A Conflict Prevention Strategy for Large and Complex Networks in Real-Time Railway Traffic Management

Pieter Vansteenwegen [a,1], Sofie Van Thielen [a], Francesco Corman [b]
[a] KU Leuven Mobility Research Centre-CIB, KU Leuven
Celestijnenlaan 300, 3001 Leuven, Belgium
[1] E-mail: pieter.vansteenwegen@kuleuven.be, Phone: +32 (0) 16 32 16 69
[b] Dep. Of Civil, Env. And Geomatic Eng., ETH Zürich
Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland

Abstract
Train timetables are built such that trains can drive without any delay. However, in real-time, unexpected events such as overcrowded platforms or small mechanical defects can cause conflicts, i.e., two trains requiring the same part of the infrastructure at the same time. Currently, such conflicts are typically resolved by experienced dispatchers. However, it is impossible for them to fully anticipate the impact of their actions on the entire network. Conflict detection and prevention tools embedded in a Traffic Management System can help them in making informed decisions. Though some advanced train movement prediction and conflict detection has been developed in the last years, there still exists a need for conflict prevention strategies capable of delivering conflict resolutions on large and complex networks based on retiming, reordering and rerouting some of the trains in real-time.
Our previous work introduced such a conflict prevention strategy that, based on offline calculations, determined which part of the network should be regarded when deciding on a conflict resolution. This work is significantly extended here by considering several new parameters for the Dynamic Impact Zone heuristic. This paper compares results on different sizes of networks, and tackles the challenges for applying the strategy on even larger networks.

Keywords
Conflict Resolution, Real-Time Railway Management, Dispatching, Large Networks

1 Introduction

Transport and mobility are important for inhabitants all over the world. Rail transport is used by many passengers to travel to their work on a daily basis. Clearly, the need for more people and goods mobility has steadily grown worldwide and this trend will continue in the future. Therefore, public transport systems will need to provide a better quality of service, in terms of frequencies, comfort, accessibility and reliability of services, along with transparent information regarding travel times and routing alternatives.

The combination of growth in mobility demand and the difficulties in building new infrastructure presses the need for utilizing the existing infrastructure at the highest possible capacity, at all times. A less costly solution is to improve the quality of the trains to decrease breakdowns and/or to increase the capacity by improving train punctuality.

Though train timetables can account for some delays occurring in real-time, unexpected events such as passenger crowding, bad weather or a small mechanical defect, can make the timetable infeasible. In order to increase the performance of railway services in practice, efficient conflict resolutions are required. These resolutions can be based on the retiming or reordering of trains, or even rerouting them. If the timetable becomes infeasible, dispatchers have to decide on the best resolution. Nowadays, they are often assisted by an advanced Traffic Management System (TMS), including train movement prediction and conflict detection. However, after a conflict is detected, dispatchers often still have to rely on their own experience to decide on the best conflict resolution. Therefore, a Conflict Prevention Module (CPM), which can be easily integrated into a TMS, is required. This module should include a Conflict Prevention Strategy (CPS) capable of resolving detected conflicts.

Our previous work introduced such a CPS that, based on offline calculations, determined which part of the network should be regarded when deciding on a conflict resolution. This work is significantly extended here by considering several new parameters for the Dynamic Impact Zone heuristic. This paper compares results on different sizes of networks, and tackles the challenges for applying the strategy on even larger networks.

Section 2 starts by explaining some important definitions required for the remainder of the paper and Section 3 discusses the related literature. Section 4 explains the simulation framework used for testing the CPS. The CPS and the different novelties are discussed in detail in Section 5. Section 6 shows the experimental results applying the new CPS on larger and more and complex networks. The paper is concluded in Section 7.

2 Definitions

This section describes all relevant definitions used in railway literature and in practice. First, the basic elements that build up a railway network are introduced. Then, it is defined how trains move through a railway network.

2.1 The Railway Network

A railway network can be considered on three different levels: the macroscopic, mesoscopic or microscopic level. The microscopic level includes all details, e.g., switches, tracks, signals. This level is important for the train drivers and dispatchers. The macroscopic level is a reduction of the microscopic level and is often only what passengers experience. The mesoscopic level lies somewhere in between the two previous levels. In this paper it is required that all timings of trains are known in full detail. Therefore, a microscopic level for the network is preferred.

A microscopic network is characterized by the signals present in the infrastructure. Signals give information to trains coming from the direction to which the signal is visible. Every signal indicates either the beginning or the end of a *block section*. The part of the infrastructure between two subsequent, similarly directed signals thus determine a block section, which is typically around 1000 meters long in Belgium.

The network can also be decomposed in zones: *station areas* and *non-station areas*. A station area includes one or more parallel platforms where passengers can embark or disembark if the train has a *stop* at this station. Before and after these platforms, there is a *switch area*, such that many possible combinations between an incoming or outgoing track and a platform can be made. This allows trains to reroute in the station area when their original platform is not available. The signals at the beginning or end of a switch area, are called the start and end points of the station area.

2.2 A Train Driving Through the Network

Nowadays, in many railway systems around the world, a train drives from signal to signal. This signal gives information about the next block section the train wants to occupy. A railway signal is comparable to ordinary traffic lights and can give a green, double yellow or red light. Green indicates that the next two block sections are free, double yellow indicates that the next block section is free, but the one after that is occupied, and red indicates that the next block section is occupied. In practice, a train has to slow down at double yellow such that it can come to a full stop at a red signal, if necessary. In this way, signals guarantee that trains can be guided safely throughout the network by giving information on the state of the block sections ahead.

According to Hansen and Pachl (2008), a block section is exclusively occupied by one train during a time interval, composed of the actual *occupation time* and safety margins before and after. The time interval during which a train blocks one block section, is called the *blocking time*. By using the blocking time theory, an accurate calculation can be performed to determine the duration of the blocking times. These blocks can then be represented in a time-distance diagram and the result is the so-called *blocking stairway*. A conflict can then be seen as where blocks belonging to different trains overlap.

2.3 Methodological Framework

As indicated by Lamorgese et al. (2017), a large gap still exists between the state-of-the-art traffic management in academic research and the state-of-the-art in practice. This shows that many challenges arise when putting academic research in practice. However, lately, a lot of effort has been put in implementing academic models in practice (see for instance Borndörfer et al., 2017). In order to close the gap between academics and practice, Corman and Quaglietta (2015) introduce a closed loop framework, which is closer to real-life situations than an open or multiple open loop. Open loop rescheduling assumes that control measures are computed and implemented once and for all, thus assuming a perfect knowledge of future traffic states. This implies that predicted and actual traffic states are equal, and that no unexpected events can occur anymore. Open loop approaches have been implemented very often in academic literature (Corman and Meng, 2013; Cacchiani et al., 2014; Pellegrini et al., 2016). An extension is the multiple open loop, where it is assumed that at some points more traffic conditions are known, and the calculations can be reconsidered with this additional information (Corman and Quaglietta, 2015). A closed loop approach calculates dispatching actions every time updated information is available, and adjusts control measures immediately (Caimi et al., 2012; Corman and Quaglietta, 2015). In this setup, new updates of information are taken into account whenever available. The implementation of the Conflict Prevention Strategy (CPS) within a closed loop framework is discussed in detail in Section 4.

3 Literature Review

An overview of some recent papers dealing with the conflict resolution problem is given in Table 1. These approaches perform conflict resolution by using: an Alternative Graph (AG), a Mixed Integer Linear Program (MILP), a Mixed Integer Program (MIP), Constraint Programming (CP), etc. According to the level of detail, we can make a distinction between microscopic, mesoscopic and macroscopic level. The models can be adapted to deal with different objective functions as can be seen in the fourth column. The model is tested on a study area, that can either be a station area (S), a control area (CA), a terminal area of a metro line (TA), a network (N), a line (L) or a railway corridor (RC). Possible control actions can be updating the train timing (retiming RT) or train order (RO) or the routes (RR).

All of these approaches describe the railway problem at a given level of detail, predict future train conflicts, compute control actions to resolve these conflicts, and evaluate using some objective function. Clearly, a variety of methods with different characteristics have been proposed over the last years. We discuss Table 1 in detail in Van Thielen (2019).

For usage in practice, a conflict resolution model should be capable of giving immediate applicable suggestions in a small time frame of merely seconds, even for very large and complex networks. Microscopic conflict resolutions are therefore preferred. Corman and Quaglietta (2015) suggest that the Conflict Resolution Problem should be tested in a closed loop fashion. Though some approaches have been tested in practice (Borndörfer et al., 2017), the size of the networks considered is always very limited.

4 Simulation Framework

This section describes how the real-time railway traffic management is modeled by a closed loop framework. In practice, real-time operations are often affected by external causes, e.g., a mechanical defect. Such external causes possibly lead to a *primary delay*, i.e. a delay that cannot be avoided during planning. Once trains start deviating from their original schedule, other conflicts can arise. A conflict occurs when (at least) two trains require the same part of the infrastructure at the same time. In this case, a dispatcher or a TMS needs to resolve the conflict at once. However, resolving a conflict requires to (locally) reroute or retime/reorder one or multiple trains. One conflict resolution can therefore cause many other conflicts in the (near) future, especially when dealing with a complex network with a high number of trains. The dispatcher or the TMS creates an updated train path plan by resolving the conflict. This new plan can then be executed until a new conflict is detected and needs to be resolved.

The whole of real-time railway traffic management can thus be divided into four major modules, as depicted in Figure 1. Firstly, there is the simulator module, resembling real-life closely by describing the current traffic states. Secondly, the conflict detection module is able to predict future train movements and detect possible conflicts. Thirdly, the conflict prevention module calculates feasible conflict resolutions for every detected conflict. This conflict resolution is then given back to a dispatcher who can still decide to follow the recommendation or not. Alternatively, an automated machine can make the decisions, e.g., always following the recommendations. This is located in the fourth module, the dispatching module.

Work	Mathematical model	Detail	Objective function	Case	Implementation area	Actions	Loop
Bettinelli et al. (2017)	Iterated greedy heuristic	Macro/micro	Combination of delay penalty, dependency breaking penalty, capacity violation penalty and detour penalty	L/S	Real-world instances	RT RO RR	Open loop
Caimi et al. (2012)	AG	Micro	Combination of reliability and punctuality	S	Berne	RT RO RR RS	Closed loop
Chen et al. (2015)	MIP	Micro	Weighted avg delay	RC	Core area of Thameslink route in London	RT RO RR	Open loop
Corman et al. (2012a)	AG	Micro	Min max sec delay	CA	Utrecht-Den Bosch	RT RO RR	Open loop
Corman et al. (2012b)	AG	Micro	Min train delays and missed connections	S	Utrecht	RT RO	Open loop
Corman and Quaglietta (2015)	AG	Micro	Min max sec delay	RC	Utrecht-Den Bosch	RT RO	Open loop, multiple open loop, closed loop
D'Ariano et al. (2008)	AG	Micro	Min max sec delay	CA	Utrecht-Den Bosch	RT RO	Open loop
D'Ariano et al. (2014)	AG/ MILP	Micro	Min max sec delay	L	East Cost Main line	RT RO RR	Open loop
Dolder et al. (2009)	Timed event graph	Micro	None (delay propagation)	N	Intercity network, Germany	RT	Closed loop
Fischetti and Monaci (2017)	AG/ MILP	Micro	Avg final delay	L	railway line nearby London	RT RO (RR)	Open loop
Josyula et al. (2018)	MILP/ Binary tree search	Micro	Min total final delay	L	Karlskrona - Tjörnarp	RT RO RR	Multiple open loop
Kecman et al. (2013)	AG	Macro/ meso	Min max sec delay	RC N	Utrecht-Den Bosch Dutch railway network	RT RO	Open loop
Lamorgese et al. (2016)	MILP	Micro/macro	Min delay cost	L	Trento-Bassano del Grappa, Foligno-Orte, Foligno-Falconara	RT RO RR	Open loop
Mannino and Mascis (2009)	MILP/ AG	Macro	Weighted deviation	TA	Milan (Sesto)	RT RO RR	Closed loop
Meng and Zhou (2011)	IP	Meso	Min total completion time of all involved trains	RC	Academic instances	RT RO RR	Open loop
Pellegrini et al. (2016)	MILP	Micro	Min total sec delay	CA	Mantes-La-Jolie junction, Rouen-Rive-Droite	RT RO RR	Open loop
Quaglietta et al. (2016)	AG	Micro	Min max sec delay	RC	Utrecht-Den Bosch	RT RO	
Rodriguez, J. (2007(@)	CP	Micro	Min max sec delay	CA	Pierrefitte-Gonesse junction	RT RO RR	Open loop
Samà et al. (2015)	AG/ MILP	Micro	Min max sec delay, min sum sec delay, min sum final delay, max punctuality, min sum weighted delays (with threshold), min deviations, min sum arrival times, min travel time trains	CA	Utrecht-Den Bosch	RT RO	Open loop
Samà et al. (2016)	MILP	Micro	Min total sec delay	CA	Rouen-Rive Droite, Lille-Flandres	RT RO RR	Open loop
Törnquist (2007)	MILP	Macro	Min total final delay, min total accumulated delay, min total delay cost	CA	Nörrkoping	RT RO	Open loop
Törnquist and Persson (2007)	MILP	Macro	Min total final delay, min total delay cost	CA	South Traffic District	RT RO	Open loop
Törnquist Krasemann (2012)	MILP	Meso	Min total final delay, min total delay cost	CA	Nörrkoping	RT RO	Open loop
Van Thielen et al. (2017)	Binary search tree	Micro	Min sec delay	CA	Brugge-Gent-Denderleeuw	RT RO RR	Closed loop
Van Thielen et al. (2018)	Data driven heuristic	Micro	Min sec delay	CA	Brugge-Gent-Denderleeuw	RT RO RR	Closed loop

Table 1: An overview of academic models for real-time railway traffic management discussed

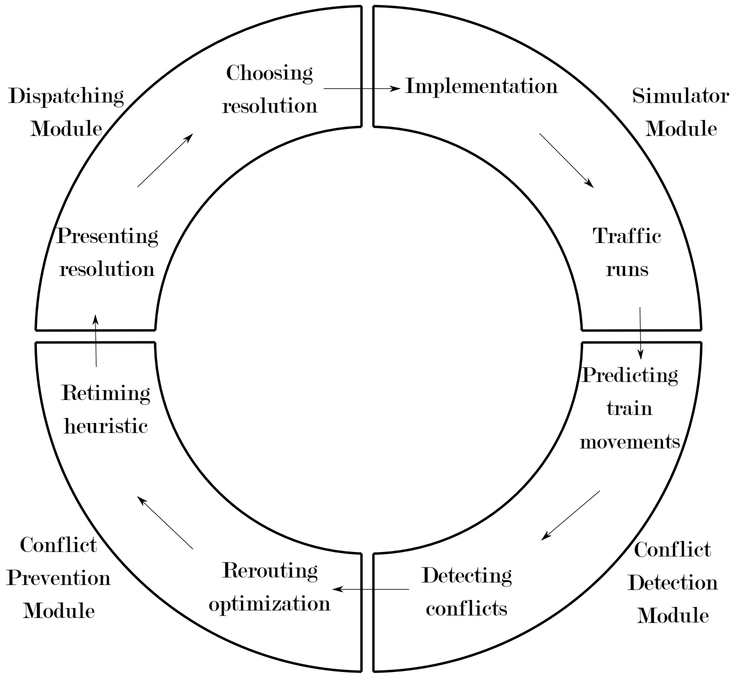


Figure 1: A schematic representation of the framework of a closed loop approach.

These different modules intercommunicate. The simulator module determines all the current states of all trains and all infrastructure parts. This module resembles reality in the sense that it takes into account stochastic dynamics of delays, and progresses synchronously in time. The simulator module starts at a given start hour, for example at 7 a.m. It will evaluate the performance of the conflict prevention module during the *simulation horizon*, which is set to 60 minutes. At the end of the simulation, several evaluation criteria such as the total secondary delay and total passenger delay of all the running trains is calculated. This is the result of many iterations of conflict detection, optimization and implementation of solution measures, in a closed loop fashion. As in real-life, it is assumed that the train information is sensed and immediately adjusted in the simulator module. It gives the necessary information on whereabouts of trains to the conflict detection module. The conflict detection module predicts and detects a certain *prediction horizon* ahead. In this paper, the prediction horizon is limited to 5 minutes because of how challenging and computationally intensive the conflict detection module is. Evidently, a TMS in practice will be capable of detecting conflicts further ahead, giving more time for calculating a good conflict resolution.

If a conflict is detected within the prediction horizon, it is sent to the conflict prevention module, where a suitable resolution is calculated. This resolution can either be based on rerouting in station areas or on retiming/reordering one of the trains. The goal is to find a resolution by minimizing a performance indicator, e.g., the total secondary delay. The objective needs to be determined beforehand by the railway infrastructure manager and/or operators. If the conflict takes place in a station area, the rerouting optimization tries to find a conflict-free solution first. If the rerouting optimization finds a feasible solution lowering the secondary delay in the station area, and results in alternative routes, then the prediction will be adapted based on the best feasible (or optimal) solution. The advantage of using rerouting is that no retiming/reordering is required and thus no additional delays are imposed. In case that the rerouting does not resolve the conflict or if the conflict is not located in a station area, the conflict needs to be resolved by finding a retiming/reordering solution. Retiming/reordering is optimized through the DIZ heuristic leading to (further) delaying one of the conflicting trains, and/or altering the order of the trains. The computed solution is then implemented by locally rerouting trains and/or changing the time and order of the trains involved in the conflict. The prediction is adapted based on this solution, possibly leading to new conflicts. This process is repeated until no conflicts are found within the prediction horizon.

This conflict resolution induces an updated train plan, including new routes and/or new orders or timings of trains. The presentation and acceptance of this resolution is dependent on the dispatching module, which can either be a dispatcher or an automated machine accepting all proposed resolutions. This dispatching module allows a dispatcher to follow the presented conflict resolution or to implement another resolution based on his/her own experience. The resolution is then sent back to the simulated reality where it can be implemented. The train plans can then be adapted according to the resolution. This will be taken into account in the conflict detection module. Since we are evaluating the performance of the conflict prevention module in this paper, we assume that the dispatching module always accepts the proposed resolutions.

These modules all have a certain computation time, because they are all in real-time. Therefore, a very advanced conflict detection module is required delivering new predictions every two seconds (or less) (Dolder et al., 2009). Also, the conflict prevention module needs to make fast decisions regarding conflict resolutions. Whenever a conflict is detected in real-time, the time required for finding a resolution and the time required for the dispatching module to accept the solution has to be taken into account. Accordingly, changes to the current situation during the calculation should not cause issues when implementing the calculated resolution. Therefore, both the rerouting optimization and the DIZ heuristic start their calculations from the expected situation a control delay after the moment of detecting the conflict. Stated otherwise, no changes to ongoing operations can be executed during the duration of the control delay. Whatever happens within this time interval after detecting the conflict, cannot be changed by the conflict prevention module. Actually, this control delay gives the conflict prevention module this exact time to calculate (and implement) a resolution.

5 Methodology

This section starts by describing our previous approach in Section 5.1. Additional improvements and extensions are then introduced in Section 5.2.

5.1 Previous Approach

The CPS consists of, on the one hand, a rerouting optimization based on a flexible job-shop problem and, on the other hand, a retiming/reordering heuristic. This heuristic examines the progress of all relevant trains for up to two options to resolve a conflict: delaying the first or second train. The first train is the first train arriving at the block section where the conflict occurs. To be useful in real-time, the progress examination should be limited in time and space in order to limit the computation time. Therefore, based on a trade-off between quality of the solution and the required computation time, a dynamic impact zone in which the progress is evaluated, is created for every conflict.

The conflict detected by TMS and sent to the CPS, is called the *initial conflict*. If this conflict is located in a station area, a rerouting optimization is started to check whether rerouting (some of the) trains reduces the overall delay. The station area is cut out of the network. For this limited network, the optimization based on a flexible job shop problem is started at the detection time plus an additional control delay, and ends when both trains have left the station area. The problem is given to Cplex with a maximum computation time in order to keep this time limited. Afterwards, alternative routes from the best feasible or optimal solution are implemented. For more detailed information on the rerouting optimization, we refer the interested reader to Van Thielen et al. (2018).

If the conflict is not resolved by the rerouting optimization, or if the conflict is not located in a station area, a resolution based on the Dynamic Impact Zone (DIZ) heuristic has to be found. This DIZ heuristic starts by selecting a suitable dynamic impact zone for the conflict. A dynamic impact zone determines which conflicts in the near future are (possibly) affected by the conflict resolution of the initial conflict. Only the trains in these conflicts should be considered during the progress examination such that the computation time remains limited, even for large networks.

The dynamic impact zone starts by considering all potential conflicts during the next half hour. An example network is shown in Figure 2, where every potential conflict is indicated with a figure. The initial conflict is depicted by a square. Conflicts can be divided into groups by considering their relation to the initial conflict. In this way, a conflict is called a *first-order conflict* if one of the trains in this conflict is also in the initial conflict. In case of the example, this means that any conflict involving T_1 or T_2 is a *first-order conflict*, and is depicted as a circle in Figure 2. *Second-order* conflicts are conflicts where at least one of the trains is involved in a first-order conflict, but it is not a first-order conflict itself. In this manner, an nth-order conflict is a conflict of which at least one train is in an $(n-1)$th-order conflict, but it is not a $(n-1)$th-order conflict itself. Second-order conflicts are depicted as diamond shapes, third-order conflicts as triangle shapes in Figure 2.

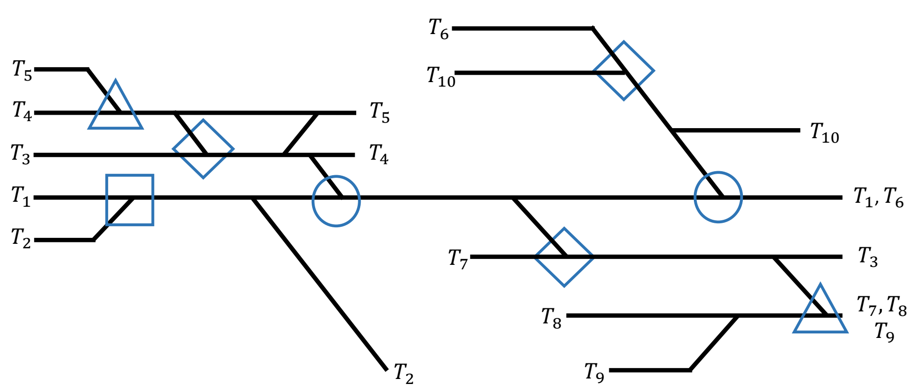


Figure 2: All potential conflicts in an example area.

As shown in Van Thielen et al. (2018), the size of the dynamic impact zone affects the computation time strongly. Therefore, the dynamic impact zone should not be too large to keep the computation time limited, but still large enough to determine a suitable solution. Therefore, offline calculations are carried out to determine which conflicts are *most likely* conflicts. After resolving 350 randomly created delay scenarios using 6 different conflict resolution strategies, the *most likely* conflicts are determined as the conflicts occurring in at least 50 % of all cases. All *most likely* conflicts are indicated by full lines in Figure 3. The dynamic impact zone, indicated with red shapes in Figure 3, is then created by including all first-order conflicts and *most likely* second-order conflicts.

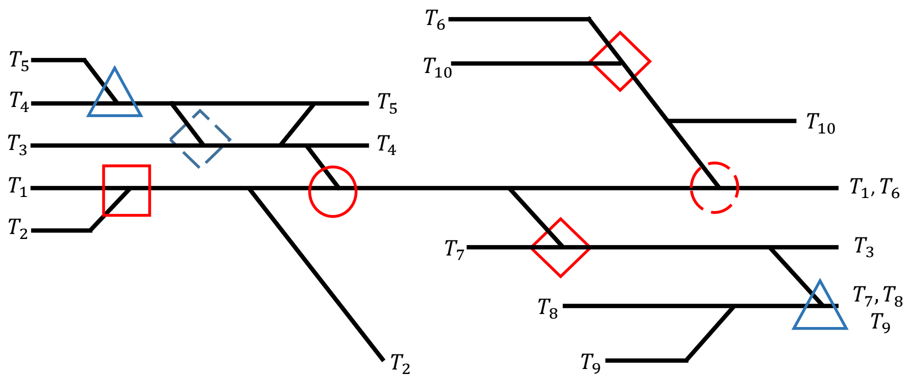


Figure 3: The dynamic impact zone for the initial conflict (indicated by the square) consists of all first-order conflicts (indicated by circles) and most likely second-order conflicts (indicated by full-lined diamonds).

After creating the dynamic impact zone, all possible conflict resolutions are determined. For every conflict resolution, a progress examination of the next half hour is started, including only trains in the dynamic impact zone. During this examination, the first-order and most likely second-order conflicts are considered, as mentioned above, and a resolution needs to be assumed. In our previous approach, it is assumed that these conflicts are solved based on FCFS. Section 5.2 describes how this can be improved. The total secondary delay of every examination is calculated and the solution leading to the lowest secondary delay is chosen.

5.2 Improvements and Extensions

Compared to the research in Van Thielen et al. (2018), several improvements and extensions are included in order to improve the CPS in both quality and computation time and to improve the performance for larger networks.

Resolving New Conflicts
Whenever a conflict is detected during the progress examination, this conflict has to be resolved. Looking at both options and branching further will be too computationally expensive. Therefore, it was first chosen to use FCFS in the progress examination (see Van

Thielen et al., 2018). Now, in order to improve the progress examination, the *immediate impact* of delaying both trains is calculated. A small example illustrates what we call the immediate impact. Figure 4 shows the routes of three trains within a progress examination of the heuristic. The path of T_1 is indicated by the red line, the path of T_2 by the green line and the path of T_3 by the purple line. During the progress examination of the heuristic, a new conflict is detected between T_1 and T_2 on block section **BS-3**. In order to determine which train to delay, the immediate impact in terms of train delay on both trains is determined first. Trains T_1 and T_2 share infrastructure on several subsequent block sections, i.e. **BS-3**, **BS-4**, **BS-5** and **BS-7**. This implies that T_1 and T_2 will be driving behind each other on all these block sections. Moreover, T_1 might be delayed due to a conflict with T_3 on **BS-7** and then further delay T_2, if T_1 drives before T_2. Therefore, in this case, it might be better to let T_2 go first. More generally, if the first train allowed to drive on a common part of the infrastructure is expected to be delayed due to another new conflict later on this common part, then the second train will also be delayed extra. Therefore, it would be better to let the other train go first. This is what we call considering the immediate impact when deciding on how to resolve a new conflict. Specifically, we first determine the set of subsequent block sections with some part of the infrastructure in common belonging to the two trains (T_1 and T_2 in the example) in the new conflict. Then, for every block section in this set (**BS-3**, **BS-4**, **BS-5** and **BS-7** in the example), we look at any other train in the dynamic impact zone that also has common infrastructure (T_3 in the example). If its occupancy based on the delay characteristics at the detection time of the new conflict (partly) coincides with the time interval of one of the two trains in the new conflict, then a delay is imposed on the train in the new conflict (T_1 in the example). This delay is then added to the delay resulting from resolving the new conflict. The option with the least total delay is chosen.

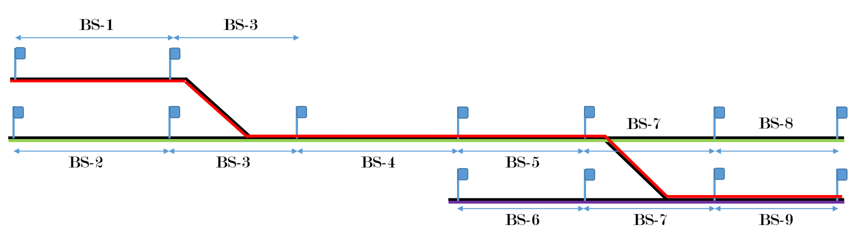


Figure 4: Three train paths on a small network: the red line indicates the path of T_1, the green line the path of T_2 and the purple line the path of T_3.

Updating Potentially Conflicting Trains

The set of all potentially conflicting trains is determined offline beforehand to reduce the computation time online. Two trains have a potential conflict if they want to use the same part of the infrastructure within a time interval of 20 minutes. Dependent on the current situation, at the time the initial conflict is detected, it is possible that these trains are not potentially conflicting anymore. In order to improve and limit the creation of the dynamic impact zone further, trains are updated as potentially conflicting if they share the same part of the infrastructure within a time interval of 10 minutes, according to their current delay. The dynamic impact zone is then created based on this updated set of potentially conflicting trains.

Adding a Maximum Distance from the Initial Conflict
The creation of the dynamic impact zone is limited by imposing a *heuristic horizon* of 30 minutes. If the network is large, this limitation might not be sufficient to control the size of the dynamic impact zone, and thus also the computation time. An additional parameter is therefore included, only searching for first-order conflicts in the dynamic impact zone located within ϵ railway km from the initial conflict.

6 Case Studies

Our proposed CPS is tested on two very large and complex networks including two or three provinces in Belgium. The several adjustments and improvements discussed in Section 5 are tested on both networks. Afterwards, the results of both networks are compared.

6.1 Study Areas

Study Area 1 (SA-1): Provinces of West and East Flanders
This rail network depicted in Figure 5 consists of two provinces in Belgium: West Flanders and East Flanders. The network is approximately 130 km long (De Panne-Puurs) and 60 km wide (Duinbergen-Lauwe). This area contains 130 stations and 11766 block sections. The largest station areas are Brugge, Gent-Sint-Pieters and Oostende. The network is considered in microscopic detail, and is considered with the timetable from 17/03/2017. Between 7 a.m. and 8 a.m., there are at most 240 trains driving in this network. There are 51 rolling stock connections between trains (turnarounds, coupling, de-coupling) taken into account.

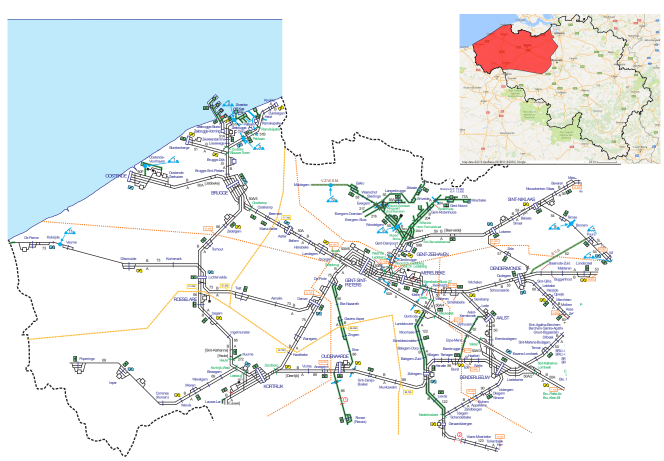


Figure 5: Study area: the provinces of West and East Flanders in Belgium.

Study Area 2 (SA-2): Provinces of Antwerp, West and East Flanders
This rail network includes three provinces of Flanders in Belgium (see Figure 6). This network is approximately 170 km long (De Panne-Zittaart) and 78 km wide (Antwerpen-

Hoeilaart). This area includes 191 station areas and 23917 block sections. The largest station areas are Gent-Sint-Pieters, Brugge, Oostende, Mechelen, Antwerpen-Centraal en Antwerpen-Berchem. Both freight and passenger trains are taken from the microscopic timetable of 17/03/2017. During the time window between 7 and 8 a.m., there are at maximum 353 trains considered and 71 rolling stock connections are provided.

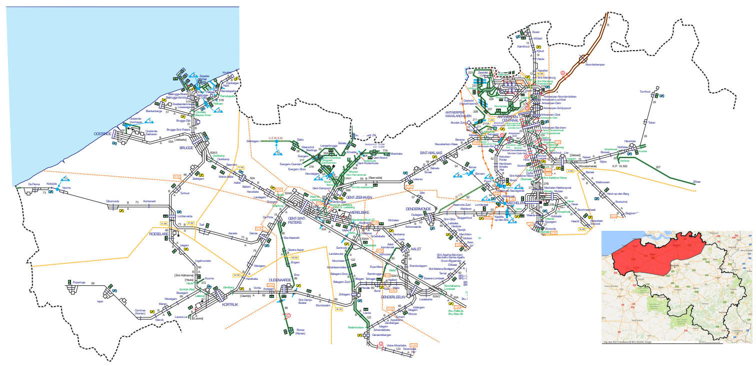


Figure 6: Study area consisting of the three provinces Antwerp, West and East Flanders.

6.2 Results

Table 2 gives an overview of the abbreviations used for the different versions of CPS. Strategy iCPS is the previous strategy from Van Thielen et al. (2018). Strategy iCPS-NC includes the new way of resolving new conflicts. Then, strategy iCPS-UPC includes updating the potential conflicts. These latter two together form the strategy iCPS-IMP. Subsequently, a maximum distance is opposed when creating the dynamic impact zone in iCPS-IMP-10km, iCPS-IMP-50km and iCPS-IMP-100km. The line in bold indicates the best obtained algorithm after performing extensive tests on both networks.

Name	Improvements			Parameters
	Include connections	New conflicts	Update pot conflicts	Max dist conflict
iCPS	✓	-	-	-
iCPS-NC	✓	✓	-	-
iCPS-UPC	✓	-	✓	-
iCPS-IMP	✓	✓	✓	-
iCPS-IMP-10km	✓	✓	✓	10 km
iCPS-IMP-50km	✓	✓	✓	**50 km**
iCPS-IMP-100km	✓	✓	✓	100 km

Table 2: Overview of the different conflict prevention strategies evaluated in this section.

As a standard, the prediction horizon is set to 5 minutes, the heuristic horizon to 30 minutes and the control delay to 60 seconds (see Van Thielen et al. (2018) for more information). The offline calculations are based on 350 runs from a delay scenario with α % of the trains delayed. The value α is randomly taken from the interval $[20\%, 80\%]$. The computation time of the rerouting optimization is limited to 30 seconds.

In order to evaluate the CPS, 20 runs from a delay scenario α are taken, where α is again randomly taken from the interval $[20\%, 80\%]$. In one run, approximately $\alpha\%$ of all trains, which are randomly chosen, are given a random delay from an exponential distribution with an average of 3 minutes and a maximum of 15 minutes. In order to compare the results on both networks, the delay scenarios are first created for the largest network SA-2. The same delay scenarios are then used as input for the network SA-1. The CPS is evaluated based on the total secondary train delay, the average and the maximum computation time. The computation time is the time required to create the dynamic impact zone and perform the progress examinations. All tests are carried out on a Intel(R) Core(TM) i7-3770 CPU 3.40GHz machine.

Table 3 shows the total secondary delay, average and maximum computation time of the different strategies. As expected, the total secondary delay increases when the network is larger. When considering the largest network, more conflicts are detected and resolved. In our simulation, trains cannot reduce their delays during operations, implying that for the largest network the delays can only increase. The improvement compared to FCFS is also somewhat smaller, but still the same order of magnitude (40-50%). The computation time of our CPS remains very similar when enlarging the network, meaning that the dynamic impact zone is well bounded.

Strategy	Train D (in min)		Average computation time (in s)		Maximum computation time (in s)	
	SA-1	SA-2	SA-1	SA-2	SA-1	SA-2
FCFS	660	843	< 0.1	< 0.1	< 0.1	< 0.1
iCPS	369 (- 44 %)	516 (- 38 %)	2.7	3.0	33.7	39.5
iCPS-NC	305 (- 54 %)	478 (- 43 %)	2.7	3.0	33.4	36.8
iCPS-UPC	370 (- 44 %)	513 (- 39 %)	2.3	2.6	32.8	36.8
iCPS-IMP	305 (- 54 %)	468 (- 44 %)	2.3	2.5	33.1	35.4
iCPS-IMP-10km	312 (- 53 %)	475 (- 44 %)	1.8	1.9	34.3	37.5
iCPS-IMP-50km	305 (- 54 %)	467 (- 45 %)	2.2	2.4	32.3	37.5
iCPS-IMP-100km	305 (- 54 %)	467 (- 45 %)	2.3	2.6	33.1	38.4

Table 3: Total secondary delay compared between SA-1 and SA-2.

Clearly, the improvements discussed in Section 5.2 assure that our new Conflict Prevention Strategies perform better both in total secondary delay as in computation time. Combining the new method of resolving new conflicts in the progress examination with updating potential conflicts (iCPS-IMP) attains the same, lower secondary delay as in iCPS-NC, while also attaining the lower computation time as in iCPS-UPC. Opposing a maximum distance from the initial conflict can keep the computation time under control, while also affecting the total secondary delay. Therefore, iCPS-IMP-50km is selected as the best strategy.

6.3 Comparison

By purely extending the network and starting from the same delay scenario, it can be examined which effects extending the network has on the secondary delays.

In order to determine whether the dynamic impact zone is robust enough for extensions to even larger networks, we look closely to the impact zones of conflicts both detected in SA-1 and SA-2 in Table 4. The size of the impact zone is expressed in the number of new conflicts, where a conflict includes two trains and the block section on which the conflict takes place. The average size of the impact zone increases when considering a larger network, leading to the higher computation time, as shown in Table 3.

Strategy	Average size IZ of SA-1	Average size IZ of SA-2	Percentage difference in size IZ
iCPS	279	329	17.9 %
iCPS-NC	274	318	15.8 %
iCPS-UPC	214	256	19.6 %
iCPS-IMP	215	252	17.2 %
iCPS-IMP-10km	80	105	31.3 %
iCPS-IMP-50km	185	216	16.8 %
iCPS-IMP-100km	211	243	15.2 %

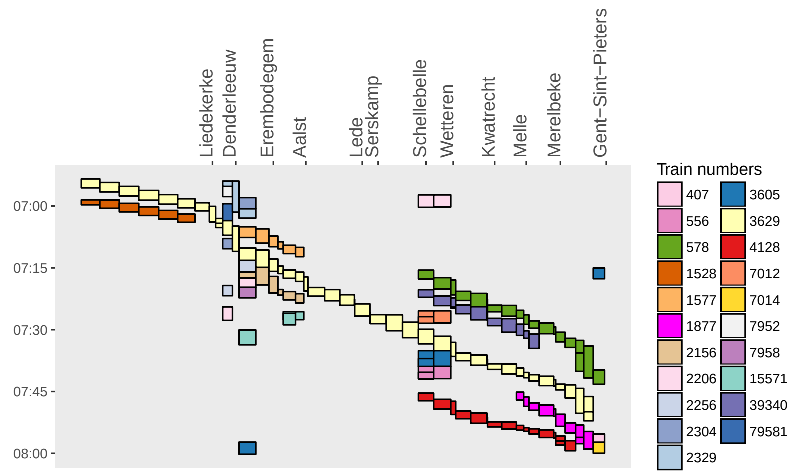
Table 4: Comparison of the size of the impact zone for the same conflicts for both networks. The size is expressed in number of conflicts, which is a couple of trains on one block section.

The strong increase in the size of the impact zone is due to the fact that several conflicts are detected close to the border with the province of Antwerp. When a conflict is located near the border with the province of Antwerp, the dynamic impact zone considered will be larger in SA-2 than in SA-1, since it is artificially bounded in SA-1 by the border of the network considered. Table 5 shows the percentage of initial conflicts located within the maximum distance (10, 50 or 100 km) from the border with the province of Antwerp. This percentage obviously increases when increasing the maximum distance. Consequently, many dynamic impact zones in SA-1 are artifically bounded by 'bumping' into the border of the province of Antwerp. This explains the slightly smaller computation time in SA-1 compared to SA-2. This 'border-effect' could only be avoided by expanding the study area until the entire network of Belgium (and then still international trains cross the borders). Nevertheless, limiting the dynamic impact zone by both the heuristic horizon and the maximum distance will be sufficient in practice to keep the computation time under control.

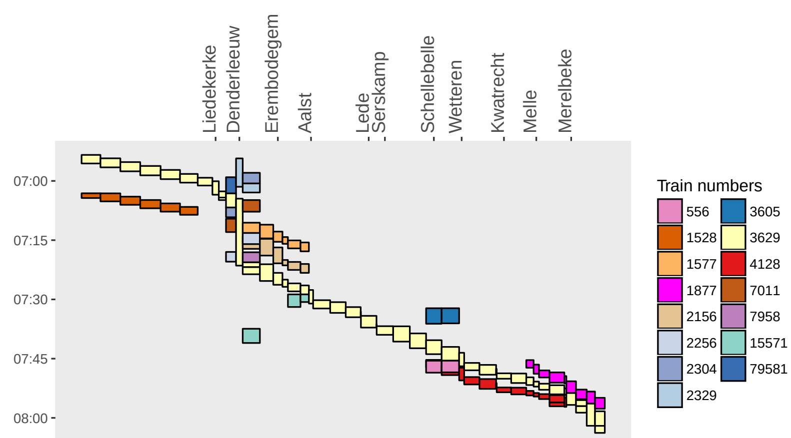
Strategy	Percentage of conflicts
iCPS-IMP-10km	5.2 %
iCPS-IMP-50km	85.9 %
iCPS-IMP-100km	99.7 %

Table 5: Percentage of initial conflicts located closer than 10, 50 or 100 km from the border with the province of Antwerp.

Figure 7 shows the time distance diagram of train 3629 and trains using the same infrastructure within a limited time window. A conflict is detected between trains 2156 and 3629 in both cases, but this conflict is resolved differently. When limiting the network to SA-1, train 2156 is delayed, whereas when the network is extended to SA-2, train 3629 is delayed.



(a) Train path of train 3629 on network SA-1



(b) Train path of train 3629 on network SA-2

Figure 7: Time-distance diagrams of train 3629 resulting from iCPS-50km on SA-1 and SA-2.

7 Conclusion and Future Research

This paper proposes several improvements and extensions to our previous conflict prevention strategy, presented in Van Thielen et al. (2018), making it applicable on larger and complex networks. The extensions allow the computation time to remain limited for real-time purposes. By comparing the same delay scenarios on different sizes of study areas, it is shown that conflicts might be resolved differently. Extending the study area leads to an increase in the total secondary delay, because more conflicts are detected and need to be resolved.

By including these additional improvements and extensions, the total secondary delay decreases, while also reducing the computation time. The basic dispatching strategy FCFS is outperformed by, on average, 40-50 %, while delivering conflict resolutions within 2.4 seconds on average.

Comparing results from both networks is difficult, because when considering the largest network, conflicts in the province of Antwerp are detected and need to be resolved as well. This leads to an increase in the total secondary delay, but also alters the current timetable and route of some trains. Moreover, many dynamic impact zones of initial conflicts considered in the smaller network are artificially bounded by the borders of that smaller network. This slightly reduces the required computation time. Nevertheless, the main conclusions remain that the conflict prevention strategy is significantly improved compared to the previous version and that the computation time can be controlled by limiting the dynamic impact zone by both the heuristic horizon and the maximum distance.

The performance of the conflict prevention strategy is tested using a simulation framework, simulating the real-time situation closely. This assures that the conflict prevention strategy can easily be embedded in a Traffic Management System, such as the one currently implemented at the Belgian railway infrastructure manager Infrabel.

A more detailed analysis would require to consider the entire network considered in practice, the whole of Belgium for instance, in order to mimic the real-time situation as closely as possible. The conflict prevention strategy can easily be applied to such (much) larger networks, but both the Simulator Module and the Conflict Detection Module should be significantly improved before running experiments on larger study areas.

References

Bettinelli, A. and Santini, A. and Vigo, D., 2017. "A real-time conflict solution algorithm for the train rescheduling problem", *Transportation Research Part B*, vol. 106, pp. 237–265.

Borndörfer, R. and Klug, T. and Lamorgese, L. and Mannino, C. and Reuther, M. and Schlechte, T., 2017. "Recent success stories on integrated optimization of railway systems", *Transportation Research Part C*, vol. 74, pp. 196–211.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L. and Wagenaar, J, An overview of recovery models and algorithms for real-time railway rescheduling. Transportation Research Part B, 63, 15–37 (2014)

Caimi, G. and Fuchsberger, M. and Laumanns, M. and Lüthi, M., 2012. "A model predictive control approach for discrete-time rescheduling in complex central railway station areas", *Computers & Operations Research*, vol. 39, pp. 2578–2593.

Chen, L. and Roberts, C. and Schmid, F., 2015. "Modeling and solving real-time train rescheduling problems in railway bottleneck sections" *IEEE Transactions on Intelligent Transportation Systems*, vol. 16(4), pp. 1896–1904.

Corman, F. and D'Ariano, A. and Pacciarelli, D. and Pranzo, M., 2012. "Optimal inter-area coordination of train rescheduling decisions", *Transportation Research Part E*, vol. 48, pp. 71–88.

Corman, F. and D'Ariano, A. and Pacciarelli, D. and Pranzo, M., 2012. "Bi-objective conflict detection and resolution in railway traffic management", *Transportation Research Part C*, vol. 20, pp. 79–94.

Corman, F. and Meng, L. A review of online dynamic models and algorithms for railway traffic control. IEEE International Conference on Intelligent Rail Transportation (ICIRT), 2013, Beijing, 128–133 (2013)

Corman, F. and Quaglietta, E., 2015. "Closing the loop in railway traffic control: framework design and impacts on operations", *Transportation Research Part C*, vol. 54, pp. 15–39.

D'Ariano, A. and Corman, F. and Pacciarelli, D. and Pranzo, M, 2008. "Reordering and Local Rerouting Strategies to Manage Train Traffic in Real Time", *Transportation Science*, vol. 42, pp. 405–419.

D'Ariano, A. and Samà, M. and D'Ariano, P. and Pacciarelli, D., 2014. "Evaluating the applicability of advanced techniques for practical real-time train scheduling", *Transportation Research Procedia*, vol. 3, pp. 278–288.

Dolder, U. and Krista, M. and Voelcker, M, 2009. "RCS - rail control system, realtime train run simulation and conflict detection on a netwide scale based on updated train positions", In: *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis, RailZurich (2009)*.

Fischetti ,M. and Monaci, M., 2017. "Using a general-purpose Mixed-Integer Linear Programming solver for the practical solution of real-time train rescheduling", *European Journal of Operational Research*, vol. 263, pp. 258–264.

Hansen, I. and Pachl, J., 2008. "Railway timetable and traffic: analysis, modelling, simulation", Eurail Press.

Josyula, S.P. and Törnquist Krasemann, J. and Lundberg, L., 2018. "A parallel algorithm for train rescheduling", *Transportation Research Part C*, vol. 95, pp. 545–569.

Kecman, P. and Corman, F. and D'Ariano, A. and Goverde, R.M.P., 2013. "Rescheduling models for railway traffic management in large-scale networks", *Public Transport*, vol. 5(1), pp. 95–123.

Lamorgese, L. and Mannino, C., 2015. "An exact decomposition approach for the real-time train dispatching problem", *Operations Research*, vol. 63(1), pp. 48–64.

Lamorgese, L. and Mannino, C. and Piacentini, M., 2016. "Optimal train dispatching by Benders'-like reformulation", *Transportation Science*, vol. 50(3), pp. 910–925.

Lamorgese, L. and Mannino, C. and Pacciarelli, D. and Törnquist Krasemann, J., 2017. Train Dispatching. In: *Handbook of Optimization in the Railway Industry, International Series in Operations Research & Management Science*, chapter 12, pp. 265–283. Springer International Publishing, Switzerland.

Mannino, C. and Mascis, A., 2009. "Optimal real-time traffic control in metro stations", *Operations Research*, vol. 57(4), pp. 1026–1039.

Meng, L. and Zhou, X., 2011. "Robust single-track dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach", *Transportation Research Part B*, vol. 45, pp. 1080–1102.

Pellegrini, P., Marlière, G. and Rodriguez, J., 2016. "A detailed analysis of the actual impact of real-time railway traffic management optimization", *Journal of Rail Transport Planning & Management*, vol. 6, pp. 13–31.

Quaglietta, E., Corman, F. and Goverde, R.M.P., 2013. "Stability analysis of railway dispatching plans in a stochastic and dynamic environment", *Journal of Rail Transport Planning & Management*, vol. 3, pp. 137–149.

Quaglietta, E., Pellegrini, P., Goverde, R.M.P., Albrecht, T., Jaekel, B., Marlière, G., Rodriguez, J., Dollevoet, T., Ambrogio, B., Carcasole, D., Giaroli, M. and Nicholson, G., 2016. "The ON-TIME real-time railway traffic management framework: A proof-of-concept using a scalable standardized data communication architecture", *Transportation Research Part C*, vol. 63, pp. 23–50.

Rodriguez, J., 2007. "A Constraint Programming Model for Real-time Train Scheduling at Junctions", Transportation Research Part B: Methodological, 41(2), 231–245.

Samà, M., Meloni, C., D'Ariano, A. and Corman, F., 2015. "A multi-criteria decision support methodology for real-time train scheduling", *Journal of Rail Transport Planning & Management*, vol. 5(3), pp. 146–162.

Samà, M. and Pellegrini, P. and D'Ariano, A. and Rodriguez, J. and Pacciarelli, D., 2016. "Ant colony optimization for the real-time train routing selection problem",*Transportation Research Part B*, vol. 85, pp. 89–108.

Samà, M., D'Ariano, A., Corman, F. and Pacciarelli, D., 2017. "A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations". *Computers & Operations Research*, vol. 78, pp. 480–499.

Törnquist, J., 2007. "Railway traffic disturbance management: an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon", *Transportation Research Part A*, vol. 41(3), pp. 249–266.

Törnquist, J. and Persson, J.A., 2007. "N-tracked railway traffic re-scheduling during disturbances", *Transportation Research Part B*, vol. 41, pp. 342–362.

Törnquist Krasemann, J., 2012. "Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances", *Journal of Transportation Research Part C*, vol. 20, pp. 62–78.

Van Thielen, S. and Corman, F. and Vansteenwegen, P., 2017. "An efficient heuristic for train rescheduling and local rerouting", *Proceedings of 7th International Seminar on Railway Operations Modelling and Analysis (IAROR), RailLille. Lille, France, 4-7 April 2017*, pp. 585–606.

Van Thielen, S., Corman, F. and Vansteenwegen, P., 2018. "Considering a dynamic impact zone for real-time railway traffic management", *Transportation Research Part B*, vol. 111, pp. 39–59.

Van Thielen, S., 2019. "Conflict prevention strategies for real-time railway traffic management", *PhD Thesis. KU Leuven*.