC:a:2:5xx]														

Table 2: Inter-departure and inter-arrival times s_i and corresponding excess journey times u per OD-pair, for optimisations over a set of 26 Belgian IC trains over a range of $F \cdot r$ passengers with assumed random incidence behaviour. OD spreading threshold=1000, no transfers, computation time = 600s.

of the value of r by 1% would result in here. One sees that the u_{opt} values in column $r = 0\%$ are much higher than those in column $r = 1\%$. This is of course the case because at $r = 0\%$, no excess journey time terms are present in the objective function. This indicates that there are quite severe waiting times expected for randomly arriving passengers if one does not model the incident waiting time at all. Table 3 shows that, compared to the excess journey time of the original timetable, an increase of 42% is expected. This indicates that in the original timetable, already quite some spreading was achieved and that in turn explains that even if u_{opt} is just a few percentages removed from u_{orig}, the penalties in the objective function on not spreading must already be delivering their result.

Table 3 shows the column sums over the 12 OD pairs, $U_{O,orig} = \sum_{i=0}^{11} u_{O_i,orig}$, $U_{O,opt} = \sum_{i=0}^{11} u_{O_i,opt}$ and similarly for the destinations D, for the different values for r. The sums for origin and destination are also calculated as $U_{O+D,orig} = U_{O,orig} + U_{D,orig}$ and similarly for the optimised schedule with index opt. This table shows that for $r = 0\%$ and $r = 1\%$ we get an increase of excess journey time compared to the original timetable of respectively 42% and 12%. For r larger than 5% our model manages a decrease of between 5% and 7%. We conclude that, for this system of 26 trains and 12 OD pairs, optimisation with our model is able to quickly (in 600 seconds) reduce the excess journey time significantly ($> 5\%$) compared to a manually constructed timetable as long as the fraction r is also significant ($\geq 5\%$). If r is very low ($\leq 1\%$), the excess journey time terms in the objective function have insufficient weight to strongly influence the solution and those few r passengers will experience excess journey times that are higher than in a manual constructed timetable. Other terms then dominate the optimisation of the total time which is still reduced. This proves that our added excess journey time model serves its basic purpose.

sol_i	r(%)	0(1)	1	5	10	50	100
sol_1	MIP gap (%)	16.6	59.4	37.9	65.2	24.9	15.7
	computation time (s)	13	102	48	182	59	242
$sol_{end,opt}$	comp. time=600s ⇒ gap (%)	0.65	1.64	1.83	2.62	4.36	5.74
	$U_{O,orig}$	23774	23774	118871	237742	1188710	2377419
	$U_{O,opt}$	**34098**	**26648**	<u>112036</u>	<u>226063</u>	<u>1101965</u>	<u>2217687</u>
	ratio	**1.43**	**1.12**	<u>0.94</u>	<u>0.95</u>	<u>0.93</u>	<u>0.93</u>
	$U_{D,orig}$	23469	23469	117344	234688	1173442	2346884
	$U_{D,opt}$	**32947**	**26386**	<u>110901</u>	<u>223033</u>	<u>1093150</u>	<u>2228194</u>
	ratio	**1.40**	**1.12**	<u>0.95</u>	<u>0.95</u>	<u>0.93</u>	<u>0.95</u>
	$U_{O+D,orig}$	47243	47243	236215	472430	2362152	4724303
	$U_{O+D,opt}$	**67045**	**53033**	<u>222936</u>	<u>449096</u>	<u>2195115</u>	<u>4445882</u>
	ratio	**1.42**	**1.12**	<u>0.94</u>	<u>0.95</u>	<u>0.93</u>	<u>0.94</u>
	excess journey time reduction (%)	**-42**	**-12**	<u>6</u>	<u>5</u>	<u>7</u>	<u>6</u>

Table 3: Total origin-, total destination- and total expected excess journey time U over the OD-pairs of table 2. Corresponding original to optimised timetable excess journey time reductions, MIP gaps and computation times. sol_1 is the first feasible solution and sol_{end} the one at the set time limit.

4.2 Comparing Soft with Hard Spreading

If one wants *perfect temporal spreading* of alternative trains, with all $s_i = T/N$, for a set of given OD pairs, we *additionally* impose

$$\forall(O, D) : \forall i \in I_N : s_i = T/N. \tag{17}$$

sol_i	r(%)	0	1	5	10	50	100
Soft spreading at Origin and Destination							
	comp. time=1200s ⇒ MIP gap (%)	0.83	3.57	7.88	8.18	4.74	5.98
	$U_{O,orig}$	0	27705	138524	277047	1385237	2770474
	$U_{O,opt}$	0	25612	**144050**	**328453**	1189351	2516351
	ratio	1.00	0.92	**1.04**	**1.19**	0.86	0.91
	$U_{D,orig}$	0	27385	136926	273852	1369259	2738519
	$U_{D,opt}$	0	25095	**140854**	**326685**	1177353	2429862
sol_{end}:opt	ratio	1.00	0.92	**1.03**	**1.19**	0.86	0.89
	$U_{O+D,orig}$	0	55090	275450	550899	2754496	5508992
	$U_{O+D,opt}$	0	50707	**284904**	**655138**	2366704	4946213
	ratio	1.00	0.92	**1.03**	**1.19**	0.86	0.90
	excess journey time reduction (%)	0	8	**-3**	**-19**	14	10
	opt. total time reduction (%)	15.42	15.34	15.17	15.03	15.31	14.96
sol_{end}:eval	post-opt. evaluation total time reduction (%)	4.6	6.26	5.68	4.81	3.71	4.92
Hard Spreading at Origin and Destination							
	comp. time=1200s ⇒ MIP gap (%)	68.07	71.56	66.9	64.75	51.64	39.36
	$U_{O,orig}$	0	27705	138524	277047	1385237	2770474
	$U_{O,opt}$	0	21060	105301	210603	1053013	2106026
	ratio	1.00	0.76	0.76	0.76	0.76	0.76
	$U_{D,orig}$	0	27385	136926	273852	1369259	2738519
	$U_{D,opt}$	0	21060	105301	210603	1053013	2106026
sol_{end}:opt	ratio	1.00	0.77	0.77	0.77	0.77	0.77
	$U_{O+D,orig}$	0	55090	275450	550899	2754496	5508992
	$U_{O+D,opt}$	0	42120	210603	421205	2106026	4212052
	ratio	1.00	0.76	0.76	0.76	0.76	0.76
	excess journey time reduction (%)	0	24	24	24	24	24
	opt. total time reduction (%)	6.89	5.64	7.93	6.79	6.84	6.28
sol_{end}:eval	post-opt. evaluation total time reduction (%)	5.24	3.68	6.02	4.96	4.61	3.72
Hard Spreading at Origin and Soft Spreading at Destination							
	comp. time=1200s ⇒ MIP gap (%)	28.15	31.0	35.59	30.45	21.59	13.59
	$U_{O,orig}$	0	27705	138524	277047	1385237	2770474
	$U_{O,opt}$	0	21060	105301	210603	1053013	2106026
	ratio	1.00	0.76	0.76	0.76	0.76	0.76
	$U_{D,orig}$	0	27385	136926	273852	1369259	2738519
	$U_{D,opt}$	0	22979	115029	229787	1176459	2278535
sol_{end}:opt	ratio	1.00	0.84	0.84	0.84	0.86	0.83
	$U_{O+D,orig}$	0	55090	275450	550899	2754496	5508992
	$U_{O+D,opt}$	0	44039	220330	440389	2229472	4384561
	ratio	1.00	0.80	0.80	0.80	0.81	0.80
	excess journey time reduction (%)	0	20	20	20	19	20
	opt. total time reduction (%)	14.55	14.41	14.06	14.24	14.23	14.7
sol_{end}:eval	post-opt. evaluation total time reduction (%)	4.83	3.78	4.07	4.23	3.63	3.27

Table 4: Effects of soft, hard and hard-soft spreading enforcement at Origin and Destination for 26 trains. OD spreading threshold=1000, no transfers.

The existing permutation matrices and objective function terms defined before can still reside in the model and continue to fulfil their purpose of correctly measuring the ordering of train alternatives and excess journey time. To investigate the effects of hard versus soft spreading, we compared three cases, where we allow 1200 seconds of optimisation time for each. The first case is one where we apply soft temporal spreading as before on origin and destination side, so by the model without equations (17). The second case is one where we enforce hard spreading via equations (17) on both O and D side. In the last case we apply hard spreading on the origin side, but soft spreading on the destination side. Note that the MIP gaps of methods with additional hard constraints are typically not reduced as quickly as methods without these. So we compare solutions for these three cases from the fair standpoint of what can be achieved in equal computing times.

Table 4 gives the results of these experiments. First of all, in the top third of the table, for soft spreading at origin and destination, it shows that a total time reduction during optimisation of around 15% is again achieved. However, when doing a post-evaluation with all excess waiting time associated with the 12 OD-pairs, we get a weaker reduction of total expected time from the original to the optimised timetable in the range of 3.71% to 6.26%.

When forcing spreading the hard way, on both origin and destination side, the middle part of Table 4 shows that we get systematic savings of 24 percent in excess journey time. This is the same for all values of r, which is logical. Since the excess journey time is forced to the minimum possible here, this is also higher than the savings of around 15% for the soft spreading technique. However, the reductions in total time range between 3.68% and 6.02%, and so, are similar to the soft spreading technique. This means that the increase in reduction from 15% to 24% in excess journey time has been lost in other time components of the objective function. In fact, more ride and dwell supplements have been added for trains between O and D for all train alternatives to match the strict requirements of equal spreading on both ends. So these trains become 'overstretched'.

To try and avoid this effect of overstretching trains, we now enforce spreading in the hard way on the origin and in a soft way on the destination side. The lower third of table 4 shows the results. We find that the excess journey time reduction is around 14.5% and the total time reduction is in the range 3.27% to 4.83%. This is comparable to the results of the approach with soft enforcement of temporal spreading in the upper third part of the table.

We conclude that the three approaches give similar reductions in total time but the approach of hard constraints on both sides gives a strong bias towards concentrating mainly on the reduction of excess journey time with the negative consequence of increasing expected ride and dwell time more than in the other two cases. For these experiments, which do not consider the component of expected transfer time, as for the highest total expected time reduction at post optimisation evaluation, there is no clear winner method yet. Table 4 shows, for each value of r, the highest percentage for this time reduction as underlined.

4.3 Adding Transfers

In the previous sections, we studied the minimisation of the excess journey time cost. We explicitly removed all transfer time costs from the objective function to avoid they would bias the obtainable reduction percentage of the excess journey time. We now add transfer time terms in the objective function and will see if our model is then also still able to reduce the total expected passenger time, including excess journey time and transfer time. The results are summarised in figure 4. To most clearly show the excess journey time component, in each case, r is set to 100%. Each of the six pictures shows a bar graph for the original timetable on the left and the optimised timetable on the right. We consider the same three cases as in the previous section and so, the organisation of in three rows of figure 4 is similar to that of table 4. The upper third represents the optimisation with soft enforcement of spreading on both O and D side. The middle third shows the optimisation with hard enforcement of spreading on both O and D side. The lower third shows results for hard enforcement of spreading on the O side and soft enforcement of spreading on the D side. The left half shows what is reached during optimisation (on the linearised objective function over the selected portion of OD-pairs for spreading and the selected transfers). The right half shows the evaluation of the original, non-linearised objective function on all OD-pairs and on all transfers and represents the end result on which each method should be evaluated.

In all cases, the dark yellow blocks are summed minimal ride and minimal dwell times.

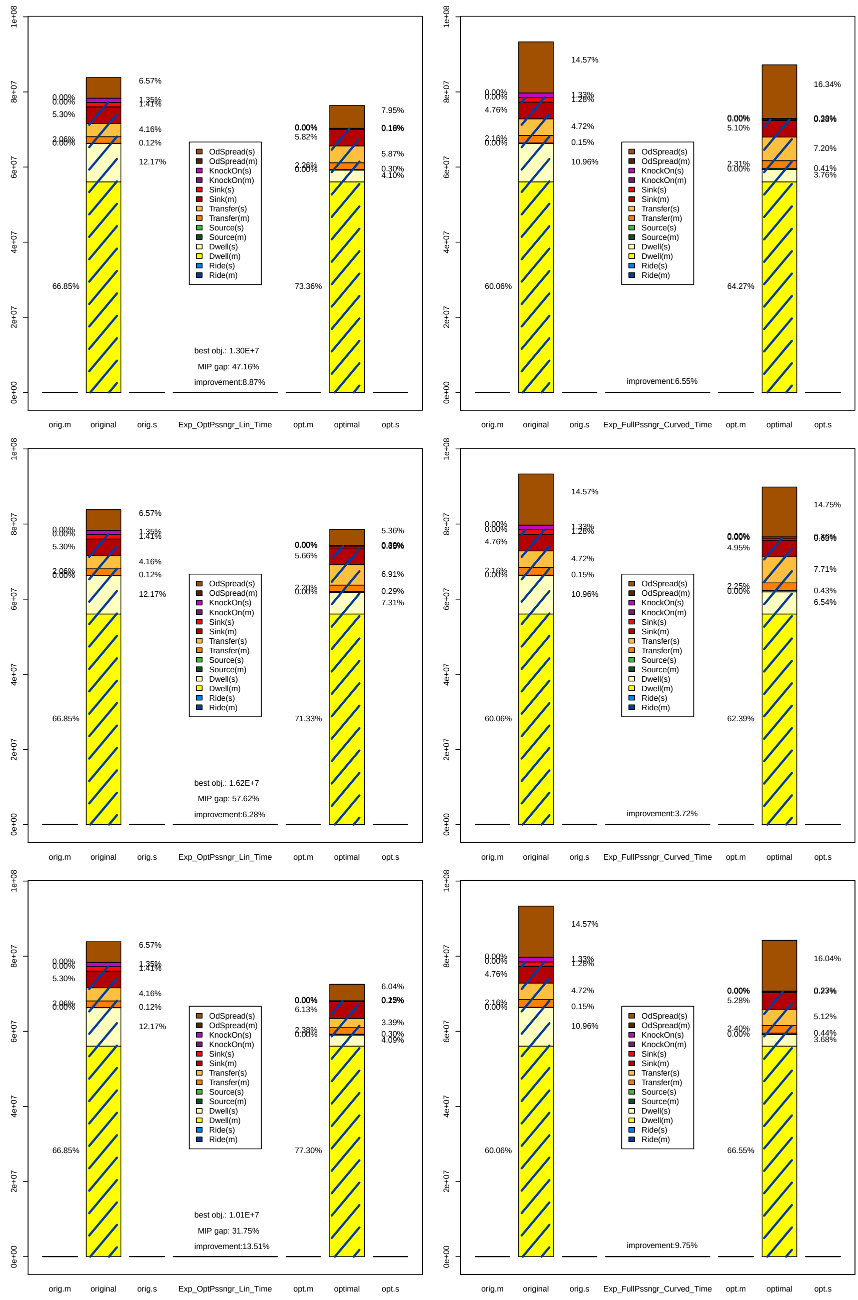


Figure 4: Effects of soft, hard and hard-soft spreading enforcement at Origin and Destination for 26 trains and adding transfers. OD Spreading threshold=1000, transfer threshold=210, computation time=1200s. Left bars are for the original timetable and right bars for the optimised timetable. Note the bigger ride and dwell expected time components on the middle row, representing hard spreading on O and D side.

These are constant and cannot be reduced during optimisation. Any blue shaded block represents an action that has its time convoluted with the 'preceding' ride action (Sels et al., 2011). The (blue shaded) light yellow part represents the summed convoluted ride and dwell supplement times. The (blue shaded) green block corresponds to time attributed to passengers entering and the red block to passengers leaving the transportation system. Orange (blue shaded) blocks represent transfer time. Dark orange is the minimum transfer time and is supposed to be 3 minutes everywhere. Light orange stands for the total expected passenger transfer time due to the transfer supplements. This includes a penalty of 1 hour in case the transfer is missed. In the left column, there are 745 transfers contributing, which are the ones considered during optimisation. In the right column, all transfers are considered during evaluation. The (not blue shaded) light purple colour indicates expected knock-on time, also passenger weighted (Sels et al., 2013a), between subsequent trains on the same infrastructure resource. Brown blocks stand for excess journey time for O and D for all 12 OD pairs together, in the left column. All 7601 OD-pairs are evaluated for excess journey time in the right column. By optimising the timetable, only the light yellow, light orange, light green, light red, purple and brown parts can be shrunk The dark yellow, dark orange, dark green, dark red parts correspond to action minima and are the same in any timetable, so cannot be shrunk.

In the upper-left figure, from the original timetable to the optimised one we get a total time reduction of 8.87% during optimisation. If we call the left bar 100, the right bar represents 100-8.87=91.13. The brown excess journey time component is slightly increased with respect to the original timetable, in absolute terms, from 6.57 out of 100 to 7.95% of 91.13 = 7.2. The transfer time is also growing from 4.16 to 5.87% of 91.13 so 5.35. However, the light-yellow-coloured ride and dwell supplements are strongly reduced from 12.17 out of 100 to 4.10% of 91.13 so 3.74. The decrease of ride and dwell supplements more than compensates the slight increase of expected excess journey time and expected transfer time. The net result is that total expected passenger journey time decreases.

In the left figure of the middle row, representing hard spreading enforcement on original and destination side, we see that the total expected time reduction is slightly less, 6.28%. So the height of the right bar-graph corresponding to the optimised timetable now represents 100-6.28=93.72. However, the brown block representing excess journey time is now reduced, from 6.57% down to 5.36% of 93.72=5.02. The orange transfer component grows from 4.16% to 6.91% of 93.72=6.48, a bigger increase compared with soft spreading enforcement. The ride and dwell supplements are reduced from 12.17 to 7.31% of 93.72 so 6.85 which is substantially more than the 3.74 reached with soft spreading enforcement. Soft enforcement reaches 8.87% total expected time reduction while hard enforcement reaches only 6.28%. Since total time reduction is the end goal, from this observation, soft enforcement is clearly preferable. The full evaluation in the right half of the middle row of figure 4 shows that also when including OD pairs and transfers with fewer passengers than the ones considered in optimisation, the results are similar and even amplified. The reduction is 6.55% for soft enforcement but only 3.72% for hard enforcement of spreading.

On the bottom row, we find the figures corresponding to the optimisation with hard spreading enforcement on the origin and soft enforcement on the destination side. The reduction of total time the optimisation achieves is a remarkable 13.51%. This is significantly better than the other approaches which reached 8.87% and 6.28% total expected time reduction. The excess journey time goes from 6.57 on 100 to 6.04 on (100-13.51)=86.49 so 5.22. In absolute terms this is very close to the 5.02 strict spreading on both sides achieves.

Expected transfer time is reduced from 4.16 on 100 to 3.39 on 86.49 so 2.93. This is almost the double reduction of the 5.35 and 6.48 we get in the other spreading approaches. Ride and dwell supplements are reduced to 4.90% of 86.49 so 3.54. Again this is better than the 3.74 and the 6.85 we saw in the other approaches. So we get better optimisation results when enforcing more hard constraints. Normally, when keeping the same objective function, imposing extra constraints to a model cannot result in a better 'optimal' solution. But here, the solver handles the model with the extra constraints *faster* than the one without. Indeed, in our cases the optimal solution is not reached yet by the solver, which is indicated by the MIP gaps presented in the left half of figure 4. From top to bottom, at 1200 seconds, soft spreading achieves a gap of 47.16%, hard spreading a gap of 57.62% and mixed spreading a gap of 31.75%. When we increase the computation time limit to 3600 seconds, the respective gaps become 41.9%, 58.33% and 31.54%. So between 1200 seconds and 3600 seconds, only the soft spreading model still significantly improves its solution. But even the best result for mixed spreading at 1200 seconds is not beaten at 3600 seconds by any of the two other methods. Note that In all cases, both for 1200 seconds and 3600 seconds, the bound is the same (within a margin of 0.3%), so gaps are comparable.

In the evaluation of all OD-pairs on excess journey time and all transfers in the right figure on the lower row, we notice a 9.75% net reduction is achieved (in 1200 seconds). Again, this is significantly better than the 6.55% and 3.72% of the other approaches. One of the reasons is that transfer time is now slightly reduced instead of significantly growing. We conclude that, once transfers are included, spreading the hard way on the origin and spreading the soft way on the destination (or probably also the reverse) is the preferable tactic for achieving the best results in the shortest time on this network of 26 trains.

4.4 Adding more OD-pairs for Spreading

For the same network of 26 Inter City trains, we now investigate the sensitivity of solution quality and computation time to the amount of OD-pairs in the model. We reduced the OD-threshold for consideration in the model of 1000 passengers per morning peak, in steps of 100, as such optimising over more and more OD-pairs. The transfer threshold is set back to 2000, hoping for low computation times. For each OD-pair, we enforce strict optimal spreading on the origin side and soft spreading via the objective function on the destination side, since this seemed the most promising technique in the previous section.

sol_i	OD-threshold for spreading	900	800	700	600	500	400
	# OD-pairs in opt.($N \geq 1$)	18	24	36	56	90	128
	# OD-pairs in opt. ($N > 1$)	14	16	22	24	30	31
sol_1	MIP gap (%)	23.2	50.1	38.2			
	computation time (s)	52	107	253			
$sol_{end,opt}$	MIP gap (%)	13.2	14.3	17.13			
	computation time (s)	1200	1200	1200	2400	2400	2400
	excess journey time reduction (%)	20.24	20.03	20.72			
	opt. total time reduction (%)	14.69	14.59	14.10			
$sol_{end,eval}$	transfer time reduction (%)	-240	-234	-239			
	post-opt. evaluation total time reduction (%)	4.38	4.65	4.72			

Table 5: Effects of adding more OD-pairs for temporal spreading to the optimisation model. 26 Inter City trains. Computation times are 1200s and 2400s.

The results are given in table 5. For the cases F=900, 800 and 700, models are constructed which perform temporal spreading of at least 2 alternative trains of respectively

14,16 and 22 OD-pairs. These models could all be optimised in 1200 seconds. Over these cases, the MIP gap achievable in 1200 seconds rises, yet the reduction of total expected waiting time stays fairly stable around 14%. The end result reduction when evaluating over all OD-pairs and all transfers increases somewhat from 4.38% over 4.65% to 4.72%.

For the cases F=600, 500 and 400, models are constructed that perform spreading on respectively 24, 30 and 34 OD-pairs with more than one alternative train. However, the imposed time limit of 2400 seconds did not suffice to solve these models.

4.5 Scaling Up to a Larger Network

We tried the approach of soft spreading for both O and D on a network of all 74 Inter City trains as they are planned for 16/12/2014. The thresholds for OD-pairs to be considered in optimisation was again set to 1000 and the transfer threshold to 210 passengers per morning peak. The required gap was set to 95%. This did yield a resulting timetable after about 26.7 hours. Optimisation reduced total expected passenger travel time by 2.76% but post evaluation on all OD-pairs for spreading and all transfers gave an increase of 6.48%. Decreasing the required gap below 95% could possibly yield a better timetable, but at the expense of more computation time. Increasing to systems with more than 74 hourly passenger trains did not result in a solution yet. Since our MILP model without excess journey time *can* be solved for all Belgian passenger trains in about *2 hours* (Sels et al., 2014), we conclude that specifically our excess journey time model is not easily scalable yet to larger networks.

5 Conclusions and Further Work

For passengers with random arrival behaviour, we integrated the expected excess journey time cost into our previous PESP based timetabling MILP model (Sels et al., 2014). This also required derivation and integration of extra constraints in our model. Our objective function which represented total expected journey time is now extended to also include excess journey time for all passengers. This paper shows that this model is indeed still able to reduce the total journey time, while it also reduces each of the separate components: ride-dwell supplements, knock-on time and excess journey time as well.

We were unable to solve the model for the network of all passenger trains in Belgium though. Possibly restricting the range of integers present in the formulation of the excess journey time cost could be a remedy for this problem. Alternatively, our high computation times could potentially be reduced by the addition of special cycle sets or a column generation approach. Also, a multi-module PESP formulation approach could be explored.

Incorporating the secondary effect of the waiting time that passengers experience when they miss the train they intended to depart with could make the model even more realistic.

The missed transfer penalty of cycle time T, currently assumed in the model, could now also be adapted as follows. If the transfer takes the transferring passenger on an OD-route that is contained in the optimised OD-pair spreading set, the N trains can be assumed to be well-spread, certainly if hard spreading is imposed. So the penalty for missing a train can be assumed to be about T/N in the average case. This would make a more realistic transfer model and the transfer time component would make up a smaller portion of the total time. Since it typically still grows during optimisation, the total time would be further reduced.

References

Barnett, A., 1974. On Controlling Randomness in Transit Operations. Transportation Science. 8, 102–116.

Bly, P. H., Jackson, R. L., 1974. Evaluation of Bus Control Strategies by Simulation. Tech. rep., Transport and Road Research Laboratory.

Bowman, L., Turnquist, M., 1981. Service Frequency, Schedule Reliability and Passenger Wait Times at Transit Stops. Transportation Research 15A (6), 465–471.

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., 2011. Optimization Problems in Passenger Railway Systems. Wiley Encyclopedia of Operations Research and Management Science 6, 3896–3905.

de Pirey, Y., 1971. Simulation of a Bus Line as a Means to Analyse and Improve Bus Transit Reliability. Master's thesis, Northwestern University, Department of Civil Engineering, Evanston, Illinois.

Fan, W., Mechemehl, B., june 2002. Characterising Bus Transit Passenger Waiting Times. Proceedings of 2nd Material Speciality Conference of the Canadian Society for Civil Engineering, June 5-8, Montreal, Quebec, Canada.

Friedman, R. R., 1976. Statistical Models of the Mean and Standard Deviation of Passenger Wait Time in Bus Transit. Master's thesis, Northwestern University, Transportation Center, Evanston, Illinois.

Frumin, M., Zhao, J., 2012. Analysing Passenger Incidence Behaviour In Heterogeneous Transit Services Using Smartcard Data and Schedule Based Assignment. Transportation Research Record: Journal of the Transportation Research Board. 2274 (1), 52–60.

Furth, P. G., Muller, T. H. J., march 2006. Service Reliability and Hidden Waiting Time: Insights from AVL Data. Transportation Research Records.

Galli, L., Stiller, S., october 2010. Strong formulations for the Multi-module PESP and a quadratic algorithm for Graphical Diophantine Equation Systems. Proceeding EAS'10 of the 18th annual European conference on Algorithms: Part I, 338–349.

Goverde, R., 1998a. Optimal Scheduling of Connections in Railway Systems. Paper presented at the 8th WTCR, Antwerp, Belgium.

Goverde, R., 1998b. Synchronization Control of Scheduled Train Services to minimize Passenger Waiting Times. Proceedings of 4th TRAIL Annual Congress, TRAIL Research School, Delft, The Netherlands.

Holroyd, E. M., Scraggs, D. A., 1966. Waiting Times for Buses in Central London. Traffic Engineering Control. 8, 158–160.

Islam, M. K., Vandebona, U., October 2010. Reliability Analysis of Public Transit Systems using Stochastic Simulation. Proceedings of the 33rd Australian Transport Research Forum Conference, 29 September - 1 October, Canberra, Australia.

Jackson, W. H., 1977. Bus Passenger Arrival Times. Master's thesis, University of Toronto, Department of Civil Engineering, Toronto, Ontario.

Jolliffe, J. K., Hutchingson, T. P., 1975. A Behavioral Explanation of the Association between Bus and Passenger Arrivals at a Bus Stop. Transportation Science. 9, 248–282.

Kroon, L., Dekker, R., Vromans, M., 2007. Cyclic Railway Timetabling: A Stochastic Optimization Approach. Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science, 41–66.

Kroon, L., Huisman, D., Abbink, E., Fioole, P.-J., Fischetti, M., Maróti, G., Schrijver, A., Ybema, R., 2009. The New Dutch Timetable: The OR Revolution. Interfaces 39, 6–17.

Liebchen, C., 2006. Periodic Timetable Optimization in Public Transport. Ph.D. thesis, Technische Universität Berlin.

Liebchen, C., 2007. Periodic Timetable Optimization in Public Transport. Operations Research Proceedings 2006, 29–36.

Nachtigall, K., 1996. Periodic network optimization with different arc frequencies. Discrete Applied Mathematics 69, 1–17.

Okrent, M. M., 1971. Effects of Transit Service Characteristics on Passenger Waiting Time. Master's thesis, Northwestern University, Department of Civil Engineering, Evanston, Illinois.

Osuna, E. E., Newell, G. F., 1972. Control Strategies for an Idealized Public Transportation System. Transportation Science. 6 (1), 57–72.

Peeters, L., 1 2003. Cyclic Railway Timetable Optimization. Ph.D. thesis, Erasmus Resaerch Institue of Management (ERIM), Rotterdam.

Schrijver, A., 1993. Spoorwegdienstregelingontwikkeling (Timetable Construction). Technical Report, CWI Center for Mathematics and Computer Science, Amsterdam.

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.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2013a. Knock-On Delay Modelling and Optimisation: The Passenger Perspective. In: Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013), December 2-4, Dresden, Germany. pp. 1–10.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., May 2013b. Expected Passenger Travel Time as Objective Function for Train Schedule Optimization. Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): Rail-Copenhagen2013, May 13-15, Copenhagen, Denmark.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2014. Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable, under review. Transportation Research Part B.

Serafini, P., Ukovich, W., 1989. A Mathematical Model for Periodic Scheduling Problems. SIAM Journal on Discrete Mathematics 2, 550–581.

Sparing, D., Goverde, R. M., Hansen, I. A., 2013. An Optimization Model for Simultaneous Periodic Timetable Generation and Stability Analysis. Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark.

Turnquist, M. A., 1978. A Model for Investigating the Effects of Service Frequency and Reliability on Bus Passenger Waiting Time. Transportation Research Records. 663, 70–73.

van der Hurk, E., Kroon, L., , Gábor, M., Vervest, P., december 2014. Deduction of Passengers' Route Choice from Smart Card Data. IEEE-ITSC2013 on Intelligent Transport Systems, to appear 2014.

Welding, P. I., 1957. The Instability of Close Interval Service. Operational Research Society. 8 (3), 133–148.

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

Zhao, J., Frumin, M., Wilson, N., Zhao, Z., 2013. Unified Estimator for Excess Journey Time under Heterogeneous Passenger Incidence Behaviour using Smartcard Data. Transportation Research, Part C. 34, 70–88.

2.7 Practical Macroscopic Evaluation and Comparison of Railway Timetables

Full paper presented at EURO Working Group on Transportation (EWGT), Delft, The Netherlands, 14-16 July 2015

Our tool for timetable optimisation, RhinoCeros, normally takes in an original timetable, extracts the line planning and *reoptimises* the timing for it, producing another timetable. It then also evaluates both the original timetable and the automatically produced 'optimised' one and writes out a comparative report. However, we can also use it in a mode that skips the optimisation and just evaluates the original timetable. By applying the evaluation function of RhinoCeros on two different input timetables, and combining the outputs, it is possible to obtain a comparative *evaluation* of two given timetables. This is what is described in the following publication for two manually created timetables supplied by Infrabel.

It is found that the second timetable (T2) reduces the total expected passenger journey time by 2.47% compared to the first one (T1). The transfer component in this total went from 10.48% of the total in timetable T1 to 7.44% of the total in timetable T2, an effective reduction in absolute terms of 31%, so transfers were better planned. From the reduction of the total planned sum of passenger weighted minima of ride and dwell times, we conclude that there is also an improvement in line planning. Roughly one third of the 2.27% has to be attributed to line planning. Knock-on delays remained the same. In total, more supplements were planned in T2 than in T1, but they did not result in less passenger time. This means that they were not added in the amount and places where they would maximally benefit passengers, as our tool, RhinoCeros would do.

This paper also explains that by utilising our method of minimising expected passenger time in practice, the quality criteria for timetables as used by other researchers are obtained. We summarise these performance criteria here and indicate why our comparative method properly evaluates them.

- A *realisable* timetable is one that plans enough time for all ride and dwell activities. This means no negative supplements on top of the minimal necessary times are planned. Many timetables in Europe are not realisable.
- A *conflict-free* timetable is one where supplements on top of the headway minimal time between any two consecutive trains are all positive. In the strict sense, this has to be analysed on a microscopic level. We and many

other researchers and practitioners use a macroscopic level approximation and require a 3 minute minimal headway time.

- When a timetable is realisable and conflict-free it is called *feasible*. Realisability, conflict-freeness and so also feasibility are deterministic properties of a timetable. This means it is possible to evaluate them without considering the (stochastic) expected primary delays and their delay propagation consequences. This also means that the answer to the question whether a timetable is feasible (or realisable or conflict-free) is ether yes or no.

- A *robust* timetable is a timetable that can absorb common primary delays. This means that in practice, trains will still run according to the timetable even if they experience common small delays. This is accounted for while planning the timetable by providing sufficient time supplements with the aim to absorb these primary delays to a certain degree.

- An *efficient* timetable is one that delivers low travel times to passengers. For this to be realised, time supplements intended to provide robustness cannot be extremely high. Efficiency and robustness are stochastic properties depending on stochastic primary delay distributions. Since our objective function contains these primary delay distributions and also their effects in the network on secondary delays, our evaluation function captures the *balance* that a timetable should realise between efficiency and robustness, so between small enough and large enough supplements respectively. For stochastic properties of a timetable, the answer to the question whether a timetable satisfies this property cannot be a simple yes or no answer and will always be 'to a certain degree'. However, when comparing two timetables, one timetable can score better on these properties than the other.

- A *stable* timetable is one where negative supplements are tolerated, as long as sufficiently positive supplements are present elsewhere in the timetable that are able to absorb the delays caused by the negative supplements. When we generate rather than compare timetables, stability is satisfied via the guarantee that our method generates (macroscopically) feasible timetables. So stability is not obtained by the objective function but by the hard constraints in our model. When we evaluate or compare timetables, non-stability is penalised by the fact that when we read in the timetable, we automatically adapt it by replacing any negative ride or dwell supplement by a zero supplement. This corresponds to what effectively happens in reality, at best, as well. The consequence is that in reality as well as in our evaluation, headways between trains may not be sufficient anymore. This will lead to a knock-on delay, both in reality and in our evaluation function and leads to increased expected passenger time, both in the model and in reality. So, on a macroscopic level, stability is properly accounted for in our evaluation. Stability is essentially a stochastic property of a timetable. One can avoid to have to analyse stability by requiring the timetable to be feasible. Stability follows from feasibility, but not the other way around.

- A *resilient* timetable is one that is able to being adjusted easily to larger delays than can be absorbed by the robustness of the timetable. So a timetable can be claimed to be resilient only if it is specified what real time interventions of traffic control (essentially real time timetable changes) will be taken in real time to make it resilient. Our reason for expecting that resilience will be better in the timetables we generate is that our timetables have a significantly lower amount of very low headway times than the existing timetables. We expect that this will mean that traffic control will have to intervene with fewer train order changes. When we evaluate or compare timetables, resilience is not directly accounted for, since we do not include the resiliency measures taken in practice.

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Transportation Research Procedia 10 (2015) 625–633

Transportation Research Procedia

www.elsevier.com/locate/procedia

18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands

Practical Macroscopic Evaluation and Comparison of Railway Timetables

Peter Sels [a,b,*], Dirk Cattrysse [a], Pieter Vansteenwegen [a]

[a] *KU Leuven, Leuven Mobility Research Centre, CIB, Celestijnenlaan 300, 3001 Leuven, Belgium*
[b] *Logically Yours BVBA, Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium*

Abstract

From the perspective of passengers, a railway timetable can be called better than another if its expected passenger time is lower in practice. So, we constructed an analytical function that evaluates a timetable on this criterion: total expected passenger time in practice. Other methods to evaluate timetables invariably describe different performance indicators: realisability, conflict-freeness, stability, efficiency, robustness, resilience, but mostly do not indicate how to score and weigh these different performance indicators. This means that when comparing two timetables, deciding which one is preferable remains hard. Our objective of expected passenger time in practice resolves these issues.

Also, compared to a simulation approach, our analytical stochastic approach has a major practical advantage. It decouples all actions (ride, dwell, transfer, knock-on) in the timetable and in doing so, can evaluate the expected time in every action separately and simply add all expected times of composing actions afterwards. So the exponential amount of combinations of primary delays over all actions that standard simulation packages explicitly iterate over is dealt with implicitly and much more efficiently. This makes that the evaluation made by our method requires less time.

Our method is applied to two timetables of all passenger trains in Belgium. Both timetables were manually planned and then put into operation in practice. With our method, we can conclude that one timetable has considerably lower total expected passenger time in practice than the other one. We also show that this is caused mainly by better passenger transfer planning, but also partly by a changed line planning. Comparison of the reported results for both timetables also suggests that advantages of each could maybe be combined.

Peer-review under responsibility of Delft University of Technology
doi: 10.1016/j.trpro.2015.09.016.

Keywords:
Comparing Cyclic Timetables, Expected Passenger Time, Objective Function Evaluation, Periodic Event Scheduling Problem (PESP)

* Corresponding author. Tel.: +32-486-956797.
E-mail address: sels.peter@gmail.com

Peer-review under responsibility of Delft University of Technology
doi: 10.1016/j.trpro.2015.09.016.

1. Introduction

Whenever a railway company updates or completely overhauls a timetable, it is highly important to evaluate the new timetable and compare it to the previous one, before the new timetable is implemented in practice. Today, evaluation methods are typically restricted to simulation methods that do not report expected passenger time but mainly focus on train time, ignoring passenger numbers in ride, dwell and transfer actions and in knock-on delays (Weston et al., 2006; Parbo et al., 2015). More recently, this shortcoming has been remedied by some (Weston et al., 2006; Kanai et al., 2011). Also, often no measuring of positive effects of temporal spreading of alternative trains is present. So these methods do not answer some relevant questions that should be asked when developing and evaluating a new timetable.

Questions to be asked about the correctness of any timetable are: Are all minimum ride, dwell, transfer and headway times respected? Questions about whether the new timetable is an improvement compared to the previous one are: Is total expected passenger time in practice reduced? Is the average probability for passengers of missing transfers reduced? Are secondary delays for passengers diminished?

In this paper, we provide a methodology and a tool to answer all these questions. The output is presented graphically, so that the effects - both in size and in sign - of differences between timetables become more visually obvious. As such, strong and weak points in the new timetable, for example better transfer planning or a worse headway planning, will be noticed quickly. Our method is applied to two different timetables, both of all passenger trains in Belgium and shows that the newer of the two timetables is indeed improved. We believe our tool is innovative in the sense that it shows how much better or worse a timetable is for passengers and on which aspects. It also pinpoints where expected passenger time is spent: during ride, dwell, transfer or knock-on actions. In an early stage, our evaluation method could indicate where to further improve the timetable and in the end it could deliver crucial arguments to convince anyone about the benefits of implementing the new timetable in practice.

Section 2 compares traditional timetable evaluation methods with our approach. Section 3 demonstrates the results of our evaluation approach applied to two manually constructed timetables, T1 and T2 as received from Infrabel, the Belgian infrastructure management company. T2 is a reworked version of T1. Due to confidentiality issues, we cannot give more details about these timetables. Sections 4 and 5 conclude this work and indicate some potential further work.

2. Comparing Timetable Evaluation Methods

Both our quality criterion for and our practical implementation of evaluating timetables are quite different from the traditional criteria and practical methods. So, in this section, we first contrast and compare them. We then show that our approach has quite some theoretical as well as practical advantages and next, explain that we cover most performance indicators present in the traditional approaches.

2.1. Traditional Approach: Estimating Performance Indicators by Simulation, mainly Microscopic

Goverde and Hansen (2013) propose a list of timetable performance indicators and derived quality levels. Widely used terms for these indicators are realisability, conflict-freeness, stability, efficiency, robustness and resilience. We are convinced that these six performance indicators should actually be considered as means to obtain the final goal of serving the passengers in the best possible way. According to us that corresponds to minimising the expected passenger time in practice. The performance indicators realisability and conflict-freeness can be checked by verifying that the minima for ride and dwell respectively for headways are respected. Together, realisability and conflict-freeness could be seen as operational feasibility. So these two criteria are easy to check. Contrary to these, the four other performance indicators depend on primary delays and on how these are propagated in the train network. Because of the stochastic nature of primary delays, timetable evaluation on these criteria is usually done via simulation, where a single simulation randomly takes one sample for each primary delay distribution and then applies these combined delays on the system and propagates them to secondary delays on affected trains. This way, simulations can calculate total delays. As an example, a low total delay then indicates a good robustness against these primary delays. For large networks, to cover all cases or rather to obtain a representative collection of all cases, this means that many simulations have to be performed.

Barber et al. (2007) give a list of available software simulation packages (e.g.: RailSys of RMCon, SIMONE of NS). More recently, quite some more software packages have become available (e.g.: LUKS of ViaCon, and OnTime of TrafIT Solutions & ViaCon). Some packages simulate on the macroscopic or mesoscopic level, but quite some also simulate on the microscopic level. This means that they will detect conflicts between trains in the timetable, also inside stations, on routings or platform tracks and via the application of time shifts, they then calculate secondary and total delays. These conflicts arise due to primary delays in practice or due to the timetable not being realisable.

2.2. Our Approach: Expected Passenger Time Evaluation and its Advantages

Our evaluation approach is entirely macroscopic. Slightly simplified, it consists of evaluating the expected passenger time in practice for every action in the timetable (ride, dwell, transfer, knock-on delay) independently. This is possible by application of the expected passenger time per action which is an analytic expression of only the time supplement variable to be assigned to that action. For a given timetable, each action's supplement is known, so also its associated expected passenger time is known. The total expected passenger time of the whole timetable is then simply the sum of all expected passenger time of each of its actions. A complete derivation of our evaluation function can be found in Sels et al. (2013b,a, 2015b), where it plays the role of objective function in an optimisation context.

The more global focus on expected passenger time in practice, also shared by Dewilde et al. (2011) instead of the direct focus on stability, efficiency, robustness and resilience has at least three advantages. (i) Since minimising expected passenger time improves stability, efficiency, robustness and resilience, it is a combined performance indicator and the problems with evaluating a timetable over multiple objectives and setting priorities for objectives are resolved. (ii) Communicating that a new timetable has some percentage less expected passenger time in practice than the previous one is a very meaningful and clear way to communicate about timetables to management and passengers. (iii) We can *separately* evaluate the expected passenger time of every planned action: ride, dwell, transfer, knock-on delay, considering only its own primary delays. Indeed, the total expected passenger time in the system is then simply the sum of all expected passenger times of its constituent components. This means that, for evaluation or optimisation, we can decouple all actions and so do not suffer from having to make combinations of samplings of primary delays over all actions. There is one underlying assumption here, and that is that primary delays of different actions are independent. But in fact, that is also what most simulation packages assume, since they allow primary delay distributions to be entered per action (ride, dwell) and in simulations then combine primary delays over different actions. Only some research, like Kroon (2008) has taken great care of not assuming any primary delay dependencies by strictly keeping together all delays as measured in practice of one train occurrence as one 'trace'. Individual delays of different traces of one train are then not combined with each other in any simulation. This means that all unknown delay dependencies existing in reality, if there would be any, are preserved in the simulation or optimisation. The idea of avoiding the problem of finding a representative set of simulation traces by taking a more analytical approach is similar to what is done in the robustness analysis tool PETER, as described in Soto y Koelemeijer et al. (2000).

Note that our evaluation function - expected passenger time in practice - is already present as the objective function of our Periodic Event Scheduling Problem (PESP) Mixed Integer Linear Programming (MILP) model as published in Sels et al. (2013b,a, 2015b). In these papers, the goal was always to automatically find a new timetable that minimises the objective function. However, in this paper, we do not optimise a timetable, but compare two already existing ones. This means that we only need to evaluate the evaluation function for both timetables and see which timetable results in the lowest value to know which one is better for passengers.

2.3. Our Timetable Evaluation Method Contains Most Traditional Performance Indicators

To show that, with our evaluation function, we cover most of the traditional performance indicators for a timetable, and one more, we discuss these performance indicators one by one and relate them to our evaluation function.

2.3.1. Realisability

First of all we evaluate both timetables on *realisability*. This means that we check whether the time planned for each ride and dwell action is at least as large as is minimally required. In other words, ride and dwell actions should be assigned positive time supplements on top of the minima required for them. Our tool performs these checks.

2.3.2. Conflict-Freeness and Stability

Conflict-freeness, also sometimes simply called feasibility, means that in the timetable, without assuming delays, there are no conflicts between trains. On the macroscopic level this means that planned headways, between train departure times on the same resource and similarly between train arrival times, are larger than an assumed safe minimum. A common value for this macroscopic minimum is 3 minutes. Stability is defined as having sufficient time supplements, on top of the headway minimum time, between subsequent trains on the same track section to absorb delays on the first train. To find out where stability bottlenecks occur, this property can be analysed over every *cycle* in the *event activity graph* corresponding to the timetable (Soto y Koelemeijer et al., 2000). The event activity graph has a node for every train arrival or departure time in a station, representing an event and has an edge for every action (ride, dwell, transfer, headway) between two events representing an activity. In our objective function of expected passenger time, we do not explicitly check every *cycle* for this quality criterion, but for every inter-train edge (knock-on or transfer edge), we penalise a supplement that is chosen too low via the expected passenger time of this edge. This guarantees that every cycle will have a reasonable amount of supplement for it to become stable. This means that we do not only answer the question if every cycle has enough supplements present on its edges but also obtain a global score for how bad the stability is in terms of the summed expected passenger knock-on time and transfer time. If for some headway edges, the timetable would not be feasible, our objective function will give a very high penalty in terms of expected knock-on time. So in our evaluation method, both conflict-freeness and stability are handled by the terms of expected passenger knock-on delay. Note that, since we treat transfer edges similarly to knock-on edges, also with a minimum of three minutes but with a different evaluation function, we also obtain a feasibility for transfers which we could call *transfer guarantee* and also obtain a *transfer stability*.

2.3.3. Efficiency versus Robustness

The total expected passenger time in practice is also the property we use to evaluate both efficiency and robustness together. Very low supplements will make the timetable efficient for passengers, but not robust against primary delays so passenger travel times will vary. Very high supplements wil make the timetable very robust but not efficient. In this sense, we think it is not very practical to try to measure efficiency and robustness separately. Our method makes the trade-off in a passenger weighted way, because both efficiency and robustness are more important for trains with many than with few passengers. As for robustness against potentially occurring primary delays, we assume the primary delays on ride, dwell and transfer actions to occur according to a negative exponential distribution, as do Meng (1991); Goverde (1998); Vansteenwegen and Van Oudheusden (2006). We further assume the average 'a' of these distributions to be a certain percentage of the minimum time of that action. The value of 'a' is typically chosen in the range of 1% to 5% Goverde (1998). In this paper, we assume $a = 2\%$.

2.3.4. Temporal Spreading of Alternative Trains

In Sels et al. (2015a), it is shown how the inter-departure waiting time between alternative trains from a passenger's first station to this passenger's arrival station depends on the temporal spreading of these trains' departure times. So, this can be different in different timetables as well. Sels et al. (2015a) also indicate, that only some categories of passengers benefit from good temporal spreading. We set the percentage of passengers that do benefit to 'r'. Traditional simulators do not report on the level of temporal spreading and to which degree it benefits passengers, because traditionally, spreading is either enforced as a hard constraint for all passengers or not at all. However, quite some research papers did concentrate on evaluating the amount of inter-departure waiting time induced by a timetable (e.g.:Zhao et al. (2013)). Our method evaluates temporal spreading with the function described in Sels et al. (2015a).

2.3.5. Resilience

A timetable that is resilient against primary delays and disruptions recovers quickly, in a number of timetable periods, to its planned state without delays thanks to (i) good supplement choices that, to some degree, prevent knock-on delays and (ii) real time interventions of dispatchers (Goverde and Hansen, 2013). This means that, to properly evaluate the expected passenger time associated with resilience, one needs to be able to model both the recovery behaviour and also the precise dispatching rules. This is work that we did not address yet.

3. Results

We now apply the method as described in section 2.2 to the timetables T1 and T2. Both timetables have a planning horizon of 1 year. They each contain around 200 hourly passenger trains. In total, for about a 1000 physical points, being stations and stop places, train arrival and departure times are specified.

3.1. Realisability

Both in timetable T1 and in timetable T2, we find some cases where the process times assigned to ride actions are smaller than the required minimal run times. Timetable T1 contains 823 such cases and timetable T2 reduced this amount to 570 cases. Table 1 shows the size and amounts of these violations for both timetables. For timetable T1 there are 320 ride edges where the assigned process time is 6 seconds lower than the minimum allowed. The total violation time for all violations together for timetable T1 is 1909 times 6 seconds, or 190.9 minutes. For timetable T2 this is 1084 times 6 seconds or 108.4 minutes. Of course, the degree to which this is problematic is proportional to the number of passengers on the ride actions where these violations occur.

Table 1. Realisability. Reduction of the number and size of minimum runtime violations from timetable T1 to timetable T2.

timetable	distribution: # actions with a violation per size of violation in seconds											weighted sum (s)	tot.#	avg. (s)
	6s	12s	18s	24s	30s	36s	42s	48s	56s	60s	66s			
T1	320	219	126	93	24	27	3	6	1	3	1	11454	823	13.9
T2	277	155	84	37	11	2	2	2				6504	570	11.4

So realisability in terms of both number and size of violations improved from timetable T1 to T2. In practice, a non-realisable timetable will not cause train collisions but will cause primary delays, even in the absence of any other external factors interfering with the train and infrastructure system. Remark that the final goal should of course be to have no such minimum time violations at all in the timetable. Note that realisability is automatically 100% achieved for any timetable that is automatically produced by the optimisation method as described in Sels et al. (2015b).

3.2. Conflict-Freeness, Stability, Efficiency, Robustness & Temporal Spreading via Total Expected Passenger Time in Practice

In fact, the minimum ride and dwell time violations mentioned in paragraph 3.1 should be solved first before one can accurately evaluate passenger time. We achieve this by, before evaluation, correcting the assigned process times in the timetable to the minimum required durations, whenever they are lower. This is also what will happen in practice.

For the case $a = 2\%$, the total expected passenger time for both timetables is shown in figure 1. The left bar indicates timetable T1 and the right bar indicates timetable T2. The vertical dimension represents expected passenger time, also for its constituent components: ride (blue), dwell (yellow), transfer (orange), knock-on (purple). For dwell and transfer time, all ride time of the ride action preceding it, is convoluted with it, which is what the blue shading refers to. On the left of each bar, the percentages (T1.m and T2.m) indicate the ratio of the total expected passenger time due to the minima (m) to its total bar height. On the right, the percentages (T1.s and T2.s) indicate the ratio of the total expected passenger time due to supplements (s) to its total bar height. For each color, the minima are shown in a darker tone of the color and the supplements in a lighter tone of the same color.

For the case $r = 0\%$, we see that the total time reduction between timetables T1 and T2 amounts to 2.47%. This is a considerable improvement. This 2.47% reduction can be attributed for 70.3% to better timetabling and for 29.7% to a lower sum of minimum ride and dwell times. This latter part of the reduction is caused by the also changed line planning associated with T1 and T2. This changed line planning can cause that more passengers could choose a route with fewer ride and dwell actions or now even have a direct connection without transfer from their origin to

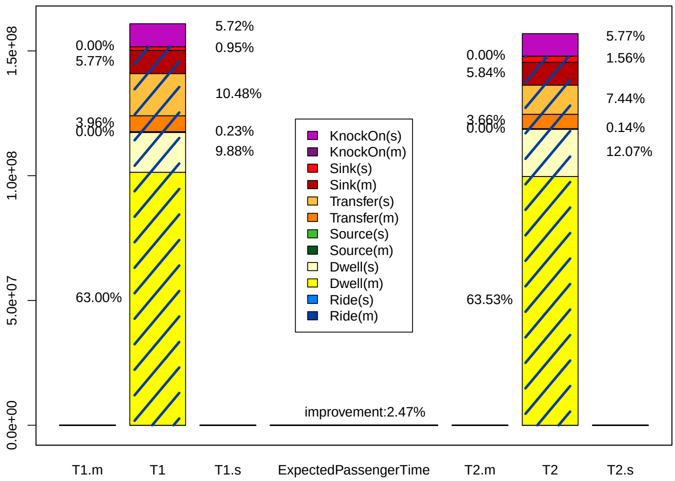


Fig. 1. Reduction of 2.47% of total expected passenger time from T1 to T2. All time units are in 6 second multiples.

destination. Other causes of the decreased minimal ride and dwell passenger time could be the reduction of run time minima by use of faster train material or the assignment of different train material on different lines (e.g. faster trains on lines with more passengers).

Figure 1 also shows that the transfer component in timetable T1 was responsible for 10.48% of the total expected passenger time, while in timetable T2 it only amounts to 7.44%. This corresponds to $1.687 \cdot 10^6$ min and $1.167 \cdot 10^6$ min respectively in absolute numbers, which is a considerable reduction of 31%. This could be due to more people having a direct connection instead of a transfer and also with better supplement planning for transfers. Figure 1 also shows that the total knock-on time did not change much between T1 and T2. It also shows that, in expected passenger time, more time is spent by passengers 'sitting out' these extra supplements. The goal of these extra supplements should be that the timetable becomes more robust and this should be visible in lower expected secondary delays. However, the purple component, representing expected knock-on delay does not decrease from T1 to T2. The expected passenger time spent in the ride and dwell supplements is 9.88% for timetable T1 and increases to 12.07% in the timetable T2. This larger expected time is caused by, on average over passengers, larger *planned* ride and dwell supplements and is intended to make the timetable more robust. If this higher robustness was obtained, it should be visible as a decrease of expected knock-on delay time that at least offsets the increased expected passenger dwell and ride time. However, as mentioned, the expected knock-on time does not decrease but remains roughly the same. Timetable T2 had as one of the major goals the improvement of punctuality over timetable T1. To obtain this, larger ride and dwell supplements were inserted. In practice, during implementation of timetable T2, it was indeed noted that punctuality, as measured by Infrabel has risen with a few percentage points compared to during implementation of timetable T1. This could be expected from the larger inserted ride and dwell supplements, but not from the expected knock-on delay which stays similar from T1 to T2. Note that Infrabel's punctuality measure currently only represents measurements at end stations of train lines and in Brussels, when these trains are at least 6 minutes late. It also does not weigh trains delays with affected passenger numbers. This could explain different findings between model and reality.

Using the method described in Sels et al. (2015a), we also evaluated both timetables on inter-departure waiting time between alternative trains and found that there is no considerable difference between these times for both timetables. This means that, when we add this time component to the total expected passenger time, the total reduction percentage will be somewhat smaller than 2.47%. However, we prefer to postpone detailed reporting of total expected inter-departure and arrival time until it is clear what the value of 'r' is in practice.

3.3. Expected Passenger Transfer Time Improvement

Because the expected transfer time for all passenger together decreased considerably form timetable T1 to timetable T2, we want to investigate how this was achieved in some more detail. We do this graphically in figure 2 by plotting different histograms of transfer times, on the top row, per transfer action, on the middle row, per transfer passenger in the planned domain and on the bottom row, per transfer passenger in the expected time domain. The left half of figure 2 again represents results for timetable T1 and the right half for timetable T2. The top row shows that there is little variation in planned time (minimum + supplement) assigned to transfer edges. The distribution is indeed quite uniform. Note that the total number of planned train time is increased from timetable T1 to T2 by $(668115 - 573426)/573426 = 16.5\%$. One should not conclude yet that this is a deterioration, since, for all time components, the *expected passenger* time domain instead of planned train time domain is the domain we should evaluate a timetable in and these results are shown in the bottom row.

The middle row shows that, when looking at the planned time assigned per transfer passenger, a pattern starts to appear. More specifically, transfers taken by many passengers tend to be assigned 'good' supplement values. 'Good' here means that the chosen supplements, together with the transfer minimum time, separate the arrival time of the feeder train and the departure time of the connecting train, are neither too small and neither too large. Too small would lead to a frequently missed transfer and our model penalises this with a waiting time of 1 hour for the next connecting train. Too large would mean that too much time has to be spent waiting at a platform for the connecting train. The middle row of figure 2 also shows that this good pattern is even better in timetable T2 than in T1. Indeed, the average planned passenger duration went down from 8.02 minutes to 7.76 minutes. This decrease of total planned passenger time (middle row) associated with an increase in total train time (top row) shows how important it is to weigh with passenger numbers. It also demonstrates that human timetablers also take this into consideration. The bottom row of figure 2 shows that in the *expected* time domain, the pattern is even more pronounced and the expected average transfer duration per passenger goes down from 19.2 minutes to 15.8 minutes. This reduction is a big improvement. The values 19.2 and 15.8 may seem large, but note that (i) this includes *all* possible (passenger weighted) transfers and not just the 'important' transfers from a restricted list and that (ii) our penalty of 1 hour for a missed transfer is a *worst case* one. In line with these findings of improved transfers is that we also calculated that, for $a = 2\%$, the average probability for a passenger to miss a transfer is reduced from 14.41% in timetable T1 to 5.51% in timetable T2. These calculations are based on a similar quick evaluation over all transfer edges, now not of the expected passenger time function per edge, but of the analytical function representing the probability of missing a transfer. The average probability of missing a transfer is then simply the passenger weighted average over all transfer edges.

4. Conclusions

As for our particular timetable evaluation methodology, we have shown that our evaluation function, expected passenger time in practice, has quite some advantages. These are essentially derived from the fact that weighing of different performance indicators is done in a natural way because all constituent components are expressed as passenger time. These advantages are (i) the full timetable evaluation problem can be decomposed in evaluating separate actions, (ii) the evaluation results of actions can be directly added, and ultimately, (iii) simulation is replaced with simple addition.

As for the application on the Belgian timetables, assuming that for each ride, dwell and transfer action, the average primary delay is 2% of the action minimum time, we can conclude that timetable T2 reduces total expected passenger time by 2.47% compared to timetable T1. This is a considerable improvement. This paper also shows that this 2.47% improvement is caused for 70.3% by timetabling improvements, chiefly by improvement of passenger transfer planning. Improved line planning is responsible for 29.7% of the 2.47% reduction. Under the same primary delay

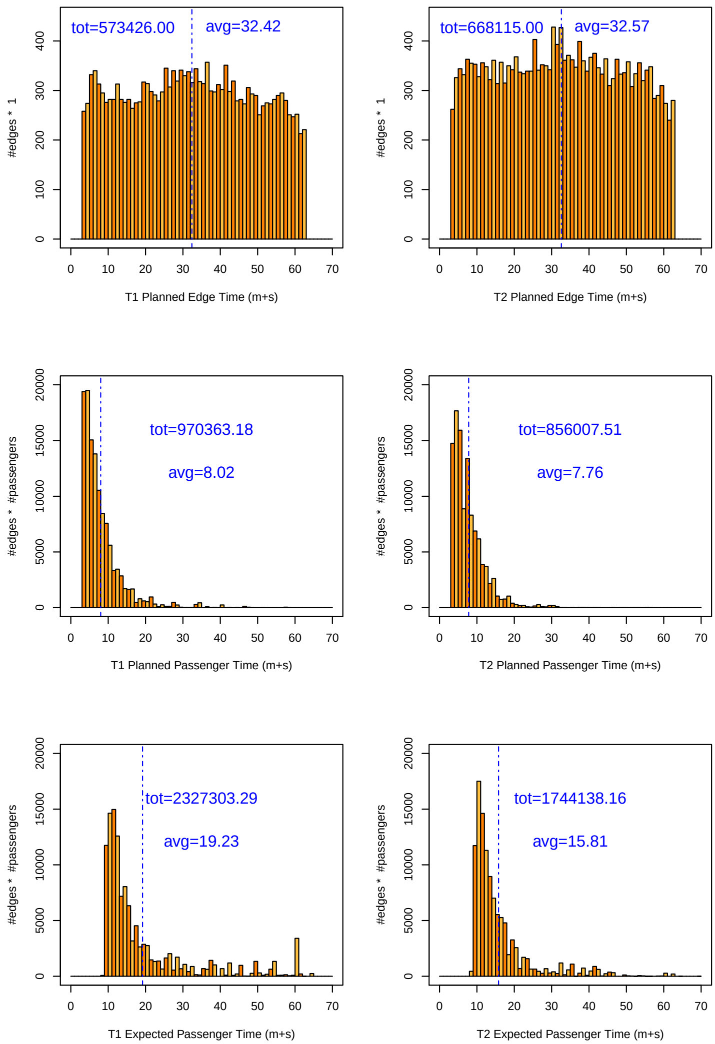


Fig. 2. Transfer time histograms for timetable T1 on the left and timetable T2 on the right. All time units are in minutes.

assumption, the average probability for a passenger of missing a transfer is reduced from 14.41% in timetable T1 to 5.51% in timetable T2. This is a huge improvement. The new timetable T2 has more expected passenger time spent in ride and dwell supplements than timetable T1. So the question arises if, in timetable T2, these supplements can be reduced to the level of timetable T1 while still keeping the advantage of the reduced time spent in transfers of timetable T2.

5. Further Work

As for our method of evaluation, evaluating the same timetables but for different values of the parameter '*a*' would be useful. Also, resilience of a timetable is not directly part of our method yet. This would require more research.

Our evaluation is conclusive about timetable T2 being preferable for passengers over timetable T1. When further manual improvement of timetable T2 is required, the question arises how to give the user feedback on how to achieve this. Answering this question is not easy. Indeed, indicating that, for example a transfer is badly planned, can be easily done. How to improve the transfer locally can also be indicated easily. However, making a local change will almost always affect other trains and it is generally hard to know if the potentially negative effect on other trains will not outweigh the desirable positive effect of the improved transfer planning. How to give hints for manual improvement could be considered in further research. However, note that via automatic timetabling optimisation, Sels et al. (2015b) were able to reduce the expected passenger time by 3.81% in 2 hours of computation time, by only changing the timetable and not the line planning. In the resulting timetable, expected knock-on time was decreased by about 50% and expected time of ride and dwell supplements was decreased, but expected transfer time increased.

References

Barber, F., Abril, M., Salido, M., Ingolotti, L., Tormos, P., Lova, A., 2007. Survey of automated systems for railway management. Technical Report. Universitat Politècnica de València.

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P., 2011. Defining Robustness of a Railway Timetable. Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011, February 16-18, Rome, Italy. .

Goverde, R., 1998. Optimal Scheduling of Connections in Railway Systems. Paper presented at the 8th WTCR, Antwerp, Belgium .

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.

Kanai, S., Shiina, K., , Harada, S., Tomii, N., 2011. An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. Journal of Rail Transport Planning & Management 1, 25–37.

Soto y Koelemeijer, G., Iounoussov, A.R., Goverde, R., Van Egmond, R.J., 2000. PETER, a performance evaluator for railway timetables, in: Computers in Railways VII, WIT Press, Southampton, UK. pp. 405–414.

Kroon, L., 2008. Stochastic improvement of cyclic railway timetables. Transportation Research Part B: Methodological 42, 553–570.

Meng, Y., 1991. Bemessung von Pufferzeiten in Anschlüssen von Reisezügen. Veröffentlichungen des Verkehrswissenschaftlichen Institutes der Rheinischen-Westfälischen Technischen Hochschule Aachen 46, 29–36.

Parbo, J., Anker, O., Prato, C.G., 2015. Passenger Perspectives in Railway Planning: A Literature Review. submitted to Transport Reviews .

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2013a. Knock-On Delay Modelling and Optimisation: The Passenger Perspective, in: Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013), December 2-4, Dresden, Germany., pp. 1–10.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2013b. Expected Passenger Travel Time as Objective Function for Train Schedule Optimization. Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark. .

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2015a. Optimal Temporal Spreading of Alternative Trains in order to Minimise Passenger Travel Time in Practice. Proceedings of 6th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailTokyo2015, March 23-26, Tokyo, Japan. .

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., 2015b. Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable. submitted to Transportation Research Part B .

Vansteenwegen, P., Van Oudheusden, D., 2006. Developing Railway Timetables that Guarantee a Better Service, journal = European Journal of Operational Research 173(1), 337–350.

Weston, P., Goodman, C., Takagi, R., Armstrong, J., Preston, J., 2006. Minimising train delays in a mixed traffic railway network with consideration of passenger delay. Proceedings of the 7th World Congress on Railway Research: WCRR2006, June 4-8, 2006, Montreal, Canada .

Zhao, J., Frumin, M., Wilson, N., Zhao, Z., 2013. Unified Estimator for Excess Journey Time under Heterogeneous Passenger Incidence Behaviour using Smartcard Data. Transportation Research, Part C: Emerging Technologies. 34, 70–88.

Chapter 3

Platforming and Routing in Stations

"Getting permission is harder than being forgiven."

– Banksy

"... yet, going for it without permission

can just hit you unforgivingly harder."

– Peter Sels

Currently, at Infrabel, when a timetable proposal is passed to them from their main operator NMBS, Infrabel adds some more trains, like international trains and freight trains from other operators. For the resulting macroscopic timetable, it needs to be checked that the train arrival and departure times it specifies can be realised also inside the stations. This depends on the available platform tracks, on the routing variants between the open lines and these platform tracks, on the dependency (crossing or equal) or non-dependency of these routing variants and on the simultaneity of trains using these platforms and dependent routing variants.

The train platforming and routing problem is essentially a vehicle routing problem (VRP). In the vehicle routing problem (Dantzig, Ramser (1959)), an optimal set of routes is to be found so that a given set of vehicles can deliver to a given set of customers. In our train platforming problem, we have a well known set of routes and what we deliver to customers is 'passengers', or essentially, the connectivity between station IN lines and station OUT lines. On top of a classical VRP we also have dependent (equal or crossing) routes that no two trains can use at any time.

The check if all trains can be platformed and routed is currently still performed manually at Infrabel. Planning for a large station without any platform or routing conflicts can take up to about two weeks even for an experienced human planner and is error prone. Research is available that shows that automation of this process is possible [3]. The question arises if we can use or/and improve this research and create a tool for Infrabel that can automate the resolution of these platform problems.

3.1 The Train Platforming Problem: the Infrastructure Management Company Perspective

Full Paper published in Journal Transportation Research, Part B: Methodological, 2014

We set up a model and implemented a tool for Infrabel, called Leopard. Leopard is based on a Mixed Integer Linear Programming (MILP) model that contains constraints forbidding any two trains to use the same platform at the same time. It also contains constraints forbidding any two trains to use dependent routing variants at the same time. The objective of Leopard is to platform and route as many train occupations as possible in the given station. We apply a filter to the train pairs, so that only for train pairs that can potentially hinder each other, constraints are formulated. This saves model creation time as well as model resolution time. The result is that Leopard can solve the average station platforming problem in about 0.25 seconds. Leopard is also useable to estimate station capacity. The user then supplies more train traffic that is expected in the future up to the point that this extra traffic cannot be platformed or routed anymore. At that point station maximum capacity is reached and is reported. When major over-saturation happens, computation times increase to some minutes per station. Using Leopard in this context gives Infrabel a station capacity estimation tool which it did not possess before.

Transportation Research Part B 61 (2014) 55–72

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

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] KU 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

ARTICLE INFO

Article history:
Received 17 December 2012
Received in revised form 14 January 2014
Accepted 17 January 2014

Keywords:
Train platforming problem
Mixed Integer Linear Programming
Train station capacity

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.

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.

* Corresponding author at: KU Leuven, Centre for Industrial Management/Traffic & Infrastructure, Celestijnenlaan 300a, 3001 Heverlee, Belgium. Tel.: +32 486956797.

E-mail address: sels.peter@gmail.com (P. Sels).

http://dx.doi.org/10.1016/j.trb.2014.01.004
0191-2615/

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 do not 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, and (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, 2011b,a) and Lusby 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 precalculate 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 s. Experiments including time shift variations required at most 400 s 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 10,000 s, 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 s.

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 s and one case had no solution in 1200 s. 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 s.

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 do not 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 min is one with 110 trains and 9 platforms on a time horizon of 2 h 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 Appendix A. 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 neighboring 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 40 km 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. (2014) 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 http://code.google.com/p/milp-logic/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 Fig. 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 Fig. 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 <u>o</u>ccupation, <u>m</u>ovement, <u>p</u>latform and <u>r</u>outing indices.

- The time $trhi_{o,m,r}$ is the <u>t</u>ime when the <u>r</u>outing has the <u>h</u>ead of the train at its beginning (<u>i</u>n side), which is also the end of the IN line.
- The time $trho_{o,p,m,r}$ is the <u>t</u>ime when the <u>r</u>outing has the <u>h</u>ead of the train at its very end (<u>o</u>ut side). This coincides with the <u>i</u>n side of the <u>p</u>latform, 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.

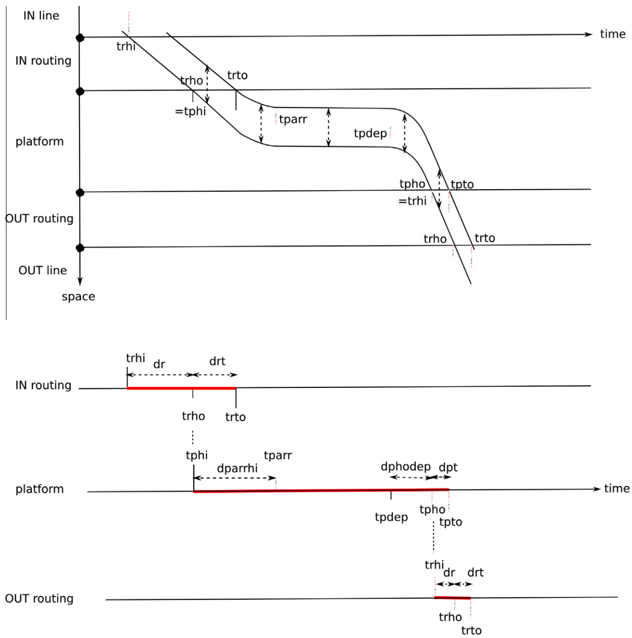


Fig. 1. Resource occupation time intervals and their durations.

- 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. Fig. 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.
  - $r2p : R \rightarrow P : r \mapsto p$ 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.
  - $m2o : M \rightarrow O : m \mapsto o$ 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 ($\Rightarrow$ middle of train is at middle of platform)
- $\forall o \in O : tpdep_o$ is the fixed departure/pass time at the platform ($\Rightarrow$ 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) Constant Low and High 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 $\equiv$ 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 Eqs. (5)–(7).

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

$$\forall o \in O : \forall p \in P_o : \begin{cases} \forall m \in M_{o,IN} : otLoLbC_{o,p} \leqslant otLoC_{o,p,m} \\ \forall m \in M_{o,OUT} : otHiUbC_{o,p} \geqslant otHiC_{o,p,m} \end{cases} \tag{5}$$

– 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:

$$\forall o \in O : \forall p \in P_o : \begin{cases} otLoLbC_o \leqslant otLoLbC_{o,p} \\ otHiUbC_o \geqslant otHiUbC_{o,p}. \end{cases} \tag{6}$$

These platform choice independent bounds $otLoLbC_o$ and $otHiUbC_o$ will be used in Eq. (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:

$$\forall o \in O : \forall m \in M_o : \forall r \in R_{o,m} \begin{cases} mtLoLbC_{o,m} \leqslant mtLoC_{o,m,r} \\ mtHiUbC_{o,m} \geqslant mtHiC_{o,m,r}. \end{cases} \tag{7}$$

The route choice independent bounds $mtLoLbC_{o,m}$ and $mtHiUbC_{o,m}$ will be used in Eq. (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 min to 3 min. The lexical preceding operator $\prec$ is used to avoid unnecessary definitions and constraints.
  - Before and separated booleans for occupation platform interval pairs:

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

In Eq. (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 Eq. (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:

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

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 $op_{m2o_m, r2p_r} = true$):

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

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 Eqs. (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

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

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

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

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:

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

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.

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*.

Station **row**	Bergen		Brugge		Denderleeuw		Gent		Leuven	
Conservative or progressive	C	P	C	P	C	P	C	P	C	P
# 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. ($r \rightarrow p$) **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. ($r \rightarrow p$) **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. ($r \rightarrow p$) **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. ($r \rightarrow p$) **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 ($r \rightarrow p$) **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 ($r \rightarrow p$) **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. ($r \rightarrow p$) **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. ($r \rightarrow p$) **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. ($r \rightarrow p$) **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. ($r \rightarrow p$) **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 ($r \rightarrow p$) **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 ($r \rightarrow p$) **3e**	0	0	19	15	15	14	26.7	26.7	−5	−9

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 Sections 4.4–4.6 are calculated prior to setting up the MILP model. For the ones in Section 4.6, Eqs. (1)–(7) are used to do this. The decision variables in Section 4.7 are added to the MILP model and their definitions (8)–(13) become linear constraints in this model, as do the constraints (14)–(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.

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.

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 min. Note that the Brugge and Gent cases are exceptions. CPLEX and Xpress do not 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 min 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 s.

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

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 2.6 GHz. Fico Xpress v7.2.1 Solver run on HP Intel Xeon Quad core 3.3 GHz. #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.

Station	Bergen		Brugge		Denderleeuw		Gent		Leuven	
Conservative or progressive	C	P	C	P	C	P	C	P	C	P
C: #P #R #O	7, 128, 58		10, 196, 87		9, 206, 91		12, 224, 563		14, 324, 160	
C: #C ∗ #V	7 k, 5 k		18 k, 9 k		13 k, 8 k		227 k, 107 k		46 k, 24 k	
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**	7199.8[1]	**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	96 k, 54 k		248 k, 96 k		140 k, 58 k		482 k, 231 k		101 k, 56 k	
CF: Solver Time Cplex (s)	138.1	847.4	**337.0**	7200.7[1]	27.7	78.2	**182.2**	7204.8[2]	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	7201.4[3]	**20.5**	**33.7**	n.a.[4]	n.a.[4]	**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	9 k, 7 k		1 k, 0.9 k		3 k, 2 k		1.0 k, 0.8 k		3 k, 2 k	
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	9 k, 7 k		3 k, 4 k		14 k, 9 k		4 k, 6 k		21 k, 9 k	
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

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 s 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 min 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 do not 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.

Appendix A. Comparison of features in TPP literature

See Tables A.1 and A.2.

Appendix B. 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 Fig. 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

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	Level	Approach	Technique	Different route variants	Platf. times var.	Cont. time model	Different		
							d_r	$v_{r,max}$	l_t
Zwaneveld et al. (1996)	S	Time & platform	MILP	Y	Y	N	Y	Y	Y
Zwaneveld (1997)		Choice graph, NPP							
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)	S, T	Human planner Inspired B& B heur.	MILP	N	Y	N	Y	Y	N
Carey and Carville (2000)	S, T	Human planner	Greedy	N	Y	N	Y	Y	N
Carey and Carville (2003)		Inspired heuristic	Heuristic						
Carey and Crawford (2005)		Human planner	Greedy	N	Y	N	Y	Y	N
		Inspired heuristic	Heuristic						
Caprara et al. (2007)	T	Time & Platform	Branch&	Y	Y				
Caprara et al. (2011a)		Choice graph, NPP	Price&cut						
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.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.

Publication	Stations tested	Traffic tested	Optimal solution	Solver times		Integrated	
				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) Delorme et al. (2001)	1 Real, Italian 1 Real, P.G.	6 Real 4 Projected	N Y	T:0.49 S: $\geqslant$1000	T:115 S:10 k	 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) Sels et al.	1 Real, P.G. 10 Real Belgian	20 Generated 10 Real + 10 Projected	Y Y	 T: 3.15 S: 374	 T:27.6 S: 2853	 Infrabel BE	 Ocapi Leopard

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 Fig. 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 min. 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. Fig. 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.

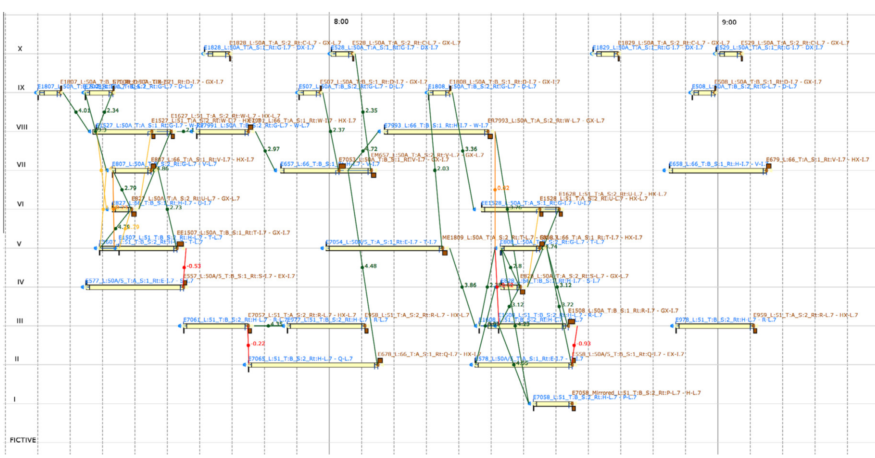


Fig. B.1. Brugge station platform occupations. Original schedule. All occupations are assigned to real platforms, but 4 cases of simultaneous dependent-route use (red lines) and 2 cases of dependent-route reuse within 1 min (dark orange lines) occur. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

P. Sels et al./Transportation Research Part B 61 (2014) 55–72 71

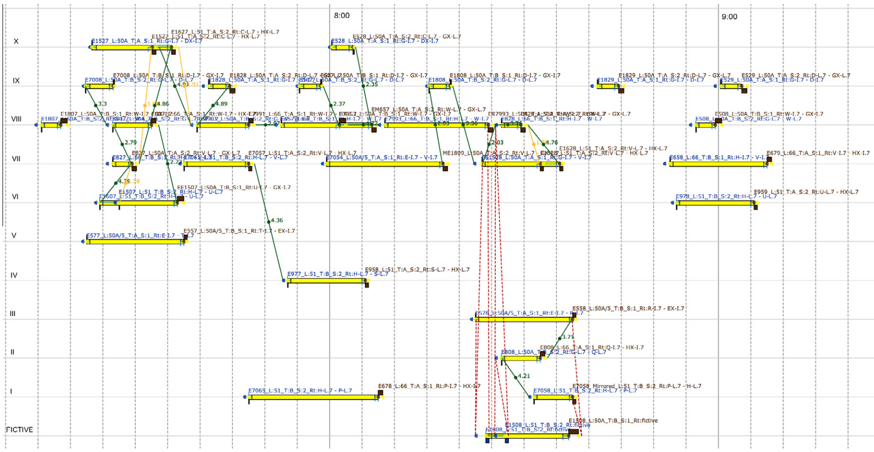


Fig. B.2. Brugge station platform occupations. Progressively optimized schedule. No cases of simultaneous dependent-route use nor of dependent-route reuse within 1 min occur. However, one occupation cannot be put on a real platform and ends up on the fictive platform.

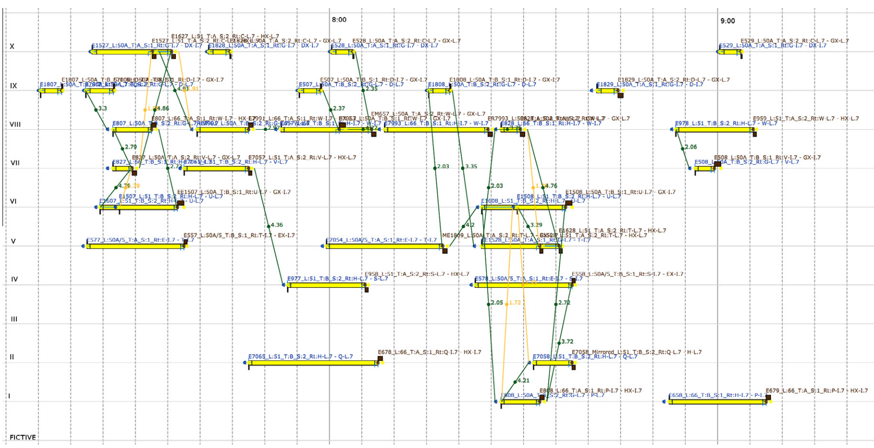


Fig. B.3. Brugge station platform occupations with manually, only 2 slightly adapted platform times. Progressively optimized schedule. All occupations are assigned to real platforms and the fastest dependent-route reuse occurs with $d_{min} = 1.7$ min, which is acceptable.

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 Fig. B.2. Dependent-route reuse lines now only have positive values for d. Here, the only ones below 2 min 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.

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 with -1 minute. The result is shown in Fig. 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 Fig. B.1.

We have shown, here for Brugge, with Fig. B.1, that conflicts exist in the current schedule, with Fig. B.2, that one must change platform times or not all trains can be platformed and with Fig. 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, pp. 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., 2007. Passenger railway optimization. Handbooks in Operations Research and Management Science, vol. 14. Elsevier, Amsterdam, pp. 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., 2014. Improving the robustness in railway station areas. European Journal of Operational Research 235 (1), 276–286.

Fuchsberger, M., 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., 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., 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., September 2012. milp-logic: A C++ MILP Solver Abstraction Layer with a C++ Boolean Modelling Layer on Top. <http://code.google.com/p/milp-logic>.

Sels, P., Dewilde, T., Cattrysse, D., Vansteenwegen, P., February 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. <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. In: Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13–15, Copenhagen, Denmark. <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., 1994, . Model Building in Mathematical Programming, 3rd ed., 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.

3.2 Automated Platforming & Routing of Trains in All Belgian Railway Stations

Full Paper submitted to Journal: Expert Systems With Applications, 2015

We use the tool Leopard as described in the previous section 3.1 and apply it to all Belgian stations. All stations pass and are resolved in 0.25 seconds on average. In total, the same amount of train occupations is platformed and routed as found in the platforming solutions performed by human planners and retrieved from the Infrabel databases. Leopard also automatically produces a spreadsheet of platforming solution indicators per station, like number of trains platformed, number of conflicts and a robustness value. For easy comparison, it juxtaposes these numbers on one line per station, both for the original and, next to it, for the optimised platform plan. Totals over all selected stations are also produced so that also the total performance of the automated platforming can be judged. Even though the robustness of the platforming plan is not part of the objective function, averaged over all stations, the robustness of the Leopard generated solutions is significantly better than the human made solutions.

This shows that Leopard can be used for both checking existing platforming plans and automatically generating new platforming plans for each station in Belgium. As such, it can also be used as a fast, automatic more microscopic feasibility check of an existing or changed macroscopic timetable. We believe that deploying and using Leopard would save Infrabel a lot of time and human error. Also, since only platforming plans without conflicts are generated, fewer delays would be occurring in stations. This would then also benefit train passengers.

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 and routing 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.

As an example, the graphical output of Leopard for a larger station is shown and discussed. We also produce an overview of key performance indicators for all stations in Belgium. We run our model for periodical and non-periodical platforming and also for platforming with or without penalties for deviating from an original platforming plan.

Platforming in practice in railway companies is often still done manually. Although in some cases simulation software is used in order to support and evaluate this manual work, we believe our optimisation system is an important step forward that allows generating platform plans more quickly and in a less error prone way.

Keywords: Train Platforming & Routing 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

Preprint submitted to Elsevier *April 23, 2016*

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.

1.5. Research Questions

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.

1.6. 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, 6 and 7 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).

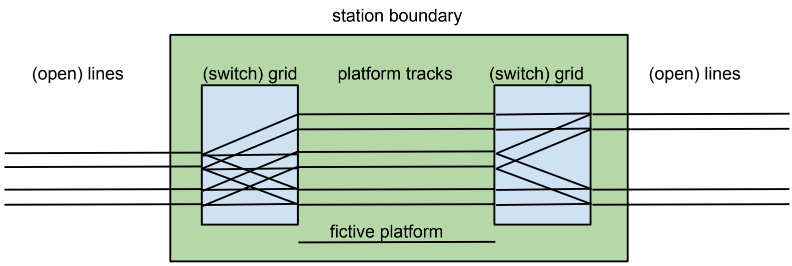


Figure 1: Platforming on Infrastructure

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. So, for this we had to use another data source. Infrabel uses the simulator LUKS of ViaCon and entered the whole Belgian macroscopic and microscopic infrastructure in this application. This includes all the routes that are possible for trains in all stations. For all these routes, speed restrictions were entered and route lengths can be deduced. LUKS can write out a file with all these routes, that includes these speed restrictions, the route lengths and the components that occur on these routes. Route components are switches, signals and tracks. Leopard then reads this file and derives which route pairs are independent and which are dependent. Two routes are dependent if they share at least one component. With this information, Leopard imposes constraints only between dependent route pairs, so that it avoids that two trains simultaneously use these routes. With the speed restriction information, the model uses the maximum speed that is allowed on the route tracks and calculates the corresponding minimal total time necessary to traverse the route. The information from the LUKS file is added to the corresponding routes from the Infrabel database and as such the platforming model for a station can be set up.

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

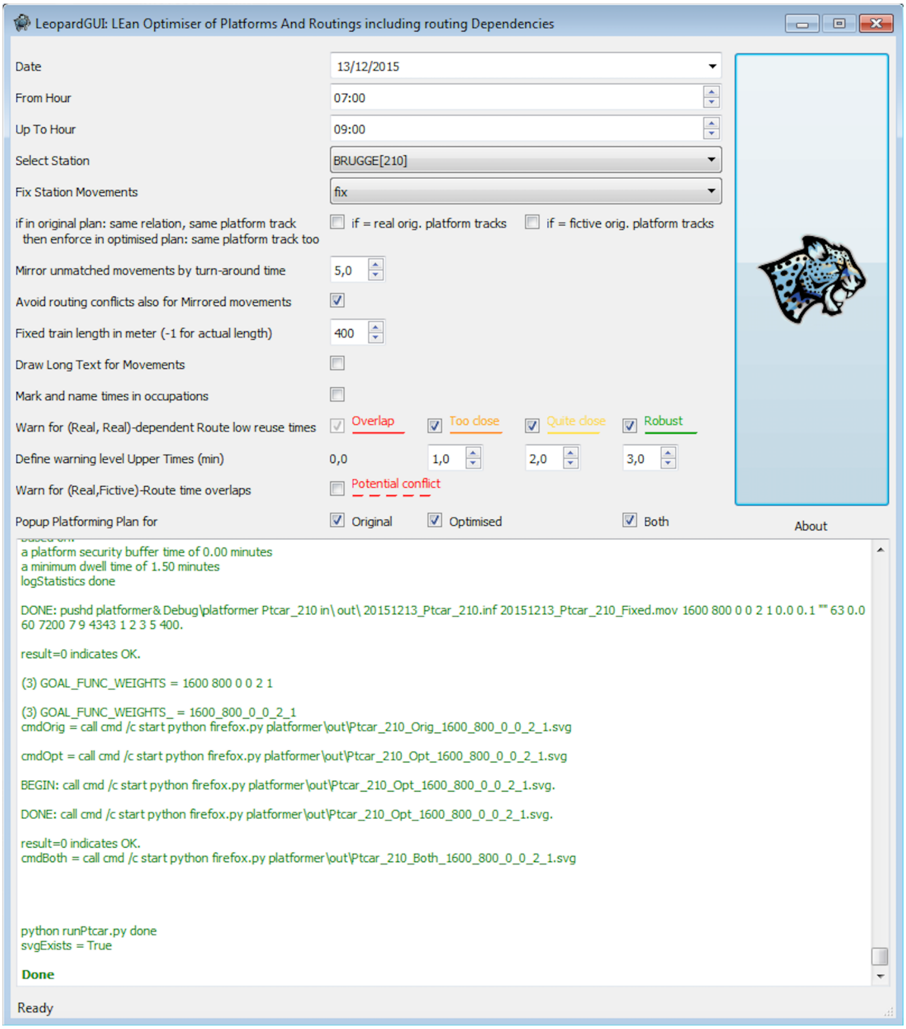


Figure 2: Leopard input GUI

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. These conditions are user selectable. 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 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. Note that train occupations on the fictive platform track in the original platform plan point to human planners either not being able to plan them on a real platform track or just not having performed the job yet. If both checkboxes are left unchecked, Leopard will not spend effort to platform any occupation pairs of the same 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.

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

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

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. It could be that one occupation cannot be put on the only possible platform for it, say platform 5 because there is another train occupying it. It could also be that the reason is a combination of conditions. For example, if the occupation f on the fictive platform were to be platformed on the still free platform 5, occupation x's out-movement would be in conflict with the in-movement of occupation f. Also much more complicated reasons involving many more occupations and nested conditions could prevent occupation f from being platformed on a real platform. We have chosen to 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.

The green text in the bottom half of figure 2 is progress log text of Leopard. It tells the user what Leopard is busy with and gives an idea of how long it will still take. In case the input database contains inconsistencies, the user will see an error there as well. The last output lines show in which directory the output files in Scalable Vector Graph (SVG) format are written to before these files are popped up as tabs in the browser. The main advantages of the SVG format are that all modern browsers can visualise these files, that it is possible to jump from one place to another when the user clicks an object and that it is possible to show information dynamically and conditionally like when a user hovers over an object. These features are used by our system as will be discussed in the next section.

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 each in respectively figure 3, 4 and 5. These are all platforming plans for the station of Namur. All three figures show 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.

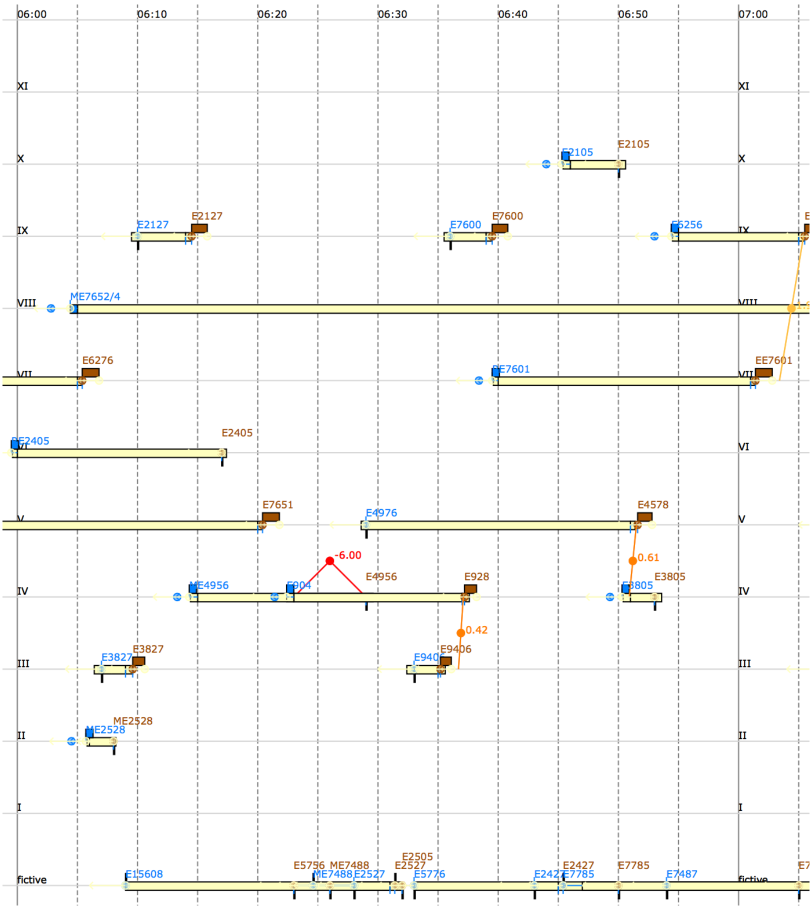


Figure 3: Namur original platform plan, created by human planners and drawn by Leopard

Figure 3 shows the manually created platform plan. In this paper, we will also call this the *original* plan. Figure 3, 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 to be considered as 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. So if train E9406 would be delayed by 0.42 minutes or more and train E928 would be on time, both trains would be using dependent routes at the same time. 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, so a dark orange line also connects these movements, giving a warning that robustness is low here. Around 07:04, a light orange line is drawn from OUT-movement EE7601 leaving from platform VII to the OUT-movement E6277 leaving from platform IX. The text '1.96' next to this light orange line indicates that 1.96 minutes are left between these movements. With the default user settings in Leopard, the following thresholds correspond to the following colours. If the time between movements on dependent routes is between 0 and 1 minute, the line colour is dark orange. If the time between movements is between 1 and 2 minutes, the line colour is light orange. If more than two minutes passes between such movement pair, the line colour is green, meaning there is no major robustness issue between these movements. 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. Note that the numbers marked at all dark orange, light orange and green lines consistently are positive. This means that there is no time overlap of the usage intervals of dependent routes. 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. Similarly to the original platforming plan in figure 3, there are also some train occupations assigned to the fictive platform track. In other words, not all trains can be assigned to real platforms and routes without generating a platform or routing conflict. The train split occupation [E15608 - E5756, E2505] and the occupation [E7487, E7670], are assigned to the fictive platform. In the original platforming plan of figure 3, these same two occupations were not assigned to a real platform either. However, in the original platform plan, four more occupations were assigned to the fictive track: ME7488, E2527, E2427 and E7785. 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 5 shows both the original and the optimised platform plan for Namur, interleaved in one picture. The original platform plan has been shifted up by half a platform division to avoid rectangle overlap between the original and optimised plans. When loading this platform plan in a browser, one can click on any rectangle of the original plan and some integrated javascript code will then scroll the picture up to the corresponding rectangle in the optimised plan and vice versa. This makes it easy to see which platform occupations have changed. Note that in quite some cases the chosen platform track in the optimised platform plan is not far removed from the chosen platform in the original platform plan. This is because the objective function for this case contained terms that penalise deviations from the platform track in the original plan. This penalty is proportional to the number of tracks deviated.

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 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

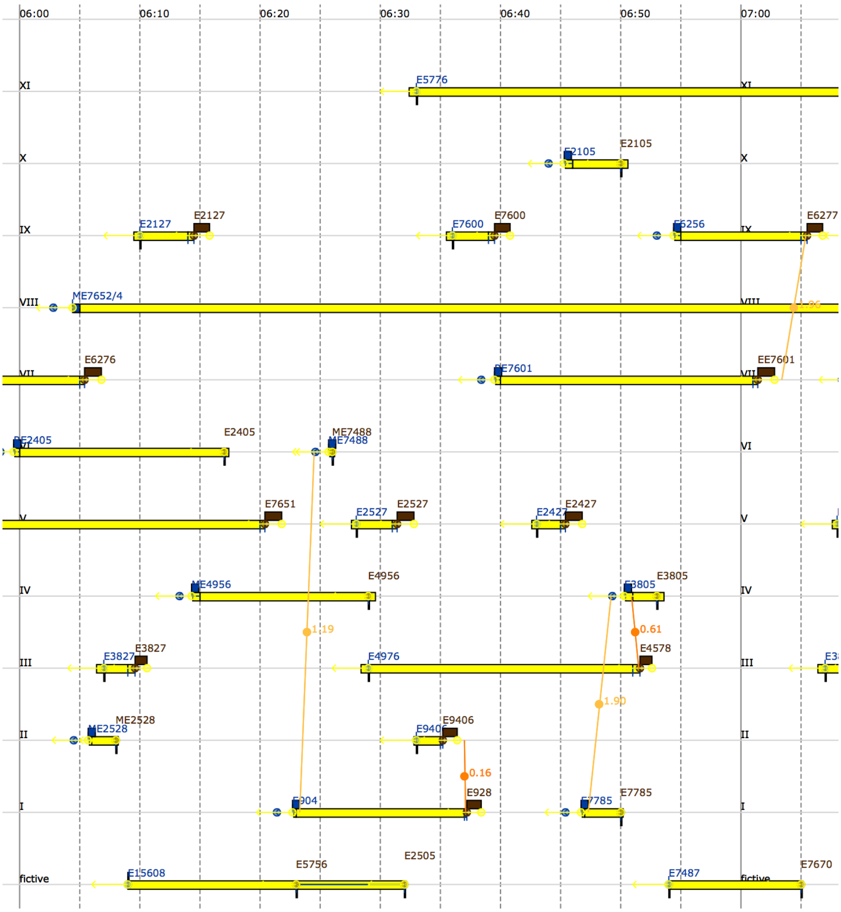


Figure 4: Namur optimised platform plan, generated and drawn by Leopard

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.

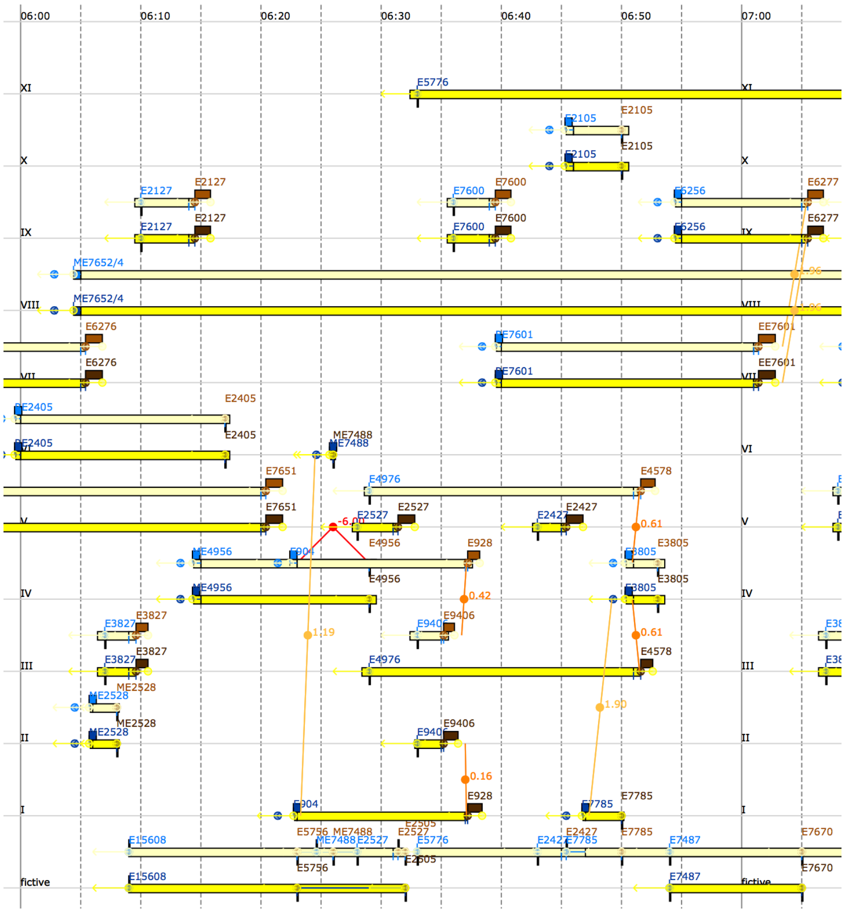


Figure 5: Namur original and optimised platform plans drawn by Leopard for human comparison

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 tables 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 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 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

Table 2: Optimisation time in seconds. Stations 1 to 8 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
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

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 sta-

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

tions: 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 5. 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.

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

#platforn	#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						

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

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						

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 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.

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 ($\leq 0'$)	158	0	0	0	0	0	0	0	0
red route conflicts ($\leq 0'$)	187	0	0	0	0	0	0	0	0
dark orange indications ($\leq 1'$)	534	175	178	151	151	131	126	147	126
light orange indications ($\leq 2'$)	1453	608	627	578	579	548	544	575	544
green indications ($\leq 3'$)	2206	1180	1243	1112	1110	1020	1025	1058	1025
robustness score	-7800	-1308	-1339	-1182	-1183	-1072	-1048	-1163	-1048

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.

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 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.

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

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.

5. Answer to the Research Questions

The previous sections discussed our platforming tool and its results in detail. This section gives a more high level visual impression of the quality of manually versus automatically platformed solutions over all stations in figure 6. This helps to answer all research questions that were raised in section 1.5.

5.1. Summarizing Comparison between Manually and Automatically Generated Platforming Plans

Figure 6 corresponds to the results of platforming traffic between 6am and 9am. For the automatic part, we took the results produced by the solver Gurobi when run on our model in progressive mode, where occupations on real and fictive platform tracks are not required to be planned periodically. This corresponds to column 3 in table 6.

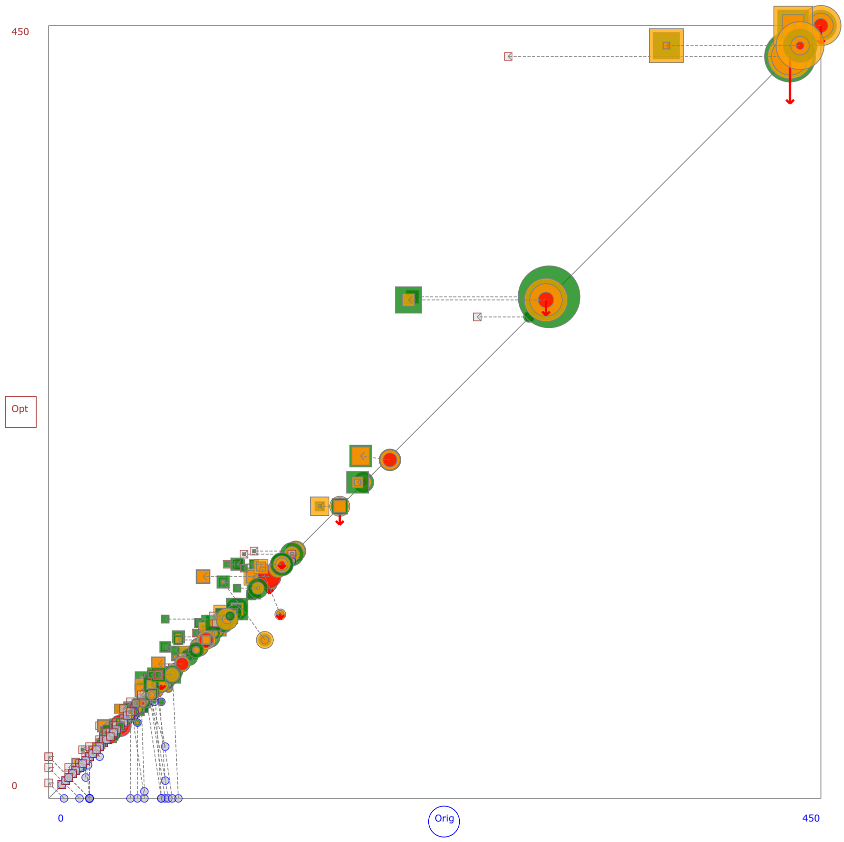


Figure 6: Comparison of achieved number of platformed occupations between original (circles) and optimised (squares) platform plans, including indications of platform conflicts (red downwards arrows), routing conflicts (red circles) and routing major robustness issues (dark orange circles or squares), minor robustness issues (light orange circles or squares) and safe routing reuse times between 2 and 3 minutes (green circles or squares).

For the original platform plans, the horizontal axis in figure 6 shows the number of total occupations and the vertical axis shows the number of manually platformed occupations. A set of concentric circles in the lower right triangular half of figure 6 represents manually platformed stations. If the center of these circles lies on the line describing $x = y$, all trains were platformed. If this center point lies below it, less than 100% of the occupations was platformed in the manual plan. For the optimised platform plans, squares instead of circles, in the upper left triangular half are used to represent platforming results for stations. The center point of a square set represents how many train occupations could be platformed by Leopard for that station. When this center point lies on the line $x = y$, 100% was automatically platformed. If $x < y$, the point lies a bit more to the left, meaning that less than 100% of occupations could be automatically platformed. The colour of each disc or square: red, dark orange, light orange and green refers to the colour of respectively, conflicts, major and minor robustness issues and safe separation occurrences, as they were explained in section 3.2. The disc and square areas are proportional to the number of these respective conflicts, major and minor occurrences and safe separation occurrences. So an original platform plan for a station that shows a red disc contains routing conflicts. A red downward pointing arrow indicates that in that station's original platforming plan, some platform conflicts exist. The length of the arrow is proportional to the number of platform conflicts occurring in that plan. So a red downward arrow suggests that the original platforming plan still has platform conflicts and should not be considered to score as close to the diagonal line as the position of the circles suggests.

What is immediately obvious from figure 6 is that only for a few stations, a significant amount of traffic cannot be platformed by Leopard while it seems to be platformed by the human planners. However, in each of these cases, a significant amount of platform and routing conflicts exists in the human created plans, as can be seen by respectively, the red downward arrows and the red circles. Leopard generated plans have no single platform nor routing conflict as can be seen by the absence of red squares and absence of red left pointing arrows. So, figure 6 makes visually clear that the cost of being strict about avoiding every single platform and route conflict in the planning is that one can sometimes not platform the full 100% of occupations. Figure 6 also shows that when there are no conflicts in the original plan, Leopard also finds a solution with a similar percentage of occupations platformed. The fact that robustness for all stations together is better for the automatically generated platform plans than for the manually generated platforming plans can be seen in figure 6 by noting that there are fewer red squares (none) than red circles and fewer/smaller dark/light orange squares than dark/light orange circles.

For each specific station, the concentric circles representing the original platform plan are connected with a dashed arrow to the squares representing the optimised platform plan. For most stations, there are no very long arrows, so no huge changes in number of occupations planned automatically compared to manually. However, some stations were not manually platformed at all. These stations are represented in figure 6 on the x axis at low values of x. The fact that manual platform plans are lacking for these stations can be due to the fact that this was not done yet at the time we evaluated the platform plans or because planners did not consider these stations were worth checking or otherwise because they forgot to check them. Of course, the automatic platformer quickly runs over all stations without exception. For most of these stations, Leopard can platform all occupations or a high percentage of them. One can indeed see that the grey circles end up as grey squares on or close to the diagonal line. For a few stations only, a low percentage, but still higher than 0%, can be platformed but for three stations, none of the occupations could be platformed. For these three stations, no routes were defined yet. This problem would be resolved if Infrabel defined routes in these stations as well.

We conclude that figure 6 nicely summarises the main aspects of a fair comparison between human and computer generated platforming plans of all stations in Belgium. It shows the percentage of platformed occupations in a way that allows visual comparison between the two plans for each station. It shows the amount of remaining platform as well as remaining routing conflicts in red in the manual plans. It shows the lack of these conflicts in the optimised plans. It also shows the amount of robustness issues in orange in both plans. It also shows whether a station was not yet platformed manually and whether a station could not be platformed automatically due to lack of defined routes. As a result it also clearly shows if all stations were platformed or not. The ideal end result for figure 6 would be that all squares of stations are displayed on the diagonal line and there is as little orange present as possible.

5.2. Answer to the Research Questions and Gained Insights

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 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 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. Leopard then makes smaller deviations only and still checks for errors.

6. Conclusions

For Infrabel, we developed a tool called Leopard 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. Without going into microscopic detail, the presence of any platforming or routing issues in a station here already indicate that the macroscopic timetable arrival and/or departure times need to be adapted in that station. Manually adapting the timetable until all stations can place all trains on real platforms is the first context of use of Leopard. When such a ‘platformable’ timetable is obtained, within Leopard, station per station, independently, various objective functions can be explored to obtain the most desirable platforming plan for each station. For example, one could choose to stay close to the platforms chosen in the original platform plan and/or one could choose to generate a platforming plan where trains of the same periodic relation are assigned to the same platform track.

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. A plot comparing several performance criteria between the manually and the automatically generated platform plans is also generated.

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. On average, over all stations, Gurobi gives a somewhat better robustness score than XPRESS, but note that robustness was not part of the objective function.

7. 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.

Chapter 4

Other Publications as First Author

During my PhD time I published more abstracts and papers. I do not include the total text of these here, but merely refer to them. They can be found online via the web-linked titles or via the references.

4.1 Deriving all Passenger Flows in a Railway Network from Ticket Sales Data

Extended abstract presented at ORBEL, Gent, Belgium, 10-11 February 2011

- Authors: Peter Sels, Thijs Dewilde, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [30]

The idea is presented that local passenger numbers riding on trains, dwelling on trains and transferring between trains can be derived from train ticket sales data and knowledge of the train network.

4.2 Calculation of Realistic Railway Station Capacity by Platforming Feasibility Checks

Extended abstract presented at Models and Technologies for Intelligent Transportation Systems (MT-ITS), Leuven, Belgium, 22-24 June 2011

- Authors: Peter Sels, Thijs Dewilde, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [40]

Our initial tool made for Infrabel that is able to calculate station capacity estimates by platforming larger and larger sets of trains is presented. Typically, the current set of trains is used and additional traffic that is expected in the future is added to it. Each time the tool is run and the solution with the maximum number of trains that can be platformed and routed without conflicts is calculated. If not all trains can be platformed or routed anymore, station capacity is exceeded. This tool was integrated at Infrabel in their planning tool OCAPI.

4.3 Automated, Passenger Time Optimal, Robust Timetabling, Using Integer Programming

Short paper presented at International Workshop on High-Speed and Intercity Railways (IWHIR), Hong Kong - Shenzhen, 19-22 July 2011

- Authors: Peter Sels, Thijs Dewilde, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [29]

The first, preliminary results of the timetabling tool RhinoCeros, written for Infrabel, on the set of all Belgian hourly trains are presented.

4.4 Automatically and Quickly Planning Platform and Route of Trains in Railway Stations with a Case Study for Mechelen Station

Full paper at International Journal Conference for Technology and Engineering (IJCTE), Bangkok, Thailand, 1-2 January 2014

- Authors: Peter Sels, Thijs Dewilde, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [34]

A case study of platforming and routing for the station of Mechelen is presented.

4.5 Light Rail, Tram and Bus co-Timetabling Minimising Passenger Travel Time in Practice

Extended abstract submitted to Triënnial Symposium of Transportation Analysis (TRISTAN), Oranjestad, Aruba, 13-17 June 2016

- Authors: Peter Sels, Niels van Oort, Sandra Nijënstein and Pieter Vansteenwegen.
- Reference: [38]

In the above mentioned timetabling work, we have consistently produced cyclic timetables for train networks that span an entire country. We now want to explore if the used methodology and tools can also be used to produce cyclic timetables for urban transport networks. The studied network is an intermodal one with three modes of transport: light rail, trams and buses. It is clear that there are some major differences. We mention some of them. (i) Buses do not have to be separated by a minimum time in the timetable model since bus drivers are taking care of avoiding collisions in real time. (ii) Traditionally, train timetables specify both arrival and departure times at stations, while bus and tram timetables usually only specify one station time. (iii) For buses and trams, other vehicles continuously share lanes or tracks, which is not the case for trains, so bigger stochastic riding time variations are expected than for trains. Even with these differences, we are hopeful that the same software will be able to generate timetables for these urban intermodal systems as well.

Chapter 5

Co-Authored Publications

During my PhD time, Thijs Dewilde was also doing research towards his own PhD. We proof read and co-authored each other's papers. This resulted in two more conference and two more journal publications with Thijs as first author. We enlist these here in sections 5.1 to 5.4.

5.1 Defining Robustness of a Railway Timetable

Full paper presented at International Association of Railway Operations Research (IAROR) Rome, Italy, 16-18 February 2011

- Authors: Thijs Dewilde, Peter Sels, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [7]

Various definitions as given in the literature are given. The function of expected passenger time in practice, including primary and secondary delays is presented. This is a function that can be evaluated over the timetable. It is concluded that the timetable that minimises this function is good for the totality of passengers, since it brings them from their origin to destination as fast as possible, statistically over many daily operations of this timetable. So by definition of robust as a concept over statistical occurrences of delays, It is also concluded that this timetable will be robust against delays, It is argued that, unlike some other definitions, this is a definition of robustness that is practically usable to compute timetables. Computing such a timetable with this method is exactly what we manage to do for this thesis, for Belgium and for Denmark.

5.2 Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming (IAROR)

Full paper at International Association of Railway Operations Research (IAROR) Copenhagen, Denmark, May 13-15 2013

- Authors: Thijs Dewilde, Peter Sels, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [9]

This paper studies the area of the Brussels North-South axis as well as the station of Antwerp. It optimises timetabling, routing and platforming in one unified approach instead of sequentially by first timetabling and then platforming together with routing as this thesis does. Obviously a unified approach can potentially generate a better optimum than a sequential one since interaction indeed occurs.

5.3 Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming (JRTPM)

Full paper in Journal: Rail Transport Planning & Management, 2013

- Authors: Thijs Dewilde, Peter Sels, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [10]

For the paper in section 5.2, Thijs Dewilde won the Young Railway Operations Research Award 2013 of IAROR. Consequently, Thijs was invited to write a reworked version of the paper for the JRTPM journal. After reviews by this journal, this is the resulting paper.

5.4 Improving the robustness in railway station areas

Full Paper published in Journal: European Journal of Operational Research, 2013

- Authors: Thijs Dewilde, Peter Sels, Dirk Cattrysse and Pieter Vansteenwegen.
- Reference: [8]

This paper extends the previous ones by adding a simulator and its results and by suggestions for improvement of the traffic flow. The case study is the train traffic bottleneck in Belgium: the Brussels area.

5.5 Unification of Research

While I looked at timetabling at the country level separately and then at platforming and routing for every station separately, Thijs essentially looked at optimisation of timetabling, platforming and routing together, at the same time, but for smaller areas. He studied the areas of the Brussels North-South axis and the station area of Antwerp. We envision that our work could possibly be integrated in the following way. It is expected that simultaneous optimisation of timetabling, platforming and routing delivers better results than the sequential optimisation of first timetabling and then platforming and routing. Therefore, first, for some critical areas, the simultaneous optimisation of Thijs could be tried. Then the total country-level timetabling problem could be tackled, with added boundary conditions imposing the earlier found local solutions at the boundaries of these critical areas. Of course, when several critical areas all impose boundary conditions, it is possible that these boundary conditions cannot be simultaneously satisfied. Finding out how to avoid this would require further research.

Chapter 6

Conclusions and Further Work

"A world without problems is an illusion,

so is a world without solutions."

– From "Puzzillusions' , by Gianni A. Sarcone

In section 6.1, we make a summary of the main results we obtained and how these are useful to Infrabel and train passengers. Of course, as always, more can be done and we make suggestions for future work in section 6.2.

6.1 Conclusions

We conclude by presenting the main results we achieved during this PhD. Subsequently, we want to recommend some practices to railway companies. We then give suggestions for further improvement of our work.

6.1.1 Results

From ticket sales data, we were able to derive local passenger numbers on all trains in all locations in the train network by implementing a routing algorithm. Thanks to parallellising this algorithm over all cores and over several machines in the network, we can perform this routing task arbitrarily quickly. The results of this algorithm are later used in our timetabling approach. We then managed to construct workable timetabling and platforming processes that are both quicker than the current manual ones at Infrabel and also both result in higher quality plans. As for timetabling, compared to the state of the art in Periodic Event Scheduling Problems (PESP) research before, we can claim that we are the first to add a complete objective function that controls all timing supplements in the whole macroscopic timetable. We are also the first to produce a timetable for an entire country from a PESP model that includes such a complete objective function. In fact we did this for both all 196 Belgian hourly passenger trains as for all 88 Danish hourly passenger trains. As for platforming, we created a model that platforms typical train traffic of a three hour window in 0.25 seconds per Belgian station on average. This is also quicker than previous research reports on similar problems. We also produce a cumulative report for all stations that compares key performance indicators of the current platform plan and the optimised platform plan. Over all stations, this report shows that the number of platformed trains is almost identical but that the robustness of the optimised platform plans is much better.

Both timetabling and platforming systems are implemented, integrated and usable at Infrabel. The resulting better plans should also result in operational benefits. Beneficial effects of the generated macroscopic timetable should be especially noticeable to passengers, who will experience quicker effective end-to-end connections and less missed transfers. According to the rules used at Infrabel on a macroscopic level, the platforming plans have no train platforming nor routing conflicts anymore and this will lead to less real time delays in practice in stations. This will also be beneficial to passengers.

Since it is the first time that a macroscopic train timetable has been generated for an entire country, or two, now is the time that railway companies *can* consider the use of our approach and its generated timetables. Since our approach also predicts a significant reduction of passenger time in practice, 3.8% for Belgium and 2.9% for Denmark, as well as probabilities on missed transfers dropping from more than 10% to less than 3%, now is also the time that they *should*. Today, passengers often complain about 'unnecessary' long travel times, frequent delays and missed connections. Adopting a method that reduces all of these phenomena, where perception *is* reality, can only be to the passenger's benefit and will reduce their complaints. Their increased satisfaction and associated increased willingness to pay for a better train service can only trickle down and be translated into beneficial effects for the train operator companies and then further for Infrabel.

6.1.2 Recommendations to Railway Companies

During our 6 years of our consulting job at Infrabel, we noticed that some processes could be improved. All three recommendations would have very large positive impact on operations. The first two are very easy to implement. The last one is harder, but tools to do this are now both developped and integrated at Infrabel and this thesis fully explains the rationale down to the details of this process.

Record and Keep Ticket Sales Data at the Station Level

For a train operator, it is very important to record and keep passenger demand data for various reasons. As shown in chapter 2, for timetabling the ticket sales data serve as input data to calculate the number of passengers on trains. We concluded that it is best to keep record of ticket sales data at the station to station level, rather than at the coarser, less accurate zone to zone level. This data should be kept as a non-symmetric origin-destination (OD) matrix. Morning and evening OD-matrices can be added together for some purposes, but the original matrices should be kept as recorded.

Improve Cooperation between Train Operator and Infrastructure Manager

From within Infrabel, it has been impossible to obtain recent ticket sales data (OD-matrix) from our main operator, NMBS. While we understand that this is sensitive information, we think it should be possible for an operator to trust that the infrastructure manager will not pass this information to third parties. If this information is not received by the railway infrastructure company, it becomes impossible for them to optimise the planning towards what is important for their customer, the operator. So it is in the train operator's benefit to pass this information.

The infrastructure company is the party responsible for the final integrated timetable. The infrastructure company is the only party that can possess the OD information of all its clients and combine (add) them. Consequently, it is the only party that can perform the global optimisation for all its clients together for the greater good of all. This includes consideration of passenger transfers between trains of two different clients.

Embrace Optimisation instead of Only Simulation

Currently, Infrabel constructs timetables manually and then *simulates* the previous timetable and the new timetable and compares punctuality results between both. In some areas, the new timetable will be better than the previous one while in other areas the opposite will be true. This is useful information to predict what will happen in reality. However, more useful questions are: (i) “How to improve the timetable in the areas that are worse off?” and (ii) “How to improve the timetable in the areas that are worse off, without losing the improvement made in the better off areas?”. These questions trigger the next question (iii) “What does ’better’ mean?”. To answer this, one needs to answer the question: “What is the objective of a timetable?”. Simulation can never give an answer to the questions of type (i), (ii). Indeed, simulation is a ’forward process’ and gives no answers to ’backwards’ questions like “If I want to reach or maximise or minimise (so ’optimise’) my objective, how do I need to change the control parameters?”. To answer this, and to answer all questions of type (i), (ii) and (iii) mentioned above, one needs *optimisation* and one needs to clearly specify an *objective (function)*. This is exactly what we did in this thesis and we highly recommend railway companies to follow the optimisation paradigm. Simulation alone will require a lot of trial and error and human effort for a medium quality result. Optimisation will lead to higher improvement according to a clearly defined objective and it will do so in a fraction of the time and with a fraction of the human effort. The results mentioned in section 6.1.1 are a clear proof of this.

6.2 Useful Further Work

The platforming problem, as defined by Infrabel, with fixed arrival and departure times, we consider as mainly solved. For timetabling, and also its integration with platforming, some further work is certainly interesting. We mention some points of possible improvements in sections 6.2.1 to 6.2.8.

6.2.1 Improvement of Input Data

"In God we trust, all others bring data."

– from"The Elements of Statistical Learning"

– by Trevor Hastie, Robert Tibshirani and Jerome Friedman

The ticket sales data we used dates from 2002. New numbers could not be obtained from NMBS. If one supposes that the relative increase of passenger demand from 2002 to now is the same for all origin-destination pairs, they will not affect the timetable that is generated. However, any inhomogeneous increase in demand will have an effect on the produced optimised timetable. It is therefore highly preferrable to have recent, dependable input data.

Currently, we still suppose the same type and scale of primary delay distributions for all trains and all network locations and on all activities; ride, dwell and transfers. More specifically, we assume that the average of the negative exponential primary delays of an activity amounts to $a\%$ of the minimum of that activity. For ride activities this minimum includes a 5% markup on the calculated minimum run time. We typically assumed this $a\%$ to be in the range from 2% to 5%. In practice, the value of $a\%$ could vary per train and per location. So, we could tune the primary delay distributions imposed during optimisation to the delay distributions we measure in practice with the timetable in operation. This seems more realistic. Note that one must also be cautious with this method because the measured delays may for some part be dependent on on the timetable itself.

6.2.2 Further Reduction of the MIP Gap

Given all assumptions mentioned in the papers, the timetables we produced for Belgium and Denmark are respectively 3.8% and 2.9% better than the original timetable in terms of the objective function. These are both solutions of the

MIP models that correspond to MIP gaps of more than 70%. A MIP gap of x% means that the solver has mathematically proven that the optimal solution is at most $x\%$ better in terms of the objective function that the solution currently found. It does not guarantee that a solution with a gap smaller than $x\%$ exists. When we give the solver more time, sometimes the MIP gap decreases very little after a few additional hours or days. We do not think that solutions that are 70% better really exist, but it is clear that the solver is unable to quickly decrease the MIP gap. We do not know the reason for this slowness. With smaller train networks as well as with bigger assumed primary delays (bigger values of 'a%'), lower values of MIP gaps are more quickly reached.

Since we think our objective function is what it must be for passengers, only methods of cleverly restricting the search space without excluding too many good solutions would be the measures to take. Since, with primary delay distributions that have a larger average ($a\%$), solutions with a lower MIP gap are more quickly reached, one could first try to find a solution with a very large $a\%$ and then use it as an initial solution of the model as generated with the real value for $a\%$. Maybe also other specific cycle constraints could help, for example cycles connecting alternative trains. Fixing certain orders for alternative trains could also potentially help speed up computation without losing feasibility or much of the optimality.

6.2.3 Addition of Boundary Conditions

Currently, the timetable is computed supposing that there are no constraints on when trains enter or leave the country. In practice, neighbouring countries have meetings, prior to changing their timetables, where they agree on time windows for all trains that pass boundaries between their countries. To provide for this in our timetabling tool, these boundary conditions could be added as additional constraints to the model. Thanks to the flexibility of Integer Linear Programming, this can be done without having to change the current constraints nor the objective function in the current model. Of course, added boundary conditions may increase or decrease the computation time. Inside the country under consideration, similar boundary conditions can also be added for sub-areas that may have a prefixed solution that was computed with another method. Likewise, boundary conditions may be needed to specify when bridges open or close at waterways. If many boundary conditions are specified, it is possible that the model becomes infeasible. In fact, it is of course good to know in advance when this is the case, since it means that a timetable with these requirements cannot be found. The boundary conditions will then have to be relaxed until the model becomes feasible.

6.2.4 Scalable Optimisation of Temporal Spreading of Alternative Trains

The excess journey time component of the total journey time of passengers is equivalent to the time passengers wait for their train at their station of departure plus the time they wait for the next mode of transport at their station of arrival. When we add this excess journey time to the objective function of our optimisation, the optimisation process becomes much slower. In fact, up to now, the largest train network we can solve with this feature enabled is one with 26 hourly trains. With more trains, computation times became impractically large. For the whole of Belgium, we need 196 hourly trains. So, more research is required to make handling this excess journey time in the objective function more efficient.

6.2.5 Further Reduction of the Expected Transfer Time Component

From the original timetable to the automatically generated optimised timetable, we typically reduce the total expected passenger travel time by some percentages like 3.8% or 2.9%. However the expected transfer component of these total times still increases from the original to the optimised timetable. Preferably, this component by itself should also be reduced, ideally without losing the decrease of total expected passenger time we already obtained or even further reducing it. Indeed, one wants short transfer times in practice, but not so short that they have a high chance of being missed. Some preliminary ideas on how to obtain this are requiring that important transfers take less than 10 minutes or take about the same time as in the original timetable. So rather than changing the objective function. this would restrict the search space. Another idea is to apply hard-soft spreading of trains on major corridors. As discussed in section 2.6, this method was beneficial for the reduction of excess journey time, but also seemed to have the side effect of restricting the increase of expected transfer time. Unfortunately, as mentioned in section 6.2.4, large networks still take too much computation time for optimisations with this extended objective function. Trying out these ideas would require further experiments.

6.2.6 Execute Multiple Reflowing and Retiming Iterations

We optimise a timetable by first flowing passengers over the network of trains in our graph model as they would choose their trains in reality. We then know an estimate for local passenger numbers on every train. Next, with these numbers fixed, we adapt the train arrival and departure times to minimise the total expected passenger journey time in practice. We arrive at a new timetable, which, when put in practice, would lead to some passengers again choosing a new route. This means that the first phase of reflowing passengers would have to be carried out again, and then the next retiming phase could be carried out. We should reiterate until passenger time does not decrease anymore before we put a timetable into practice.

6.2.7 Coupling of Timetabling and Platforming

Our platforming tool can currently automatically platform and route a maximum possible number of trains in all stations for the current timetable as it is present in the Infrabel databases. We could also perform platforming and routing for all stations for the automatically generated new timetable if these tools would be connected. This has not been done yet. It is a matter of additional coding rather than extra research work.

When timetabling and platforming are coupled up, the issue arises of how to resolve the situation of a station where not all trains can be platformed. The arrival or/and departure times of unplaced trains or/and trains hindering unplaced trains can be slightly shifted in time. To properly organise this and certainly when wanting to construct a system that can guarantee to always find a solution for this, information needs to be passed from the platforming to the timetabling tool and back again. We currently have no such system in place. Recently, research tools came into being, that are able to do this during multiple iterations between the macro timetabling level and micro platforming and routing level and have been demonstrated to work for smaller areas [9, 6, 1]. These approaches could also work for larger areas or could at least be a source of inspiration.

When connecting our timetabling tool to our platforming tool, the opportunity arises to pass along the number of passengers on trains entering and leaving a station. This information could be used in the objective function of the platforming model. As such preference could be given to platform and route important trains over less important trains. Additionally, the number of passengers on all transfers in the station could be passed to the platforming tool and the platformer could try to place train couples on nearby platform tracks when many people transfer between these trains. Currently all trains are still considered to have the same importance and transfers do not play a role in platform selection.

6.2.8 Further Cooperation on Timetabling

We are interested in further cooperation with rail companies like Infrabel or/and NMBS. We believe we have an approach to timetabling that will benefit them. Our approach is automated, which will save them planning time and effort. We have also shown that with our timetable, thanks to its better allocation of time supplements, passengers will save 3.8% of passenger time for Belgian case. Apart from that, our method is also guaranteed to produce timetables without run time violations and without train conflicts, which will save trains and their passengers even more travel time. Our simulations of both the original and the optimised timetables with independent commercial tools confirm that train punctuality would also significantly increase.

Bibliography

[1] Nikola Bešinović, Roberto Roberti, Egidio Quaglietta, Valentina Cacchiani, Paolo Toth, and Rob Goverde. Micro-macro approach to robust timetabling. *Proceedings of 6th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailTokyo2015, March 23-26, Tokyo, Japan.*, pages 1–21, 2015. pages 10, 11, 234

[2] Alain Billionnet. Using Integer Programming to Solve the Train Platforming Problem. *Transportation Science*, 37(2):213–222, 2003. pages 11

[3] Alberto Caprara, Laura Galli, and Paolo Toth. Solution of the Train Platforming Problem. *Transportation Science*, 45(2)(2):246–257, 2011. pages 11, 174

[4] Alberto Caprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. *Passenger Railway Optimization. Handbooks in Operations Research and Management Science*, volume 14. Elsevier, Amsterdam, 2007. pages 2

[5] Dorotea De Luca Cardillo. k L-List τ Colouring of Graphs. *European Journal of Operational Research*, 106(1)(1):160–164, 1998. pages 11

[6] Thijs Dewilde. *Improving the robustness of a railway system in large and complex station areas*. PhD thesis, Katholieke Universiteit Leuven (KU Leuven), Leuven, Dec 2014. pages 10, 234

[7] Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Defining Robustness of a Railway Timetable. *Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailRome2011, February 16-18, Rome, Italy.*, 2011. pages 223

[8] Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. 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., 2013. pages 225

[9] Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Robust Railway Station Planning: An Interaction Between Routing, Timetabling and Platforming. *Journal of Railway Transport Planning & Management (JRTPM). (Young Railway Operations Research Award 2013 of IAROR.)*, 3:68–77, 2013. pages 10, 224, 234

[10] Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Improving the robustness in railway station areas. *European Journal of Operational Research*, 235:276–286, 2014. pages 10, 224

[11] R.M.P. Goverde and Ingo Hansen. Performance indicators for railway timetables. *Proceedings of IEEE International Conference on Intelligent Rail Transportation: ICIRT2013, August 30-September 1, 2013, Beijing, China.*, pages 301–306, 2013. pages 6

[12] Rob M.P. Goverde. The Max-Plus Algebra approach to railway timetable design. In B. Mellitt, R.J. Hill, J. Allan, G. Sciutto, and C.A. Brebbia, editors, *Computers in Railways VI*, pages 339–350. WIT Press, Ashurst, 1998. pages 7

[13] Peter Großmann, Steffen Hölldobler, Norbert Manthey, Karl Nachtigall, Jens Opitz, and Peter Steinke. Solving Periodic Event Scheduling Problems with SAT. In He Jiang, Ding Wei, Ali Moonis, and Wu Xindong, editors, *Advanced Research in Applied Artificial Intelligence*, volume LNAI 7345, page 166–175. Springer-Verlag Berlin Heidelberg, 2012. pages 7

[14] Dennis Huisman, Leo Kroon, Ramon M. Lentink, and Michiel Vromans. Operations research in passenger railway transportation. *Statistica Neerlandica*, 59(4):467–497, 2005. pages 2

[15] Leo Kroon, Rommert Dekker, Maróti Gábor, Mathijn Retel Helmrich, and Michiel Vromans. Stochastic Improvement of Cyclic Railway Timetables. Technical report, Erasmus Research Institute of Management (ERIM), 2006. pages 7

[16] Leo Kroon, Dennis Huisman, Erwin Abbink, Pieter-Jan Fioole, Matteo Fischetti, Gábor Maróti, Alexander Schrijver, Adri Steenbeek, and Roelof Ybema. The New Dutch Timetable: The OR Revolution. *Interfaces*, 39:6–17, 2009. pages 7

[17] Leo Kroon, H. Edwin Romeijn, and Peter J. Zwaneveld. Routing Trains Through Railway Stations: Complexity Issues. *European Journal of Operational Research*, 98(3)(1):485–498, 1997. pages 11

[18] Michael Kümmling, Peter Großmann, Karl Nachtigall, Jens Opitz, and Reyk Weiß. The State-of-the-art Realization of Automatic Railway Timetable Computation. In *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013), December 2-4, Dresden, Germany.*, pages 1–10, 2013. pages 7

[19] Christian Liebchen. *Periodic Timetable Optimization in Public Transport*. PhD thesis, Technische Universität Berlin, 2006. pages 7, 8

[20] Christian Liebchen. Periodic Timetable Optimization in Public Transport. *Operations Research Proceedings*, 2006:29–36, 2007. pages 7

[21] Richard M. Lusby, Jesper Larsen, Matthias Ehrgott, and David Ryan. Railway track allocation: models and methods. *OR Spectrum 2011*, 33(4):843–883, 2011. pages 2

[22] Karl Nachtigall. Periodic network optimization with different arc frequencies. *Discrete Applied Mathematics*, 69:1–17, 1996. pages 7

[23] Karl Nachtigall and Jens Opitz. Solving Periodic Timetable Optimisation Problems by Modulo Simplex Calculations. In Matteo Fischetti and Peter Widmayer, editors, *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*, volume 9 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. pages 7

[24] Michiel A Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B*, 30(6):455–464, 1996. pages 7, 8, 79

[25] Leon W.P. Peeters. *Cyclic Railway Timetable Optimization*. PhD thesis, Erasmus Resaerch Institute of Management (ERIM), Rotterdam, 2003. pages 8

[26] Marco Rinaldi, Farzad Fakhraeiroudsari, Francesco Viti, and Chris Tampère. Improving the accuracy of OD estimation from traffic counts employing a partial observabillity maximizing methodology. *Annual meeting of the transportation board edition:95 location:Washington D.C. date:11-14 January 2016*, 2016. pages 18

[27] Alexander Schrijver and Adri Steenbeek. Spoorwegdienstregelingontwikkeling (Timetable Construction). *Technical Report, CWI Center for Mathematics and Computer Science, Amsterdam*, 1993. pages 7

[28] Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Automated Platforming & Routing of Trains in All Belgian Railway Stations, under review. *Expert Systems With Applications.* pages 12

[29] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Automated, Passenger Time Optimal, Robust Timetabling, using Mixed Integer Programming. In *Proceedings of the 1st International Workshop on High-Speed and Intercity Railways (IWHIR), Hong Kong*, volume 2, pages 87–92, 2011. pages 220

[30] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Deriving all Passenger Flows in a Railway Network from Ticket Sales Data. *Booklet of Abstracts of 25th Conference of Operational Research Belgium (ORBEL): Orbel2011, February 10-11, Gent, Belgium.*, 2011. pages 219

[31] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. 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.*, 2011. pages 8

[32] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. A Passenger Knock-On Delay Model for Timetable Optimisation. In *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2013), December 2-4, Dresden, Germany.*, pages 1–10, December 2013. pages 4, 8

[33] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. 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.*, 2013. pages 4, 8

[34] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Automatically and Quickly Planning Platform and Route of Trains in Railway Stations with a Case Study for Mechelen Station. *Proceedings of the International Conference on Civil and transport Engineering (ISAET): ISAET2014, January 1-2, Bangkok, Thailand.*, 2014. pages 221

[35] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Towards a Better Train Timetable for Denmark, Reducing Total Expected Passenger Time. *Proceedings of the 13th Conference on Advanced Systems in Public Transport (CASPT2015), 19-23 July 2015, Rotterdam, The Netherlands.*, July 2015. pages 4, 8

[36] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Optimal Temporal Spreading of Alternative Trains in order to Minimise Passenger Travel Time in Practice. *Proceedings of 6th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailTokyp2015, March 23-26, Tokyo, Japan.*, February 2015. pages 8

[37] Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable, accepted. *Transportation Research Part B*, 2015. pages 4, 8

[38] Peter Sels, Niels van Oort, Sandra Nijënstein, and Pieter Vansteenwegen. Light Rail, Tram and Bus co-Timetabling Minimising Passenger Travel Time in Practice. *Proceedings of the 9th Trienial Symposium on Transportation Analysis (TRISTAN): TRISTAN IX, June 13-17, Oranjestad, Aruba.*, 2016. pages 221

[39] Peter Sels, Pieter Vansteenwegen, Thijs Dewilde, Dirk Cattrysse, Bertrand Waquet, and Antoine Joubert. The train platforming problem: The infrastructure management company perspective. *Transportation Research Part B*, 61:55–72, 2014. pages 12

[40] Peter Sels, Bertrand Waquet, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Calculation of Realistic Railway Station Capacity by Platforming Feasibility Checks. In *Proceedings of the 2nd International Conference on Models and Technologies for Intelligent Transport Systems (MT-ITS 2011), June 22-24, Leuven, Belgium*, number 031, pages 1–4, 2011. pages 220

[41] Paolo Serafini and Walter Ukovich. A Mathematical Model for Periodic Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2:550–581, 1989. pages 6, 39, 76

[42] Daniel Sparing, Rob M.P. Goverde, and Ingo A. Hansen. An Optimization Model for Simultaneous Periodic Timetable Generation and Stability Analysis. *Proceedings of 5th International Seminar on Railway Operations Modelling and Analysis (IAROR): RailCopenhagen2013, May 13-15, Copenhagen, Denmark*, 2013. pages 7

[43] Evelien van der Hurk, Leo Kroon, , Maróti Gábor, and Peter Vervest. Deduction of Passengers' Route Choice from Smart Card Data. *IEEE-ITSC2013 on Intelligent Transport Systems, to appear 2014*, Dec 2014. pages 17

[44] Pieter Vansteenwegen and Dirk Van Oudheusden. Developing Railway Timetables that Guarantee a Better Service. *European Journal of Operational Research*, 173(1)(1):337–350, 2006. pages i

[45] Pieter Vansteenwegen and Dirk Van Oudheusden. Decreasing the passenger waiting time for an intercity rail network. *Transportation Research Part B*, 41(4)(4):478–492, 2007. pages i

[46] Peter J. Zwaneveld, Leo Kroon, H. Romeijn, Marc Salomon, Stephane Dauzère-Perès, Stan P. M. Van Hoesel, and Harrie W. Ambergen. Routing Trains Through Railway Stations: Model Formulation and Algorithms. *Transportation Science*, 30(1):181–194, 1996. pages 11

Curriculum Vitae

Peter Sels obtained his Master of Science in Electrical Engineering at Group T, Institute of Technology, Leuven, Belgium in 1994 and his Master of Science in Computation at Oxford University, U.K. in 1996. He worked as a digital chip design engineer and C++ programmer constructing chip design tools at Easics. Subsequently, he was a Logically Yours consultant to Altran De Valck Brussels, Alcatel Space Antwerp, Camco Technologies Leuven and Philips Research Eindhoven. From 2009 to 2015 he was a consulted to the Belgian railway infrastructure management company Infrabel as well as performed research towards his PhD at KU Leuven on the topics of macroscopic cyclic timetabling and train platforming and routing in stations. He currently consults to N-SIDE. His research interests are cyclic timetabling, energy market design, balancing and clearing.

List of Publications

Articles in Internationally Reviewed Academic Journals

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2013) "Robust railway station planning: An interaction between routing, timetabling and platforming". In: *Journal of Rail Transport Planning & Management* 3(3), pp. 68–77.

Sels, P., Vansteenwegen P., Dewilde T., Cattrysse D., B. Waquet and A. Joubert. (2014) "The train platforming problem: The infrastructure management company perspective". In: *Transportation Research Part B: Methodological* 61, pp. 55–72.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2014) "Improving the robustness in railway station areas". In: *European Journal of Operational Research* 235(1), pp. 276–286.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2015) "Reducing the Passenger Travel Time in Practice by the Automated Construction of a Robust Railway Timetable". In: *Transportation Research Part B: Methodological.*

Sels, P., Cattrysse D. and Vansteenwegen P. (2016) "Automated Platforming & Routing of Trains in All Belgian Railway Stations". Submitted to: *Expert Systems And Applications.*

Article in Academic Book, Internationally Recognised Scientific Publisher

Sels, P., Dewilde T., Vansteenwegen P. and Cattrysse D.. (2011) "Automated, Passenger Time Optimal, Robust Timetabling, using Integer Programming". In: *Proceedings of the 1st International Workshop on High-speed and Intercity Railways (IWHIR 2011)*. Ed. by Y.-Q. Ni and X.-W. Ye. Vol. 148. Lecture Notes in Electrical engineering. Springer-Verlag Berlin Heidelberg, pp. 87–92.

Papers at International Scientific Conferences and Symposia, Published in Full in Proceedings

Sels, P., B. Waquet, Dewilde T., Cattrysse D. and Vansteenwegen P. (2011) "Calculation of Realistic Railway Station Capacity by Platforming Feasibility Checks". In: *Proceedings of 2nd International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. Ed. by F. Viti, C. Tampère and B. Immers. Leuven, Belgium.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2011) "Deriving all Passenger Flows in a Railway Network from Ticket Sales Data". In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome)*. Rome, Italy.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2011) "Defining robustness of a railway timetable". In: *Proceedings of 4th International Conference on Railway Operations Modelling and Analysis (RailRome)*. Rome, Italy.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2012) "Solving the train routing problem to decrease knock-on delay at station level". In: *Proceedings of 10th International Conference on Applied Mathematical Optimization and Modelling (APMOD)*. Ed. by L. Suhl, G. Mitra, C. Lucas, A. Koberstein and L. Beckmann. Paderborn, Germany, pp. 256–261.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2013) "A Passenger Knock-On Delay Model for Timetable Optimisation". In: *Proceedings of 3rd International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. Ed. by T. Albrecht, B. Jaekel and M. Lehnert. Dresden, Germany, pp. 377–387.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2013) "Expected

Passenger Travel Time for Train Schedule Evaluation and Optimisation". In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann and J. Pachl. Copenhagen, Denmark.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2013) "Robust railway station planning: An interaction between routing timetabling and platforming". In: *Proceedings of 5th International Conference on Railway Operations Modelling and Analysis (RailCopenhagen)*. Ed. by I. Hansen, O. Nielsen, A. Landex, J. Törnquist Krasemann and J. Pachl. Copenhagen, Denmark. Young Railway Operations Research Award 2013 of the International Association of Railway Operations Research (IAROR).

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2013) "Exploiting the potential of the available capacity in railway station areas to improve the robustness". In: *Proceedings of eighth Triennial Symposium on Transportation Analysis (TRISTAN)*. San Pedro de Atacama, Chile.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2013) "Optimal Temporal Spreading of Alternative Trains in order to Minimise Passenger Travel Time in Practice". In: *Proceedings of 6th International Conference on Railway Operations Modelling and Analysis (RailTokyo)*, Tokyo, Japan.

Sels, P., Cattrysse D. and Vansteenwegen P. (2015) "Practical Macroscopic Evaluation and Comparison of Railway Timetables". In: *Proceedings of 18th Euro Working Group on Transportation (EWGT)*, 14-16 July 2015, Delft, The Netherlands.

Sels, P., Meisch K., Møller T., Parbo J., Dewilde T., Cattrysse D. and Vansteenwegen P. (2015) "Towards a Better Train Timetable for Denmark Reducing Total Expected Passenger Time". In: *Proceedings of 13th Conference on Advanced Systems in Public Transport, (CASPT)*, 19-23 July 2015, Rotterdam, The Netherlands.

Meeting Abstracts, Presented at International Scientific Conferences and Symposia, Published or not Published in Proceedings or Journals

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2011) "Deriving all Passenger Flows in a Railway Network from Ticket Sales Data". In: *Proceedings of 25th Annual conference of the Belgian Operations Research Society (ORBEL)*. Ghent, Belgium, pp. 110–111.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2011) "Defining robustness of a railway timetable". In: *Proceedings of 25th Annual conference of the Belgian Operations Research Society (ORBEL)*. Ghent, Belgium, pp. 108–109.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2011) "Knock, Knock, Knock-on on railway networks". In: *Proceedings of 19th Conference of the International Federation of Operational Research Societies (IFORS)*. Melbourne, Australia, pp. 53.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2012) "Improving the robustness of the railway system in Brussels". In: *Proceedings of 26th Annual Conference of the Belgian Operations Research Society (ORBEL)*. Brussels, Belgium, pp. 55–56.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2012) "On the improvement of robustness in railway station areas". In: *Proceedings of 25th European Conference on Operational Research (EURO)*. Vilnius, Lithuania.

Dewilde T., **Sels P.**, Cattrysse D. and Vansteenwegen P. (2013) "Adapting the train schedule due to planned infrastructure unavailability in complex railway stations". In: *Proceedings of International Conference on Operations Research (OR2013)*. Rotterdam, the Netherlands, pp. 63-64.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2014) "Automatically and Quickly Planning Platform and Route of Trains in Railway Stations with a Case Study for Mechelen Station". In: *Proceedings of 2014 International Journal Conference on Civil and Transport Engineering (IJCCTE)*. Bangkok, Thailand, pp 6-10.

Sels, P., Dewilde T., Cattrysse D. and Vansteenwegen P. (2014) "Automatically and Quickly Planning Platform and Route of Trains in Railway Stations: ". In: *Proceedings of 20th Conference of the International Federation of Operational Research Societies (IFORS)*. Barcelona, Spain.

Dewilde T., Burggraeve S., Ameye T., **Sels P.**, Cattrysse D. and P. Vansteenwegen. (2014) "The impact of preprocessing for solving the train routing problem". In: *Proceedings of 28th Annual conference of the Belgian Operations Research Society (ORBEL)*. Mons, Belgium, pp. 110–111.

Sels, P., Dewilde T., Vansteenwegen P. and Cattrysse D.. (2015) "Light Rail, Tram and Bus co-Timetabling Minimising Passenger Travel Time in Practice". In: *Proceedings of 9th Triennial Symposium on Transportation Analysis (TRISTAN)*. Oranjestad, Aruba.

Other Publications

Burggraeve, S., Dewilde T., Ameye T., **Sels P.** and Vansteenwegen P. (2014) "Definitie van robuustheid voor treinsystemen" (Defining robustness of a railway timetable) (in Dutch). Published in *Tijdschrift Vervoerswetenschap*.

Open Sourced Software

Sels, P. (2012) "milp-logic: a C++ MILP Solver Abstraction Layer with a C++ Boolean Modelling Layer on Top". URL: https://github.com/PeterSels/milp-logic/.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF MECHANICAL ENGINEERING
CENTRE FOR INDUSTRIAL MANAGEMENT, TRAFFIC AND INFRASTRUCTURE
Celestijnenlaan 300 box 2442
B-3001 Leuven
sels.peter@gmail.com
http://www.cib.kuleuven.be