Improving passenger robustness by taking passenger numbers and recurring delays explicitly into account on the tactical level

Sofie Burggraeve [a, 1], Thijs Dewilde [a], Peter Sels [a, b, c] Pieter Vansteenwegen [a]
[a] KU Leuven, Centre of Industrial Management / Traffic and Infrastructure,
Celestijnenlaan 300 Box 2422, 3001 Leuven, Belgium
[1] E-mail: sofie.burggraeve@kuleuven.be, Phone: +32 (0) 32 16 372023
[b] Logically Yours BVBA,
Plankenbergstraat 112 Bus L7, 2100 Deurne, Belgium
[c] Infrabel, Department of Rail Access, Fonsnylaan 13, 1060 Brussels, Belgium

Abstract
In this paper, we consider nearly saturated station areas whose limited capacity is one of the main reasons of delay propagation. Our goal is to improve, during the planning phase, the total travel time in practice of all passengers in the railway network. Therefore, passenger numbers and recurring delays of trains are explicitly taken into account in the optimization process. By maximizing the (passenger- and/or delay-) weighted spreading between trains in a fixed time window, potential conflicts that affect many passengers are avoided which is beneficial for the total real travel time of all passengers. Using our approach, the passenger robustness of Brussels, Belgium's main railway bottleneck, can be improved by 11%.

Keywords
Passenger robustness, Recurring delays, Timetable, Simulation

Introduction

Convincing more commuters to use public transportation is a classic strategy for increasing the sustainability of a transportation system. One key concern of (potential) railway passengers is the reliability of their travel time in practice (König, & Axhausen (2002)). This travel time in practice differs from the planned travel time as a result of primary delays of trains (caused by a high number of passengers during peak hours, mechanical failures, weather conditions, etc.) and secondary delays of trains, which are delays caused by delays of other trains. Since primary delays have external causes, they cannot be prevented by appropriate planning, but secondary delays can. Note that enough supplements in the planned travel times of the trains and enough buffer time between trains that use the same infrastructure, make a timetable immune against (propagating) delays. However, another key concern of (potential) railway passengers is the length of their travel time in practice. Preferably, they want to reach their destination as quickly as possible. Clearly this contradicts with adding large supplements and buffers. Therefore, Dewilde, Sels, Cattrysse, & Vansteenwegen (2011) introduced a new and comprehensive definition of robustness: a timetable and routing plan which minimize total passenger travel time in practice (in the presence of frequently occurring small delays) are called (passenger) robust. A passenger robust timetable will thus address both concerns of (potential) passengers.

In practice, however, the evaluation of all passenger travel times for different delay scenarios during timetable, routing and platform optimization is computationally very expensive for dense railway networks. In (Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014); Kroon, Huisman, & Maróti (2007)), the authors deal with this limitation by using an objective function that maximizes the time spans between any pair of trains. Simulation shows that striving for this simpler goal function does improve the passenger robustness. However, this alternative objective function shifts the focus from passengers to trains during optimization. For instance, consider the ordered train pairs (A, B) and (A, C) and suppose that there is only one passenger on train B and there are 1000 passengers on train C. The time spans between A and B, and A and C are equally important with respect to this alternative objective function which maximizes the time spans between any pair of trains. It should be noted, however, that the contribution to the total passenger travel time in practice of a knock-on delay of train A on train C will be a thousand times larger than that of a knock-on delay of train A on train B. To deal with this shortcoming, we adapt the approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) in order to take passenger numbers explicitly into account. This changes both the objective function used during optimization as well as the local search heuristic itself.

As the robustness definition of Dewilde, Sels, Cattrysse, & Vansteenwegen (2011) is based on real travel times, it could be used in the whole transport sector, for example for intermodal transport of passengers, for car traffic, etc. In this article, however, we restrict ourselves to passenger transport in a railway system. We also restrict our research to the timetabling and route planning. Although, for example, also network design, on the structural level, and real-time interventions, on the operational level, have an impact on the real travel times of railway passengers and thus on the robustness of the railway system. The considered delays are restricted to frequently occurring small delays, so large disturbances are also not considered.

Our research concentrates on complex and highly used station areas, which are in itself limited railway networks. Goldratt, & Cox (1986) argued that this approach of first making a good planning for the bottlenecks in the network and thereafter extending this planning to the whole network, is profitable. In our approach, we consider delays of trains upon entering the station area as primary delays. However, it could be that this entering delay is in fact a knock-on delay (secondary delay) incurred somewhere outside the restricted network, but anyway, the delay cannot be avoided by a better planning of the limited railway network. Historical data on these entering delays can be used to determine the typical delay distribution of each train when entering the station area (or, more generally, the railway network). This information can be used to address another shortcoming of the previously used objective function that maximizes the time span between any two trains, namely that all trains are treated equally and thus independently of their typical delays. For example, suppose that train X is on average five minutes late and trains Y and Z are always on time. According to the objective function of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014), the time span between trains X and Z is equally important as that between trains Y and Z. Nevertheless, a time span of five minutes between X and Z will very often be too short, while a five minute time span will always be sufficient between Y and Z. We will address this shortcoming explicitly in the objective function and the optimization approach. We show that this further improves the passenger robustness.

Extensive experiments are conducted in order to evaluate the positive and negative effects of both extensions separately and also the way in which these will interact. We used the dense railway station area of Brussels (Belgium) as a practical case study.

First, the state of the art is described. The approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) is contained in the second section to be able to point out the differences with our new method, which is also presented in this section. Then the results of the performed case studies are discussed and finally we summarize and conclude the paper in the last section.

1 State of the art

A railway system is typically planned in several steps that are dependent on the planning horizon (Huisman, Kroon, Lentink, & Vromans (2005)). This paper focuses on the timetabling and the route planning, which both are situated on the tactical level (medium long period planning). An extensive overview of the train routing problem at station level can be found in the survey paper of Lusby, Larsen, Ehrgott, & Ryan (2011). We refer to this paper for more details and references about these problems. Furthermore, robust timetabling is also covered by a large set of papers (Cacchiani, & Toth (2012)) in which each author captures the notion of robustness in their own way. Dewilde, Sels, Cattrysse, & Vansteenwegen (2011) and Oudt (2013) both present an overview of interpretations of robustness used by railway researchers. Dewilde, Sels, Cattrysse, & Vansteenwegen (2011) also present a new practical and comprehensive definition of robustness:
"A passenger robust railway system minimizes the total real weighted travel time of the passengers, in case of frequently occurring small disturbances."

The goal of our research is to improve the service towards the passengers by providing short and reliable real travel times. We strive towards this goal by producing a timetable and routing plan that is passenger robust according to the cited definition.

Another objective that focuses on the passengers is the minimization of the average passenger delay. This average passenger delay can differ from the average train delay to a large extent, as not all trains transport the same number of passengers and because a passenger can need multiple trains for his trip such that he can be delayed as a consequence of a missed transfer. Other research that focuses on passengers is for example described in Vansteenwegen, & Van Oudheusden (2006); Sels, Dewilde, Cattrysse, & Vansteenwegen (2013); Engelhardt-Funke, & Kolonko (2004); Takeuchi, Tomii, & Hirai (2007); Liebchen, Schachtebeck, Schöbel, Stiller, & Prigge (2010). The work of Vansteenwegen, & Van Oudheusden (2006) improves passenger robustness by only focusing on transfers, the work of Sels, Dewilde, Cattrysse, & Vansteenwegen (2013) derives the knock-on delays as passengers experience them in practice and they include these passenger numbers and knock-on delays in their objective function, Engelhardt-Funke, & Kolonko (2004) concentrate on the evaluation of infrastructure changes from the viewpoint of passengers, Takeuchi, Tomii, & Hirai (2007) use passenger criteria in simulation but provide no optimization algorithm and Liebchen, Schachtebeck, Schöbel, Stiller, & Prigge (2010) improve passenger service by increasing the nominal travel time, but this is not achievable in our research as we work with densely used bottlenecks.

This article presents a new approach to take recurring delays explicitly into account in timetabling and routing planning without using a stochastic approach as in (Kroon, Dekker, & Vromans (2007); Kroon, Maróti, Helmrich, Vromans, & Dekker (2008)). These latter approaches build in a simulation module into the optimization algorithm, which make them computationally limited.

2 Approach

In order to improve passenger robustness, we start from the approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014). They developed an algorithm that indirectly improves the passenger robustness of an existing timetable by performing small changes to this timetable, in this way their approach only incurs a small cost to the railway operator and the railway infrastructure manager. As we will modify this algorithm, we discuss their approach in more detail. We first describe their objective function and thereafter the different parts in their optimization algorithm. Then our modifications to this approach are explained.

2.1 Framework model

The original algorithm consists of three modules: the routing module, the timetabling module and the platforming module, which iteratively improve the spreading of the trains and thus the passenger robustness. Their resulting timetable is evaluated with a simulation model, in which small delays drawn from an exponential distribution based on historical delays are added to the arrival and departure times of the trains (10 000 runs). These delays cause that train trips cannot pass off as planned, which results in knock-on delays and prolonged real travel times. This simulation model makes it possible to compare the passenger robustness of a new timetable and a reference timetable.

Goal function

Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) argue that directly optimizing passenger robustness makes the approach complicated and time consuming, as real travel times of passengers are to be known, which require delay propagation computations. Instead of directly optimizing passenger robustness, their algorithm strives to maximize the sum of the shortest buffer time between each pair of trains. If two trains use the same infrastructure element on their route, then the time span between the reservations of that infrastructure element is defined as a buffer time between these two trains. The shortest buffer time is where both trains drive closest after each other if they drive in the same direction, or where they cross each other if they drive in different directions. This goal function can be represented by:

$$\sum_{t,t' \in T} C_{(t,t')}, \tag{1}$$

where T is the set of all trains in the considered railway system, t and t' represent two different trains ($t, t' \in T$) and $C_{(t,t')}$ is defined as

$$C_{(t,t')} = \begin{cases} 15 & \text{if } B_{(t,t')} \leq 0 \quad \text{(conflict);} \\ \frac{1}{B_{(t,t')}} & \text{if } 0 < B_{(t,t')} \leq 15; \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $B_{(t,t')}$ is the shortest buffer time between trains t and t'. If trains t and t' have no common infrastructure, then $B_{(t,t')}$ is assigned the value infinity. If the routing plan and timetable cause a conflict between trains t and t', then $B_{(t,t')}$ is assigned a value smaller than 0 and $C_{(t,t')}$ is set to 15. To improve the value of this goal function, trains can be spread in space and time. For example, choosing another route for a train, can cause that two trains no longer overlap in space and thus that their minimal buffer time becomes infinite and the corresponding cost becomes zero. Another example is a small shift in arrival and departure times of a train, such that the minimal buffer time between this train and another train increases with the length of this shift. Of course, before carrying out these changes, one has to check whether the new route or the shift in arrival and departure times does not deteriorate the shortest buffer times in between the adapted train and other trains.

Remark further that in this goal function the shortest buffer time between any train pair is equally important, independently of the number of passengers on these trains and the typical delays of both trains.

Routing module

The input for this module is a set of possible routes for each train and the arrival and departure times in each station. In this module, a route is assigned to each train such that the total cost in equation (1) is minimized by a spreading in space. The aim that there are no conflicts is implemented as a hard constraint. First the shortest buffer time between each two routes (of different trains) is calculated. Then, for the assignment itself, the model below, (3)-(8), is implemented in C++ and solved by using Cplex 12.6.

Let R_t be the set of routes for train t. Define for all train-route combinations $(t, r) \in T \times R_t$ the non-negative, continuous variable $z_{(t,r)}$ that represents the spreading cost of assigning route r to train t. The variables $x_{(t,r)}$ are binary variables which indicate whether train-route combination (t, r) is selected or not.

$$\text{Minimize} \sum_{t \in T} \sum_{r \in R_t} z_{(t,r)}, \tag{3}$$

subject to

$$\sum_{t' \in T} \sum_{r' \in R_{t'}} C_{(t,r),(t',r')} \big(x_{(t,r)} + x_{(t',r')} - 1\big) \le z_{(t,r)} \qquad \forall (t,r) \in T \times R_t, \tag{4}$$

$$\sum_{r \in R_t} x_{(t,r)} = 1 \qquad \forall t \in T, \tag{5}$$

$$x_{(t,r)} + \sum_{r' \in R^c_{t'}} x_{(t',r')} \le 1 \qquad \forall (t,r) \in T \times R_t, t' \in T \setminus \{t\}, \; R^c_{t'} = \{r' \in R_{t'} : B_{(t,r),(t',r')} = 0\}, \tag{6}$$

$$z_{(t,r)} \ge 0 \qquad \forall (t,r) \in T \times R_t, \tag{7}$$

$$x_{(t,r)} \in \{0, 1\} \qquad \forall (t,r) \in T \times R_t. \tag{8}$$

For each train-route combination, constraint (4) sets a lower bound for the corresponding term in the objective function: if both route r and route r' are selected for trains t and t' respectively, then the value of the decision variables of these train-route combinations ($z_{(t,r)}$ and $z_{(t',r')}$) has to be at least the cost corresponding to their shortest buffer time. Constraint (5) provides that exactly one route is assigned to every train. Constraint (6) is the hard constraint that prevents that conflicting routes can be assigned. Constraints (7) and (8) are the sign restrictions and type settings for the decision variables.

This model is an extended version of a node packing model. The node packing constraint is represented by constraint (6). The continuous variables $z_{(t,r)}$ are created to linearize the more intuitive objective function by using Kaufman's method (Kaufman, & Broeckx (1978)):

$$\sum_{(t,r)\in T\times R_t,(t',r')\in T\times R_{t'}} C_{(t,r),(t',r')}x_{(t,r)}x_{(t',r')}. \tag{9}$$

By assigning the most convenient set of routes, the cost function is improved by a better spreading in space.

Timetabling module

The input for this module are the routes that were fixed in the routing module for each train. The minimal buffer times between trains can be further increased by changing the arrival and departure times of trains in the stations and by changing the order of the trains. However, changes in arrival and departure times may not cross the fixed time window to assure the frequencies of the trains. Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) argue that an exact model would be (too) large and complex for the case studies they consider, as it would require the inclusion of the minimum buffer times for all possible variations in arrival and departure times for all trains. That is why they use a heuristic approach: they evaluate step by step the effect of a shift in the arrival and departure times of one train, a combined shift of arrival and departure times of more trains and an order swap of two trains. If necessary, conflicts can be resolved by applying multiple parallel changes before evaluating the entire move. This also allows to check the potential of a change. In case a change incurs no conflict and the cost function improves, the timetable is adapted. The timetabling module also contains a smallest ascent procedure to escape from local optima.

In the timetabling module, the cost function is improved by a better spreading in time.

Platforming module

Both in the routing module as in the timetabling module, the platform of each train in each station is fixed. In this module, a set of trains is selected for which a platform change is considered and carried out only if it improves the objective function (1). In order to assess the potential of a platform change, newly created timetabling opportunities are included.

Using a new platform and a new route towards this platform improves the cost function by a better spreading in space. Since also promising timetable changes are taken into account, the cost function also improves by a better spreading in time.

Interaction of the modules

The algorithm starts by performing the routing module. Thereafter the timetabling module is started. If there is no or hardly no improvement by changing the arrival and departure times and the orders of trains, then the platforming module is applied and thereafter again the routing module. Conversely, if the timetabling module produces a considerable improvement, then the platforming module is skipped and the routing module is directly performed again. This loop is repeated until there is no improvement anymore or until a certain number of repetitions is reached. An overview of this algorithm is presented in Figure 1. The internal timetabling module represents the timetable module that is called inside the timetabling module or the platforming module to assess the potential of timetable or platform changes.

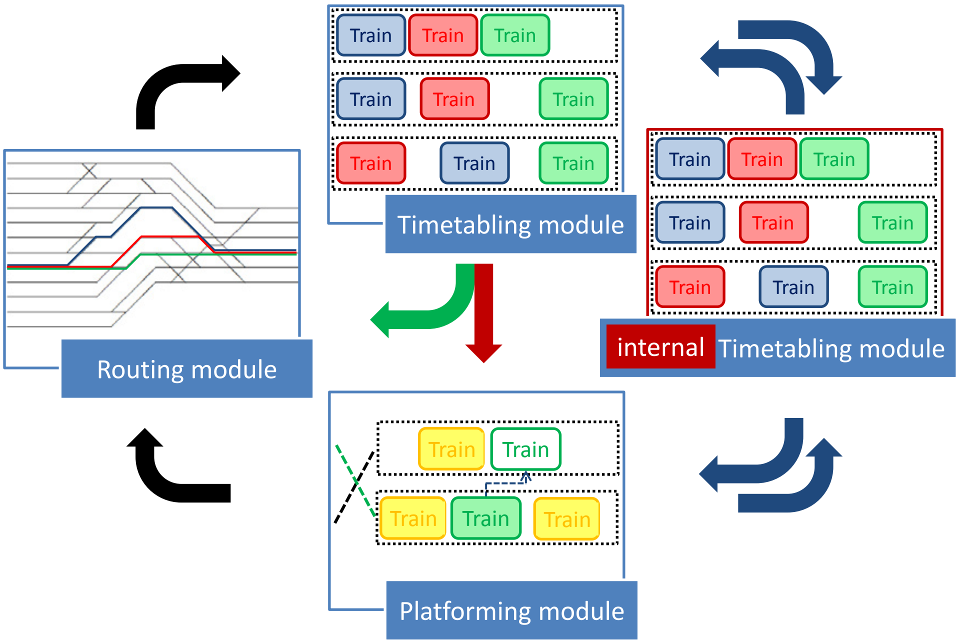

Figure 1: Overview of the optimization algorithm of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014).

2.2 Introducing passenger numbers

To improve the total real travel time in practice of all passengers, it seems more appropriate to make trains with a higher occupation less liable to delays than hardly occupied trains. We use this observation to improve the algorithm sketched above.

One way to make a train less liable to delays is to increase the buffer time between this train and the trains that use the same infrastructure before this train. As the algorithm of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) strives to increase the shortest buffer times between train pairs, an easy improvement could be achieved by giving priority to train pairs with a short buffer time and a high occupation rate of the second train (the train that uses the shared infrastructure the latest).

A well-known method to prioritize is to use weights. The costs $C_{(t,t')}$ and $C_{(t,r),(t',r')}$ in the objective function (1) and (9) make it clear which train pairs and train-route combinations have the shortest buffer times and thus the most negative impact on the goal function (1). We now want to assign weights to the shortest buffer times between trains, such that the size of these weights indicates the following: the higher this new weight the more passengers are affected by a delay of the first train. The new weights thus reveal the number of passengers whose real travel times have a higher chance to become longer than the planned travel times in case of a delay of the first train. Thus the higher this weight the more important it is to spread the corresponding trains well. To achieve this, we set these new weights equal to the number of passengers that travel with this second train on the moment that the train uses the shared infrastructure together with the number of passengers that board on this train in one of the stations after the common infrastructure. If there are different places in the network where both trains use the same infrastructure, the moment where the shortest buffer time between the two trains is reached, is chosen to count the passengers. If two trains do not share infrastructure, then there is no second train, and the weight is set to 0, just like the cost incurred by the 'infinite buffer time' between these two trains. The definition of these weights is based on the assumption that the travel time of the passengers that are on the second train on and after the moment that the shortest buffer time is reached has the highest chance to be lengthened. The travel time of the passengers on the first train will not be affected by a short buffer time between this train and the next train in case of a delay of this train, that is why we don't take them into account in the weights.

Implementation

We now explain how these new weights are practically implemented in the algorithm of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014). For each pair of train-route combinations $\big((t,r),(t',r')\big)$, the place in the network where the shortest buffer time appears can easily be determined by computing the points in time that these trains enter and leave the shared infrastructure elements. At this place we then determine the number of passengers that are on the second train or will board in an upcoming station onto this train.

We will indicate the number of passengers on the second train of train-route combinations (t,r) and (t',r') as $P_{(t,r),(t',r')}$ (or as $P_{(t,t')}$ if the routes are fixed). Remark that $P_{(t,r),(t',r')}$ depends on the routes as the place where the minimal buffer time is reached can change if one of both trains takes another route, such that the number of passengers can be different as passengers can enter and leave the second train in stations in between both places. An example is provided in Figure 2. Suppose that train A releases the common infrastructure on the left of the network three minutes before train B enters the network and has a dwell time of one minute in the station and train B has a dwell time of two minutes in the station. Then, the buffer time between A and B before they reach the station is three minutes and after the station the buffer is increased to four minutes in the left part of the picture, assuming that the blocking time of A is the same for the section before and after the station. Thus in the left part of the picture, the shortest buffer time between both trains is three minutes and achieved before the station and the assigned weight will be the sum of the number of passengers that are on this moment on train B and the number of passengers that will board on train B in the station. In the right part of the picture, the shortest buffer time between trains A and B is four minutes and the critical place is located after the station, so the assigned weight will be number of passengers that are on that moment on the train. Both weights differ by the number of passengers that leave train B in the station. What is more, selecting another route for one of the two trains can even cause that these routes of both trains are no more overlapping such that there is no meaningful assignment of a place where the minimal buffer time is reached.

First, the weights can be introduced into *objective function (1)* as follows:

$$\sum_{t,t' \in T} C_{t,t'} P_{t,t'}, \tag{10}$$

which is now a weighted cost function. The negative impact of short buffer times between trains for which the second train transports many passengers becomes explicit.

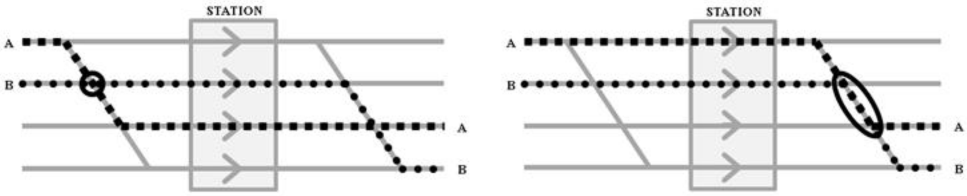

Figure 2: Example: Suppose that the arrival time of A at the station is at $t = 0$ and that of B on $t = 3$, the dwell time of A is 1 minute and that of B is 2 minutes. Let r be the route of train A on the left, r' the route of train A on the right and r'' the route of train B. The weight $P_{(A,r),(B,r'')}$ will differ from the weight $P_{(A,r'),(B,r'')}$ by the number of passengers that leave train B in the station as the shortest buffer time in the left part of the picture is reached before the station and in the right part of the picture after the station.

In *the routing module*, the weights can be added to the model in the same way as the costs. Only constraint (4) changes and becomes:

$$\sum_{t' \in T} \sum_{r' \in R_{t'}} C_{(t,r),(t',r')} P_{(t,r),(t',r')} \big(x_{(t,r)} + x_{(t',r')} - 1 \big) \leq z_{(t,r)}. \tag{11}$$

This implies that the value of $z_{(t,r)}$ will be at least the product of $C_{(t,r),(t',r')}$ and $P_{(t,r),(t',r')}$ if train-route combinations (t, r) and (t', r') are assigned to trains t and t' respectively. Thus the higher $P_{(t,r),(t',r')}$, the less plausible it is that train-route combinations (t, r) and (t', r') will both be selected.

In the *timetabling module* the goal function can be improved by changing the arrival and departure times of the trains and the order of trains. The former change can shift the place where the shortest buffer time occurs and thus the weight coming from the passenger numbers can change as passengers can enter and leave the train in stations between the old and new place. Further, the latter change has also an impact on $P_{(t,t')}$, because only the number of passengers on the second train of a train pair counts, such that after an order swap the number of passengers on the other train becomes important.

Another modification of the algorithm of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) is related to the selection of train pairs which will be considered for timetable adaptations. In Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) this selection is based on the shortest buffer times. We adapt this selection procedure by taking also the passenger numbers of the second train into account. We select the train pairs according to their value of $C_{(t,t')} P_{(t,t')}$. In dealing with the selected train pairs, we also adjust the order in which train pairs are considered: in the algorithm of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) the selected train pairs are dealt with in a random order, we treat them according to their negative impact on the goal function (10).

We left *the platforming module* as it was, but we used the altered objective function (10) to evaluate considered platform changes.

2.3 Introducing recurring delays

Suppose that a single train regularly enters a restricted railway area with a delay. We will investigate if it is favorable to take this recurring delay explicitly into account in the planning of the restricted network. We could argue that the buffer times between this train and trains that use the same infrastructure shortly after this train have to be at least as large as the mean recurring delay of that first train. Indeed, otherwise the second train of the considered train pair is often halted at the common infrastructure to avoid a conflict. But of course the fixed time window in which all trains has to be planned limits this time span. The time that the second train has to wait while the first train clears the common infrastructure, causes a lengthening of the real travel time for all passengers on that second train.

Suppose now that the second train of a train pair is recurrently delayed when it enters the restricted network. In this case, the buffer time between the two trains is lengthened by the time length of the recurring delay. Thus it becomes less urgent to deal with that buffer time in the optimization algorithm. So, not only the mean recurring delay of the first train has an impact on the priority of the buffer time between a train pair, but also the mean recurring delay of the second train has to be taken into account. A recurring delay of the first train makes the buffer time between a train pair more critical, while a recurring delay of the second train makes it less critical (Sels, Dewilde, Cattrysse, & Vansteenwegen (2013)).

We want to include both opposing effects in the optimization algorithm. Observe that the recurring delays of both trains affect the time span between a train pair. This is why we will again assign weights to the buffer times to indicate the need for lengthening that buffer time. A buffer time will get a weight bigger than 1 if it becomes more important to lengthen that buffer time as a consequence of the recurring delays. A buffer time will get a weight smaller than 1 if it becomes less important to lengthen that buffer time. For example, if the mean recurring delay of the first train is higher than that of the second train, which increases the chance of a conflict to occur, then that buffer time becomes more critical and thus gets a weight bigger than 1.

Implementation

We represent the mean recurring delay for train t as D_t. We first give some examples to offer an interpretation of the weights that we will assign to the buffer times to include the effect of recurring delays. Suppose that the buffer time between trains t and t' is five minutes and that the mean recurring delay of train t, which uses first the common infrastructure, is one minute and that train t' is never delayed. Then the "altered" buffer time is 20% shorter ($\frac{D_t}{B_{(t,t')}} = \frac{1}{5} = 20\%$). We could now find it $\frac{1}{1-20\%}(> 1)$ times more important to increase the buffer time between t and t'. Alternatively, suppose that train t' has a mean recurring delay of two minutes and that train t is never delayed. Then the altered buffer time is 40% longer, such that we could find it $\frac{1}{1+40\%}(< 1)$ times less important to further increase this buffer time. If train t now has a mean recurring delay of one minute and train t' has a mean recurring delay of two minutes, then the buffer time between t and t' is on average lengthened with 20% and also here we could find it now $\frac{1}{1+20\%}(< 1)$ less important to further increase this buffer time. The relative importance that we assign to the buffer times in these three situations, to indicate how much more or less important it is to increase the planned buffer time, can be embedded in one formula:

$$D_{(t,t')} = \frac{1}{1 - \frac{D_t - D_{t'}}{B_{(t,t')}}}, \tag{12}$$

where $D_{(t,t')}$ is the relative importance assigned to the buffer times when recurring delays are taken into account and where we suppose, without loss of generality, that train t uses the common infrastructure before train t'. We make the restriction that the weighted cost can never be bigger than the cost of a conflicting train pair:

$$\forall t, t' \in T : D_{(t,t')} C_{(t,t')} \leq 15. \tag{13}$$

Remark that formula (12) is not meaningful in the following cases:

- if $B_{(t,t')} \leq 0$, we assign $D_{(t,t')}$ the value 1, such that the weighted cost $D_{(t,t')} C_{(t,t')} (= 1 \cdot 15 = 15)$ remains maximal;
- if $B_{(t,t')} > 0$ and $\frac{D_t - D_{t'}}{B_{(t,t')}} > 1 \Leftrightarrow B_{(t,t')} - D_t + D_{t'} < 1$, in this case the mean recurrent delay of the first train exceeds the buffer time summed up with the mean recurring delay of the second train, so that both trains are in conflict taking the mean recurrent delays into account), so here we assign $D_{(t,t')}$ the value $15 \cdot B_{(t,t')}$, such that $D_{(t,t')} C_{(t,t')} = 15 \cdot B_{(t,t')} \frac{1}{B_{(t,t')}} = 15$, which equals the cost of a conflicting train pair;
- if t and t' have no common infrastructure, the value of $D_{(t,t')}$ is of no importance as $C_{(t,t')} = 0$ in this case, we assign it the value 1.

If we now assign $D_{(t,t')}$ as weights to the cost corresponding to the planned buffer times between trains, then buffer times that are in risk to be decreased by the impact of the recurring delays of both trains will get a weight larger than one and thus will get more priority to be increased by the algorithm. And the other way round, buffer times that are lengthened by the impact of the recurring delays of both trains, will get a weight smaller than one and thus will get less priority to be increased by the algorithm. The new objective function where, for this experiment, we neglect the impact of the passengers, but only take the impact of the recurring delays on the buffer times into account, becomes:

$$\sum_{t,t' \in T} D_{(t,t')} C_{(t,t')}. \tag{14}$$

This is again a weighted cost function.

Remark now that if we put in the formulas for $D_{(t,t')}$ and $C_{(t,t')}$ for a train pair with $0 < B_{(t,t')}$ and $B_{(t,t')} - D_t + D_{t'} > 1$, that we get:

$$D_{(t,t')} C_{(t,t')} = \sum_{t,t' \in T} \frac{1}{1 - \frac{D_t - D_{t'}}{B_{(t,t')}}} \frac{1}{B_{(t,t')}} \tag{15}$$

$$= \frac{1}{B_{(t,t')} - D_t + D_{t'}}, \tag{16}$$

$$\tag{17}$$

where the denominator of this last formula represents in fact the "altered" buffer time: the buffer time that is left in between both trains if the mean recurrent delays are taken into account.

So to implement these altered buffer times in the algorithm, we use weights again. These weights play the same role in the optimization algorithm as the weights based on the passenger numbers, so we refer to the previous paragraph for the description of that implementation.

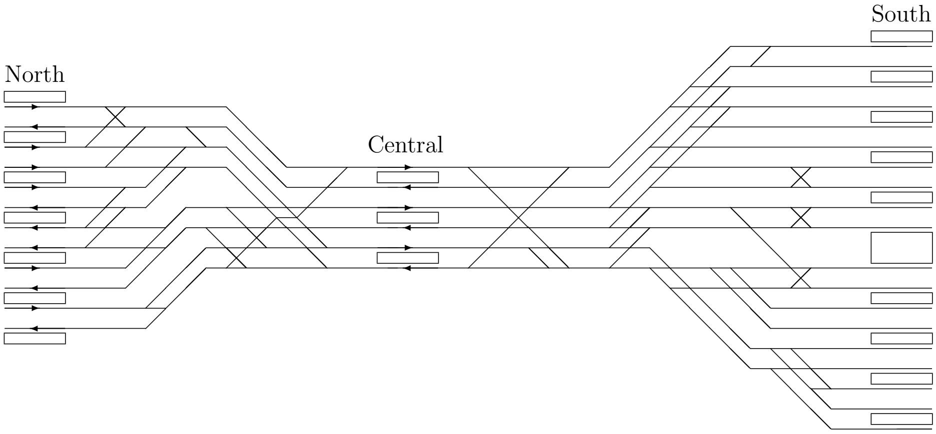

Figure 3: The core of Brussel's dense railway area

3 Results

We tested our approach on a case study that is based on that of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014), but we extended the dense railway area of Brussels such that it also includes the beginning of the open tracks, the outer grids, and the entrances to the shunt yards. This area contains three out of five of Belgium's busiest stations and is a real bottleneck. Indeed, The 19 platform tracks of Brussel-Zuid are connected via 3 tunnels with 6 platform tracks in Brussel-Centraal to 12 platform tracks in Brussel-Noord. In addition of the tracks to a shunting place, Brussel-Zuid en Brussel-Noord have several tracks that lead trains from all over the country to and from Brussels, which cause the need for a lot of switches in this area. The core of the considered area is represented in Figure 3. The case study contains 84 trains that pass through this area during peak hour in the morning (7h-8h). These trains almost saturate the network, which makes this area interesting to test the presented approach. We assume that each train enters the network with a delay. The delay for each train follows an exponential distribution with a known average value. The average value for each train is assigned randomly between zero and eight minutes.

Both in the simulation model of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) as in the optimization algorithm that takes recurrent delays into account, it is assumed that trains are delayed by their mean recurrent delay upon entering the network. In the optimization model, we take this mean recurring delay into account, in the simulation model this mean recurring delay of a train defines the exponential distribution from which the simulated entering delay of that train is drawn. These simulated delays then represent initial disturbances (primary delays). Furthermore, the same passenger numbers are used in the simulation model as in the optimization algorithm that takes passenger numbers into account.

Table 1 represents an overview of the simulation results. The columns in Table 1 correspond to the instances that are solved: the reference system for which spreading was not an issue (**Ref**); the resulting timetable and routing plan obtained by the algorithm of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) (**Dewilde et al.**); the resulting timetable by taking passenger numbers into account in the approach ($\mathbf{P}_{(\mathtt{t},\mathtt{t}')}$); the resulting timetable by

Table 1: Overview of the simulation results: the new approaches improve passenger robustness up to 11.33%.

Timetable	Ref	Dewilde et al.	$\mathbf{P}_{(\mathbf{t},\mathbf{t}')}$	$\mathbf{D}_{(\mathbf{t},\mathbf{t}')}$	$\mathbf{P}_{(\mathbf{t},\mathbf{t}')}\mathbf{D}_{(\mathbf{t},\mathbf{t}')}$
Passenger robustness	$4.15 \cdot 10^6$	$3.84 \cdot 10^6$	$3.77 \cdot 10^6$	$3.72 \cdot 10^6$	$3.68 \cdot 10^6$
% *improvement*		*7.47%*	*9.16%*	*10.36%*	*11.33%*
Total knock-on delay per hour for all trains (minutes)	211.53	145.31	142.17	136.38	138.40
% *improvement*		*31.30%*	*32.79%*	*35.53%*	*34.57%*
Percentage extra delayed trains	47.71%	37.36%	38.53%	36.98%	37.66%

taking recurring delays into account ($\mathbf{D}_{(\mathbf{t},\mathbf{t}')}$) and the resulting timetable by taking both passenger numbers and recurring delays into account ($\mathbf{P}_{(\mathbf{t},\mathbf{t}')}\mathbf{D}_{(\mathbf{t},\mathbf{t}')}$). The reference timetable and routing plan that is used is the same as the one that is used by Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014), but the passenger numbers and mean recurring delays in the simulation differ. Therefore, the results in the column of Dewilde et al. have been recalculated for these passenger numbers and delays. The optimization algorithm of each of these approaches takes 60 to 90 minutes and the simulation model takes a few seconds.

We don't compare the objective function values as we use different weights in the objective function for all instances. The first row in Table 1 presents the (passenger) robustness value for the instances. The lower this value, the better, as it represents the total weighted travel time of all passengers. We see that our new approaches further improve the passenger robustness by $1.69 - 3.86\%$ compared with the approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014). In comparison with the reference timetable and routing plan, the passenger robustness improves with $9.16 - 11.33\%$. The second row presents the total knock-on delay of all trains in one hour, thus the lower this value the less knock-on delays for trains. Also here, the new approaches give an improvement of $1.49 - 4,23\%$ compared with the results from Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014). The third row present the percentage of trains that leave the network with a larger delay than that they entered the network. The number of extra delayed trains in the approaches where passenger numbers are taken into account slightly deteriorate compared to the approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014). This is because the optimization approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014) focuses on trains: by maximizing the spreading between trains, the number of trains that gets delayed diminishes. However, an approach that takes passenger numbers into account strives to avoid delays for passengers: it could be more profitable to delay a number of trains with few passengers than one train with many passengers. Furthermore, we see that the approach that only takes recurrent delays into account further diminishes the number of extra delayed trains compared to the approach of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014).

We also investigated the effect of first applying the optimization algorithm that takes passenger numbers into account on the reference timetable and routing plan and thereafter the optimization algorithm that takes recurring delays into account on the resulting timetable and routing plan of the previous optimization (Table 2: **1st:** $\mathbf{P}_{(\mathbf{t},\mathbf{t}')}$, **2nd:** $\mathbf{D}_{(\mathbf{t},\mathbf{t}')}$). The passenger robustness is slightly improved compared to the value of each of the approaches separately.The total knock-on delay per hour improves largely and there is a small improvement in the number of extra delayed trains. More or less the same observation holds for applying the optimization algorithms in the other order (Table 2: **1st:** $\mathbf{D_{(t,t')}}$, **2nd:** $\mathbf{P_{(t,t')}}$). The results are presented in Table 2 where only the last two columns contain new information in comparison with Table 1.

Table 2: Overview of the simulation results of the combination of taking both recurring delays and passenger numbers into account.

Timetable	**Original**	**Dewilde et al.**	$\mathbf{P_{(t,t')}D_{(t,t')}}$	**1st:** $\mathbf{P_{(t,t')}}$, **2nd:** $\mathbf{D_{(t,t')}}$	**1st:** $\mathbf{D_{(t,t')}}$, **2nd:** $\mathbf{P_{(t,t')}}$
Passenger robustness	$4.15 \cdot 10^6$	$3.84 \cdot 10^6$	$3.68 \cdot 10^6$	$3.71 \cdot 10^6$	$3.74 \cdot 10^6$
% improvement		*7.47%*	*11.33%*	*10.60%*	*9.88%*
Total knock-on delay per hour for all trains (minutes)	211.53	145.31	138.40	133.66	134.04
% improvement		*31.30%*	*34.57%*	*36.81%*	*36.63%*
Percentage extra delayed trains	47.71%	37.36%	37.66%	37.13%	36.64%

4 Conclusion

In this paper, new approaches to improve the passenger robustness in railway bottlenecks are introduced. These approaches build further on the algorithm developed by Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014), which improves the passenger robustness by a better spreading of trains in a fixed time window. A first new approach takes passenger numbers explicitly into account in the production of a timetable and routing plan by using weights in the objective function and an alternative selection of train pairs to be spread. The second approach uses information of recurring delays. By making the spreading of trains more or less important dependent on the mean recurrent delays of both trains, a structural delay of the first train will systematically decrease the time span between both trains and a delay of the second train will increase the time span between both trains. The third approach combines the other two approaches. Computational results for the Brussels' area, the largest bottleneck in the Belgian railway network, show that these approaches further improve passenger robustness up to 11.33%, which is in itself an improvement of 3.86% compared to the results of Dewilde, Sels, Cattrysse, & Vansteenwegen (2013, 2014).

A thought for further research is to include also mean dwell delays in the optimization model in addition to the mean recurring delays upon entering the network and to include structural changes to the algorithm to take passenger numbers or recurring delays into account.

References

Cacchiani, V., Toth, P. (2012). "Nominal and robust train timetabling problems", *European Journal of Operational Research*, vol. 219, pp. 727-737.

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 (RailRome2011)*, Rome, Italy.

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P. (2013). "Robust railway station planning: An interaction between routing, timetabling and platforming", *Journal of Rail Transport Planning & Management*, vol. 3, pp. 68-77.

Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P. (2014). "Improving the robustness in railway station areas", *European Journal of Operational Research*, vol. 235, pp. 276-286.

Engelhardt-Funke, O., Kolonko, M. (2004). "Analysing stability and investments in railway networks using advanced evolutionary algorithms", *International Transactions in Operational Research*, vol. 11, pp. 381-394.

Goldratt, E.M., Cox, J. (1986). "The Goal: A Process of Ongoing Improvement", *Nord River Press*.

Huisman, D., Kroon, L., Lentink, R., Vromans, M., 2005. "Operations research in passenger railway transportation", *Statistica Neerlandica*, vol. 59, pp. 467-497.

Kaufman, L., Broeckx, F. (1978). "An algorithm for the quadratic assignment problem using Bender's decomposition", *European Journal of Operational Research*, vol. 2, pp. 207-211.

König, A., Axhausen, K.W. (2002). "The reliability of the Transportation System and its Influence on the Choice Behaviour", In: *Proceedings of the 2nd Swiss Transportation Research Conference*, Monte Verit, Switzerland.

Kroon, L.G., Huisman, D., Maróti, G. (2007). "Railway Timetabling from an Operations Research Perspective", In: *Econometric Institute Report EI2007-22*.

Kroon, L.G., Dekker, R., Vromans, M.J.C.M (2007). "Cyclic Railway Timetabling: a stochastic Optimization Approach", *Lecture Notes in Computer Science"*, vol. 4359, pp. 41-66.

Kroon, L., Maróti, G., Helmrich, M.R., Vromans, M., Dekker, R. (2008). "Stochastic improvement of cyclic railway timetables", *Transportation research part B*, vol. 42, pp. 553-570.

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S., Prigge, A. (2010). "Computing delay resistant railway timetables", *Computers & Operations Research*, vol. 37, pp. 857-868.

Lusby, R., Larsen, J. Ehrgott, M., Ryan, D. (2011). "Railway track allocation : Models and methods", *OR Spectrum*, vol. 33, pp. 843-883.

Oudt, S. (2013). "A Robust Railway System, A theoretical theoretical approach to enhancing the robustness of the railway system in the Netherlands", *Master thesis Science and Business Management (August 2013)*, Universiteit Utrecht, Utrecht.

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

Takeuchi, Y., Tomii, N., Hirai, C. (2007). "Evaluation method of robustness for train schedules", *Quarterly Report of Railway Tecnical Research Institute*, vol. 48, pp. 197-201.

Vansteenwegen, P., Van Oudheusden, D. (2006). "Decreasing the passenger waiting time for an intercity rail network", *Transportation Research Part B: Methodological*, vol. 41, pp. 478-492.