

Optimizing Periodic Stability in Railway Timetables: A Microscopic Model for Networks with a Macroscopic Comparison

Florian Fuchs ^a, Bernardo Martin-Iradi ^a, Francesco Corman ^{a,1}

^a Institute for Transport Planning and Systems, ETH Zürich, 8092 Zürich, Switzerland

¹ E-mail: francesco.corman@ivt.baug.ethz.ch, Phone: +41 44 633 33 50

Abstract

We present a novel microscopic model for railway timetabling designed to maximize periodic stability in mixed single- and multi-track networks. Unlike conventional approaches based on the Periodic Event Scheduling Problem (PESP), our model provides a detailed infrastructure representation with flexible routing and nuanced conflict resolution, enhancing adaptability to real-world constraints and facilitating practical implementation by operators. To ensure scalability, we integrate a Satisfiability Modulo Theories (SMT)-based approach, which efficiently narrows feasible cycle time bounds, enabling the model to handle large-scale networks. Validated on operational data from the Rhätische Bahn network—a Swiss railway with complex infrastructure—the microscopic model consistently yields lower minimal cycle times than its macroscopic counterpart. The comparative analysis also offers insights into the trade-offs between model detail, computational efficiency, and achievable cycle times across diverse operational scenarios. These findings underscore the importance of infrastructure abstraction and the careful consideration of operational and commercial interdependencies for optimal stability in complex railway networks.

Keywords

Railway timetabling, Stability, Microscopic model, Vehicle circulation, Satisfiability Modulo Theories

1 Introduction

Maintaining timetable stability in densely operated railway networks is crucial to ensure that minor disruptions do not propagate. Stability is essential for operational reliability and service quality (Lusby et al., 2018). Periodic timetables, commonly used in passenger networks, introduce additional complexity since stability must account for the cyclic nature of schedules. A periodically stable timetable absorbs unexpected delays, allowing operations to recover (Goverde and Hansen, 2013). This paper focuses on optimizing periodic stability of a railway timetable.

To achieve periodic stability, state-of-the-art approaches primarily stem from the Periodic Event Scheduling Problem (PESP) Serafini and Ukovich (1989). PESP-based models enforce periodic constraints, ensuring consistent event intervals, making them well-suited for large-scale network applications (Peeters, 2003). However, traditional PESP models typically rely on macroscopic representations that simplify infrastructure details. While PESP can theoretically be applied to any level of infrastructure granularity, its practical applications remain predominantly macroscopic Caimi et al. (2017). This limitation restricts the ability to model complex railway operations accurately, such as flexible train routing, platform assignments, and dynamic overtaking scenarios (Sparing and Goverde, 2017; Zhang and Nie, 2016).

In response to these limitations, we propose a novel microscopic timetabling model to optimize periodic stability in complex railway networks. Our approach explicitly models detailed infrastructure elements, including track segments, platform assignments, and flexible train ordering in mixed single- and double-track sections. This enables a more precise representation of real-world operational constraints, surpassing the abstraction level of macroscopic models.

Unlike traditional PESP formulations, our model follows a time-expanded scheduling approach with synchronization constraints, making it structurally similar to job-shop scheduling models (Mannino and Nakkerud, 2023). This formulation enables fine-grained control over train sequencing while maintaining computational efficiency.

Given the computational challenges of microscopic modeling, we incorporate a Satisfiability Modulo Theories (SMT)-based algorithm to enhance scalability. By efficiently using SMT to narrow bounds on feasible cycle times, we address large-scale networks with detailed infrastructure requirements, a challenge

for conventional microscopic models. This method enables optimized timetables with manageable computational resources, even for extensive railway networks.

A unique feature of our model is the integration of vehicle circulation constraints, introducing dependencies between trains based on their operational sequences. While vehicle circulation has been studied in periodic timetabling (Lieshout, 2021), its integration with minimal cycle time optimization at a microscopic level is a novel contribution. This feature ensures stability while maintaining operational feasibility.

Additionally, we validate our model using a real-world case study based on the extensive Swiss railway operator Rhätische Bahn (RhB) network. This network presents a challenging mix of single- and multi-track sections, branches, and various service types, including long-distance, regional, and car shuttle trains. These diverse characteristics make RhB an ideal setting for evaluating the effectiveness of our model in capturing operational constraints and ensuring periodic stability.

Our contributions are threefold:

1. **Microscopic Timetabling Model:** Development of a microscopic model for maximizing periodic stability, incorporating flexible routing and detailed infrastructure elements following a job-shop scheduling-based structure.
2. **SMT-based Scalability:** Application of an SMT-based algorithm to efficiently narrow cycle time bounds, enabling the model to handle larger, more detailed networks than traditional microscopic approaches.
3. **Model Comparison and Validation:** A comparative evaluation of microscopic and macroscopic models, validated on the Swiss Rhätische Bahn network, showing the modeling and methodology benefits as well as practical insights into stability trade-offs and routing strategies.

The remainder of this paper is organized as follows: Section 2 reviews background concepts and existing stability models in railway timetabling. Section 3 details our proposed microscopic model together with the macroscopic version for comparison, and the SMT-based solving method. Section 4 presents the case study, experimental setup, and results. Finally, Section 5 summarizes key findings and discusses implications for railway network planning.

2 Background

Periodic railway timetabling aims to construct schedules that repeat cyclically while ensuring operational stability. Stability is crucial in densely operated networks, where small delays can propagate if the timetable lacks sufficient flexibility to absorb disturbances (Lusby et al., 2018). A fundamental metric for assessing periodic stability is the minimum cycle time, defined as the shortest period within which all scheduled operations can be executed without conflicts (Goverde, 2007). The cycle time ratio, which compares the minimum cycle time to the nominal period, further determines whether a schedule can be sustained under periodic conditions.

2.1 Models for Periodic Timetabling

The introduction mentions that PESP models are commonly used to formulate periodic railway timetabling problems. Most applications of PESP rely on macroscopic network representations, where train movements are abstracted through time-based headway constraints rather than explicit infrastructure details. While this facilitates large-scale optimization, it limits the ability to model flexible routing, dynamic overtaking, or detailed platform assignments (Sparing and Goverde, 2017; Zhang and Nie, 2016).

Several extensions have enhanced PESP to better account for infrastructure constraints. The Track-Choice PESP (Wüst et al., 2019) introduces routing flexibility, while the Turn-Sensitive PESP (Masing et al., 2023) accounts for overtaking and directional constraints. Infrastructure-Aware PESP (Bortoletto et al., 2023) explicitly models track-based constraints, whereas the Flexible Infrastructure Assignment PESP (Bortoletto et al., 2024) generalizes infrastructure allocation across multiple configurations. Despite these advancements, PESP-based approaches remain constrained by their reliance on modular periodicity constraints, which impose structural limitations when optimizing for minimal cycle time.

Beyond PESP, other periodic timetabling models have been proposed. Constraint-based formulations such as Heydar et al. (2013) incorporate multiple train types and explicitly minimize cycle time instead of relying on periodic constraints. Hybrid macroscopic-microscopic approaches, such as Bešinović et al.

(2016), generate macroscopic timetables but refine them using microscopic feasibility checks. These models balance computational efficiency with infrastructure detail but often rely on time-based train separation rather than explicit resource occupation.

A key distinction between macroscopic and microscopic periodic timetabling lies in train separation modelling. Macroscopic approaches, commonly used in PESP, enforce time-based train separation by imposing minimum headway constraints on successive train events (Figure 1(a)). While computationally efficient, this approach may fail to capture conflicts in shared track infrastructure. Microscopic models, in contrast, rely on resource-based separation, explicitly tracking infrastructure occupation through blocking time theory (Figure 1(b)). This method provides a more detailed conflict resolution mechanism, particularly for networks with single-track sections or high-density operations (Pachl, 2018). While technically, one could use any conflict avoidance approach on any level of detail, for the sake of clarity, in the remainder of the paper, we will refer to models relying on headway constraints as macroscopic and models using resource-based separation as microscopic.

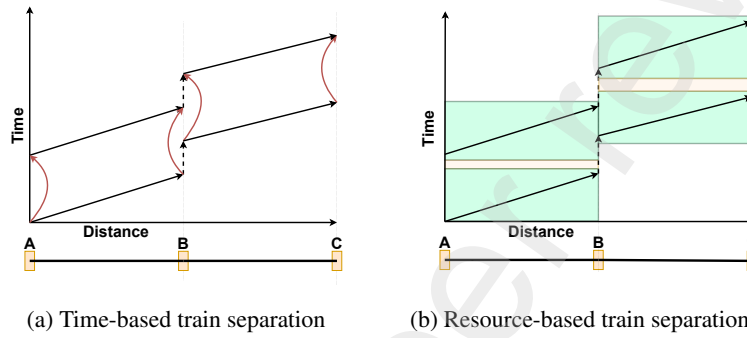


Figure 1: Comparison of time-based and resource-based train separation methods. (a) Time-based separation, common in macroscopic timetabling; (b) Resource-based separation, used in microscopic timetabling.

2.2 Optimizing for Periodic Stability

Many periodic timetabling models aim to generate feasible cyclic schedules but do not explicitly optimize for stability. Stability-oriented models focus on minimizing the cycle time, ensuring that timetables are robust against operational delays. Early work by Bergmann (1975) introduced a single-track stability model optimizing cycle time for homogeneous fleets, while Heydar et al. (2013) extended this to heterogeneous train types. Petering et al. (2016) incorporated dynamic platform assignments and overtaking operations, improving timetable flexibility.

PESP-based models have also evolved toward stability optimization. Sparing and Goverde (2017) proposed a cycle-time minimization approach within PESP, integrating running time adjustments for improved network performance. Zhang and Nie (2016) introduced heuristic-based stability improvements, demonstrating how timetable adjustments can influence periodic stability.

Alternative approaches focus on robustness rather than pure cycle time minimization. Flexible PESP (Caimi et al., 2011) introduces adaptive time bounds to accommodate variability, while Robust PESP (Gorrigk, 2015) optimizes headway buffers to mitigate delay propagation. These models prioritize resilience over strict cycle time minimization, offering a trade-off between periodicity and operational flexibility.

This paper presents a microscopic periodic timetabling model that integrates detailed infrastructure representations, routing, and vehicle circulation constraints while ensuring scalability through a Satisfiability Modulo Theories (SMT)-based optimization approach.

3 Problem Description and Formulation

We aim to optimize the routing and scheduling of a set of periodic trains $l \in \mathcal{L}$ over time. Given all primary trains within a single period, the objective is to create a timetable that can be repeated indefinitely, ensuring that operations recur consistently after each period T .

3.1 Routing Trains

In addition to assigning timestamps, we must select each train's routes across the infrastructure network. For this purpose, we use a *Train Flow Network* (TFN). An example of a TFN is shown in Figure 2.

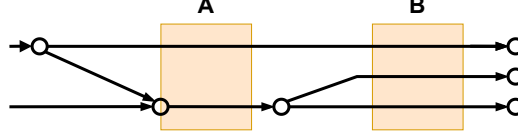


Figure 2: A Train Flow Network (TFN) for an example train with five paths.

The TFN is represented as a directed acyclic graph with nodes $n \in \mathcal{N}$ and arcs $w \in \mathcal{W}$. Each node represents a possible routing or decision point in the infrastructure, and each arc represents a possible track segment or path between nodes. We enforce flow conservation to ensure that each train selects exactly one path from its origin to its destination without cycles. To simplify the notation, we aggregate the routing decisions of all trains \mathcal{L} in a single TFN, and denote $\mathcal{N}^{\text{Source}}$ and $\mathcal{N}^{\text{Sink}}$ to the set of source and sink nodes for all trains. We introduce binary variables $x_w \in \{0, 1\}$ to indicate whether a train uses a particular link $w \in \mathcal{W}$. The flow conservation constraints are defined as:

$$\sum_{w \in \rho(n)^+} x_w - \sum_{w \in \rho(n)^-} x_w = \begin{cases} 1 & \text{if } n \in \mathcal{N}^{\text{Source}}, \\ -1 & \text{if } n \in \mathcal{N}^{\text{Sink}}, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

Where $\rho(n)^+$ and $\rho(n)^-$ are the sets of outgoing and incoming arcs at node n , respectively. This ensures that for each train, exactly one route is chosen from the source node to the sink node, representing the train's path through the network.

3.2 Modeling Itinerary and Commercial Constraints

In addition to route selection, we need to schedule the trains by determining the timings of various events. We model these decisions using events and activities within an *Event Activity Network* (EAN). Events are denoted as $e \in \mathcal{E}$, each associated with a specific timestamp t_e . Activities are denoted as $a \in \mathcal{A}$, representing the relationships between events, such as running times between stations or dwell times at stations.

The EAN is closely linked to the Train Flow Network (TFN) introduced earlier. Each arc $w \in \mathcal{W}$ in the TFN corresponds to a potential segment of a train's route and is associated with events and itinerary activities in the EAN. This ensures that the EAN contains all possible paths through the network. When a specific route is selected within the TFN (i.e., when $x_w = 1$ for an arc), only the corresponding itinerary activities in the EAN are enforced, reflecting the chosen path, while non-selected activities can be ignored. Similar to the TFN, all activities across trains \mathcal{L} are aggregated into a single EAN without loss of generality.

For each node n in the TFN, representing a location or decision point, there is an event e in the EAN corresponding to the train's arrival at or departure from that location. For each arc w in the TFN, representing a possible movement from one node to another, there is an activity a in the EAN representing the traversal of that segment. Each activity $a = (i, j)$ connects an origin event i and a destination event j , with associated time bounds $[lb_a, ub_a]$, specifying the minimum and maximum allowed time differences between i and j :

$$lb_a \leq t_j - t_i \leq ub_a, \quad \forall a = (i, j) \in \mathcal{A}. \quad (2)$$

We classify the activities into two categories: Itinerary and commercial activities.

Itinerary activities (\mathcal{A}^{ITI}) represent the technical requirements for running and dwelling times based on the chosen route. They are directly linked to the route choice variables x_w from the TFN. An itinerary

activity a is only active if the corresponding arc $\alpha(a)$ (where $\alpha(a)$ is the function mapping activity a to its associated arc w in the TFN) is part of the selected route ($x_{\alpha(a)} = 1$).

The time bounds $[lb_a, ub_a]$ for these activities are determined by physical and safety constraints, such as minimum running times. To ensure that itinerary activities are only enforced when the corresponding route is selected, we introduce the following constraint:

$$lb_a \cdot x_{\alpha(a)} \leq t_j - t_i \leq ub_a \cdot x_{\alpha(a)} + M \cdot (1 - x_{\alpha(a)}), \quad \forall a = (i, j) \in \mathcal{A}^{\text{ITI}}, \quad (3)$$

where M is a sufficiently large constant. This constraint effectively deactivates the itinerary activity when $x_{\alpha(a)} = 0$.

Commercial activities (\mathcal{A}^{COM}) capture service-related requirements, such as scheduled arrival and departure times, minimum and maximum dwell times at stations, and connections for passenger transfers or vehicle circulation. Commercial activities are always active regardless of the route chosen, as they represent the desired service characteristics. The time bounds for commercial activities are defined similarly:

$$lb_a \leq t_j - t_i \leq ub_a, \quad \forall a = (i, j) \in \mathcal{A}^{\text{COM}}. \quad (4)$$

An illustrative example of how commercial and itinerary constraints are modelled within the EAN is shown in Figure 3.

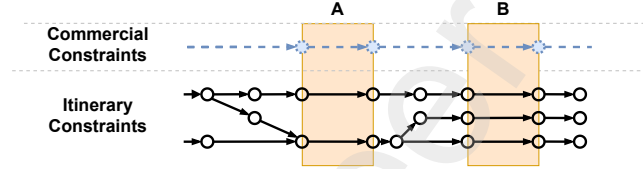


Figure 3: Commercial and itinerary constraints for an example train, modelled as activities in the EAN and linked to the TFN.

To ensure that commercial constraints are valid independently of the chosen track, we cluster all station entry and exit events into a single node for each train in the EAN (Figure 4). This allows the commercial and itinerary requirements to be combined without additional model modifications. We perform this clustering for each train $l \in \mathcal{L}$.

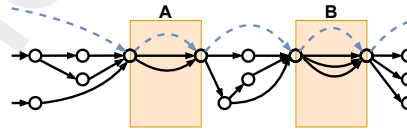


Figure 4: Clustering of events that are station entry or exit for one train. This ensures that all commercial requirements are met independently of the chosen track.

3.3 Modeling Vehicle Circulation

We enforce vehicle circulation by ensuring that specific sequences of services are covered by the same vehicle. This is achieved by imposing a constraint that limits the time between the arrival of one service and the departure of the next. This constraint is modeled as a commercial activity. Let $\mathcal{A}^{\text{VEH}} \subset \mathcal{A}^{\text{COM}}$ denote the set of activities representing vehicle transfers between consecutive services. As a subset of commercial constraints, these activities must satisfy Constraint (4):

$$lb_a \leq t_j - t_i \leq ub_a, \quad \forall a = (i, j) \in \mathcal{A}^{\text{VEH}}. \quad (5)$$

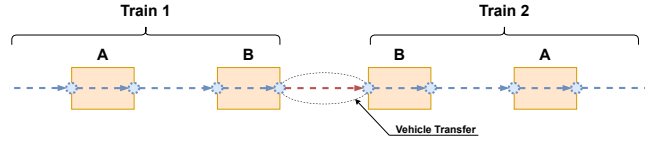


Figure 5: Illustration of vehicle circulation constraints. The diagram shows two consecutive train services (Train 1 and Train 2) assigned to the same vehicle.

Figure 5 provides a visualization example of a vehicle transfer activity. Without loss of generality, we incorporate Constraint (5) into Constraint (4), considering only the latter in the final formulation. Since these activities are independent of routing decisions, the TFN remains an acyclic graph.

3.4 Modeling Infrastructure Occupation

A key challenge in timetabling is ensuring that trains do not conflict with shared infrastructure resources. Conflicts occur when two trains attempt to use the same resource simultaneously. An example conflict is given in Figure 6. To prevent such conflicts, we model the occupation of resources by defining entry and exit events for each train's use of a resource path.

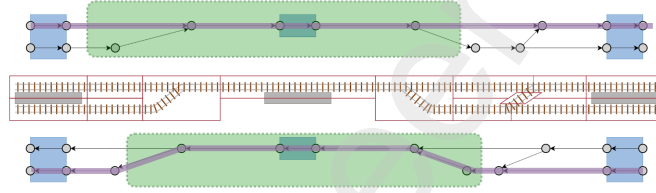


Figure 6: A potential conflict between two trains as they require the same resources (marked in green) along the single-track corridor.

Each train $l \in \mathcal{L}$ along its route visits a sets of resource paths (a continuous sequence where the train occupies the same resource). The set \mathcal{P} captures all resource paths across all trains. For each resource path $p \in \mathcal{P}$, we define the following events:

- **Entry time** t_p^{entry} : The time at which the train begins to occupy resource path p . This corresponds to the first event along the path in topological order. Therefore, $t_p^{\text{entry}} = t_e$, where e is the first event associated with path p in the Event Activity Network (EAN).
- **Exit time** t_p^{exit} : The time at which the train finishes using resource path p . This corresponds to the last event along the path in topological order. Thus, $t_p^{\text{exit}} = t_{e'}$, where e' is the last event associated with path p in the EAN.

These events are part of the EAN, and their associated times t_p^{entry} and t_p^{exit} represent when the train begins and ends its occupation of the resource.

Since trains have flexible routes, not all resource paths are necessarily part of the selected route. To model this, we introduce a binary variable z_p , where $z_p = 1$ if resource path p is used by the train, and 0 otherwise. To link z_p to the route selection, we define z_p based on the route choice variables x_w , which indicate whether the train uses a specific link w in the infrastructure. The relationship between z_p and x_w is expressed as:

$$z_p \geq \sum_{w \in \mathcal{W}_p} x_w - |\mathcal{W}_p| + 1, \quad (6)$$

where \mathcal{W}_p is the set of arcs (links) associated with resource path p . This constraint ensures that $z_p = 1$ only if all links in \mathcal{W}_p are selected, indicating that the entire resource path p is used.

To prevent conflicts between trains on shared resources, we consider pairs of resource paths (p, p') that use the same resources and could overlap in time. For each such pair $(p, p') \in \mathcal{K}$, we introduce precedence binary variables $y_{(p,p')}$ that are 1 if the train using resource path p clears the resource before the train using resource path p' occupies it, and 0 otherwise. We enforce the following disjunctive constraint to ensure that both precedence conditions cannot hold simultaneously:

$$y_{(p,p')} + y_{(p',p)} \leq 1, \quad (7)$$

This constraint ensures that either $y_{(p,p')} = 1$ or $y_{(p',p)} = 1$, but not both, thus avoiding any conflict.

To ensure that the precedence constraints are only enforced when both resource paths p and p' are used, we activate them with the following condition:

$$y_{(p,p')} + y_{(p',p)} \geq z_p + z_{p'} - 1, \quad (8)$$

This ensures that if both resource paths p and p' are used ($z_p = z_{p'} = 1$), then one of the precedence conditions ($y_{(p,p')} = 1$ or $y_{(p',p)} = 1$) must hold, preventing resource conflicts.

We further define separation constraints to ensure the required time gap between the trains' entry and exit times on conflicting resource paths. For resource paths p and p' , the separation constraints are:

$$t_{p'}^{\text{entry}} - t_p^{\text{exit}} \geq \delta - M \cdot (1 - y_{(p,p')}), \quad (9)$$

$$t_p^{\text{entry}} - t_{p'}^{\text{exit}} \geq \delta - M \cdot (1 - y_{(p',p)}), \quad (10)$$

where t_p^{entry} and t_p^{exit} are the entry and exit times for resource path p , $t_{p'}^{\text{entry}}$ and $t_{p'}^{\text{exit}}$ are the entry and exit times for resource path p' , δ is the minimum separation time required between trains on the same resource, and M is a sufficiently large constant used to linearise the disjunctive constraints.

These constraints ensure that if $y_{(p,p')} = 1$, train p must clear the resource at least δ time units before train p' begins to use it. Similarly, if $y_{(p',p)} = 1$, train p' must clear the resource at least δ time units before train p uses it.

3.5 Introducing Periodicity to Obtain a Cyclic Timetable

A cyclic timetable, also known as a periodic timetable, is one in which the schedule repeats every fixed period T . For instance, the timetable will repeat every hour if $T = 1$ hour. Events recur in each period, so if a train departs at time t in the first period, it will depart again at times $t + kT$, where k is a non-negative integer representing the repetition index. Figure 7 illustrates an example of a periodically repeated train, with time windows displayed for each repetition.

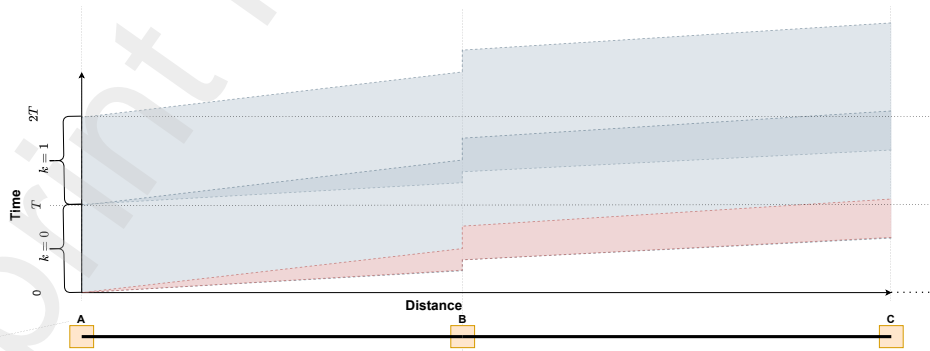


Figure 7: A periodically repeated train with the bounds on all events visualised. The red slot shows the bounds for the initial departure at $t_e = 0$ at location A. As we deal with a periodic timetable, the departure at A recurs each period. Thus, we can expand the red slot into the first blue slot for $k = 0$ and the second (top) blue slot for $k = 1$.

Consequently, we must constrain the timing of each train's origin events to ensure they occur within their designated periods. Let $\mathcal{E}_k^{\text{origin}}$ denote the set of origin events at cycle repetition k . We define the time bounds for these origin events as follows:

$$T \cdot k \leq t_e \leq T \cdot (k + 1), \quad \forall e \in \mathcal{E}_k^{\text{origin}}, \quad \forall k \in \{0, 1, \dots, \hat{k} - 1\} \quad (11)$$

These constraints ensure that each origin event occurs within its respective period. Subsequent events are implicitly constrained by the activity bounds and periodicity constraints, preventing their times from becoming arbitrarily large.

To create a perfectly cyclic timetable, the schedule must repeat exactly every period T . This requires that the time difference between an event and its k -th repetition is $k \cdot T$, where $k \in \mathbb{Z}_{\geq 0}$.

Since some trains may take longer than one period T to complete their journeys, the timetable must be rolled out over multiple periods. Let Δ denote the maximum run time of any train from its origin to its destination. The number of periods \hat{k} required to achieve a fully periodic schedule is then calculated as:

$$\hat{k} = \left\lceil \frac{\Delta}{T} \right\rceil + 1. \quad (12)$$

This expansion requires the timetable to be extended over \hat{k} periods, with each train being replicated $\hat{k} - 1$ times to account for its operations in subsequent periods. As shown by Zhang et al. (2019), satisfying the condition in Equation (12) ensures that a sufficient number of copies are included to capture all potential conflicts. Thus, once the timetabling problem is solved over \hat{k} periods, the timetable can be rolled out for as many repetitions as needed without risking unresolved conflicts.

We introduce an operator that designates the repetition index for each event to model the periodicity. Let $k \in \{0, 1, \dots, \hat{k} - 1\}$ represent the repetition index, with $k = 0$ corresponding to the initial event in the first period. The departure time of the k -th repetition of a train is then given by $t^{(k)} = t^{(0)} + kT$, where $t^{(0)}$ is the departure time in the initial period.

To ensure that all repeated trains adhere to the cyclic timetable, we introduce periodicity activities \mathcal{A}^{PER} . These activities link events across different periods, enforcing that the time difference between an event and its repetition is $k \cdot T$. For any event e repeated over k periods, the periodicity constraint is defined as:

$$t_{e^{(k)}} = t_{e^{(0)}} + kT, \quad \forall e \in \mathcal{E}, \quad \forall k \in \{0, 1, \dots, \hat{k} - 1\}. \quad (13)$$

Additionally, spatial periodicity must be maintained, ensuring that each repeated train follows the same route as in the initial period. To enforce this, the route choice variables x_w are linked across the repetitions. Specifically, for all repetitions k , we set:

$$x_w^{(k)} = x_w^{(0)}, \quad \forall w \in \mathcal{W}, \quad \forall k \in \{1, \dots, \hat{k} - 1\}. \quad (14)$$

This constraint guarantees that the route choice for each train remains identical across all periods, thereby maintaining spatial periodicity. All train repetitions utilize the same sequence of infrastructure elements as in the initial period.

In summary, $t_{e^{(k)}}$ denotes the time of event e in the k -th repetition, and $x_w^{(k)}$ is the route choice variable for arc w in repetition k . By enforcing both temporal (13) and spatial (14) periodicity constraints, we ensure that the timetable repeats exactly every period T , with trains following consistent routes in each repetition.

Note on Notation: For simplicity, we denote variables without a period index (e.g., x_w and t_e) to represent the base case of $k = 0$, where the timetable period is not explicitly repeated. In cases where periodicity constraints are applied, such as in the temporal and spatial periodicity sections, we include the repetition index k (e.g., $x_w^{(k)}$ and $t_e^{(k)}$) to indicate subsequent periods.

3.6 The Cyclic Microscopic Railway Timetabling Problem (C-MicroRTP)

The C-MicroRTP integrates train operations' temporal and spatial periodicity while considering infrastructure usage and conflict avoidance. The model is formulated as follows:

$$\min \sum_{a=(i,j) \in \mathcal{A}^{\text{COM}}} (t_j - t_i), \quad (15a)$$

$$\text{subject to:} \quad (15b)$$

$$\sum_{w \in \rho(n)^+} x_w = 1, \quad \forall n \in \mathcal{N}^{\text{Source}}, \quad (15c)$$

$$\sum_{w \in \rho(n)^+} x_w = \sum_{w \in \rho(n)^-} x_w, \quad \forall n \in \mathcal{N} \setminus (\mathcal{N}^{\text{Source}} \cup \mathcal{N}^{\text{Sink}}), \quad (15d)$$

$$z_p \geq \sum_{w \in \mathcal{W}_p} x_w - |\mathcal{W}_p| + 1, \quad \forall p \in \mathcal{P}, \quad (15e)$$

$$lb_a \cdot x_{\alpha(a)} \leq t_j - t_i \leq ub_a \cdot x_{\alpha(a)} + M \cdot (1 - x_{\alpha(a)}), \quad \forall a = (i, j) \in \mathcal{A}^{\text{ITI}}, \quad (15f)$$

$$lb_a \leq t_j - t_i \leq ub_a, \quad \forall a = (i, j) \in \mathcal{A}^{\text{COM}}, \quad (15g)$$

$$T \cdot k \leq t_e \leq T \cdot (k + 1), \quad \forall e \in \mathcal{E}_k^{\text{origin}} \quad \forall k \in \{0, 1, \dots, k - 1\} \quad (15h)$$

$$t_{e(k)} = t_{e(0)} + kT, \quad \forall e \in \mathcal{E}, \quad \forall k \in \{0, 1, \dots, \hat{k} - 1\}, \quad (15i)$$

$$x_w^{(k)} = x_w^{(0)}, \quad \forall w \in \mathcal{W}, \quad \forall k \in \{1, \dots, \hat{k} - 1\}, \quad (15j)$$

$$y_{pp'} + y_{p'p} \leq 1, \quad \forall \{p, p'\} \in \mathcal{K}, \quad (15k)$$

$$y_{pp'} + y_{p'p} \geq z_p + z_{p'} - 1, \quad \forall \{p, p'\} \in \mathcal{K}, \quad (15l)$$

$$t_{p'}^{\text{entry}} - t_p^{\text{exit}} + \delta \geq M(y_{pp'} - 1), \quad \forall \{p, p'\} \in \mathcal{K}, \quad (15m)$$

$$t_p^{\text{entry}} - t_{p'}^{\text{exit}} + \delta \geq M(y_{p'p} - 1), \quad \forall \{p, p'\} \in \mathcal{K}, \quad (15n)$$

$$t_e \in \mathbb{R}^+, \quad \forall e \in \mathcal{E}, \quad (15o)$$

$$t_p^{\text{entry}}, t_p^{\text{exit}} \in \mathbb{R}^+, z_p \in \{0, 1\}, \quad \forall p \in \mathcal{P}, \quad (15p)$$

$$y_{pp'}, y_{p'p} \in \{0, 1\}, \quad \forall \{p, p'\} \in \mathcal{K}, \quad (15q)$$

$$x_w \in \{0, 1\}, \quad \forall w \in \mathcal{W}. \quad (15r)$$

Since all the constraints have been explained in the previous sections, we omit a detailed explanation of the equations in Equation 15. Here, we defined the objective in Equation 15a such that the total duration of all commercial constraints is minimized.

3.7 Extending C-MicroRTP to Variable Cycle Microscopic Timetabling Problem (VC-MicroRTP)

The C-MicroRTP optimizes the timetable, assuming a fixed period T . To generalize this, we introduce a variable period duration ω in place of T , resulting in the Variable Cycle Microscopic Railway Timetabling Problem (VC-MicroRTP). Our objective is now to minimize ω . If a timetable is feasible for C-MicroRTP with a fixed period T , we can assume ω to be within the range $\omega \in (0, T]$. Thus, the variable ω is constrained such that the minimal period satisfies all defined constraints while reducing the total cycle time.

To extend the model, we replace T with ω in all periodic constraints so that instead of $k \cdot T$, we now have $k \cdot \omega$. Specifically, the temporal periodicity constraint (15i) is replaced by:

$$t_{e(k)} = t_{e(0)} + k\omega, \quad \forall e \in \mathcal{E}, \quad \forall k \in \{0, 1, \dots, \hat{k} - 1\}. \quad (16)$$

Furthermore, we have to adjust Equation (11) to restrict the time windows of the origin events to account for a variable period duration by replacing T with ω , leading to Equation (17), which replaces constraint (15h) in (15).

$$\omega \cdot k \leq t_e \leq \omega \cdot (k + 1), \quad \forall e \in \mathcal{E}_k^{\text{origin}} \quad \forall k \in \{0, 1, \dots, k - 1\} \quad (17)$$

These two modifications ensure consistency between the periodicity constraints and the newly defined variable period.

3.8 Variable Cycle Macroscopic Timetabling Problem (VC-MacroRTP)

We develop a macroscopic variant of the VC-MicroRTP to compare microscopic and macroscopic models. The primary difference between these models lies in the infrastructure representation and mechanism to prevent train conflicts. The macroscopic model utilizes an aggregated infrastructure representation, managing conflicts through headway constraints rather than blocking time theory as in the microscopic model. Figure 8 provides an example of the aggregation process used for itinerary constraints.

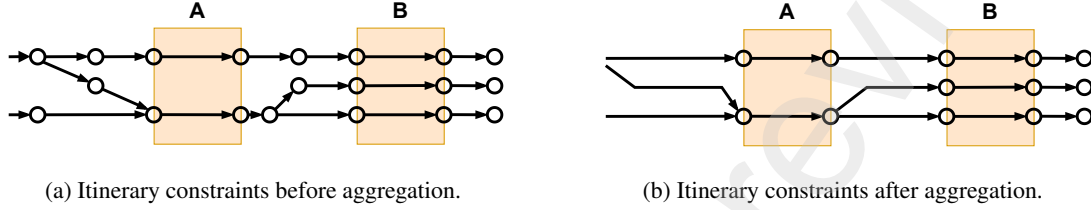


Figure 8: Example of itinerary constraint aggregation for an example train.

All itinerary activities between stations are aggregated to convert the microscopic Event Activity Network (EAN) to a macroscopic representation. However, we retain the routing options to allow flexibility, as depicted in Figure 8. Traditional, macroscopic periodic timetable models, such as PESP, generally do not accommodate alternative routing options (Caimi et al., 2017). Recent studies, however, indicate that incorporating routing alternatives (Fuchs et al., 2022) or track choice (Masing et al., 2023) can lead to higher-quality solutions. Additionally, Sparing and Goverde (2017) extended PESP to enable more accurate modelling of overtaking manoeuvres.

Based on these findings, we have included routing flexibility in our VC-MacroRTP formulation. Given that only adjustments to the EAN and a shift from blocking time conflicts to headway constraints are required, we omit the detailed formulation of VC-MacroRTP here.

3.9 Tightening the Bounds of the Minimal Cycle Time using Satisfiability Modulo Theories (SMT)

The challenge of finding minimal cycle times in timetable optimization is well-documented, as demonstrated by Sparing and Goverde (2017), who proposed a specialized algorithm to obtain tighter bounds on ω within a PESP-based model. To address this challenge, we introduce an algorithm to derive improved bounds for ω before invoking the Mixed Integer Program (MIP) solver. This algorithm consists of two main steps: a binary search to establish an upper bound on ω , and an interval-based search to refine the lower bound.

Both search steps involve transforming the model from a MIP into a Boolean Satisfiability Problem (SAT) extended by difference constraints, enabling us to find a solution using Satisfiability Modulo Theories (SMT). This transformation uses the SMT solver applied by Leutwiler and Corman (2022). For additional details on SMT, readers are referred to that work.

Binary Search for the Upper Bound The binary search for an upper bound on ω leverages the feasibility of any solution with $T \in [0, T_{\max}]$ as an upper bound. By conducting a binary search within this range, we efficiently narrow down possible values of T that serve as an upper bound for ω .

Interval Search for the Lower Bound To obtain a lower bound on ω , we relax the strict temporal periodicity constraint of the model in Equation 15, defining the periodicity bounds as $\underline{T} \leq T \leq \bar{T}$, where \underline{T} and \bar{T} denote the lower and upper bounds, respectively. This relaxation replaces the equality constraint in Equation 13 with the inequalities shown in Equations 18 and 19:

$$t_{e(k)} \leq t_{e(0)} + k\bar{T}, \quad \forall e \in \mathcal{E}, \quad \forall k \in \{0, 1, \dots, \hat{k} - 1\}. \quad (18)$$

$$t_{e(k)} \geq t_{e(0)} + k\underline{T}, \quad \forall e \in \mathcal{E}, \quad \forall k \in \{0, 1, \dots, \hat{k} - 1\}. \quad (19)$$

If the SMT model with interval $[\underline{T}, \overline{T}]$ is found infeasible, we can conclude that no feasible solution exists for any fixed T within this interval. This allows us to iteratively improve the lower bound on ω through a linear search. The search procedure is initialized with $\underline{T} = \omega_{\min}$, $\overline{T} = \omega_{\max}$, and search width $d = \overline{T} - \underline{T}$. Suppose the problem is found to be infeasible. In that case, we increase the lower bound by setting $\underline{T} = \overline{T}$ and adjust the interval to $\overline{T} = \min(\omega_{\max}, \overline{T} + d)$. In cases where the model is feasible, we halve the search interval width $d = d/2$ and set $\overline{T} = \min(\omega_{\max}, \overline{T} + d)$. This iterative search continues until \underline{T} and \overline{T} converge to a single value.

Additionally, in cases where the SMT solver finds a feasible solution for a relaxed periodicity interval with a very small width $d = \overline{T} - \underline{T}$, we may attempt to recover a strictly periodic solution through a repair procedure. Specifically, we use the SMT solution as a warm start and temporarily limit the event times to the bounds applied in SMT in the strictly periodic MIP model. This strategy effectively limits the MIP's search space to a neighborhood around the near-periodic solution. If the MIP solver identifies a strictly periodic and feasible solution within this window, it can be used to tighten the upper bound on ω as it might yield an improved feasible solution.

Moreover, the bounding procedure itself can serve as an approximation scheme. Once the upper and lower bounds on ω fall within a user-defined tolerance, the procedure can be terminated, and the best known solution returned. This provides a guaranteed approximation of the optimal cycle time with provable bounds on suboptimality.

4 Case Study

This case study investigates the application of our microscopic model (MICRO) and compares it against a macroscopic representation (MACRO) of the same problem. We conduct a series of experiments to assess the impact of different constraints on periodic stability and to analyze the transferability of solutions between the two models.

4.1 Infrastructure and Timetable

Using the original microscopic network, the case study is based on infrastructure data from the Swiss railway company *Rhätische Bahn* (RhB). Figure 9 shows the layout and associated line plan. The timetable consists of periodic passenger and car shuttle train services, operating at a cycle time of one hour ($T = 1$ h). For each line in the network, two trains (one in each direction) are scheduled per cycle, except for the car shuttles, which operate at a frequency of twice per hour.

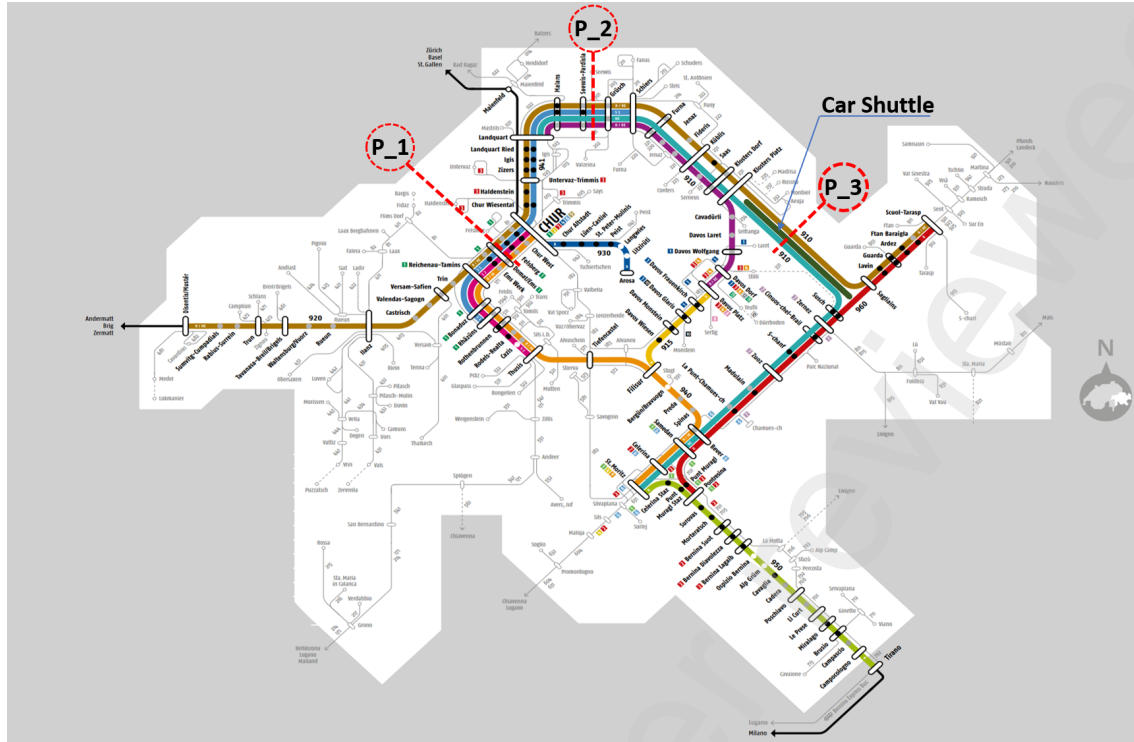


Figure 9: Line plan of RhB as operated in 2022, consisting of 11 lines, with a period duration of 1 h. The car shuttle operates at a frequency of two, while all other services operate at a frequency of one. Additionally, the partitions P_1 , P_2 , and P_3 are marked. All lines that cross the partition marking are part of the corresponding partition.

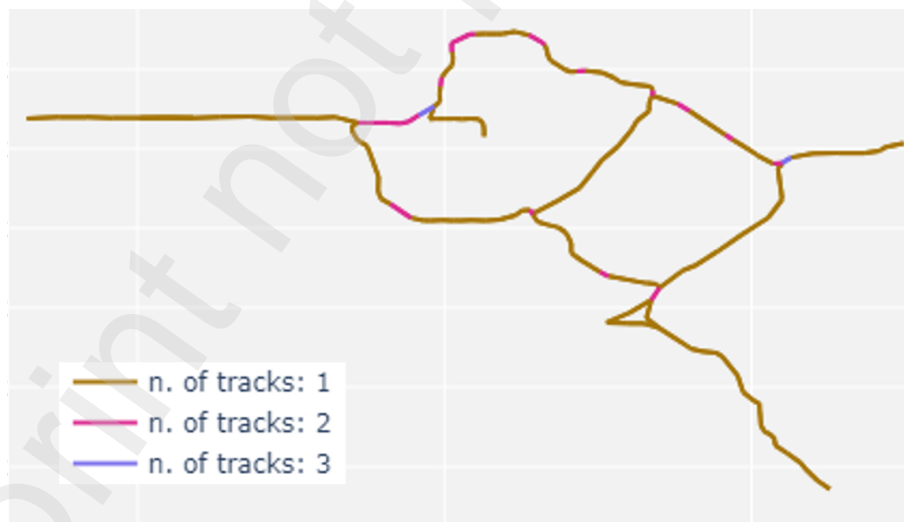


Figure 10: Number of tracks by section. Overall, the network covers 102 stations and extends for 385 km.

RhB's operational data define each train's minimal running times, potential train routes, and stopping patterns. This includes vehicle circulations and transfer requirements. Parameters such as reoccupation times and minimal headway times are directly derived from RhB's operational practices without modifications. For the macroscopic model, in case of two following trains outside of stations, we define a minimum headway time of 120 seconds, as this value has been successfully used in previous studies on this network

(Fuchs et al., 2022). If two trains travel in opposite directions, we define a minimum headway of 60 seconds. Tracks within stations (i.e., dwelling activities) require a minimum headway of 60 seconds, regardless of whether the trains meet head-on or follow each other.

The case study network predominantly comprises single-track sections, as shown in Figure 10. Consequently, train meetings or overtakings can only occur in stations or sections with two or more tracks. Both microscopic and macroscopic EANs are constructed for this study. The macroscopic EAN is derived by aggregating the microscopic EAN, as detailed in Section 3.8. The study evaluates the entire network (*All*) and three partitions, marked in Figure 9 as *P_1*, *P_2*, and *P_3*.

To provide a comprehensive overview of the case study's scope, we report key size metrics for the EAN and infrastructure across all partitions and the entire network. These metrics include the number of trains, resources (e.g., track segments), potential conflicts, routing decisions, links, and nodes. Table 1 summarizes these statistics. Note that the reported values account for all trains departing in the first period before the subsequent roll-out.

Table 1: Key size metrics for the Event Activity Network and infrastructure by partition.

Instance	# Trains	# Resources	# Conflicts	# Routing Decisions	# Links	# Nodes
<i>P_1</i>	10	1072	40407	900	6732	5983
<i>P_2</i>	10	1176	29696	565	4644	4204
<i>P_3</i>	8	692	11098	220	2032	1866
<i>All</i>	28	2026	72951	1307	10613	9567

The models were implemented in Python using the *OpenBus* framework (Fuchs and Corman, 2019), with Gurobi 12.0.1 as the MIP solver (Gurobi Optimization, LLC, 2023). All experiments were conducted on a server with 16 CPU cores (Intel Xeon Gold 6248) and 32 GB of RAM. We optimise to a gap of 0.0 %.

4.2 Effect of Different Commercial Constraint Configurations on Minimal Cycle Time

In the first assessment, we aim to assess the effect of varying commercial constraints, as has been done in previous studies (Sparing and Goverde, 2017; Zhang and Nie, 2016). We conducted experiments with four distinct configurations. Each configuration is defined by two key parameters, **L** and **G**, which control the permissible slack and flexibility in the schedule:

L : This parameter dictates the maximum allowable deviation in running time between two commercial stops. For example, a value of **L = 1.1** indicates that the schedule can accommodate up to 1.1 times the minimal running time between two commercial stops.

G : This parameter defines the total slack allowed for the entire train journey from terminal to terminal. A value of **G = 1.1** specifies that the total journey time should not exceed 1.1 times the minimal running time required for the complete trip.

We evaluate the network as well as the three partitions for four sets of parameters: (**L: 1.1, G: 1.1**), (**L: 1.2, G: 1.15**), (**L: 1.3, G: 1.2**) and (**L: 1.4, G: 1.25**). The resulting cycle ratio, defined by $\omega/3600$ s are given in Table 2.

Table 2: Minimal cycle ratio found by MICRO and MACRO by partition and configuration.

Configuration	Model	P_1	P_2	P_3	All
(L: 1.1 G: 1.1)	MICRO	0.646	0.664	0.711	0.714
	MACRO	0.667	0.723	0.652	-
(L: 1.2 G: 1.15)	MICRO	0.646	0.651	0.711	0.711
	MACRO	0.667	0.703	0.632	0.705
(L: 1.3 G: 1.2)	MICRO	0.646	0.636	0.711	0.711
	MACRO	0.667	0.630	0.620	0.705
(L: 1.4 G: 1.25)	MICRO	0.646	0.631	0.711	0.711
	MACRO	0.667	0.607	0.609	0.705

The results in Table 2 show that the effect of increased slack on the minimal cycle ratio varies by model and partition. In the MICRO model, the ratio remains stable across all slack levels, particularly for P_1 and P_3, indicating limited sensitivity to slack adjustments. In contrast, the MACRO model shows a decreasing trend in cycle ratio for P_2 and P_3 as slack increases, suggesting that additional slack allows the model to better exploit routing and timing flexibility.

Notably, the full network is infeasible under MACRO for the most restrictive slack configuration ($L : 1.1, G : 1.1$), confirming that some timetables cannot be operated without violating operational margins. When comparing models, MACRO yields lower cycle ratios in P_2 and P_3 under relaxed constraints, while MICRO performs better in P_1.

4.3 Impact of Fixed vs. Flexible Routing on Minimal Cycle Time

In this experiment, we investigate the impact of restricting routing options on minimal cycle time, aiming to understand the value of flexible routes in timetable optimization. By removing all routing alternatives except the default ones, we constrained the trains to follow their predefined routes. This configuration was evaluated using the same four slack parameter sets ($(L: 1.1, G: 1.1)$, $(L: 1.2, G: 1.15)$, $(L: 1.3, G: 1.2)$, and $(L: 1.4, G: 1.25)$). Table 3 presents the minimal cycle-time ratios achieved with fixed routing and the differences from the flexible-routing results in Table 2.

Table 3: Minimal cycle ratio found by MICRO and MACRO by partition and configuration, with fixed routing. Values in brackets indicate differences from Table 2, while "–" indicates that no stable solution could be found.

Configuration	Model	P_1	P_2	P_3	All
$(L: 1.1, G: 1.1)$	MICRO	–	–	–	–
	MACRO	–	–	–	–
$(L: 1.2, G: 1.15)$	MICRO	0.822 (+0.176)	–	–	–
	MACRO	0.836 (+0.169)	–	–	–
$(L: 1.3, G: 1.2)$	MICRO	0.822 (+0.176)	0.842 (+0.206)	0.842 (+0.131)	0.898 (+0.188)
	MACRO	0.836 (+0.169)	0.857 (+0.228)	0.857 (+0.237)	0.929 (+0.224)
$(L: 1.4, G: 1.25)$	MICRO	0.806 (+0.160)	0.842 (+0.211)	0.842 (+0.131)	0.898 (+0.188)
	MACRO	0.832 (+0.166)	0.857 (+0.250)	0.857 (+0.248)	0.868 (+0.163)

The results show that the minimal cycle time increases across all tested partitions and slack parameters when routing flexibility is removed. This increase highlights the role of routing flexibility in optimizing for lower cycle times, as restricting routes reduces the model's ability to exploit alternate paths and minimize train conflicts.

A notable observation is that for the two most restrictive slack configurations ($L: 1.1, G: 1.1$) and $(L: 1.2, G: 1.15)$, both MICRO and MACRO models fail to find stable solutions for the entire network and some partitions, indicating that the fixed routes are insufficient to meet a one-hour cycle under tighter constraints. This lack of feasible solutions emphasizes the importance of routing flexibility, especially under stringent operational constraints.

For the more relaxed configurations ($L : 1.3, G : 1.2$) and $(L : 1.4, G : 1.25)$, both models produce stable solutions across all partitions. MICRO achieves consistently lower cycle times at the partition level, benefiting from its finer-grained conflict representation. However, for the full network, MACRO achieves a lower cycle ratio (0.868 vs. 0.898) in the most relaxed setting.

In summary, the analysis indicates that flexible routing significantly enhances the ability to reduce the minimal cycle time, particularly under stringent slack constraints. With more slack, fixed routing becomes feasible but generally results in higher cycle times, underscoring the trade-offs between routing flexibility and cycle time.

4.4 Effect of Vehicle Circulation Constraints on Minimal Cycle Time

In contrast to the previous assessment, this experiment introduces additional constraints by incorporating vehicle circulation dependencies. Vehicle circulation constraints enforce that a train can only depart from the first station if the corresponding rolling stock is available. When this rolling stock is provided by

another, terminating train, this creates interdependencies among trains. As with the previous configurations, we evaluate the entire network and the three partitions across the four slack parameter sets. Table 4 shows the minimal cycle ratios observed with vehicle circulation constraints, along with the differences from the baseline results without vehicle circulation (Table 2).

Table 4: Minimal cycle ratio found by MICRO and MACRO by partition and configuration, with vehicle circulation. Values in brackets show differences from Table 2.

Configuration	Model	P.1	P.2	P.3	All
(L: 1.1 G: 1.1)	MICRO	0.822 (+0.176)	0.679 (+0.015)	0.725 (+0.014)	0.931 (+0.217)
	MACRO	0.837 (+0.170)	0.816 (+0.093)	0.696 (+0.044)	inf (–)
(L: 1.2 G: 1.15)	MICRO	0.822 (+0.176)	0.679 (+0.029)	0.725 (+0.014)	0.835 (+0.124)
	MACRO	0.837 (+0.170)	0.737 (+0.034)	0.696 (+0.064)	inf (–)
(L: 1.3 G: 1.2)	MICRO	0.822 (+0.176)	0.679 (+0.043)	0.725 (+0.014)	0.835 (+0.124)
	MACRO	0.837 (+0.170)	0.737 (+0.108)	0.696 (+0.076)	0.855 (+0.149)
(L: 1.4 G: 1.25)	MICRO	0.822 (+0.176)	0.679 (+0.048)	0.725 (+0.014)	0.835 (+0.124)
	MACRO	0.837 (+0.170)	0.699 (+0.092)	0.696 (+0.087)	0.855 (+0.149)

The results in Table 4 reveal that the impact of vehicle circulation constraints is partition-dependent. For *P.1* and *ALL*, the minimal cycle ratios increase substantially, indicating that vehicle circulation constraints introduce significant challenges in these network sections. This reflects the added complexity and interdependence imposed by circulation requirements, which limit scheduling flexibility and increase minimal cycle times.

In contrast, partitions *P.2* and *P.3* exhibit minimal or no changes in cycle ratios, with values either remaining constant or experiencing only marginal increases. This suggests that the infrastructure configurations and train movements within these partitions are less constrained by vehicle circulation dependencies, likely due to reduced inter-train dependencies or lower rolling stock requirements.

Another important observation is that for the least restrictive slack configurations (**L: 1.3, G: 1.2** and **L: 1.4, G: 1.25**), the MICRO model maintains stable solutions across all partitions. Notably, for the most restrictive configurations (**L: 1.1, G: 1.1** and **L: 1.2, G: 1.15**), the MACRO model fails to find feasible solutions for the entire network, while the MICRO model still returns feasible results.

These findings indicate that vehicle circulation constraints can significantly affect the minimal cycle time, particularly in more interconnected network sections. As observed in *P.1* and the entire network, vehicle interdependencies restrict scheduling flexibility, emphasizing balancing circulation constraints with infrastructure capacity to achieve lower cycle times.

4.5 Combined Effects of Constraints on Minimal Cycle Time

This section analyses the cumulative impact of progressively adding constraints—flexible routing, vehicle circulation, and transfer constraints—on the minimal cycle time. For this analysis, we focus on the entire network (partition *ALL*), examining the incremental effects on cycle time when transitioning from flexible routing alone to a fully constrained model. Passenger transfer constraints, in addition to vehicle circulation, introduce further dependencies between trains, impacting schedule stability. These transfers are modeled as commercial constraints, ensuring that the time between the arrival of one train and the departure of the connecting train lies within predefined limits. The corresponding bounds were derived from operational data provided by RhB.

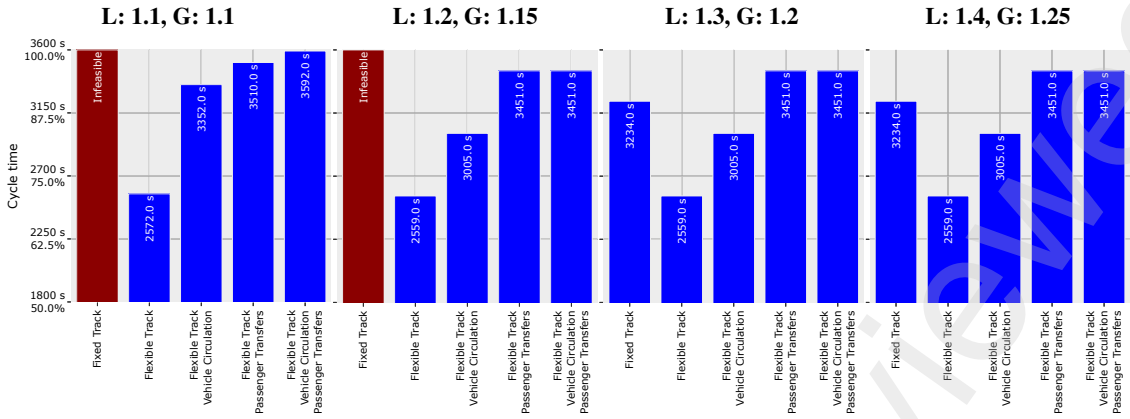


Figure 11: Barplots illustrating the impact of different constraints on the minimal cycle time across the whole network when using MICRO. Each configuration is labelled with L and G parameters, reflecting the slack levels for the particular scenario.

Figure 11 illustrates the effects of adding constraints on cycle time for the MICRO model. To help interpret the chart, we walk through one configuration, ($L : 1.1, G : 1.1$), where:

1. The *fixed track* configuration is infeasible, as no solution satisfies the one-hour cycle bound.
2. Allowing *flexible routing* reduces the cycle time to 2572 s, showing the benefit of route choice.
3. Adding *vehicle circulation* increases the cycle time to 3352 s.
4. Including *passenger transfers* raises it further to 3510 s.
5. Finally, enforcing both *vehicle circulation and transfers* results in a minimal cycle time of 3592 s, leaving just 8 s below the one-hour threshold.

This trend is consistent across all configurations: each added constraint leads to a longer cycle time. The gain from flexible routing is progressively offset by the structural dependencies introduced by circulation and transfer constraints.

In the most constrained scenarios (rightmost bars in Figure 11), the resulting minimal cycle times range between 3451 s and 3592 s, corresponding to an operational reserve of just 0.2% to 4.1%. This is notably lower than recommendations found in the literature, such as the 7–10% reserve suggested by [Goverde and Hansen \(2013\)](#).

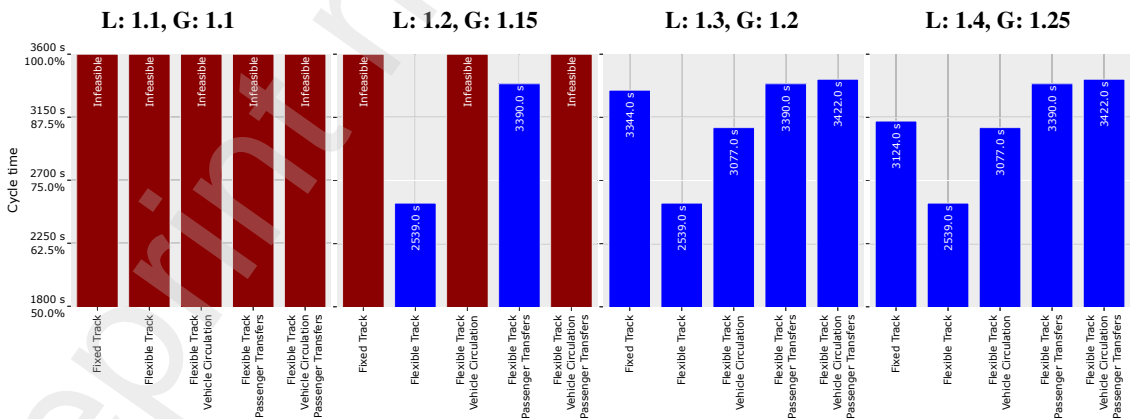


Figure 12: Barplot to assess the impact of different constraints on the minimal cycle time considering the whole network when using MACRO. Colors are consistent with Figure 11.

As shown in Figure 12, no feasible solution exists for the most restrictive slack configuration ($L : 1.1, G : 1.1$), regardless of the set of constraints applied. Even with flexible routing, the limited slack prevents a stable timetable once vehicle circulation and passenger transfers are introduced.

For the more relaxed configurations, the MACRO model yields periodically stable timetables, with cycle times increasing progressively as additional constraints are added. Across all slack levels, the baseline with flexible routing alone consistently results in a minimal cycle time of 2539 s. Adding vehicle circulation raises this to 3390 s, and including passenger transfers further increases it to 3422 s—independent of the specific slack configuration ($L : 1.2\text{--}1.4, G : 1.15\text{--}1.25$). Notably, scenarios with tighter slack limits remain infeasible once all constraints are active.

The worst-case cycle time of 3422 s corresponds to a reserve of just 4.9%, which remains below the 7–10% operational buffer recommended in the literature (Goverde and Hansen, 2013). Although this reserve is slightly higher than in the MICRO model under the same scenario (approximately 3451 s), the difference between models is no longer consistent. The MACRO model achieves comparable—or occasionally better—cycle times in fully constrained settings when sufficient slack is available.

In summary, flexible routing is essential to ensure feasibility, while adding vehicle circulation and transfer constraints substantially increases the minimal cycle time.

4.6 Relationship Between Line Length and Minimal Cycle Ratio

A network's minimal cycle time can never be shorter than that of its individual lines. To supplement our network-wide assessment, we investigate the correlation between line length—characterized by the minimal allowed running time from terminus to terminus—and the minimal cycle ratio. Since individual lines typically lack transfer connections, we focus on three configurations: (1) flexible tracks without vehicle circulation, (2) fixed tracks without vehicle circulation, and (3) fixed tracks with vehicle circulation. Each line is evaluated under these configurations across varying slack levels. Figure 13 presents scatter plots for both the MICRO and MACRO models, including a linear trend line from ordinary least squares regression.

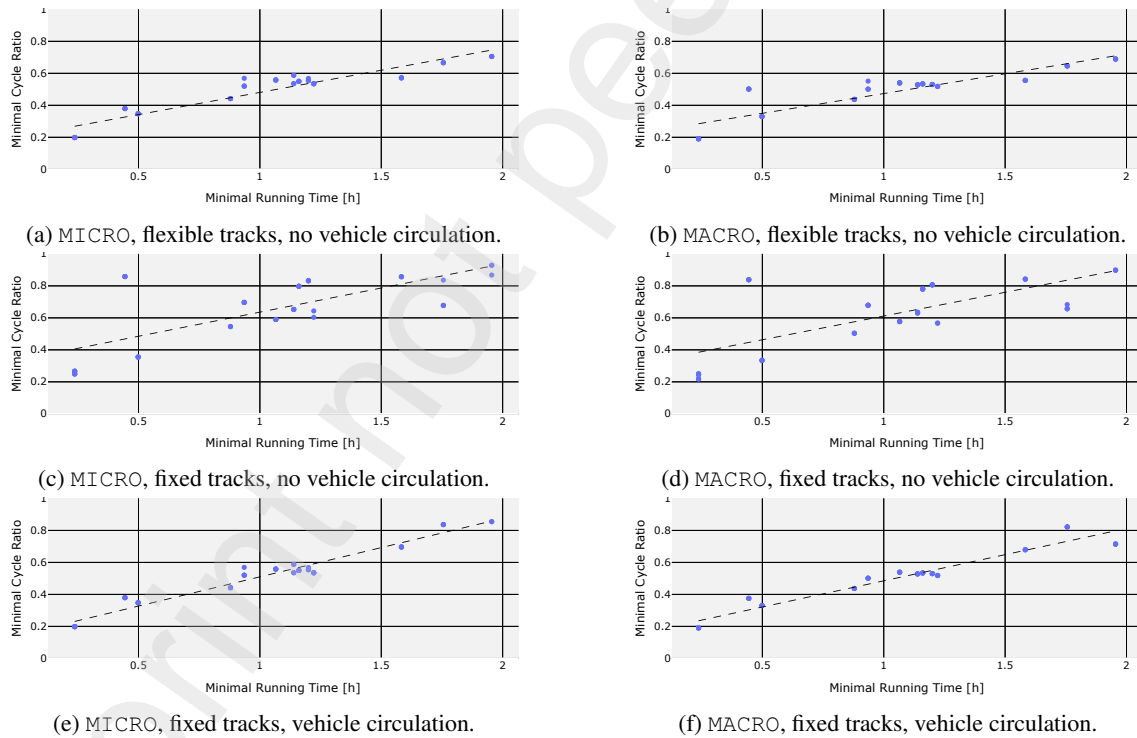


Figure 13: Scatterplots of the minimal cycle ratio (ω/T_0) vs minimal running time for each line by model and set of constraints.

Figure 13 shows a consistent trend across models and configurations: longer minimal running times are generally associated with higher minimal cycle ratios, indicating that long lines offer limited flexibility in reducing the cycle time. In configurations including vehicle circulation, lines with running times approaching 2 h tend to yield minimal cycle ratios close to one. This reflects a structural limitation in timetabling optimization, as these lines inherently restrict how low the cycle time can be pushed.

An additional observation concerns the fixed-track configurations (Figures 13c and 13d), which exhibit more erratic behavior around the trend line. This deviation likely results from constraints imposed by single-track sections in combination with fixed routes, limiting opportunities to optimize meeting locations and thus increasing variability in achievable cycle ratios.

4.7 Evaluation of Computation Time and Solution Method

We analyse the performance of the MIP and SMT-based solution approaches concerning their ability to compute tight lower and upper bounds on the minimal cycle time ω . Table 5 summarises results for four full-network configurations under the basic setting with flexible train routing. Each row corresponds to a different slack configuration, and we report lower bounds (LB), upper bounds (UB), optimality gap (GAP), and runtime for both MIP and SMT.

Configuration	Model	MIP				SMT			
		LB [s]	UB [s]	GAP [%]	Time [s]	LB [s]	UB [s]	GAP [%]	Time [s]
(L: 1.1, G: 1.10)	MICRO	2481.0	–	–	>20000	2571.7	2572.0	0.01	2379
	MACRO	2400.0	–	–	>20000	3600.0	inf	inf	13
(L: 1.2, G: 1.15)	MICRO	2481.0	–	–	>20000	2557.9	2559.0	0.04	17066
	MACRO	2539.0	–	–	>20000	2539.0	2539.0	0.00	1536
(L: 1.3, G: 1.20)	MICRO	2481.0	–	–	>20000	2557.9	2559.0	0.04	17816
	MACRO	2539.0	–	–	>20000	2539.0	2539.0	0.00	713
(L: 1.4, G: 1.25)	MICRO	2481.0	–	–	>20000	2557.9	2559.0	0.04	15994
	MACRO	2539.0	–	–	>20000	2539.0	2539.0	0.00	538

Table 5: Performance comparison of MIP and SMT methods across slack configurations: lower/upper bounds, optimality gap, and runtime.

Across all configurations, the plain MIP solver fails to provide feasible solutions within the 20,000-second time limit, while the SMT-based approach consistently delivers tight lower and upper bounds in a fraction of the time. These findings are consistent with prior work by [Sparing and Goverde \(2017\)](#), who also observed that solving such models with MIP alone often results in excessive runtimes and large optimality gaps. This further motivates using dedicated bounding procedures to improve performance before applying the MIP solver.

To evaluate runtime across a broader range of settings, we report in Figure 14 the computation time needed to solve each instance optimally. Results are grouped by model type and constraint configuration, covering different infrastructure partitions and feature combinations. The tags TR, PT, VC, and FT¹ indicate enabled constraints in each scenario.

¹TR: Train Routing, PT: Passenger Transfers, VC: Vehicle Circulation, FT: Fixed Tracks.

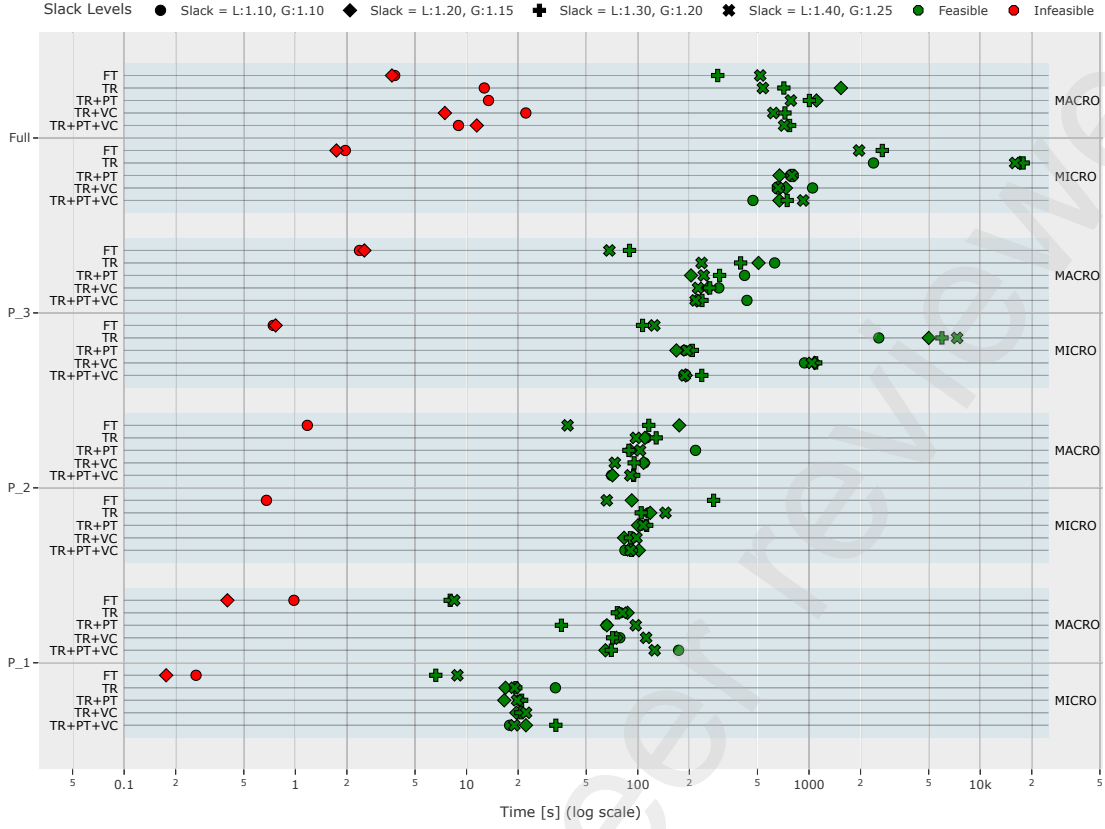


Figure 14: Computation time (log scale) to solve each configuration optimally, grouped by model and constraint combination.

Figure 14 shows that both models solve smaller instances (e.g., P_1) quickly, but runtime increases sharply with network size. Interestingly, MICRO often outperforms MACRO on small and mid-sized instances. However, for the full partition and larger configurations, MACRO becomes more efficient, if sufficient slack is available while otherwise it fails to find a stable solutions. Moreover, instances with additional constraints such as PT or VC tend to require less time, likely because the added structure reduces ambiguity in train scheduling decisions. Infeasible cases are typically detected early and solved much faster. All instances can be solved to optimality in less than five hours.

4.8 Comparison of Stability and Solution Transferability between MICRO and MACRO

To assess the relative stability of the two models, we compare the minimal cycle times ω achieved across all feasible configurations. Figure 15 presents the distribution of differences between MICRO and MACRO for each tested scenario.

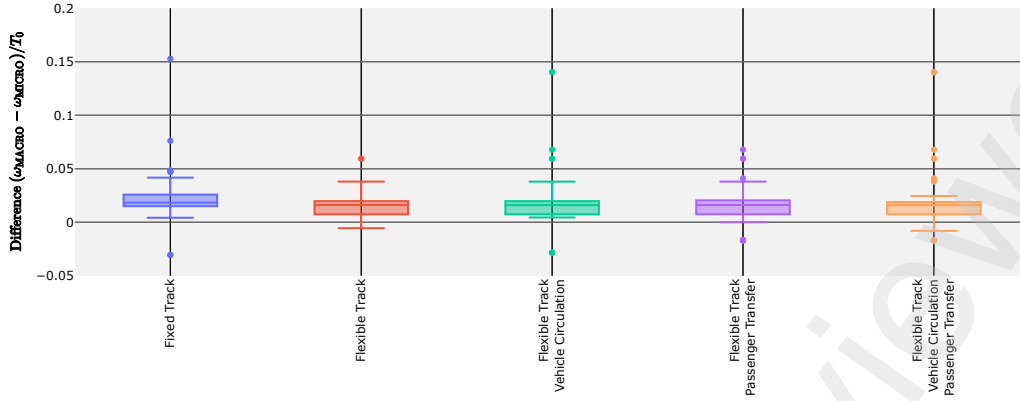


Figure 15: Boxplot of the difference between the minimal ω found by MICRO and MACRO by constraint selection across all slack levels and partitions ($n = 16$).

The results reveal a consistent trend: MICRO typically achieves lower or equal minimal cycle times compared to MACRO. This advantage is particularly pronounced in configurations with dense infrastructure or additional operational constraints, such as vehicle circulation or fixed tracks. The finer infrastructure resolution of MICRO enables it to exploit routing flexibility and manage conflicts more effectively, especially in single-track sections or complex interaction patterns.

In contrast, the aggregated nature of MACRO limits its ability to precisely model infrastructure bottlenecks and operational conflicts, resulting in more conservative cycle times. While both models follow similar trends with increasing slack, MACRO consistently yields higher cycle ratios. These findings indicate that MICRO offers greater robustness under tight operational conditions.

Beyond comparing solution quality, we also analyse whether solutions can be transferred between the two model types. Ideally, feasible solutions from one model could serve as valid or near-valid candidates in the other. However, this is not always possible due to structural differences, particularly in conflict modelling.

To examine this, we introduce a relaxed formulation with soft precedence constraints. Binary variables $\lambda_{pp'}$ indicate whether a conflict between train paths p and p' is violated. The model minimises the number of such violations while respecting all other timetable constraints.

We extend the MIP formulation by relaxing the disjunctive precedence constraints (15m) and (15n), and replacing the original objective (15a). For each potential conflict pair $\{p, p'\} \in \mathcal{K}$, we introduce a binary variable $\lambda_{pp'} \in \{0, 1\}$ that indicates whether a violation occurs. The modified constraints permit overlaps when $\lambda_{pp'} = 1$, and the objective counts the total number of such violations.

The following constraints define the relaxed model:

$$\min \sum_{\{p, p'\} \in \mathcal{K}} \lambda_{pp'}, \quad (20a)$$

$$t_{p'}^{\text{entry}} - t_p^{\text{exit}} + \delta \geq M(y_{pp'} - 1) + M \cdot \lambda_{pp'} \quad \forall \{p, p'\} \in \mathcal{K}, \quad (20b)$$

$$t_p^{\text{entry}} - t_{p'}^{\text{exit}} + \delta \geq M(y_{p'p} - 1) + M \cdot \lambda_{pp'} \quad \forall \{p, p'\} \in \mathcal{K}, \quad (20c)$$

$$\lambda_{pp'} \in \{0, 1\} \quad \forall \{p, p'\} \in \mathcal{K}. \quad (20d)$$

All other constraints remain as defined in the original model. This soft-conflict relaxation can be applied to both the VC-MicroRTP and VC-MacroRTP formulations. We evaluate how many precedence violations occur when importing a solution from one model into another. Results are shown in Figure 16.

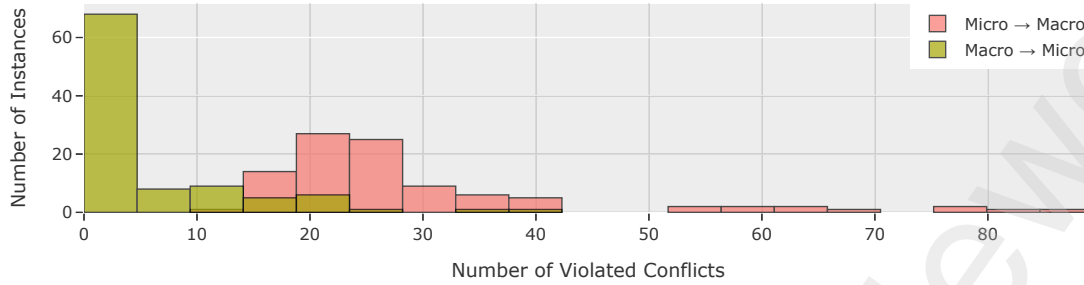


Figure 16: Histogram of the number of violated precedence constraints when transferring solutions between MICRO and MACRO using the soft conflict model.

The histogram reveals a distinct asymmetry: MACRO-to-MICRO transfers often yield feasible or near-feasible results, while the reverse is rarely the case. This outcome is expected. MACRO solutions, shaped by a coarser infrastructure representation, tend to respect broader safety margins. When mapped to MICRO, they can often be realised without modification. Conversely, MICRO solutions exploit detailed infrastructure features and tighter conflict modelling to minimise cycle time. When transferred to MACRO, these fine-grained timings often violate headway or routing assumptions, leading to infeasibility.

These findings underline the asymmetry in model fidelity and generalisability. MACRO can serve as a useful starting point for MICRO optimisation, but MICRO solutions are often too tightly optimised to be directly accepted in the aggregated MACRO model without adjustment.

5 Conclusion

This study introduced a novel microscopic timetabling model (MICRO) for optimizing periodic stability in complex railway networks alongside a macroscopic counterpart (MACRO) for comparison. Key methodological contributions include integrating detailed infrastructure elements, flexible routing, vehicle circulation, and passenger transfer constraints in MICRO and an SMT-based search technique for efficiently refining cycle time bounds. These advancements enable a more precise assessment of stability and computational efficiency in railway timetabling.

Our findings show that both models accommodate complex scheduling constraints but yield different minimal cycle times, with MICRO generally producing lower values due to its detailed infrastructure representation. Vehicle circulation constraints significantly increased cycle times by adding interdependencies that restrict flexibility, suggesting that flexible vehicle circulation strategies, as proposed by Lieshout (2021), could be beneficial for stability.

Flexible routing improved stability by reducing conflicts, though its impact was limited by the RhB network's predominantly single-track configuration and additional constraints. Individual line characteristics, such as long running times, also impose structural limits on network-wide cycle time reductions, highlighting the need to account for these factors.

While our model is based on real infrastructure and train schedules, it does not optimize for the same objectives as the current RhB timetable, making direct comparisons non-trivial. However, a qualitative analysis suggests that our optimized timetables exhibit a more structured periodic pattern, mainly due to the explicit enforcement of cyclic constraints. Differences between the micro- and macroscopic models are primarily observed in train sequencing and overtaking opportunities, as the microscopic model offers greater flexibility in route assignment.

The SMT-based search algorithm effectively refined cycle time bounds and improved solution times, particularly in complex scenarios. Unexpectedly, MACRO did not result in shorter computation times; instead, MICRO performed better, likely due to its more straightforward constraint handling.

While solutions from MACRO generally required significant adjustments to be applicable in MICRO, this underscores the potential benefits of selecting models that align with the level of infrastructure detail required. For railway operators, these findings offer hope that detailed infrastructure modelling and proper constraint handling can lead to stable, feasible schedules in complex networks.

Several additional aspects should be considered before transitioning our approach towards practical railway applications. First, an interactive decision-support system would allow railway planners to iteratively refine constraints, such as modifying transfer connections, enforcing mandatory stops, or adjusting infrastructure use dynamically. Second, integrating a more flexible vehicle circulation model would enable dynamic rolling stock allocation, improving operational feasibility. Finally, incorporating delay management mechanisms could enhance timetable robustness against real-world perturbations.

In conclusion, this study reiterates the importance of infrastructure representation and model choice in timetabling optimization. Both models offer valuable insights, and future work could refine the SMT-based search for scalability and explore model adjustments to improve solution transferability between detailed and aggregated infrastructure representations.

References

- Bergmann, D. R. (1975). Integer programming formulation for deriving minimum dispatch intervals on a guideway accommodating through and local public transportation services. *Transportation Planning and Technology*, 3(1):27–30.
- Bešinović, N., Goverde, R. M. P., Quaglietta, E., and Roberti, R. (2016). An integrated micro-macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87:14–32.
- Bortoletto, E., Lindner, N., and Masing, B. (2023). Periodic Timetabling with Cyclic Order Constraints. In Frigioni, D. and Schiewe, P., editors, *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023)*, volume 115 of *Open Access Series in Informatics (OASICS)*, pages 7:1–7:18, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- Bortoletto, E., van Lieshout, R. N., Masing, B., and Lindner, N. (2024). Periodic Event Scheduling with Flexible Infrastructure Assignment. In Bouman, P. C. and Kontogiannis, S. C., editors, *24th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2024)*, volume 123 of *Open Access Series in Informatics (OASICS)*, pages 4:1–4:18, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- Caimi, G., Fuchsberger, M., Laumanns, M., and Schüpbach, K. (2011). Periodic railway timetabling with event flexibility. *Networks*, 57(1):3–18.
- Caimi, G., Kroon, L., and Liebchen, C. (2017). Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4):285–312.
- Fuchs, F. and Corman, F. (2019). An open toolbox for integrated optimization of public transport. In *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 1–7.
- Fuchs, F., Trivella, A., and Corman, F. (2022). Enhancing the interaction of railway timetabling and line planning with infrastructure awareness. *Transportation Research Part C: Emerging Technologies*, 142:103805.
- Goerigk, M. (2015). Exact and heuristic approaches to the robust periodic event scheduling problem. *Public Transport*, 7(1):101–119.
- Goverde, R. M. P. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, 41(2):179–201.
- Goverde, R. M. P. and Hansen, I. A. (2013). Performance indicators for railway timetables. In *2013 IEEE International Conference on Intelligent Rail Transportation Proceedings*, pages 301–306.
- Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual.
- Heydar, M., Petering, M. E., and Bergmann, D. R. (2013). Mixed integer programming for minimizing the period of a cyclic railway timetable for a single track with two train types. *Computers & Industrial Engineering*, 66(1):171–185.
- Leutwiler, F. and Corman, F. (2022). A logic-based benders decomposition for microscopic railway timetable planning. *European Journal of Operational Research*, 303(2):525–540.
- Lieshout, R. N. V. (2021). Integrated periodic timetabling and vehicle circulation scheduling. *Transportation Science*, 55:768–790.
- Lusby, R. M., Larsen, J., and Bull, S. (2018). A survey on robustness in railway planning. *European Journal of Operational Research*, 266(1):1–15.
- Mannino, C. and Nakkerud, A. (2023). Optimal train rescheduling in oslo central station. *Omega*, 116:102796.
- Masing, B., Lindner, N., and Liebchen, C. (2023). Periodic timetabling with integrated track choice for railway construction sites. *Journal of Rail Transport Planning and Management*, 28:100416.
- Pachl, J. (2018). *Railway Operation and Control*. VTD Rail Publishing, Mountlake Terrace, USA, 4th edition.
- Peters, L. W. P. (2003). *Cyclic railway timetable optimization*, volume 22. Erasmus University Rotterdam, Rotterdam.
- Petering, M. E. H., Heydar, M., and Bergmann, D. R. (2016). Mixed-integer programming for railway capacity analysis and cyclic, combined train timetabling and platforming. *Transportation Science*, 50(3):892–909.
- Serafini, P. and Ukovich, W. (1989). A Mathematical Model for Periodic Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581.
- Sparing, D. and Goverde, R. M. P. (2017). A cycle time optimization model for generating stable periodic railway timetables. *Transportation Research Part B: Methodological*, 98(434):198–223.
- Wüst, R., Bütikofer, S., Ess, S., Gomez, C., Steiner, A., Laumanns, M., and Szabo, J. (2019). Periodic timetabling with ‘track choice’-

persp based on given line concepts and mesoscopic infrastructure. In *Operations Research Proceedings 2018*, pages 571–578. Springer.

Zhang, X. and Nie, L. (2016). Integrating capacity analysis with high-speed railway timetabling: A minimum cycle time calculation model with flexible overtaking constraints and intelligent enumeration. *Transportation Research Part C: Emerging Technologies*, 68:509–531.

Zhang, Y., Peng, Q., Yao, Y., Zhang, X., and Zhou, X. (2019). Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and Alternating Direction Method of Multipliers methods. *Transportation Research Part B: Methodological*, 128:344–379.