



Delft University of Technology

Stable and robust train routing in station areas with balanced infrastructure capacity occupation

Bešinović, Nikola; Goverde, Rob M.P.

DOI

[10.1007/s12469-019-00202-3](https://doi.org/10.1007/s12469-019-00202-3)

Publication date

2019

Document Version

Final published version

Published in

Public Transport

Citation (APA)

Bešinović, N., & Goverde, R. M. P. (2019). Stable and robust train routing in station areas with balanced infrastructure capacity occupation. *Public Transport*, 11(2), 211-236. <https://doi.org/10.1007/s12469-019-00202-3>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Stable and robust train routing in station areas with balanced infrastructure capacity occupation

Nikola Bešinović¹ · Rob M. P. Goverde¹

Accepted: 27 April 2019 / Published online: 17 May 2019
© The Author(s) 2019

Abstract

Routing trains through busy railway station layouts is an important part of the time-tabling process. For each train, a feasible route has to be determined to provide reliable operations, given the arrival and departure times at stations. In this paper, we propose a model for stable and robust train routing with the goal to minimize capacity occupation and maximize robustness. We define a multi-objective optimization problem and provide the heuristic RouteCare based on a max-plus automata model and a delay propagation model. We consider microscopic infrastructure to guarantee practical feasibility. The performance of the proposed algorithm is demonstrated on real-life instances of the Dutch railway network. The generated solutions outperformed the variants of RouteCare that independently maximize stability or robustness by 10.4% and 9.5%, respectively. In addition, RouteCare showed that even for the same number of resources used, a more robust route plan can be found that uses the station capacity more efficiently.

Keywords Railway · Routing · Conflict-free · Stable · Robust · Max-plus algebra

1 Introduction

Timetable planning becomes more challenging with increasing demands and needs more attention and more effort from traffic planners. The need for more reliable planning is constantly growing. In particular, stations are locations in the network that are usually recognized as bottlenecks where most of the delays are being generated. In addition, intensive use of infrastructure resources (e.g., track sections, switches, crossings) requires periodic and frequent preventive maintenance to keep a high level of service. These challenges raise the following questions: 1. *How to incorporate stability measures in the planning process to obtain an acceptable level of stability?* 2. *How to assign more robust routes with less interdependencies?* 3.

✉ Nikola Bešinović
n.besinovic@tudelft.nl

¹ Department of Transport and Planning, Delft University of Technology, Delft, The Netherlands

How to increase the availability of the infrastructure and reduce the required frequency of maintenance? Answering all these questions serve the single goal of providing a better and more reliable service to passengers.

To quantify the quality of scheduling solutions, Goverde and Hansen (2013) proposed design parameters such as efficiency, feasibility, stability and robustness. Timetable *efficiency* is the ability of trains to have travel times as short as possible. Timetable *feasibility* is the ability of all trains to adhere to their scheduled train paths, which are conflict-free considering microscopic details, i.e., all trains can proceed undisturbed by other traffic. Timetable *stability* is the ability of a timetable to absorb initial and primary delays so that delayed trains return to their scheduled train paths without rescheduling. This is directly connected to the infrastructure occupation. If the occupation is higher, then the remaining time allowances are lower and provide a less stable timetable. Timetable *robustness* is the ability of a timetable to withstand design errors, parameter variations, and changing operational conditions. We use these definitions of timetable indicators and translate them to the station level when designing route plans. To satisfy these design indicators, we need to consider different mathematical models both on a microscopic and macroscopic level of detail (Goverde et al. 2016). A microscopic level considers the detailed infrastructure (block lengths, curvatures, gradients) and signalling system, while a macroscopic level considers relevant timetable points (e.g., stations, junctions, bridges) as nodes and tracks between them as arcs. If microscopic details were not considered but instead default minimum headway norms are used, then we refer to normative or macroscopic feasibility.

In this paper, we consider an extension of the train routing problem (TRP) that was introduced by Zwaneveld et al. (1996). The TRP consists of choosing an appropriate route for each train arriving or departing in a station area and platform assignment where each train is assigned to one station platform track. The input to the TRP is usually a predefined macroscopic timetable that includes a number of trains and the scheduled arrival and departure times in the station. The output of the TRP is a *route plan*, which represents a set of compatible train routes consisting of an infrastructure itinerary assigned to each train at the appropriate time according to the timetable. In addition, a route plan is defined on the microscopic level.

The TRP attracted a lot of attention in the past and different approaches to model the TRP are found in the literature, particularly focusing on determining macroscopically feasible and efficient train routes. Carey and Carville (2003) gave a heuristic approach similar to the manual planning methods currently used in Great Britain. Zwaneveld et al. (1996, 2001) suggested formulating the TRP as a node packing problem, while Lusby et al. (2011a) proposed the set packing formulation. Zwaneveld et al. (1996, 2001) defined a TRP on a conflict graph, which models all conflicting train routes. While doing so, they considered only aggregated train routes due to the size of problem instances, and thus cannot guarantee the conflict-freeness of the produced route plan.

Caimi et al. (2011) defined the problem as a multicommodity flow problem. They considered the TRP on a Resource Tree Conflict Graph which computes a route plan that is conflict-free at the microscopic level and gave a computationally efficient model. Corman et al. (2009) proposed an Alternative Graph formulation

for rescheduling and rerouting using an incompatibility graph and blocking times. The incompatibility graph models all routes that cannot be used simultaneously. All incompatible routes are then included in the Alternative Graph, which also requires sequence-dependent setup times to model the sectional-release route locking principle. A more efficient aggregated route formulation is furthermore introduced which computes slightly conservative microscopically conflict-free solutions with small extra buffer times. An extensive overview of the state-of-the-art is given in Sels et al. (2014) for periodic routing and Lusby et al. (2011b) and Cacchiani et al. (2014) for non-periodic cases. In our paper, the focus is on the periodic TRP version, but the approach may easily be adjusted to the non-periodic problem as well.

It has been commonly accepted that delays of trains are unavoidable in real-time operations as being subject to internal and external stochastic disturbances. In order to absorb these disturbances in real-time, a route plan should contain time supplements in the train process times and buffer times between pairs of consecutive train movements. These time allowances help to reduce delays and make the route plan more robust. Therefore, robustness of route plans is a necessary feature; however, it has been considered rarely in the literature. Caimi et al. (2005), Caprara et al. (2010) and Dewilde et al. (2014) introduced robustness implicitly by rewarding extended buffer times between trains as a common way to improve the robustness. Herrmann (2006) introduced three other robustness measures for the TRP as in Zwaneveld et al. (1996) which are variations on maximizing (clique/edge/node) weights in the conflict graph. These measures resemble an explicit robustness measure based on stochastic disturbance distributions. Caimi et al. (2005) and Herrmann used fixed-point heuristics to solve the TRP. Note that Herrmann referred to these measures as stability measures. In our paper, we define robustness explicitly as reducing the cumulative train delays in the station area. This robustness notion has been used in Kroon et al. (2008) for improving timetable robustness, but has not been used in train routing problems so far.

According to Goverde and Hansen (2013), stability is regarded as an important design indicator. A common way to express stability, which we also adopt in this paper, is through infrastructure capacity occupation. Capacity occupation norms for station occupation are proposed by the International Railway Association UIC (UIC 2013) and as such should be respected in the planning processes. In the literature, these norms are also known as stability norms. For example, having a route plan with an infrastructure occupation rate of a platform track of 90% of the time can not be accepted as a good route plan. Capacity occupation in a periodic timetable can be defined by the minimum cycle time μ , which represents then a stability measure (Heidergott et al. 2005; Goverde 2007). The minimum cycle time is the minimal period length in which the periodic execution of arrival and departure events is feasible after compressing the events as much as possible. Therefore, we want to minimize the cycle time μ to achieve this. Let us assume a route plan with a scheduled cycle time T . The minimized μ provides $T - \mu$ time allowances (i.e., time supplements and buffer times) on the critical cycle and more everywhere else, i.e., on non-critical cycles. For more details on timetable stability evaluation, we refer to Goverde (2007, 2010), and on capacity assessment, we refer to Bešinović and Goverde (2018). Burkolter (2005) developed a Petri-net model that minimizes the

cycle time to find a stable timetable structure (e.g., a train sequence). The proposed model applies two heuristics, local search and simulated annealing. The feasibility is additionally checked on the microscopic level by the model of Herrmann (2006). Petering et al. (2015), Zhang and Nie (2016) and Sparing and Goverde (2017) solve a related problem to Burkolter (2005) for bigger networks. They developed variants of exact minimal cycle time optimization (MCTO) models to find an optimal timetable structure. Bešinović and Goverde (2016) proposed a two-stage stability-to-robustness model that integrates MCTO and further designs an actual stable and robust timetable by effectively distributing time allowances. So far, no implementations of using the minimum cycle time as a stability measure for route plans exists in the literature.

The use of individual infrastructure elements plays a considerable role in traffic operations, because the occupation of each element determines how frequent maintenance should be undertaken. Each resource, particularly movable elements like switches, needs regular maintenance and replacement due to their lifetime cycle. The lifetime depends on multiple factors that may be train-, track- or usage-related. Some of these relevant factors are trains' axle load, total tonnage, soil quality, maintenance and component renewal, train speed, number of operations (i.e., switch movements), and number of trains in facing or in trailing direction (Zwanenburg 2009). Closing a track section to maintain a switch often needs some traffic adjustments, which causes local changes in train routes, possible platform changes or even cancelling trains, and above all, evokes passenger inconvenience. Therefore, it is highly beneficial to spread the usage of infrastructure elements as equally as possible. The parameters defining the resource occupation can be the total blocking time and/or the number of resources used in a station. In our paper, we adopt the latter.

We introduce an extension to the original TRP and define it as the robust train routing problem (RTRP). Given a microscopic infrastructure and signalling system, macroscopic timetable (arrival and departure times), a set of alternative routes and train dynamics characteristics, determine a route plan including platform assignment that provides conflict-free, stable and robust operations with minimized and evenly distributed infrastructure use. To compute conflict-free solutions, we consider a much higher level of (infrastructure) details compared to e.g. Zwaneveld et al. (1996, 2001), which is already NP-hard. Together with additional objectives such as robustness, i.e. explicitly considering many disturbance scenarios, and an even infrastructure use, our considered problem requires many new variables, extra constraints, and a stochastic model formulation. In our opinion, this makes the newly created problem significantly more difficult to solve even for small instances. Therefore, we opted to apply a heuristic approach to find feasible, stable and robust route plan solutions.

In this paper, we propose a new microscopic multi-objective approach for the RTRP to improve train routing and platform assignment, which incorporates the important aspects of feasibility, stability, robustness, and balanced infrastructure occupation. We introduce the multi-start local search heuristic *RouteCare* to solve the RTRP. *RouteCare* is based on two models for capacity assessment and robustness evaluation. The former involves a max-plus automata model, and the latter integrates a delay propagation model. One of the main advantages of the max-plus

automata model is the ability to represent the infrastructure resources and blocking times of these resources corresponding to blocking time stairways. Other than the time-event graph max-plus algebra models (Heidergott et al. 2005; Goverde 2007), in the max-plus automata, both the start and end time of blocking each resource by each train are taken into account. In this way, conflicts between trains are naturally forbidden, and this enables to compute capacity occupation by a given route plan. Feasibility is guaranteed by explicitly modelling track use and considering all operational constraints of the infrastructure and signalling system. Stability is achieved by minimizing total infrastructure occupation. Robustness is enforced by reducing the expected train delays. An even resource use is encouraged by distributing trains over more resources. A case study of a typical station area in the Netherlands shows the potential of the RouteCare algorithm by improvements in the obtained route plans compared to the original route plans. The approach represents a valuable addition to decision support tools in planning processes. In particular, RouteCare may be used in the planning of basic route plans, and also for short-term adjustments due to scheduled infrastructure maintenance and construction works.

The main contributions of this paper are fourfold. First, we propose a new RTRP formulation that for the first time explicitly includes stability and uses the minimum cycle time as a stability measure for the RTRP. Second, the capacity assessment model provides a microscopically conflict-free solution while no additional constraints for modelling headway constraints are needed. Third, the proposed model considers a more even use of infrastructure resources which may lead to generating more acceptable route plans and reduce required maintenance works. Fourth, the model incorporates capacity and robustness assessment directly in the optimization process of the TRP. As a result, planners gain an extended knowledge about the created route plan as soon as it has been produced. This is an improvement to the current general practice, where a capacity and robustness assessment is undertaken a posteriori or not at all in the design steps.

The remainder of the paper is organized as follows. Section 2 gives the definitions and notation used in the paper. Section 3 presents the methodology for solving the TRP. In particular, first the necessary data preprocessing and adopted modelling assumptions are given in Sects. 3.1 and 3.2. Then, models for capacity assessment, robustness evaluation and a multi-objective notion of the RTRP are described in Sects. 3.3, 3.4 and 3.5. The heuristic RouteCare and the corresponding permutation rules are elaborated in Sects. 3.6 and 3.7. Section 4 gives results of computational experiments, which is followed by conclusions in Sect. 5.

2 Definitions and notation

The *microscopic infrastructure* topology graph $G = (X, S)$ represents the network at the level of homogeneous behavioral sections (i.e. sections with a constant value of speed limit, gradient and curvature). Each arc (in railway terms infrastructure section) in $S \subseteq X \times X$ is described with its attributes such as length and constant characteristics of speed limit, gradient, and radius. The microscopic nodes $x \in X$ represent both points in which these characteristics change and infrastructure element changes

like boundaries of blocks, switches, and platform track sections. Next to the infrastructure, we denote the set of *trains* by Z and typically denote a train by $z \in Z$.

A *resource* represents a subset of infrastructure elements that can be exclusively allocated to a single train at a given time. In practice, this is a block section or an interlocking section such as switch, crossing, single or double slip. The set of resources is denoted by R , with a typical element $r \in R$. We also define the subset of resources that belong to platform tracks (tracks where a train stops) as R_{platform} .

An *itinerary* represents a sequence of resources r that can be used by a train to traverse through a station and it is denoted as j . The set of all itineraries in a station is denoted J . An itinerary may define an inbound movement, i.e. from the station entry to the platform track, outbound movement, i.e. from the platform track to exit of the station, and through movement from entry to exit of the station. Typically, when a train is stopping, its itinerary is split into inbound and outbound.

A *train route* defines an assignment of an itinerary to a train. A train route p_{zj} is an itinerary $j \in J$ assigned to train $z \in Z$. A train route p_{zj} is described by a subset of used resources p_{zjr} , where r is a resource at the route. We define the set of all possible routes for a train z as P_z . Further, the set of all train routes over all trains is denoted as P . Each train route blocks resources for a limited time that can be computed using the blocking time theory (Hansen and Pachl 2014). In addition, a *train line* is a train service that is defined with train type, stopping pattern, frequency and train routes through stations.

A *route plan* consists of assigned routes for all trains including inbound and outbound routes and selected platform tracks within a station and is denoted by \bar{P} . In addition, the set of used resources in \bar{P} is established as $\bar{R} = \bigcup_{p_{zj} \in \bar{P}} p_{zjr}$, where $j \in J$, $r \in p_{zj}$, $\bar{R} \subseteq R$ and $\bar{P} \subseteq P$.

The *blocking time* of a resource is the time during which the resource is solely dedicated to a single train and cannot be used by any other. A blocking time consists of an approach, running and clearing time, corresponding to a train running from the approach signal to the block signal, traversing the block itself, and running out of the block until the train has completely left the block over its entire train length (Hansen and Pachl 2014). In addition, the blocking time includes setup and release times, as well as the driver sight and reaction time. The blocking time of a resource r on a route p_{zj} is denoted as b_{zjr} and is defined with its starting and ending time, $b_{zjr} = (s_{zjr}, f_{zjr})$, for all $z \in Z$, $j \in J$, and $r \in j$. A *blocking time stairway* of a train route is computed for a given train-dynamic speed profile based on the scheduled running time on the (macroscopic) network level. For this purpose, we use the microscopic timetabling models developed in Bešinović et al. (2017). A blocking time stairway for a train route p_{zj} is defined as $B_{zj} = (s_{zj}, f_{zj})$, where s_{zj} and f_{zj} are vectors of start and end times of all resources r over the train route p_{zj} .

Note that in our approach it is not fundamental that the timetable is periodic. However, the timetables in the Netherlands are planned as such, so we assume throughout the paper that all route plans are also periodic with a *scheduled cycle time* T .

Capacity occupation of a station is defined as the share of time that the train routes from a given route plan occupy the infrastructure in a given time period.

Capacity occupation may be determined as a summation of critical blocking times in the station network. A critical blocking time is one that affects the value of the capacity occupation if its size changes. These critical blocking times together form a so-called *critical cycle*. Resources that are included in the critical cycle are referred to as critical resources, and the minimum cycle time of a route plan is the weight of the critical cycle, i.e., the sum of its arc weights.

Platform occupation is the total blocking time allocated to trains stopping at a certain platform. The occupation of a single resource r , such as a platform track, is denoted as B_r and computed as the sum of the blocking times over this resource by all routes in a route plan.

Table 1 summarises the notation used throughout the paper, while a detailed introduction of each is given in the following sections.

3 Modeling the RTRP

3.1 Preprocessing

The preprocessing consists of the following procedures which are briefly explained.

1. Import all possible route alternatives.
2. Reduce the set of alternative routes.
3. Compute the blocking time stairways for each train route.
4. Derive the pairs of conflicting train routes.

In Procedure 1, the input for each train line is a set of train routes that are implemented in the interlocking system which represent all possible routing alternatives. In theory, we may incorporate all existing routes that connect a station entry/exit with all station platform tracks, but in practice the set of preferable platforms is commonly reduced to a smaller number of platforms depending on the station layout and previous planning decisions.

For a pair of origin and destination nodes there may exist an ample number of defined alternative routes. Therefore, during experimental testing, route reduction rules (Procedure 2) may be implemented to reduce computation time, as in Zwaneveld et al. (1996) and Dewilde et al. (2014). We tested the reduction carried out in Zwaneveld et al. (1996).

In Procedure 3, we use a microscopic model previously developed in Bešinović et al. (2017) for computing blocking time stairways for all possible train routes using the detailed infrastructure defined by the graph G . First, running times are computed for scheduled arrival and departure times from the timetable in a way that time supplements are equally distributed along the train run. Then, blocking time stairways are computed for all train routes in the station.

Procedure 4 defines a set of conflicting train routes, which is done by finding conflicts between two train routes. A conflict is determined as an overlap (in time and space) between blocking times of two routes. For example, let us consider train

Table 1 Notation of the RTRP

Notation	Description
X	Set of infrastructure nodes
S	Set of infrastructure arcs
$G = (X, S)$	Infrastructure graph
Z	Set of trains
R	Set of resources
$R_{platform}$	Set of resources belonging to platform tracks
P	Set of all alternative train routes
\bar{P}	Set of selected train routes in a route plan
\bar{R}	Set of used resources in \bar{P}
z	Trains
j	Itineraries
r	Resources
p_{zj}	Train route of a train z over itinerary j
r_{zjk}	Resource k of a p_{zj}
s_{zj}, f_{zj}	Vectors of start and end times of a blocking time stairway for train z on itinerary j
B_{zj}	Blocking time stairway for train z on itinerary j (tuple of two vectors)
s_{zjr}, f_{zjr}	Start and end times for train z on itinerary j at resource r
B_r	Total blocking time over all routes using resource r
C	Set of conflicting train route pairs
K	Set of critical resources from the critical cycle
T	Scheduled cycle time
H	Time horizon for the delay propagation model
V	Set of events in station areas, e.g., entry, exit, arrival, departure events
A	Set of activities, e.g., running, dwelling times, headways
$EAN = (V, A)$	Event-activity network
l_{ij}	Lower bound for process (i, j)
Q	Number of replications
q	Replication
h	Time
$D_i(h)$	Observed delay at h -th period of the i -th event
k_{ij}	Modulo parameter
$\delta^{(q)}$	One disturbance replication (scenario)
α, β, γ	Weight factors of the objective function
N	Number of trains in route plan \bar{P}

routes p_{zj} and p_{qm} , where z and q are trains in Z and j and m are corresponding itineraries in J , respectively. Also, assume that train z precedes train q within the station. The corresponding blocking time stairways are B_{zj} and B_{qm} . Let r_c denote a conflicting resource. Then, the conflict between two train routes are identified as:

$$c_{zj,qm} = \{(p_{zj}, p_{qm}) | \exists r_c \in p_{zj} \cup p_{qm}, s_{zjr_c} < s_{qmr_c}, f_{zjr_c} > s_{qmr_c}\}, \forall z, q \in Z, j, m \in J.$$

If $c_{zj,qm} = \emptyset$, then there is no conflict between routes p_{zj} and p_{qm} . The set of conflicting train route pairs C is defined as:

$$C = \bigcup_{z,j \in Z, m \in J} c_{zj,qm}.$$

3.2 Assumptions

We make three assumptions within the RTRP formulation. First, we assume that an initial route plan is given and feasible for all $z \in Z$. The input to the RTRP is a route plan \bar{P} which is the output from a previous planning step, the timetable design. By using a timetabling approach developed in Goverde et al. (2016), our generated timetable is microscopically conflict-free, and so is our initial \bar{P} . Second, we assume that the weight of a train route equals the running time through the station. Third, we do not consider passenger connections in stations. To solve the given multi-objective RTRP, we introduce a heuristic algorithm in the following sections. In particular, Section 3.3 describes a max-plus automata model to compute the capacity occupation, Sect. 3.4 gives a delay propagation model to evaluate robustness, Sect. 3.5 presents the objective function and Sect. 3.7 combines all previous in a heuristic algorithm.

3.3 Capacity assessment: max-plus automata model

We propose a max-plus automata model for assessing the capacity occupation and refer to it as the capacity assessment (CA) submodel. The output of the CA submodel consists of 1) the capacity occupation μ , 2) the set of critical resources K , i.e., the resources that determine the critical cycle, 3) the total blocking times of single resources, B_r , for all resources $r \in \bar{R}$, 4) the platform occupation, B_r , for all platform resources $r \in R_{platform}$, and 5) the number of resources θ used by the route plan. All these output characteristics of a route plan are exploited afterwards to modify the set of chosen routes. In essence, a max-plus automata model considers blocking time stairways directly by using start and finish times of each block in the stairway. Also, the model does not allow overlap between two stairways. Thus, minimum headways between trains are modelled explicitly and the output is conflict-free. Graphically, the max-plus automata model resembles a tetris game where a falling block represents a blocking time stairway.

Gaubert and Mairesse (1999) introduced max-plus automata which combine elements of the theory of heaps-of-pieces and max-plus algebra. An initial application of max-plus automata in railways is given in Van Egmond (1999). Max-plus algebra is an idempotent semiring over the union of real numbers and $\varepsilon = -\infty$, $\mathbb{R}_{\max} = \mathbb{R} \cup \{\varepsilon\}$, and is equipped with the two binary operations maximum (\oplus) and addition (\otimes). Let a and b be real scalars or ε , then the operations are defined as

$$a \oplus b = \max(a, b) \quad \text{and} \quad a \otimes b = a + b.$$

The element $\varepsilon = -\infty$ is the neutral element for \oplus and absorbing for \otimes . The element $e = 0$ is the neutral element for \otimes . Properties of max-plus algebra are similar

to conventional algebra. We refer to Goverde (2007) and Heidergott et al. (2005) for more details on max-plus algebra with applications to railways.

A max-plus automaton is a triple (\bar{P}, \bar{R}, M) . Here, \bar{P} is a finite set of tasks representing a route plan. A task $p \in \bar{P}$ represents a train route p_{zj} . In the remainder of the section, we omit indices to keep the notation clearer. \bar{R} is a finite set of resources (as defined in Sect. 3.1). Lastly, M is a linear transformation $\bar{P} \rightarrow \mathbb{R}_{\max}^{\bar{R} \times \bar{R}}$, which is uniquely specified by a finite family of matrices $M(p)$, $p \in \bar{P}$, as defined in the next paragraph.

Vectors $s(p)$ and $f(p)$ represent the start and end times of the blocking time stairway s of train route p , respectively; and are $|\bar{R}|$ -dimensional row vectors with an element equal to ϵ if a resource is not used. Matrix $M(p)$ represents a blocking time stairways of train route p in max-plus automata notation, and simultaneously equals the capacity occupation of p . Matrix $M(p)$ is computed as:

$$M_{ij}(p) = \begin{cases} e, & i = j, i \notin \bar{R}(p), \\ f_j(p) - s_i(p), & i, j \in \bar{R}(p), \\ \epsilon, & \text{otherwise.} \end{cases} \quad (1)$$

Route plan \bar{P} is represented as $\bar{P} = \{p_1, \dots, p_n\}$, where n is the number of routes in \bar{P} . Therefore,

$$M(\bar{P}) = M(p_1 \cdots p_n) = M(p_1) \otimes \cdots \otimes M(p_n),$$

where $p_1 \cdots p_n$ is concatenation of train routes in \bar{P} and $M(\bar{P})$ is computed sequentially by adding one route at a time.

We introduce the upper contour of the route plan \bar{P} , $x(\bar{P})$, which is computed as

$$x(\bar{P}) = M(\bar{P}) \otimes x(e), \quad (2)$$

where $x(e)$ is an $|\bar{R}|$ -dimensional row vector with all entries equal to ϵ .

In order to determine the earliest possible start of the next cycle, we add the route of the first train in the cycle p_1 to the route plan as $\bar{P}p_1$. The difference $f(p_1) - s(p_1)$ represents a vector of blocking times of p_1 (from the next period) over all resources. The capacity occupation $\mu(\bar{P})$ is computed as

$$\mu(\bar{P}) = \min_{r \in \bar{R}} [x(\bar{P}p_1) - (f(p_1) - s(p_1))]_r, \quad (3)$$

where schedule $\bar{P}p_1$ is a schedule for one time period including the first train service p_1 that belongs to the next period and the minimum is taken over all resources $r \in \bar{R}$, which correspond to the vector entries.

The set of critical resources K is constructed step-wise, after adding each train route p_z to the heap of previously added routes $M(p_1 \cdots p_{z-1})$. A resource of p_z that satisfies

$$f(p_z) - s(p_z) = x(p_1 \cdots p_z) - x(p_1 \cdots p_{z-1})$$

is considered as a critical resource and is added to K . In other words, if the difference between upper contours after adding route p_z is exactly the blocking time of that route $f(p_z) - s(p_z)$ on some resources, then such a resource is a critical resource. Once all train routes from \bar{P} are included in M , all critical resources are known.

The total blocking time of each resource r is computed as:

$$B_r = \sum_{z \in Z, j \in J} (f(p_{zjr}) - s(p_{zjr})), \quad \forall r \in \bar{R}.$$

Blocking times of platform tracks $r \in R_{platform}$ are contained in \bar{R} , so we do not explicitly recompute them. In addition, the number of unique resources used θ for a given route plan is obtained as $\theta = |\bar{R}|$. Note that the model complexity, the size of the M matrices, depends on the route choices and number of resources used \bar{R} and not on the station layout complexity.

Algorithm 1 summarizes the capacity occupation model. Given is a timetable and the corresponding route plan \bar{P} that includes all selected train routes $p_z \in \bar{P} \subset P$. First, all individual matrices M are computed for each train route p in \bar{P} . Next, matrix $M(\bar{P}p_1)$ is created by adding one route at the time. While doing so, the critical resource r is determined in each iteration and assigned to the set of critical resources K . Then, the CA submodel computes the capacity occupation for a station by using (2) and (3). Finally, the output consists of the capacity occupation μ , the critical resources K , total number of resources used θ , and the resource occupation B_r for all $r \in \bar{R}$. The output of Algorithm 1 depends on the order trains are scheduled in the timetable and thus its routes added to the heap. Note that in case that two trains arrive/depart simultaneously to/from the station (i.e., have rolling stock or passenger connection, but not infrastructural) they can be modelled as one piece to maintain their dependency (see Bešinović and Goverde 2018 for more details).

Input: timetable, route plan \bar{P}
Output: $\mu(\bar{P})$, K , B_r for all $r \in R_{platform}$, θ
 generate $M(p_z)$ for all $p_z \in \bar{P}$
for $z = 2$ to $|Z| + 1$
 compute $M(p_1 \dots p_z) := M(p_1 \dots p_{z-1}) \otimes M(p_z)$
 determine critical resource r and set $K := K \cup \{r\}$
endfor
 compute $x(\bar{P}p_1)$
 compute $\mu(\bar{P})$
 compute B_r for all $r \in \bar{R}$
 compute $\theta(\bar{P})$

Algorithm 1 Pseudo code of the CA submodel

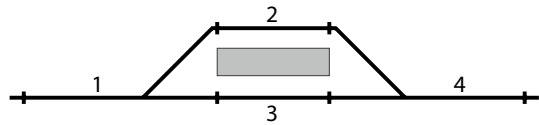
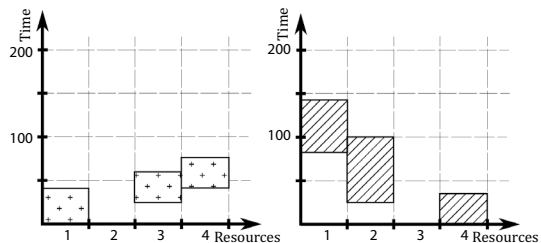
For better understanding of the max-plus automata model, we give a numerical example of two train routes a and b running through a simple station (Fig. 1). Routes a and b make a route plan $\bar{P} = ab$. Train route a uses resources 1, 3 and 4, while b uses 4, 2 and 1. The set of used resources is $\bar{R} = \{1, 2, 3, 4\}$. Their blocking times are defined with vectors of starting and ending times in Table 2.

Figure 2 visualises the capacity occupation of train routes a and b . Using (1), matrices M for each train route are computed:

$$M(a) = \begin{bmatrix} 40 & \epsilon & 60 & 75 \\ \epsilon & e & \epsilon & \epsilon \\ 15 & \epsilon & 35 & 50 \\ 0 & \epsilon & 20 & 55 \end{bmatrix}, \quad M(b) = \begin{bmatrix} 60 & 20 & \epsilon & -45 \\ 115 & 75 & \epsilon & 10 \\ \epsilon & \epsilon & e & \epsilon \\ 140 & 100 & \epsilon & 35 \end{bmatrix}.$$

Table 2 Blocking times for train routes a and b

Route r	$s(r)$ (s)	$f(r)$ (s)
a	$[0, \epsilon, 25, 40]$	$[40, \epsilon, 60, 75]$
b	$[80, 25, \epsilon, 0]$	$[140, 100, \epsilon, 35]$

Fig. 1 Example station layout**Fig. 2** Resource occupation of train routes a and b 

The matrix M for trains ab is computed as

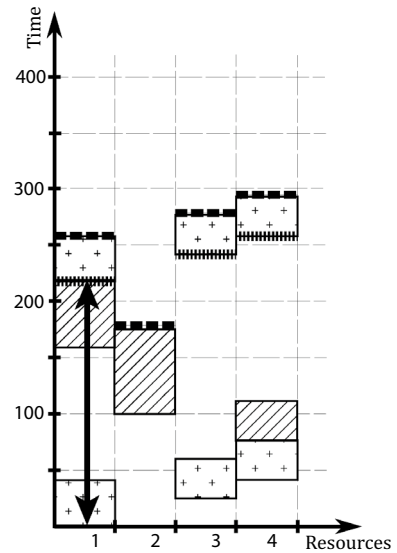
$$M(ab) = M(a) \otimes M(b) = \begin{bmatrix} 215 & 175 & 60 & 110 \\ 115 & 75 & \epsilon & 10 \\ 190 & 150 & 35 & 85 \\ 175 & 135 & 20 & 70 \end{bmatrix}.$$

The upper contour of the route plan ab is then computed as $x(\bar{P}a) = x(aba) = M(aba) \otimes x(e) = (255, 175, 315, 330)^T$. Here a is added to represent the first train from the following period. The capacity occupation is computed using (3) as

$$\mu(\bar{P}) = \mu(ab) = \min_r [x(aba) - (f(a) - s(a))]_r = \min_r \left(\begin{bmatrix} 255 \\ 175 \\ 315 \\ 330 \end{bmatrix} - \begin{bmatrix} 40 \\ \epsilon \\ 35 \\ 35 \end{bmatrix} \right)_r = 215s,$$

where the minimum is taken over the vector entries. Figure 3 shows the final result of the capacity assessment of route plan \bar{P} . Dashed lines give the upper contour which equals $x(\bar{P}a) = (255, 175, 315, 330)^T$. The double arrow depicts the capacity occupation $\mu(\bar{P})$ which equals 215 s. There is one critical resource since all routes touch at resource 1, $K = \{1\}$. The total number of resources used is $\theta = 4$ and platform occupation is 75 s and 35 s for each platform (resources 2 and 3), respectively.

Fig. 3 Capacity occupation of ab . Dashed lines give the upper contour $x(\bar{P}a)$. The double arrow depicts the capacity occupation $\mu(\bar{P})$



3.4 Robustness evaluation: delay propagation model

We implement a simulation-based model as robustness evaluation and refer to it as the RE submodel. The RE submodel simulates train operations in a station area under initial disturbances. Disturbances are defined as small deviations from the planned process times and are assumed to be known and given as input.

We consider a horizon of H hours to capture the propagation over a longer period of time. In transport networks with dense traffic, it may commonly happen that a train from 1 h causes a delay to a following train in a next period. We assume that all train orders are fixed as in the timetable, so no traffic control measures within the station area are taken into account. By doing so, we concentrate on the timetable itself.

We deploy a macroscopic model of a station area with several events for each train: entry to the station, arrival and departure to/from a platform and exit from the station. Also, all dependencies of train routes are maintained and mapped from the microscopic network used in the CA submodel to the macroscopic network. This level of abstraction allows to develop a fast RE submodel capable of performing a number of Monte Carlo simulations. We create an event-activity network $EAN = (V, A)$, where vertices correspond to events $i \in V$ with corresponding event time v_i , i.e., given scheduled arrival, departure or pass-through times. Arcs $a \in A$ represent process times such as running, dwell, transfer of headway times. Each arc is described with a pair of events $a = (i, j)$. For modeling the RE submodel, we use precedence constraints

$$v_j - v_i \geq l_{ij}. \quad (4)$$

where l_{ij} is the minimum process time over arc (i, j) , computed using models from Bešinović et al. (2017). So, (4) gives a lower bound on the process times between two adjacent events.

The robustness evaluation model is based on a Sample Average Approximation Method, see Shapiro (2000). We assume that the primary disturbances are described by a non-negative random vector variable Δ that assigns non-negative disturbance values to each activity. Let δ be a replication from Δ and $D(v, \delta)$ is defined as the average weighted delay when route plan v is performed subject to disturbances in δ . Thus, we want to find minimum D^* as

$$D^* = \text{minimize}\{ED(v, \Delta)\}.$$

To solve D^* , we approximate it by (1) selecting a fixed number of realizations $|Q|$, and (2) making a random sample of Q vectors $\delta^{(q)}$ of primary disturbances. Thus, we compute the average cumulative delay D_Q^* as:

$$D_Q^* = \text{minimize} \left\{ \frac{1}{|Q|} \sum_{q=1}^Q D(v, \delta^{(q)}) \right\}.$$

The robustness evaluation model is realized as an optimization model that minimizes the average cumulative delay of the route plan \bar{P} , while $D_i(h)$ is a single delay of one event i in the h -th hour. The RE submodel is formulated as follows:

$$D_Q^* = \text{minimize} \left\{ \frac{1}{|Q|} \sum_{q=1}^Q D(v, \delta^{(q)}) \right\} \quad (5)$$

subject to

$$v_j^{(q)}(h + k_{ij}) - v_i^{(q)}(h) \geq l_{ij} + \delta_{ij}^{(q)}(h), \quad \forall (i, j) \in A, \forall h \in H, \quad (6)$$

$$v_i^{(q)}(h) \geq v_i + hT, \quad \forall i \in V, \forall h \in H, \quad (7)$$

$$v_i^{(q)}(h) - v_i - hT \leq D_i(h), \quad \forall i \in V, \forall h \in H \quad (8)$$

Here, variable $v_i^{(q)}(h)$ defines the simulated event time of event i in hour h with respect to the input disturbances and k_{ij} defines the hour shift of events i and j . For example, if i is scheduled before j then $k_{ij} = 0$, otherwise $k_{ij} = 1$. Constraint (6) assures that precedence constraints are satisfied for the minimum process times and a given disturbance $\delta_{ij}^{(q)}(h)$ in the h -th hour from event i to j . Constraint (7) guarantees that a train does not depart earlier than scheduled. Note that the scheduled event time of periodic event i after h hours is $v_i + h \cdot T$ with $T = 60$ min. Finally, (8) defines all delay variables $D_i(h)$ as non-negative.

The RE submodel tests a given tentative route plan \bar{P} against a set of replications $q \in Q$, and returns the robustness measure $\bar{D}(\bar{P}) = D_Q^*$. From the RE formulation

can be seen that maximizing robustness equals minimizing the average cumulative delay, which we use as a robustness measure in Sect. 4. Also, we create an ordered list of the train routes that are affected the most, i.e., received the most delays, by summing the delays for each train line separately.

3.5 Objective functions for the RTRP

In order to address the posed questions, we define multiple objective functions.

As the first objective, a route plan with small capacity occupation allows more buffer times to be spread between trains. In other words, such a route plan tends to be more robust. We choose to minimize cycle time μ to get the maximal achievable time allowance.

As the second objective, we introduce the robustness of a route plan and it is based on the delay propagation. In particular, the goal is to separate train routes in time as much as possible to minimize possible train delays. Train event times are considered as fixed as they are the result of a previous planning step—timetabling on a network level. Thus, in our model, time supplements are given and fixed, while buffer times between routes depend essentially on the route choice that has been made. Choosing certain routes, the knock-on effects of delays from one train to another can be significantly reduced or even avoided if two routes are independent. Thus, we want to find a route plan that minimizes the average train delay \bar{D} .

The third objective considers balancing the use of single infrastructure resources. Since our focus is on a dominant passenger network, we can assume that all trains are of a similar tonnage, and thus, we may translate the expected lifetime of a resource to the number of train runs over it. In this way, a reduced number of scheduled routes over a single resource would extend its lifetime and reduce the maintenance frequency and the required (partial) station closures for these works. To achieve this, we maximize the number of θ resources used in a station area. As a consequence, some used routes may be longer and/or slower; however, such route plan can significantly reduce the need for maintenance closures. Note that we also choose not to use efficiency in the objective function. Having another objective criterion makes it more difficult to balance and weigh all criteria correctly, and it makes the results also more difficult to interpret. This is acceptable for the TRP as running times do not differ much due to already reduced train speeds in a station area. Still, route efficiency is encouraged within permutation rules by giving priority to faster routes (see Sect. 3.7). Most importantly, a possible slight increase in running times will always be outweighed by the reduced need of maintenance closures of single resources.

We summarize the multi-objective problem for solving the RTRP as

$$\text{cost}(\bar{P}) = \min_P \{ \alpha\mu + \beta\bar{D} + \gamma\theta \}, \quad (9)$$

where α , β and γ are weight factors of three objectives to normalize the costs of single objectives.

First, the stability measure is of a deterministic nature, while the robustness measure is stochastic. Second, the former considers only the critical cycle, while the latter focuses on the buffer times, i.e., the delay reduction potential of the buffer times, also between all train lines of the route plan. When optimizing only on stability, i.e., over the critical cycle, good robustness over the complete route plan is not necessarily achieved. Reversely, if only robustness is considered, stability may suffer by high capacity occupation of a certain platform (resource). Thus, we need both measures to find the best overall solutions. The importance of a multi-objective approach is quantified in Sect. 4.

3.6 RTRP heuristic

We propose the heuristic *RouteCare* for solving the RTRP. *RouteCare* is a greedy multi-start local search heuristic that is based on the capacity assessment model and the delay propagation model. *RouteCare* returns the best found robust route plan *RRP*. Algorithm 2 gives an outline of *RouteCare*.

The initialization phase consists of assigning the input route plan \bar{P} as the best current *RRP* and its cost is the cost of \bar{P} , $c(RRP) := c(\bar{P})$. *RouteCare* consists of several steps that are repeated iteratively until it performs an N_{iter} number of iterations. The algorithm may also terminate before reaching N_{iter} if stagnation of the algorithm is observed. We assume that the algorithm stagnates when the total cost of a solution does not change for a given number of iterations *stagMax*. In order to prevent premature stagnation and move away from a possible local minimum, we execute a re-start when a solution is unchanged for a *randMax* number of iterations. To this end, we define a counter *stag* that tracks the number of iterations without solution improvement. If *stag* = *randMax*, then the route plan \bar{P} is randomized.

At each iteration, we receive a route plan \bar{P} with the evaluated corresponding cost $c(\bar{P})$. First, a tentative route plan \bar{P} is assessed on infrastructure occupation $\mu(\bar{P})$ using the max-plus automata model from Sect. 3.3. Then, the algorithm evaluates the robustness of the proposed route plan \bar{P} by using the delay propagation model from Sect. 3.4 and checks the number of used resources $\theta(\bar{P})$. Consecutively, the total cost $c(\bar{P})$ is obtained. If a new \bar{P} has a lower cost than *RRP* then \bar{P} becomes the new *RRP*.

Finally, we permute \bar{P} by selecting N candidate routes to be exchanged based on the rules defined in Sect. 3.7. One can choose a value for N between 1 and $|Z|$. Note that if $N = |Z|$ then the new \bar{P} represents a completely new set of routes. An alternative route plan is evaluated again in the new iteration.

3.7 Route permutations

In this section, we define rules to remove unfavorable routes from a current \bar{P} and substitute them with alternative ones for a given train line. Therefore, we define a set of rules for excluding a route from \bar{P} and likewise, a set of rules for selecting an alternative route to be inserted in the \bar{P} . The exclusion rules are based on the output of the capacity assessment and robustness evaluation. The inclusion rules

Input: timetable, initial route plan \bar{P} , N_{iter} , $stagMax$, $randMax$, α, β, γ
Output: robust route plan RRP
Initialize $RRP := \bar{P}$; $iter := 1$; $stag := 0$
while $iter \leq N_{iter}$ OR $stag = stagMax$
 Compute capacity occupation $\mu(\bar{P})$, critical resources K , number of resources $\theta(\bar{P})$ (Section 3.3)
 Evaluate robustness to determine the robustness cost $\bar{D}(\bar{P})$ (Section 3.4)
 Compute cost of \bar{P} : $c(\bar{P}) := \alpha\mu(\bar{P}) + \beta\bar{D}(\bar{P}) + \gamma\theta(\bar{P})$
 if $c(\bar{P}) < c(RRP)$
 Update $RRP := \bar{P}$ and $c(RRP) := c(\bar{P})$
 Update $stag := 0$
 else
 Update $stag := stag + 1$
 end if
 if $stag = randMax$
 randomize \bar{P}
 end if
 Permute N routes in \bar{P} (Section 3.7)
 Update $iter := iter + 1$
end while

Algorithm 2 Pseudo code of RouteCare algorithm

are derived from the corresponding exclusion ones together with the constraints that provide a realizable route and satisfy the conflict-freeness of chosen routes.

We first give a set of exclusion (E) rules.

Rule E1. Exclude a train route with one or more resources on the critical cycle.

A route that is part of a critical cycle is a good candidate for exclusion. Braking a chain of critical events may lead to a change in capacity occupation. If a new route is chosen wisely, this rate may be reduced.

Rule E2. Exclude a route that participates in the critical cycle with the most critical resource, i.e., the one with the highest capacity occupation.

This rule relaxes the utilization of the resource with the highest capacity occupation. As such, it provides a diversification in used resources and thus leads to a more dispersed utilization of station resources.

Rule E3. Exclude a route that uses a platform track with the highest occupation.

Resources that correspond to platform tracks may not necessarily be in the critical cycle. However, having more train lines scheduled to a single platform sets more dependencies between them, and in general leads to less resistance to possible delays. Therefore, we would like to use station platform tracks in a more even manner and to achieve this by distributing trains over all available platform tracks.

Rule E4. Exclude a route that received most delays in the robustness evaluation. In the RE submodel, an ordered list of routes with total delays is generated, so we take the most sensitive route and seek for its alternative. Since one of the objectives in the model is reducing delays, excluding the least robust route may improve the route plan.

Rule E5. If more than one route exists that satisfies a certain exclusion rule, then exclude the route among the possible alternatives with a probability proportional to its weight, i.e., a route with lower weight has a higher probability to be selected.

Note that E-rules may be mutually exclusive. Therefore, during the experimentation phase, in each iteration one E-rule is chosen randomly with all rules treated equally. The weight of a route is the running time over it.

Inclusion (I) rules are given as follows. First, we define rules that naturally follow exclusion rules E1 to E3, thus representing their counterparts. All I-rules correspond to a train considered for exclusion by E-rules. In other words, I-rules are applied on the set of routes of the train z , that has been predetermined by an E-rule.

Rule I1. Include a route from P_z that does not use a resource in the critical cycle.

Rule I2. Include a route from P_z that does not use any critical resource.

Rule I3. Include a route from P_z that does not use the highest utilized platform track.

Rule I4. Include a route from P_z that ensures a conflict-free run considering all other routes in the route plan, i.e., select a route that satisfies

$$(p_{zj}, p_{qm}) \notin C, q \in Z, z \neq q,$$

over all other train routes q in the route plan \bar{P} .

Since one of the objectives is to deliver a conflict-free route plan, this rule opts only for an alternative route that does not create a conflict with other routes in \bar{P} .

Rule I5. If more than one route exists that satisfies a certain insertion rule, then include a route among the possible alternatives with a probability proportional to its weight, i.e., the route with the lowest weight has a higher probability to be chosen.

In terms of the nature of the defined rules and their relation to the objectives, we may say that rule pairs E1–I1, E2–I2 and E3–I3 relate to the capacity occupation constraints and the critical cycle time. Recall that due to the modelling nature of max-plus automata in the CA submodel, it is not needed to model headways explicitly. Furthermore, E4 encourages the robustness of a route plan. The pair of rules E3–I3 encourages more evenly distributed usage of resources, which is particularly valuable for the maintenance planning. Finally, rule I4 maintains the conflict-freeness of a given solution. Therefore, we may conclude that by using all defined rules, we obtain a RTRP model, and if we exclude E4 then we get the relaxation to the original TRP model.

4 Computational results

We tested RouteCare on a medium-sized station in the Netherlands, Den Bosch. The outline of the station is given in Fig. 4. The station includes eight station tracks of which five are equipped with a platform to provide boarding and alighting of passengers. The remaining station tracks are used for trains passing through the station without stopping. As a common practice, station tracks are divided in groups per direction of traffic, that is, two platform tracks for trains running in one direction and the remaining three for the opposite one. In the basic scenario, 12 trains are serving the station every 30 min. Therefore, a cycle time of $T = 30$ min was considered. For the robustness evaluation, we simulated one day of service, $H = 32$ (between 6 a.m.–22 p.m.). In the RE submodel, $|Q| = 30$ replications are performed. The delay distribution over all process times is assumed to be exponential. For the

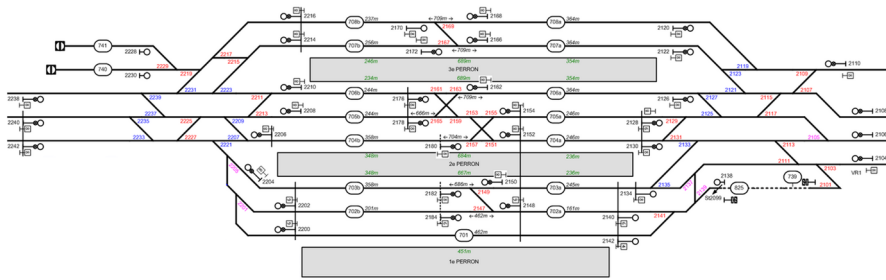


Fig. 4 Station Ht layout (source: sporenplan.nl)

parameter of the exponential distribution, we adopted 5% of the process times (Yuan and Hansen, 2007). The parameters of RouteCare are defined after initial empirical experiments on the algorithm performance of finding better route plan solutions. In particular, $N = 1$, $randMax = 20$, $N_{iter} = 500$ and $stagMax = 40$. The parameters of the objective function (9) are defined based on the relative importance of each objective. The importance of the three objectives is defined by the lexicographic order: stability, robustness, and even infrastructure use. The first two objectives, capacity occupation and robustness are of the same nature, both expressed in seconds, while capacity occupation is typically of a bigger order of magnitude for the experiments performed. Therefore, α and β are both adopted to be 1. The third objective, the number of resources, is expressed in non-dimensional units, and typically takes lower values than both the capacity occupation and robustness, thus we adopted also $\gamma = 1$. We generated 40 alternative routes for all trains, thus there was no need for applying Procedure 2 in the preprocessing stage. We used an Intel E7 with 2.6 GHz processor for testing purposes.

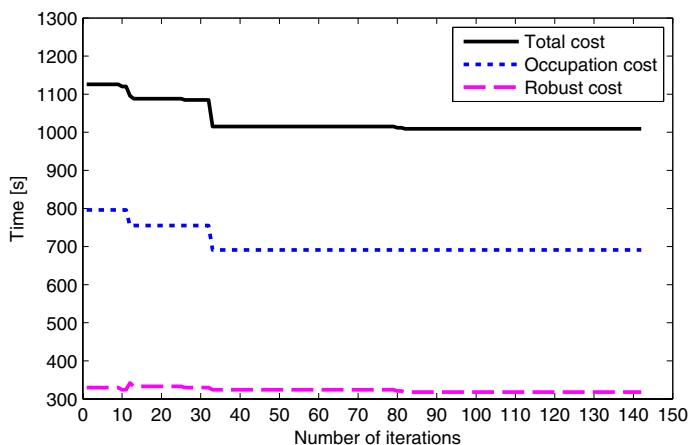
The performance of RouteCare in terms of solution quality and algorithm stability was tested first with the results given in Sect. 4.1. In addition, we compared the two submodels CA and RE to evaluate their effects individually to the obtained solutions in Sect. 4.2. Initially, RouteCare was run with only the CA submodel active, while RE was evaluated a posteriori to determine the robustness of the solution. In the second run, RouteCare used only the RE submodel, while CA was performed a posteriori. By doing this, we could quantify the effect of considering only one objective function on the quality of the resulting route plans. After determining the benefits of the model, we undertook extensive computational analyses for different scenarios whose results are reported in Sect. 4.3.

4.1 Performance of the RouteCare heuristic

In this section, we give results on the stability of the RouteCare algorithm developed for solving the RTRP. To produce the initial macroscopic timetable, we used the timetabling model developed by Bešinović et al. (2016). Then we applied RouteCare to compute the route plan for the station. We reran the algorithm 30 times to evaluate

Table 3 Results of the RTRP heuristic

	Mean	Standard deviation
Total cost	1089	7.95 (0.73%)
Number of iterations	125	39.46 (31.57%)

**Fig. 5** Performance of RouteCare

its algorithmic stability. Table 3 shows the statistics of the obtained results. Particularly, it gives the average and the standard deviation of the total cost and number of iterations needed to obtain a solution. We also compared the solution of RouteCare with the original route plan as considered in the timetabling denoted as *OriginalRP*. To that purpose, *OriginalRP* was evaluated using the objectives in RouteCare—capacity occupation, average cumulative delays and number of resources. It can be observed from Table 3 that RouteCare performed good throughout successive simulations, as it always found an improved solution.

Although it did not always end up to a single value, the obtained total cost has a limited standard deviation of only 0.73%. The average number of necessary iterations before reaching a solution was 125, while in some simulation runs the solution has been found after 100 iterations. In all cases, the model needed considerably less than N_{iter} iterations. Therefore, we can say that the average performance of RouteCare was satisfactory for the given test case.

Figure 5 shows the improvement of the total cost (solid line) for one run of RouteCare with both costs for the CA (dotted line) and RE (dashed line) submodels. It also gives intermediate costs of capacity occupation and robustness. It can be observed that in the first ten iterations the capacity occupation was relatively high, but noticeable improvements were obtained in the beginning, due to decreasing the total delay of a route plan. Around iteration $iter = 35$, we see another significant reduction in capacity occupation. At $iter = 80$, we observe a small robustness improvement, which was the last change in the solution cost. The model continued iterations until it reached one of the stopping criteria *stagMax*, so the final solution was obtained

Table 4 Rules applied in submodels of RouteCare

	E1–I1	E2–I2	E3–I3	E4	I4	E5–I5
CA submodel	+	+	+	–	+	+
RE submodel	–	–	–	+	+	+
CA and RE	+	+	+	+	+	+

Table 5 Results of individual submodels for $|Z| = 12$ trains

RouteCare	Capacity occupation $\mu(RRP)$	Cumulative delay $\bar{D}(RRP)$	No. of resources θ	Total cost $c(RRP)$
Only CA	609	531	70	1210
Only RE	803	318	66	1187
CA and RE	690	333	66	1089

Bold values represent the best results

after 142 iterations. The total cost was $c(RRP) = 1089$, where $\mu(RRP) = 690$ s, $\bar{D}(RRP) = 333$ s and $\theta = 66$. We observed improvements in the newly generated route plan RRP when comparing with OriginalRP. The results for OriginalRP were $\mu(OriginalRP) = 696$, $\bar{D}(OriginalRP) = 354$ and $\theta(OriginalRP) = 48$. So, the capacity occupation was reduced with 6 s, while average delay with 21 s. In addition, the new solution has a more diverse use of infrastructure resources. Thus, we may say that RouteCare may improve existing route plans.

4.2 Performance analysis: CA vs. RE submodels

In this section, we give a comparison of the produced route plans when only one of the CA or RE submodels is applied as well as the complete RouteCare with both submodels being active. Using only one submodel entails a subset of E- and I-rules that are applicable. Table 4 gives an outline of the used rules for each case considered.

Table 5 shows the results for all three cases. Namely, the capacity occupation, average cumulative delay as robustness cost, number of resources used, and the total cost. Bold numbers in the table indicate the value that was the aim of the optimization. For example, when only the CA submodel is used, then we obtain the *RP* that is optimized (minimized) for capacity utilization, while robustness is evaluated a posteriori. The second case is tackled in a similar way, while the last considered a multi-objective optimization with jointly minimizing capacity utilization, maximizing robustness (minimizing average cumulative delay) and maximizing the number of resources.

When using only the CA submodel, we obtained the lowest capacity usage of 609 s. However, a posteriori evaluation of robustness showed much greater cumulative delay compared to the other two cases, more than 160%. This means that

focusing solely on minimizing infrastructure occupation does not necessarily mean a better route plan robustness.

In the second case, when the model only maximizes robustness, $\bar{D}(RRP) = 318$ s is achieved. However, the capacity occupation is remarkably higher than the one in the first case with in total 194 which is an increase of 31%. Applying only the RE submodel resulted in a total cost of 1187 which was slightly smaller than when applying only the CA submodel. However, since RE does not consider the use of resources, it ended up with four resources less than the CA case.

Comparing the complete RouteCare with the RE submodel, we observe that with just a minor increase of the average cumulative delay, from 318 to 333 s, we obtained a significantly reduced capacity occupation of $\mu(RRP) = 690$ instead of 803. In addition, both models use an equal number of resources $\theta = 66$. This means that by including both objectives in RouteCare, a route plan of overall better quality is obtained. Looking at the total costs of the three, we concluded that the multi-objective setup performed the best with a solution that is better than the CA submodel by 117 (10.4%) and the RE submodel by 98 (9.5%). RouteCare showed that even for the same number of resources used, a more robust route plan can be found that uses more efficiently the station capacity.

In general, if a route planning is done by respecting only the robustness objective, it may lead to a significantly higher capacity occupation. Since the latter would be evaluated only afterwards, it may result in discarding such a route plan due to exceeding the stability norms. On the other hand, it is not sufficient to design a route plan that focuses only on minimizing the infrastructure occupation. Such a route plan may have an ample amount of time allowances but allocated inefficiently. Thus, distributing them in a smart way is also highly important. Therefore, it may be concluded that it is important to include multiple objectives in the optimization model in order to obtain principally better solutions for the route plan.

4.3 Testing RouteCare on different timetable instances

In this section, we present the results of random scenarios and timetables generated. We tested 24 scenarios with different numbers of train lines ranging between 9 and 14 operating within the cycle time $T = 30$ min. In these tests, we adopted $N_{iter} = 200$ and $stagMax = 40$. The characteristics of scenarios and the obtained results are given in Table 6.

Changing the number of trains does not necessarily mean a change in capacity occupation. Thus, the critical cycle may remain the same with an increasing (or decreasing) number of scheduled routes. In other words, an added route may not use resources on the critical cycle nor generate a new one. For example, this was observed between scenarios *sc1* and *sc2*, the capacity occupation remained the same while the number of trains is different, 14 and 13, respectively. However, the robustness cost for the latter case was reduced which is a result of less trains in the station.

The robustness cost shows better correspondence when changing the number of train routes in the station. In other words, more trains implies more dependencies between train routes and thus, less buffer times in between. In addition, the

Table 6 Additional computational results

Scenario	No. of trains $ Z $	Capacity occupation $\mu(RRP)$	Total delay $\bar{D}(RRP)$	No. of resources θ	Total cost $c(RRP)$	No. of iterations
sc1	14	826	125	65	1016	65
sc2	13	826	117	64	1007	62
sc3	13	599	115	74	788	64
sc4	12	599	107	74	780	89
sc5	11	654	94	74	822	44
sc6	10	599	94	72	765	72
sc7	9	608	80	64	752	59
sc8	13	746	105	66	917	71
sc9	12	631	110	62	803	105
sc10	13	821	115	72	1008	93
sc11	13	904	108	68	1080	47
sc12	12	651	107	66	824	77
sc13	11	651	94	64	809	62
sc14	10	484	92	61	637	85
sc15	9	160	80	62	302	50
sc16	12	746	103	71	920	45
sc17	10	631	97	62	790	78
sc18	12	821	94	67	982	109
sc19	12	631	110	69	810	70
sc20	13	690	118	71	879	120
sc21	13	890	115	70	1075	58
sc22	13	609	185	69	863	47
sc23	13	826	110	70	1006	61
sc24	13	821	176	74	1071	65

trend of increased total cost with an increasing number of train routes is more evident. As expected, the scenarios with a smaller number of train routes used usually less resources, while the ones with more trains used more resources. For most of the scenarios, 21 out of 24, RouteCare finished in less than 100 iterations, as shown in the last column.

We notice that the same number of trains does not necessarily generate the same costs, neither capacity occupation nor average delay. For example, the capacity occupation for scenarios with 13 train routes varies between 609 and 890 s, while the average delay varies between 110 and 185 s. Thus, we may conclude that the capacity occupation does not only depend on the station layout and number of trains but also on the way how the infrastructure has been utilized in terms of scheduled dwell times, corridors served by train lines, or types of trains. In addition, it depends on scheduled train operations such as trains

passing through, originating or terminating in the station. So, all these criteria influence the capacity occupation heavily.

5 Conclusions

This paper presented a new microscopic multi-objective model for solving the robust train routing problem. The developed multi-start local search heuristic RouteCare improves feasible, stable and robust route plans with even use of infrastructure resources. The model incorporates two submodels for capacity assessment based on max-plus automata theory and robustness evaluation based on the delay propagation algorithm. The feasibility is guaranteed as the input to the model was computed using a microscopic model for computing running times and blocking time stairways. Stability and robustness are increased by minimizing the cycle time (capacity occupation) and minimizing the expected total delays. Finally, the model maximizes the number of different resources used which reduces the need for maintenance closure of resources and thus, for alternative traffic adjustments.

RouteCare was tested on a medium-size station in the Netherlands. The results showed high efficiency of RouteCare, i.e., the algorithm reported good performance over multiple successive runs while the obtained solution was always within 0.73% of the best found solution. The two submodels were tested individually to evaluate their distinctive performances and also compared with the complete RouteCare. Results revealed that it is highly important to consider multiple objectives in order to obtain principally better overall solutions for the route plan in a station area. In particular, for the test case, RouteCare achieved a 9.5% improvement compared to the model that included only the robustness objective. These results stressed the importance of using multi-objective approaches to generate better route plans. The extensive computational analysis for different input timetables showed that the capacity occupation and the corresponding total cost are also heavily influenced by the way the station capacity is utilized like train mix and preferred platforms or scheduled dwell times and not only by the number of scheduled trains and the station layout.

As part of future research, we may assess the efficiency of exclusion and inclusion rules individually. The results may give an insight for further simultaneous usage of rules to further improve the performance of RouteCare. As new indicators of infrastructure utilization, we may consider how resources are used, e.g. number of switch operations, axle load and resource age. We also plan to develop an exact model for solving the RTRP, which would allow to quantify the improvements achieved by RouteCare. Moreover, we will test the importance of the implemented submodels on the resulting route plan by varying the corresponding weight factors. In addition, multiple Pareto-optimal solutions can be provided to planners to be selected which would increase the quality of the implemented route plans and reduce time needed to generate them. Overall, RouteCare provided very promising results and could be considered as a valuable support tool to railway planners in designing better railway services.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: theory. Algorithms and applications. Prentice Hall, Englewood Cliffs
- Bešinović N, Goverde RMP (2016) Improving robustness of railway timetables: a new two-stage approach. In: The ninth triennial symposium on transportation analysis (TRISTAN IX), 12–17 June 2016, Aruba
- Bešinović N, Goverde RMP (2018) Capacity assessment in railway networks. In: Handbook of optimization in the railway industry. International Series in Operations Research and Management Science, vol 268. Springer, Cham, pp 25–45
- Bešinović N, Goverde RMP, Quaglietta E, Roberti R (2016) Micro-macro approach to robust timetabling. *Transp Res Part B Methodol* 87:14–32
- Bešinović N, Quaglietta E, Goverde RMP (2017) Microscopic models and network transformations for automated railway traffic planning. *Comput Aided Civ Infrastruct Eng* 32(2):89–106
- Burkoltter DM (2005) Capacity of railways in station areas using Petri nets. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich
- Cacchiani V, Galli L, Toth P (2014) A tutorial on non-periodic train timetabling and train platforming problems. *EURO J Transp Logist* 4(3):285–320
- Caimi G, Burkoltter D, Herrmann T (2005) Finding delay-tolerant train routings through stations. In: Fleuren H, den Hertog D, Kort P (eds) Operations research proceedings, vol 2004. Springer, Berlin, pp 136–143
- Caimi G, Chudak F, Fuchsberger M, Laumanns M, Zenklusen R (2011) A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transp Sci* 45(2):212–227
- Caprara A, Galli L, Kroon LG, Maróti G, Toth P (2010) Robust train routing and online re-scheduling. In: 10th Workshop on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS '10)
- Carey M, Carville S (2003) Scheduling and platforming trains at busy complex stations. *Transp Res Part A* 37:195–224
- Corman F, Goverde RMP, D'Ariano A (2009) Rescheduling dense train traffic over complex station interlocking areas. In: Ahuja RK, Möhring RH, Zaroliagis CD (eds) Robust and online large-scale optimization: models and techniques for transportation systems. Springer, Berlin, pp 369–386
- Dewilde T, Sels P, Cattrysse D, Vansteenwegen P (2014) Improving the robustness in railway station areas. *Eur J Oper Res* 235(1):276–286
- Gaubert S, Mairesse J (1999) Modeling and analysis of timed petri nets using heaps of pieces. *IEEE Trans Autom Control* 44(4):683–697
- Goverde RMP (2007) Railway timetable stability analysis using max-plus system theory. *Transp Res Part B* 41(2):179–201
- Goverde RMP (2010) A delay propagation algorithm for large-scale railway traffic networks. *Transp Res Part C* 18(3):269–287
- Goverde RMP, Hansen IA (2013) Performance indicators for railway timetables. In: IEEE international conference on intelligent rail transportation 2013 (ICIRT), Beijing, August 30–September 1 2013, pp 301–306
- Goverde RMP, Bešinović N, Binder A, Cacchiani V, Quaglietta E, Roberti R, Toth P (2016) A three-level framework for performance-based railway timetabling. *Transp Res Part C Emerg Technol* 67:62–83
- Hansen IA, Pacht J (2014) Railway timetabling and operations. Eurailpress, Hamburg

- Heidergott B, Olsder GJ, van der Woude J (2005) Max plus at work: modeling and analysis of synchronized systems: a course on max-plus algebra and its applications. Princeton University Press, Princeton
- Herrmann TM (2006) Stability of timetables and train routings through station regions. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich
- Kroon LG, Romeijn HE, Zwaneveld PJ (1997) Routing trains through railway stations: complexity issues. *Eur J Oper Res* 98(3):485–498
- Kroon LG, Maroti G, Retel Helmrich MJ, Vromans MJCM, Dekker R (2008) Stochastic improvement of cyclic railway timetables. *Transp Res Part B Methodol* 42(6):553–570
- Lusby RM, Larsen J, Ryan D, Ehrgott M (2011a) Routing trains through railway junctions: a new set-packing approach. *Transp Sci* 45(2):228–245
- Lusby RM, Larsen J, Ehrgott M, Ryan D (2011b) Railway track allocation: models and methods. *OR Spectr* 33(4):843–883
- Petering MEH, Heydar M, Bergmann DR (2015) Mixed-integer programming for railway capacity analysis and cyclic, combined train timetabling and platforming. *Transp Sci* 50(3):892–909
- Sedgewick R, Wayne K (2011) Algorithms, 4th edn. Addison-Wesley, New York
- Sels P, Vansteenwegen P, Dewilde T, Cattrysse D, Waquet B, Joubert A (2014) The train platforming problem: the infrastructure management company perspective. *Transp Res Part B Methodol* 61:55–72
- Shapiro A (2000) Stochastic programming by Monte Carlo simulation methods, Stochastic Programming E-Print Series
- Sparing D, Goverde RMP (2017) A cycle time optimization model for generating stable periodic railway timetables. *Transp Res Part B Methodol* 98:198–223
- UIC, Code 406: capacity. International Union of Railways, Paris (2013)
- Van Egmond RJ (1999) Railway capacity assessment, an algebraic approach. TRAIL studies in transportation science series S99/2
- Yuan J, Hansen IA (2007) Optimizing capacity utilization of stations by estimating knock-on train delays. *Transp Res Part B* 41:202–217
- Zhang X, Nie L (2016) Integrating capacity analysis with high-speed railway timetabling: a minimum cycle time calculation model with flexible overtaking constraints and intelligent enumeration. *Transp Res Part C Emerg Technol* 68:509–531
- Zwanenburg WJ (2009) Modelling degradation processes of switches and crossings for maintenance and renewal planning on the Swiss railway network. Ph.D. Thesis, EPFL, Lausanne
- Zwaneveld PJ, Kroon LG, van Hoesel CPM (2001) Routing trains through a railway station based on a node packing model. *Eur J Oper Res* 128:14–33
- Zwaneveld PJ, Kroon LG, Romeijn HE, Salomon M, Dauzere-Peres S, van Hoesel CPM, Ambergen HW (1996) Routing trains through railway stations: model formulation and algorithm. *Transp Sci* 30:181–194

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.