# Solving Integrated Periodic Railway Timetabling with Satisfiability Modulo Theories: A Scalable Approach to Routing and Vehicle Circulation

Florian Fuchs[1], Bernardo Martin-Iradi[1], Francesco Corman[1,*]

[1]Institute for Transport Planning and Systems (IVT), ETH Zürich, 8092 Zürich, Switzerland

**Abstract**

This paper introduces a novel approach for jointly solving the periodic Train Timetabling Problem (TTP), train routing, and Vehicle Circulation Problem (VCP) through a unified optimization model. While these planning stages are traditionally addressed sequentially, their interdependencies often lead to suboptimal vehicle usage. We propose the *VCR-PESP*, an integrated formulation that minimizes fleet size while ensuring feasible and infrastructure-compliant periodic timetables.

We present the first Satisfiability Modulo Theories (SMT)-based method for the VCR-PESP to solve the resulting large-scale instances. Unlike the Boolean Satisfiability Problem (SAT), which requires time discretisation, SMT supports continuous time via difference constraints, eliminating the trade-off between temporal precision and encoding size. Our approach avoids rounding artifacts and scales effectively, outperforming both SAT and Mixed Integer Program (MIP) models across non-trivial instances.

Using real-world data from the Swiss narrow-gauge operator RhB, we conduct extensive experiments to assess the impact of time discretisation, vehicle circulation strategies, route flexibility, and planning integration. We show that discrete models inflate vehicle requirements and that fully integrated solutions substantially reduce fleet needs compared to sequential approaches. Our framework consistently delivers high-resolution solutions with tractable runtimes, even in large and complex networks.

By combining modeling accuracy with scalable solver technology, this work establishes SMT as a powerful tool for integrated railway planning. It demonstrates how relaxing discretisation and solving across planning layers enables more efficient and implementable timetables.

**Keywords:** Timetabling; Routing; Vehicle Circulation; Satisfiability Modulo Theories; Transportation

*Corresponding author: francesco.corman@ivt.baug.ethz.ch

# 1. Introduction

Efficient public transportation planning is crucial for urban mobility, and railway systems are one of its core modes of transport. As depicted in Fig. F1, railway planning consists of strategic, tactical, and operational phases (Bussieck et al. 1997, Lusby et al. 2018). At the strategic level, network and line planning define the overall service structure, determining routes, stops, and long-term infrastructure development. Tactical planning involves timetable design, vehicle assignment, and workforce allocation, whereas operational planning focuses on real-time traffic management to mitigate disruptions and maintain service reliability. Vehicle circulation represents an intermediate planning stage between timetabling and detailed vehicle scheduling, where the focus is on determining the total number of vehicles needed rather than assigning specific rolling stock to individual trips.

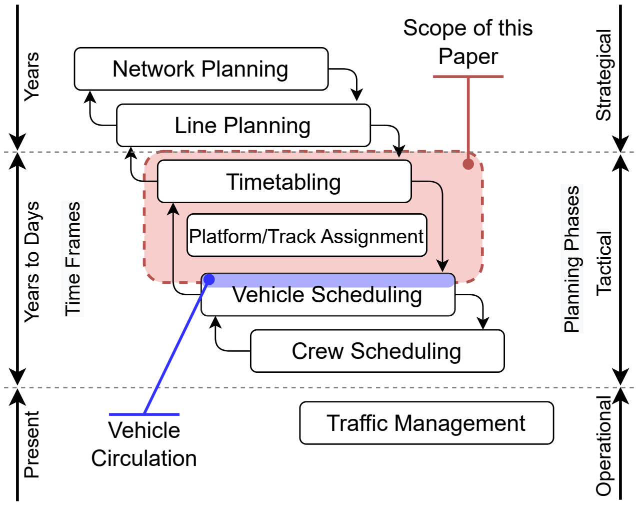



Figure F1: Illustration of railway planning stages, positioning vehicle circulation as an intermediate step between timetabling and vehicle scheduling.

Although planning these phases is traditionally handled sequentially, their interdependence suggests potential benefits from integrated optimization (Schiewe 2020). Joint optimization of timetabling and vehicle scheduling can reduce fleet requirements and improve resource utilization, highlighting the need for models that incorporate multiple planning aspects simultaneously. However, solving these integrated problems using traditional Mixed Integer Program (MIP) methods remains challenging due to computational complexity and achieving optimal solutions for large instances.

Recent studies (Borndörfer et al. 2020, Matos et al. 2020, Gattermann et al. 2016) have shown that solution methods based on the Boolean Satisfiability Problem (SAT) can outperform MIP solvers on planning problems such as the Train Timetabling Problem (TTP). Despite its advantages, SAT methods rely on the discretisation of time, which increases the size of the problem encoding as the time resolution becomes finer, posing scalability challenges. Satisfiability Modulo Theories (SMT) emerges as a promising alternative to address the scalability limitations. SMT extend SAT solvers by incorporating difference constraints, enabling the handling of continuous time without discretisation. This alternative allows for more scalable and precise solutions compared to traditional SAT-based methods.

This paper addresses all these limitations by, on one side, integrating timetabling, train routing, and vehicle circulation into a unified framework referred to as the Periodic Event Scheduling Problem with Vehicle Circulation and Routing (VCR-PESP). This breaks the sequential structure and focuses on minimizing the number of vehicles needed to operate a cyclic timetable. Conversely, we implement an SMT-based approach for solving the integrated problem at a large scale.

Key contributions of this research include:

- **Integrated problem formulation:** We introduce the VCR-PESP, a unified model integrating periodic timetabling, infrastructure-aware routing, and vehicle circulation within a single optimization framework.
- **Scalable solution approach:** We develop a novel SMT-based model that supports continuous time and avoids discretisation. Compared to traditional SAT and MIP methods, it achieves improved scalability and runtime performance.
- **Real-world computational study:** We apply our method to real-world data from the Swiss narrow-gauge operator RhB, demonstrating substantial fleet size reductions and efficient solver performance across different network and time resolutions.
- **Systematic evaluation of integration effects:** We quantify the impact of discretisation, routing flexibility, and vehicle sharing on vehicle requirements, illustrating the operational benefits of fully integrated planning approaches.

Our paper highlights the importance of integrating vehicle circulation into timetabling. It emphasizes the computational benefits of SMT, offering a framework for addressing the challenges of large-scale public transportation planning.

The paper is organised as follows: Section 2 provides a literature review covering timetabling, vehicle circulation, and train routing. Section 3 details the methodology, focusing on integrating these components using SMT. Section 4 presents and places the computational results in context. Finally, Section 5 summarises the findings and outlines directions for future research.

# 2. Literature Review

This section reviews the stages of railway system planning, focusing on integrating periodic timetabling and vehicle circulation. We discuss recent advancements in optimization techniques and identify challenges that this paper seeks to address.

## 2.1 Periodic Timetabling and Extensions, Including Routing

As mentioned in Section 1, railway planning is divided into stages, with different stages handled sequentially due to their complex nature. In particular, the TTP in the timetabling stage is a key planning problem, as it defines the arrival and departure times of all services and ensures that these do not conflict given the existing infrastructure and operational requirements.

A common approach for finding timetables for passenger trains in a network is to define a periodic timetable that repeats over a certain period. The *Periodic Event Scheduling Problem* (PESP), introduced by Serafini and Ukovich (1989), is the foundation of periodic timetabling. It ensures that train schedules repeat in cycles while satisfying operational constraints. The TTP framework has been widely applied in railway planning, balancing passenger demand, infrastructure capacity, and operational feasibility (Peeters 2003). While PESP provides a structured optimization model, its reliance on modular constraints imposes limitations when incorporating additional degrees of freedom, such as train routing and infrastructure constraints.

Several extensions have sought to enhance PESP by improving timetable flexibility. Studies by Gattermann et al. (2016) and Robenek et al. (2016a) introduce passenger route adjustments, while Liebchen (2004) investigate timetable symmetry constraints. More recent work has focused on integrating track choice into periodic timetabling. Wüst et al. (2019) extend PESP to incorporate flexible train routing decisions, allowing trains to adapt to different infrastructure configurations. Similarly, Masing et al. (2023) explore routing adaptability in railway construction settings.

(Bortoletto et al. 2023) introduce the Infrastructure-aware PESP, where the model is formulated using explicit track-based constraints, and the Flexible Infrastructure Assignment PESP (Bortoletto et al. 2024), generalizing infrastructure allocation across multiple configurations.

Periodic timetabling can also be modeled using alternative models to the PESP. Constraint-based formulations like the one presented in Heydar et al. (2013) incorporate multiple train types and explicitly minimize cycle time instead of relying on periodic constraints. Robenek et al. (2016b) consider a model between cyclic and acyclic timetabling and display the interest of keeping regularity for a fixed set of lines instead of all lines to solve the problem on a network level. The model is solved using a simulated annealing method and verified on the Israeli railway network. Martin-Iradi and Ropke (2022) use a time-space graph to formulate a macroscopic TTP with train routing and solve it using a column-generation-based matheuristic.

### 2.2 Integration of Vehicle Circulation into Timetabling

Vehicle circulation is critical in railway operations (Borndörfer et al. 2018), as it determines how rolling stock is allocated across scheduled services. Traditional approaches treat vehicle circulation as a separate problem, first establishing a timetable and then assigning vehicles, often resulting in inefficient fleet usage (Goossens et al. 2006). More recent research has demonstrated that integrating vehicle circulation into TTP can lead to reductions in fleet size while maintaining timetable feasibility (Lieshout 2021). Optimization models can minimize turnaround times and dead-heading by considering rolling stock constraints at the timetabling stage, improving overall vehicle utilisation.

Examples beyond railway include integrated models for bus operations that optimize fleet costs or passenger transfers, such as Ibarra-Rojas et al. (2014), Fonseca et al. (2018), and Schmid and Ehmke (2015). While these approaches explore valuable trade-offs between vehicle usage, robustness, and transfers, they typically assume fixed routing or ignore infrastructure constraints, making them less applicable to railway models.

MIP formulations are widely used for periodic timetabling and vehicle circulation problems due to their flexibility in constraint modeling (Goerigk and Liebchen 2017, Herrigel et al. 2018). However, due to their inherent complexity, MIP models become computationally intractable for large-scale instances.

Even without additional constraints such as routing or rolling stock optimization, PESP is known to be NP-hard (Peeters 2003). This computational intractability has motivated the development of new solution techniques, relying on Boolean Satisfiability Problem (SAT).

### 2.3 SAT-based Methods in Transport Planning

Recent studies have shown that solving the TTP without VCP using Boolean Satisfiability Problem (SAT)-based solvers outperforms commercial ones for MIPs, especially when the objective is to find feasible solutions (Großmann et al. 2012, Großmann 2016, Kümmling et al. 2015, Fuchs et al. 2022). Borndörfer et al. (2020) propose a concurrent solver for the TTP that integrates SAT, MIP, and domain-specific heuristics, demonstrating the benefit of combining logical and mathematical programming approaches for solving PESP instances.

SAT-based methods encode scheduling constraints as logical formulas and leverage efficient SAT solvers to determine feasible solutions. However, SAT formulations rely on time discretisation, which increases encoding size and computational effort as resolution granularity improves (Großmann 2011). While this can be mitigated through coarse time steps, such approximations may lead to suboptimal or infeasible solutions in high-resolution timetables.

Previous extensions of SAT include the work by Gattermann et al. (2016), who augment SAT-based timetabling by including passenger routing preferences in the form of soft constraints, resulting in a MaxSAT formulation. Matos et al. (2020) further enhance this approach by introducing reinforcement learning to guide the search in MaxSAT, improving performance on public benchmark instances. While these methods demonstrate the adaptability of logic-based frameworks to include user-centric objectives, they do not yet support integration with routing and vehicle constraints, which introduces additional modeling and computational challenges.

These developments illustrate a shift from classical SAT solving toward hybrid and learning-based strategies, highlighting the potential of logic-oriented approaches to model passenger-centric objectives. However, such approaches have not yet been extended to integrate routing or vehicle constraints, which introduces additional structural complexity.

### 2.4 Research Gap

While prior research has addressed periodic timetabling extensions, routing flexibility, and vehicle circulation separately, their joint optimization remains an open challenge. Most existing models treat routing independently of rolling stock constraints or optimize vehicle circulation, assuming fixed train paths (Caimi et al. 2017).

Moreover, although MIP and SAT-based methods have successfully solved isolated components of the problem, they struggle with scalability when considering large-scale instances with high-resolution schedules. In particular, SAT-based models require time discretisation, which increases encoding size and computational complexity as resolution improves (Großmann 2011). While coarse time steps can reduce problem size, they may lead to suboptimal or infeasible solutions.

This paper formulates an integrated optimization model that addresses these challenges. The model jointly considers TTP, routing flexibility, and vehicle circulation at a mesoscopic resolution—offering more detail than macroscopic models while remaining computationally tractable. We leverage Satisfiability Modulo Theories (SMT) as a scalable solution framework. SMT extends SAT by incorporating difference-logic constraints, allowing continuous-time formulations without discretisation (Armando et al. 2004, Leutwiler and Corman 2022). This extension enables efficient solving of integrated problems while maintaining high temporal precision.

## 3. Methodology

This section presents the methodology for modeling and optimizing train routes, event timings, train sequencing, and vehicle transitions within a periodic timetable. We introduce the Event Activity Network (EAN), the foundational structure for modeling events, activities, and interdependencies. The EAN provides a structured representation of the timetabling problem, linking routing and scheduling decisions.

Building on the EAN, we describe three distinct approaches to solving the problem: a MIP formulation, an SMT encoding, and a SAT-based version. Each approach builds on the EAN with selectable activities, enabling us to solve and compare the three.

## 3.1 Introducing the Event Activity Network (EAN)

The EAN is a standard data structure to model periodic timetabling problems. It consists of a set of nodes $\mathcal{E}$ representing *events* (e.g., train arrivals or departures) and a set of edges $\mathcal{A}$ representing *activities* that capture constraints between these events (Liebchen and Möhring 2004). While activities are traditionally binding and consistently enforced, enabling routing flexibility or optional passenger and vehicle transfers requires defining some activities as *selectable*—their constraints apply only if specific conditions are met. This concept supports modeling infrastructure-dependent routing (Fuchs et al. 2022) and conditional transfers (Kroon et al. 2014), and is essential for integrated optimization across planning stages.

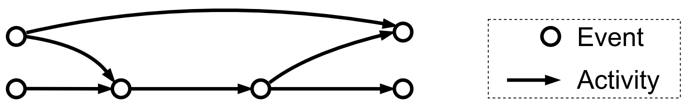



Figure F2: Example EAN with six events and six activities.

The EAN aims to assign a timestamp $t_e \in [0, T)$ to each event $e \in \mathcal{E}$, where $T$ is the period of the timetable. For each activity $a \in \mathcal{A}$, there is a duration $\delta_a$ bounded by a lower limit $\delta_a^{\min}$ and an upper limit $\delta_a^{\max}$, such that $\delta_a^{\min} \leqslant \delta_a \leqslant \delta_a^{\max}$. The duration $\delta_a$ is derived from the scheduled times of the two connected events. If $i$ and $j$ denote the origin and destination events of activity $a = (i, j)$, then the duration is calculated as:

$$\delta_a = t_j - t_i + k_a \cdot T, \quad \forall a = (i, j) \in \mathcal{A} \tag{1}$$

where $k_a \in \mathbb{Z}$ adjusts for cases where $t_i > t_j$ due to train schedules being longer than the cycle period.

The EAN includes three types of events: *arrival*, where trains arrive at a stopping stop, *departure*, where trains depart from a stopping stop, and *passing*, where a train traverses a given location. It also supports six distinct activity types:

1. *Trip activities* for train movements between stations,
2. *Dwell activities* for stopping at stations,
3. *Headway activities* to ensure safe and feasible separation of trains using shared infrastructure,

4. *Regularity activities* to enforce even spacing for high-frequency services,

5. *Commercial activities* to meet service requirements such as maximum time between commercial stops,

6. *Vehicle-transfer activities* to model vehicle handovers between terminating and originating trains at termini (Kroon et al. 2014, Lieshout 2021).

These activities are explained in detail in the remainder of this section.

## 3.2 Including Train Routing

To optimize train routes, we extend the EAN to include multiple routing options per train service, following the approach of Fuchs et al. (2022). Routes consist of *dwell* and *trip* activities, which are modeled within the EAN. An example of itinerary activities for a single train is depicted in Fig. F3.

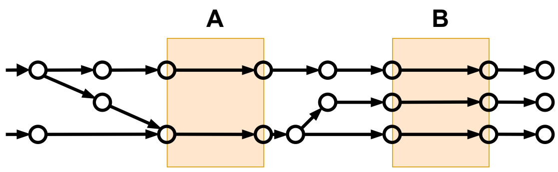



Figure F3: Commercial and itinerary constraints for an example train visiting two stops, A and B, modeled as activities in the EAN.

To construct this extended EAN, we begin with a line plan specifying the stations each train serves. Each train's infrastructure and routing options are derived from this station sequence, allowing the EAN to represent the train's feasible movements through the network. The resulting network includes one weakly connected component for each train, which consists of only *dwell* and *trip* activities. These are organized into a Train Flow Network (TFN), a graph with node set $\mathcal{V}$ and arc set $\mathcal{W}$ representing itinerary options. An example TFN is shown in Fig. F4.

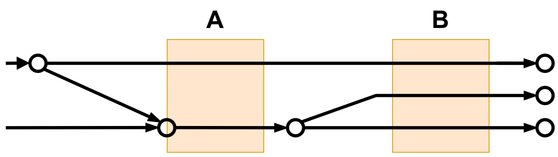


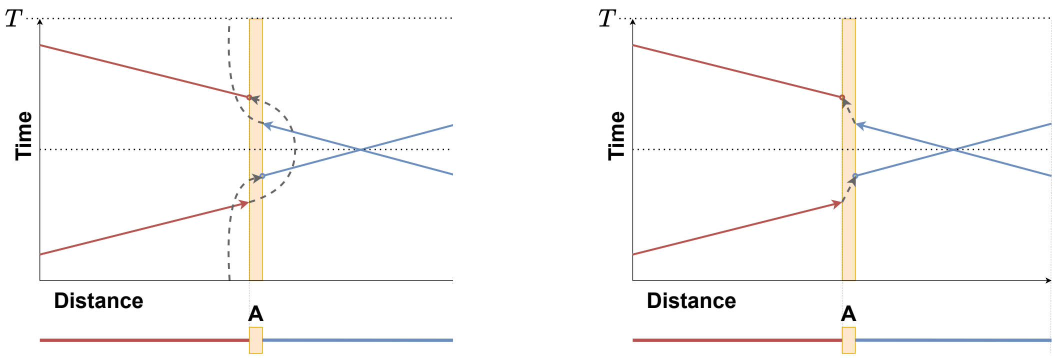

Figure F4: A TFN example for a train service with five possible paths.

The sets $\mathcal{V}^{\text{Source}}$ and $\mathcal{V}^{\text{Sink}}$ denote artificial source and sink nodes for the set of train services to schedule, and are a subset of $\mathcal{V}$. The TFN (as well as the EAN) is a directed and acyclic graph, meaning that a path from source to sink will define a valid route for a train service. To compute the chosen route, we define binary variables $x_v \in \{0,1\}$ to indicate whether a node

$v \in \mathcal{V}$ is visited, and $x_w \in \{0,1\}$ to indicate whether an arc $w \in \mathcal{W}$ is selected. As the TFN is a reduced version of the EAN, each event $e \in \mathcal{E}$ and activity $a \in \mathcal{A}$ can be mapped to the related nodes and links in the TFN. Once the train routes are selected, the EAN is completed by incorporating all remaining activities, such as *headway* and *vehicle-transfer* activities. This completion ensures that the final model respects operational and commercial constraints while preventing conflicts.

### 3.3 Adding Vehicle Circulation

Optimized vehicle circulation ensures the efficient use of rolling stock. To model vehicle movements effectively, we extend the EAN to include vehicle circulation links at termini. These links represent the transfer of vehicles between train services at their origin and destination stations. By modeling these links as activities within the EAN, we can seamlessly apply the same framework for routing trains to vehicle circulation.




(a) Vehicle circulation restricted to individual lines.

(b) Vehicles shared across lines.

Figure F5: Illustration of vehicle circulation strategies using two lines (i.e., red and blue): per-line circulation (a) and shared vehicle circulation (b).

Figure F5 illustrates the two primary vehicle circulation strategies. In the *per-line* strategy (Fig. F5a), vehicles are restricted to operating within the same line, effectively isolating vehicle pools for each train line. In the *shared circulation* strategy (Fig. F5b), vehicles can transfer freely between lines, enabling a more flexible allocation of rolling stock. The choice of strategy may have significant implications for vehicle requirements and operational flexibility, as suggested by the example in Fig. F5 and demonstrated in our experimental results.

For each terminating train service, the model considers all possible links that transfer vehicles to originating train services. These links are represented as binary variables $x_u \in \{0,1\}$, where $u \in \mathcal{U}$ denotes a vehicle circulation link. To ensure a feasible matching between train arrivals and departures, we impose the following constraints:

$$\sum_{u \in \mathcal{U}_e^{\text{Out}}} x_u = 1, \quad \forall e \in \mathcal{E}^{\text{Start}}, \tag{2}$$

$$\sum_{u \in \mathcal{U}_e^{\text{In}}} x_u = 1, \quad \forall e \in \mathcal{E}^{\text{End}}, \tag{3}$$

where the events of all originating and terminating train services are denoted by the sets $\mathcal{E}^{\text{Start}}$ and $\mathcal{E}^{\text{End}}$, respectively, $\mathcal{U}_e^{\text{Out}}$ is the set of outgoing circulation links associated with event $e$, and $\mathcal{U}_e^{\text{In}}$ is the set of incoming circulation links. These constraints ensure that each arrival event at a terminus matches precisely one departure event, forming a valid vehicle transfer. Due to the cyclic characteristic of the timetable, we do not need to account for the start and end of vehicle operations.

## 3.4 Defining Selectable Activities

We follow the concept of *selectable* and *non-selectable* activities introduced by Fuchs et al. (2022), and we extend this approach by treating all activities $a \in \mathcal{A}$ as *selectable*. A *selectable* activity is one whose time constraints $\delta_a \in [\delta_a^{\min}, \delta_a^{\max}]$ are enforced only if associated routing or circulation decisions activate the activity. Otherwise, the activity remains inactive and its duration unconstrained (i.e., $t_a \in \mathbb{R}$). As mentioned in Section 3.2, each activity $a \in \mathcal{A}$ has associated elements in the TFN, which group together a set of itinerary nodes $v \in \mathcal{V}_a \subseteq \mathcal{V}$, links $w \in \mathcal{W}_a \subseteq \mathcal{W}$, or vehicle transfers $u \in \mathcal{U}_a \subseteq \mathcal{U}$, and the activation of the activity depends on the utilization of such elements.

To formalize this relationship, for each activity $a \in \mathcal{A}$, we define its relevant components $\Omega_a$ as the union of the sets $\mathcal{V}_a$, $\mathcal{W}_a$, and $\mathcal{U}_a$:

$$\Omega_a = \mathcal{V}_a \cup \mathcal{W}_a \cup \mathcal{U}_a. \tag{4}$$

The sets $\mathcal{V}_a$, $\mathcal{W}_a$, and $\mathcal{U}_a$ are activity-specific. For example:

- A *trip* activity depends solely on the corresponding infrastructure link $w \in \mathcal{W}$, with $|\mathcal{U}_a| = 0$, $|\mathcal{V}_a| = 0$, and $|\mathcal{W}_a| = 1$.
- A *headway* activity depends on two infrastructure nodes $v \in \mathcal{V}$ traversed by the trains, with $|\mathcal{U}_a| = 0$, $|\mathcal{V}_a| = 2$, and $|\mathcal{W}_a| = 0$.
- A *vehicle-transfer* activity depends on a single vehicle transfer, with $|\mathcal{U}_a| = 1$, $|\mathcal{V}_a| = 0$, and $|\mathcal{W}_a| = 0$.

The duration $\delta_a$ of an activity $a \in \mathcal{A}$, is by default constrained by the time bounds $\delta_a^{\min}$ and

$\delta_a^{\max}$ if all associated components $\omega \in \Omega_a$ are active. If at least one component is not selected ($x_\omega = 0$), these bounds are relaxed using large constants $B_a^{\min}$ and $B_a^{\max}$:

$$\delta_a \geqslant \delta_a^{\min} - (|\Omega_a| - \sum_{\omega \in \Omega_a} x_\omega) \cdot B_a^{\min}, \quad \forall a \in \mathcal{A}, \tag{5}$$

$$\delta_a \leqslant \delta_a^{\max} + (|\Omega_a| - \sum_{\omega \in \Omega_a} x_\omega) \cdot B_a^{\max}, \quad \forall a \in \mathcal{A}. \tag{6}$$

These equations ensure that the bounds on $\delta_a$ are only enforced when the activity is active, while inactive activities are effectively unconstrained.

This unified treatment of selectable activities eliminates the need to explicitly distinguish between *selectable* and *non-selectable* activities, as all activities are inherently treated as selectable. For example, independent of routing decisions, *commercial* activities are modeled with $|\Omega_a| = 0$. Consequently, their time bounds must always be respected. For notation simplicity, we unify the definition of our selection variable $x_\omega$ for each element $\omega \in \Omega_a$ and activity $a \in \mathcal{A}$.

By linking the activation of activities to the TFN, we effectively reduce redundancy in the model. For instance, headway activities are automatically deactivated if only one train uses the relevant infrastructure, and redundant vehicle-transfer options are excluded based on the chosen routes. This approach prevents over-constraining the problem, ensuring feasibility while maintaining flexibility for optimization.

### 3.5 MIP Formulation

As a final step before formulating the MIP, we define the objective function, which minimizes the number of vehicles required to operate a periodic timetable. In the VCR-PESP, each train service must be part of a feasible vehicle sequence that loops back periodically, forming a closed path along a cycle in the EAN. These vehicle cycles consist of alternating commercial activities (i.e., scheduled train trips) and vehicle-transfer activities (i.e., transitions at terminus stations).

Let $\mathcal{S} \subset \mathcal{A}$ be the set of commercial activities and $\mathcal{U} \subset \mathcal{A}$ the set of selected vehicle-transfer activities. When a valid timetable is constructed, each vehicle must follow a path through a sequence of activities in $\mathcal{S} \cup \mathcal{U}$ that forms a cycle of total duration equal to an integer multiple of the period $T$. This property reflects the cyclic nature of the timetable, where vehicles repeat the same circulation pattern every period.

An illustrative example is shown in Fig. F6, highlighting two such vehicle cycles. Each cycle consists of alternating activities with duration $\delta_a = t_j - t_i + k_a T$. Because each event appears exactly once as a predecessor and once as a successor in the cycle, all $t_j - t_i$ terms cancel when summing over the cycle. The total duration of the cycle reduces to $\sum_{a \in \text{cycle}} k_a \cdot T$, and the number of vehicles needed to operate the cycle is thus equal to $\sum_{a \in \text{cycle}} k_a$.

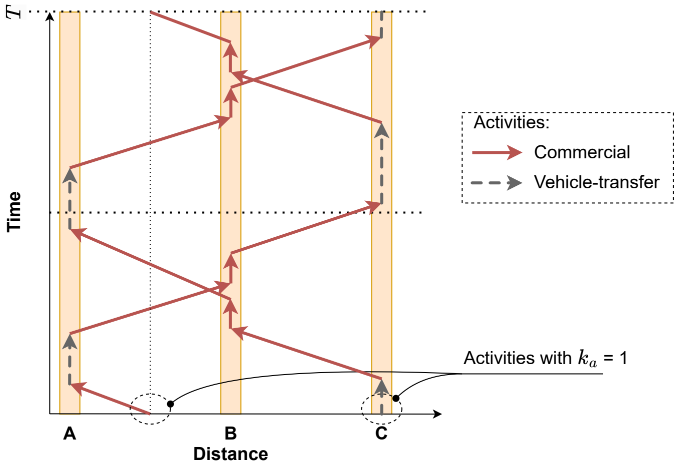



Figure F6: Example illustration of two cycles in which vehicles circulate. Each cycle has a total length of $T$. Since there are two such cycles, the total vehicle requirement is 2.

Consequently, the total number of vehicles required across all cycles in the solution is given by:

$$\sum_{a \in \mathcal{S} \cup \mathcal{U}} k_a$$

This sum reflects the objective function of the VCR-PESP model and is minimized to obtain the most efficient vehicle circulation.

Table T1: Consolidated Variable and Set Definitions for VCR-PESP

| Symbol | Type | Definition |
|---|---|---|
| $\mathcal{V}$ | Set | Set of itinerary nodes in the TFN. |
| $\mathcal{V}^{\text{Source}}$, $\mathcal{V}^{\text{Sink}}$ | Set | Set of artificial source and sink nodes for train services in the TFN. |
| $\mathcal{W}$ | Set | Set of itinerary links in the TFN |
| $\mathcal{U}$ | Set | Set of vehicle circulation links. |
| $\mathcal{U}_e^{\text{In}}$,$\mathcal{U}_e^{\text{Out}}$, | Set | Set of outgoing (resp. incoming) circulation links associated with event $e \in \mathcal{E}^{\text{End}}$ (resp. $e \in \mathcal{E}^{\text{Start}}$). |
| $\mathcal{S}$ | Set | Set of commercial activities of train services. |
| $\mathcal{E}$ | Set | Set of events (i.e., arrival, departure, passing) in the EAN. |
| $\mathcal{E}^{\text{Start}}$, $\mathcal{E}^{\text{End}}$ | Set | Set of originating and terminating train service events. |
| $\mathcal{A}$ | Set | Set of activities (e.g., trip, dwell, headway, regularity, commercial, vehicle-transfer) in the EAN. |
| $\mathcal{V}_a$, $\mathcal{W}_a$,$\mathcal{U}_a$ | Set | Sets of itinerary nodes and links, and vehicle circulation links associated with activity $a \in \mathcal{A}$. |
| $\Omega_a$ | Set | Union of sets $\mathcal{V}_a, \mathcal{W}_a, \mathcal{U}_a$ for activity $a \in \mathcal{A}$. |
| $\alpha^+(v), \alpha^-(v)$ | Set | Set of outgoing (resp. incoming) nodes from (resp. to) $v \in \mathcal{V}$ directly connected by a link in $\mathcal{W}$. |
| $T$ | Parameter | Period duration. |
| $\delta_a^{\min}$ | Parameter | Lower bound for duration of activity $a \in \mathcal{A}$. |
| $\delta_a^{\max}$ | Parameter | Upper bound for duration of activity $a \in \mathcal{A}$. |
| $B_a^{\min}$, $B_a^{\max}$ | Parameter | Bounds for relaxed duration of non-selected activity $a \in \mathcal{A}$. |
| $x_\omega$ | Variable | Binary variable for selection of element $\omega \in \Omega_a$ of activity $a \in \mathcal{A}$. |
| $t_e$ | Variable | Timestamp of event $e \in \mathcal{E}$ |
| $\delta_a$ | Variable | Duration of activity $a \in \mathcal{A}$ |
| $k_a$ | Variable | Period adjustment for activity $a \in \mathcal{A}$ |

We present in (7) the formulation of the VCR-PESP. The notation is summarized in Table T1

$$\min \sum_{a \in \mathcal{S} \cup \mathcal{U}} k_a \tag{7a}$$

$$\text{subject to: } \delta_a \geqslant \delta_a^{\min} - (|\mathcal{V}_a| - \sum_{v \in \mathcal{V}_a} x_v + |\mathcal{W}_a| - \sum_{w \in \mathcal{W}_a} x_w) \cdot B_a^{\min}, \qquad \forall a \in \mathcal{A}, \tag{7b}$$

$$\delta_a \leqslant \delta_a^{\max} + (|\mathcal{V}_a| - \sum_{v \in \mathcal{V}_a} x_v + |\mathcal{W}_a| - \sum_{w \in \mathcal{W}_a} x_w) \cdot B_a^{\max} \qquad \forall a \in \mathcal{A}, \tag{7c}$$

$$\delta_a = (t_j - t_i) + k_a \cdot T, \qquad \forall a = (i,j) \in \mathcal{A}, \tag{7d}$$

$$\sum_{w \in \alpha^+(v)} x_w = \sum_{w \in \alpha^-(v)} x_w = x_v, \qquad \forall v \in \mathcal{V}, \tag{7e}$$

$$\sum_{w \in \alpha^+(v)} x_w = 1, \qquad \forall v \in \mathcal{V}^{\text{Source}} \tag{7f}$$

$$\sum_{u \in \mathcal{U}_e^{\text{Out}}} x_u = 1, \qquad \forall e \in \mathcal{E}^{\text{Start}}, \tag{7g}$$

$$\sum_{u \in \mathcal{U}_e^{\text{In}}} x_u = 1, \qquad \forall e \in \mathcal{E}^{\text{End}}, \tag{7h}$$

$$\text{variables: } t_e \in [0, T), \qquad \forall e \in \mathcal{E}, \tag{7i}$$

$$\delta_a \geqslant 0, \qquad \forall a \in \mathcal{A}, \tag{7j}$$

$$x_\omega \in \{0, 1\}, \qquad \forall \omega \in \Omega_a, \forall a \in \mathcal{A}, \tag{7k}$$

$$k_a \in \mathbb{Z}, \qquad \forall a \in \mathcal{A} \tag{7l}$$

The objective function in (7a) minimizes the vehicle count, as suggested by Lieshout (2021). Constraints (7b) and (7c) ensure both the correct activation of elements and the duration window of activities. The PESP constraint, given in (7d), establishes a direct relationship between the duration of an activity and the timing of its associated events. Constraint (7e) guarantees flow conservation in the TFN, and together with Constraint (7f), it enforces that exactly one route is chosen for each train service, thus requiring a unique path for each train within the timetable. Vehicle circulation is managed through constraints (7g) and (7h). These ensure that for each event that marks the start or end of a train's journey, precisely one vehicle circulation activity is chosen, as proposed by Lieshout (2021). The nature of the decision variables used in the model is defined in (7i), (7j), (7k), and (7l).

### 3.6 Transformation to SMT

The VCR-PESP can quickly become intractable for state-of-the-art commercial solvers. To solve (7), we present an alternative approach using Satisfiability Modulo Theories (SMT), which integrates Boolean Satisfiability Problem (SAT) with difference constraints. This approach allows us to encode the model formulation (7) into a combination of Boolean formulas and arithmetic constraints. Specifically, the Boolean formula is expressed in its Conjunctive Normal Form (CNF) and consists of Boolean variables $q \in \mathcal{Q}$, which form clauses composed of literals. Each clause is a disjunction of literals, where each literal represents either a variable or its negation, ensuring that at least one of the included variables is assigned the required value for the clause to hold. All clauses in the CNF must be satisfied in a valid assignment to the variables. This structure allows efficient constraint propagation and satisfiability checking (Biere et al. 2009).

The arithmetic component of the SMT formulation consists of difference constraints, which enforce relationships between integer variables. For a given activity $a = (i, j)$ these constraints take the following form:

$$t_i - t_j \geqslant \delta_a \vee \neg q, \tag{8}$$

Where $t_i$ and $t_j$ are event times, $\delta_a$ is a minimum required separation, and $q$ is a Boolean variable controlling whether the constraint must hold. This encoding allows conditional constraints, meaning that if $q$ is set to `False`, the constraint is effectively deactivated. In contrast, the Boolean term $\vee\neg q$ is omitted for mandatory constraints that must always be respected. For further background on the extension of SAT with difference constraints, we refer the reader to Armando et al. (2004).

Since the SMT encoding requires that all variables involved in Boolean operations are binary, we preprocess the input EAN to ensure that all activity durations $\delta_a$ fall within the range $[0, T]$, thus converting all periodicity variables $k_a$ into binary representations. To achieve compliance with this condition, we employ the method proposed by Peeters (2003). This transformation splits any activity with a non-binary $k_a$ into two or more activities, ensuring that each resulting activity can be represented with a binary $k_a$.

### 3.6.1 Boolean Encoding of the Train Flow Network (TFN)

Next, we describe how to encode the problem in SMT. Initially, we define some helper functions before delving into encoding the MIP as outlined in Eq. (7).

$$\texttt{encode-at-most-one}(\mathcal{Q}) := \bigwedge_{\forall q_i, q_j \in \mathcal{Q},\ i<j} (\neg q_i \vee \neg q_j) \tag{9a}$$

$$\texttt{encode-at-least-one}(\mathcal{Q}) := \bigvee_{q \in \mathcal{Q}} q \tag{9b}$$

$$\begin{aligned} \texttt{encode-exactly-one}(\mathcal{Q}) := \quad & \texttt{encode-at-least-one}(\mathcal{Q}) \\ & \wedge \texttt{encode-at-most-one}(\mathcal{Q}) \end{aligned} \tag{9c}$$

The `encode-at-most-one` function (Eq. (9a)) ensures that no more than one variable in a set is assigned the value `True`. Conversely, `encode-at-least-one` (Eq. (9b)) ensures at least one variable in a set is assigned the value `True`. Combining these two functions, `encode-exactly-one`

(Eq. (9c)) guarantees that exactly one variable in a set is assigned the value `True`.

Next, we focus on encoding the TFN and vehicle circulation cardinality constraints. This encoding step involves defining a SAT variable $q \in \mathcal{Q}$ for each $v \in \mathcal{V}$ node, $w \in \mathcal{W}$ link, and $u \in \mathcal{U}$ vehicle circulation link in the TFN.

$$\begin{aligned}\texttt{encode-TFN }(\mathcal{V}) := \bigwedge_{\forall v \in \mathcal{V}} (\neg q_v \bigvee_{y \in \alpha^+(v)} (q_y \wedge \neg q_v) \bigvee_{y \in \alpha^-(v)} q_y \\ \wedge\, \texttt{encode-at-most-one}(\mathcal{Q}_{(\alpha^+(v))}) \\ \wedge\, \texttt{encode-at-most-one}(\mathcal{Q}_{(\alpha^-(v))})) \quad (10a)\end{aligned}$$

$$\texttt{encode-one-train-path} := \bigwedge_{\forall c \in \mathcal{C}} \texttt{encode-exactly-one}(q_v : v \in \mathcal{V}_c) \quad (10b)$$

To encode the network, we can use the same approach as given by Fuchs et al. (2022), which encodes flow balance in Eq. (10a) and then requires one path per train in Eq. (10b).

### 3.6.2 Encoding of the Event Activity Network (EAN)

Having encoded the TFN, we now turn our attention to encoding the EAN for the SMT equivalent of model (7). We begin by defining SAT variables $\mathcal{Q}$ and time variables. Each event $e \in \mathcal{E}$ is associated with a positive time variable $t_e$.

$$\texttt{encode-event}(\mathcal{E}) := \bigwedge_{\forall e \in \mathcal{E}} (t_e - t_0 \geqslant 0) \wedge (t_0 - t_e \geqslant -T) \quad (11a)$$

Each activity $a = (i, j) \in \mathcal{A}$ links two events and has an associated period offset $k_a \in \{0, 1\}$, due to the preprocessing step described in the MIP formulation in Section 3.5. This binary nature allows us to represent each activity using exactly two precedence directions: either $t_j \geqslant t_i$, corresponding to $k_a = 0$, or $t_j < t_i$, which implies $k_a = 1$. To encode this behavior, we introduce two Boolean variables for each activity: $q_a$ to represent the case where $t_j \geqslant t_i$, and $\hat{q}_a$ to describe the inverted precedence $t_i > t_j$. These two cases are mutually exclusive and collectively exhaustive, ensuring that precisely one is active when the activity is selected. The corresponding difference constraints are then conditionally enforced using the variables $q_a$ and $\hat{q}_a$, as shown in Eqs. (12a) and (12b). Finally, Eq. (12c) ensures that precisely one of the two precedence directions is activated whenever the activity is selected (i.e., when all elements in $\Omega_a$ are active).

$$\texttt{encode-minimal-delta}(\mathcal{A}) := \bigwedge_{\forall a \in \mathcal{A}} ((t_j - t_i \geqslant \delta_a^{\min}) \vee \neg q_a) \wedge ((t_i - t_j \leqslant (T - \delta_a^{\min})) \vee \neg \hat{q}_a) \tag{12a}$$

$$\texttt{encode-maximal-delta}(\mathcal{A}) := \bigwedge_{\forall a \in \mathcal{A}} ((t_i - t_j \geqslant -\delta_a^{\max}) \vee \neg q_a) \wedge ((t_j - t_i \leqslant -(T - \delta_a^{\max})) \vee \neg \hat{q}_a) \tag{12b}$$

$$\texttt{activate-precedence}(\mathcal{A}) := \bigwedge_{\forall a \in \mathcal{A}} ((q_a \vee \hat{q}_a) \vee \bigwedge_{\omega \in \Omega_a} \neg q_\omega) \tag{12c}$$

### 3.6.3 Encoding the Vehicle Circulation

After encoding the TFN and EAN, we need to encode the vehicle circulation before focusing on transforming the encoded satisfiability problem into a minimization task. Therefore, we enforce that each start event $e \in \mathcal{E}_{\text{Start}}$ and each end event $e \in \mathcal{E}_{\text{End}}$ is connected by exactly one transfer arc:

$$\texttt{encode-vehicle-circulation-start}(\mathcal{E}_{\text{Start}}) := \bigwedge_{\forall e \in \mathcal{E}_{\text{Start}}} \texttt{encode-exactly-one}(\mathcal{Q}_{(\alpha^+(e))}) \tag{13a}$$

$$\texttt{encode-vehicle-circulation-end}(\mathcal{E}_{\text{End}}) := \bigwedge_{\forall e \in \mathcal{E}_{\text{End}}} \texttt{encode-exactly-one}(\mathcal{Q}_{(\alpha^-(e))}) \tag{13b}$$

These Eqs. (13a) and (13b) ensure the correct transfers of vehicles by enforcing flow conservation and exclusivity at transfer events, thereby guaranteeing that each scheduled train service is connected to a feasible vehicle cycle, consistent with the operational rules defined in the `SMT` model.

$$\texttt{encode-vehicle-count}(n, \mathcal{S} \cup \mathcal{U}) := \texttt{sequence-counter}(n, \hat{P}) \quad \text{where} \quad \hat{P} = \{\hat{p}_a \mid a \in \mathcal{S} \cup \mathcal{U}\} \tag{14}$$

As outlined in Section 3.5, the number of vehicles required to operate a feasible periodic timetable corresponds to the sum over all $k_a \in \{0, 1\}$ for selected commercial and vehicle-transfer activities $a \in \mathcal{S} \cup \mathcal{U}$. In the `SMT` formulation, each binary variable $\hat{p}_a$ encodes whether the corresponding precedence is inverted, i.e., whether $k_a = 1$. Consequently, Eq. (14) applies the `sequence-counter` encoding (Sinz 2005) to the set of literals $\hat{p}_a$, thereby enforcing an upper bound $n$ on the number of vehicles. This reformulates the vehicle circulation problem as a feasibility check under a fixed fleet size constraint.

### 3.7 Translation to SAT

To enable a direct comparison with the SMT and MIP models, we formulate a SAT encoding of the periodic timetabling problem. This encoding follows the approach of Fuchs et al. (2022), which extends the method of Großmann (2011) to handle train routing, and we further adapt it to incorporate vehicle circulation constraints.

For brevity, we do not explicitly detail the encoding of train sequencing, headway constraints, or vehicle circulation, as these follow the structure already established in Fuchs et al. (2022). Instead, we outline the key aspects distinguishing the SAT model from SMT and MIP.

The transformation consists of two primary steps. First, all event times are encoded using an *order encoding*, representing integer time values as Boolean variables. Second, all difference constraints—previously formulated in SMT—are expressed in propositional logic. The resulting model ensures consistency across train movements while maintaining routing flexibility.

Vehicle circulation constraints are incorporated analogously to the SMT model by enforcing `exactly-one` constraints for train handovers at terminal stations: each terminating train selects precisely one outgoing vehicle-transfer activity, and each originating train selects exactly one incoming activity. To encode the vehicle count, we introduce indicator literals $\hat{p}_a$ for each vehicle-related activity $a \in \mathcal{S} \cup \mathcal{U}$, representing whether the activity uses the cyclic wrap-around (i.e., $k_a = 1$). These literals are used in conjunction with a `sequence-counter` encoding (Sinz 2005) to enforce an upper bound $n$ on the number of wrap-arounds—and hence vehicles. This reformulates the vehicle circulation problem as a feasibility check under a fixed fleet size constraint, consistent with the SMT model.

# 4. Results

The results presented in this section are based on three sets of experiments designed to evaluate the efficiency of the proposed modeling approaches and assess the impact of various problem characteristics. First, we describe the implementation of the models (Section 4.1), and present the case-study and the set of instances derived from the Swiss railway network (Section 4.2), and compare the computational performance of Mixed Integer Programming (MIP), Boolean Satisfiability (SAT), and Satisfiability Modulo Theories (SMT) solvers varying the time discretisation granularity (Section 4.3). Second, we investigate how discretisation affects vehicle requirements under two circulation strategies: vehicles restricted to the same line versus shared use across all trains (Section 4.4). Third, we analyze the effect of train routing flexibility on vehicle counts, comparing fixed and flexible routing scenarios with the same two circulation strategies (Section 4.5). Finally, we study the benefit of the proposed integrated problem against sequential equivalent planning procedures (Section 4.6).

## 4.1 Implementation

The proposed models are implemented using the *OpenBus* framework (Fuchs and Corman 2019), ensuring consistency across all solving methodologies. We consider three optimization approaches: MIP, SAT, and SMT, each using a computing server equipped with four CPU cores (Intel Xeon Gold 6248) and 32 GB of RAM for all computational experiments. To leverage parallelization, all solvers utilize four threads.

The MIP formulation is implemented using Gurobi 12.0.1 (Gurobi Optimization, LLC 2025), with four solver threads assigned. The SAT formulation is implemented using Glucose 4.1 (Audemard and Simon 2018) via the PySAT package (Ignatiev et al. 2018). A portfolio strategy utilizes all four cores, where each core runs an independent solver instance initialized with a different random seed (Balyo et al. 2015). The SMT solver extends the approach of Leutwiler and Corman (2022), employing a portfolio-based strategy similar to the SAT approach. The solvers do not share any state or information, as they work on independent search spaces, with the first to terminate providing the final result.

We solve the SAT and SMT problems using an *ascending linear search* to determine the minimal vehicle count. First, a lower bound on the number of required vehicles is computed based on relaxed circulation constraints, neglecting headway constraints. This relaxation provides an initial lower bound for subsequent iterations. The model is then solved incrementally, starting from this bound and increasing the vehicle count $n$ step by step. If the model is infeasible for a given $n$, the vehicle count is incremented by one, and the lower bound is updated until feasibility is attained. Once a feasible solution is found, it is guaranteed optimal, as during this linear search, all instances with fewer vehicle counts have been proven to be infeasible. Thus, we can conclude the procedure.

## 4.2 Infrastructure and Instances

For our case study, we used data provided by Rhaetian Railway (RhB), a Swiss railway company operating most of the railway lines of the canton of Grisons. Following a methodology similar to Fuchs et al. (2022), we modeled the network at a mesoscopic level, as many sections consist of a single track. The infrastructure spans 380 km, with a complex terrain and operational restrictions. Technical running times were calculated using the same procedures as those employed by RhB, ensuring that the instances reflect realistic railway operations.

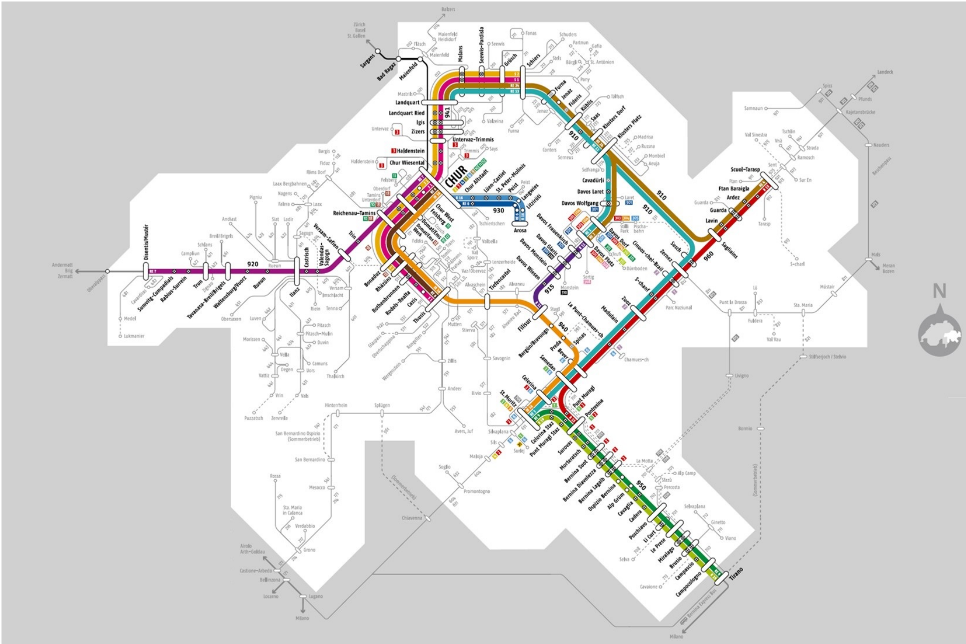


Figure F7: The current line plan for RhB (RhB 2023).

Using the current line plan (see Fig. F7), we generated a series of problem instances ranging from 1 to 10 lines, ensuring that the selected line plans remained connected. The characteristics of these instances, including the number of events, activities, headway constraints, and routing alternatives, are summarized in Table T2.

Each instance represents an increasing level of complexity, with additional lines introducing more routing alternatives, activities, and constraints. The number of vehicle transfer links varies depending on the vehicle-sharing policy:

- `No-Sharing`: Vehicles remain restricted to operating within their assigned service, meaning they cannot transfer between different lines.
- `Full-Sharing`: Vehicles can be shared across different lines (without dead-heading), allowing for a more flexible assignment and potentially reducing the number of required vehicles.

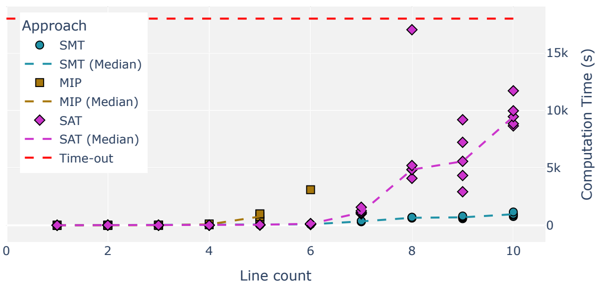
Table T2: Instance characteristics by number of lines.

| Lines | Events | Activities | Headway Constraints | Itinerary Activities | Routing Alternatives | Vehicle Transfers | |
|---|---|---|---|---|---|---|---|
| | | | | | | `No` | `Full` |
| 1 | 160 | 1038 | 812 | 144 | 34 | 2 | 2 |
| 2 | 424 | 2799 | 2234 | 385 | 95 | 4 | 6 |
| 3 | 608 | 4316 | 3515 | 541 | 124 | 6 | 12 |
| 4 | 1011 | 12317 | 10986 | 889 | 196 | 8 | 14 |
| 5 | 1363 | 22134 | 20302 | 1232 | 287 | 10 | 22 |
| 6 | 1975 | 38640 | 36059 | 1753 | 405 | 12 | 32 |
| 7 | 2379 | 43343 | 40243 | 2114 | 490 | 14 | 36 |
| 8 | 2891 | 53404 | 49681 | 2535 | 573 | 16 | 42 |
| 9 | 2989 | 54339 | 50478 | 2619 | 590 | 18 | 46 |
| 10 | 3291 | 58425 | 54144 | 2887 | 644 | 20 | 50 |

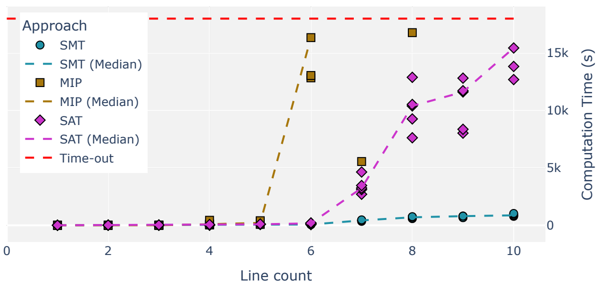
As shown in Table T2, the number of events and activities increases with the number of lines, naturally resulting in more headway constraints, itinerary activities (dwells and trips), and routing alternatives. Vehicle transfer options increase with instance size, especially when allowing vehicles to circulate across different lines. This flexibility, denoted as *Full* in the table, is expected to reduce the overall fleet size compared to the *No* case, where vehicles are restricted to individual lines.

## 4.3 Comparison of Performance

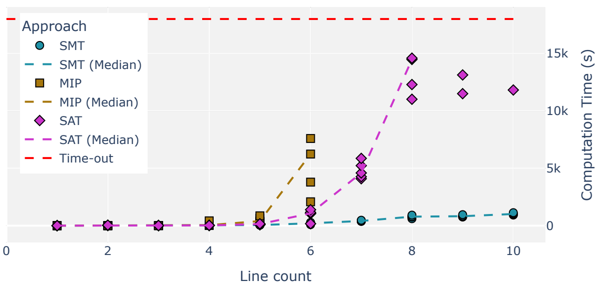
To evaluate the performance of the three approaches—Mixed Integer Programming (`MIP`), Boolean Satisfiability (`SAT`), and Satisfiability Modulo Theories (`SMT`)—we solve test instances derived from subsets of the `RhB` line plan. Each instance in Table T2 is computed once under the `Full-Sharing` and `No-Sharing` policies. To assess the impact of time granularity, we solve each instance at four different discretisation levels: 6, 3, 2, and 1 seconds. To account for performance variability, each solver is executed five times per instance with different random seeds and a time limit of 5 hours. The plots below show the median computation times per approach. We also indicate the timeout threshold and highlight scaling behavior as the instance size increases.



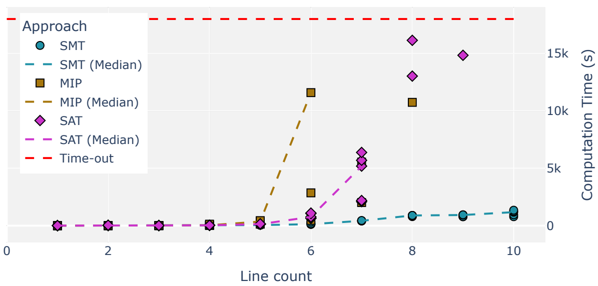

(a) 6-second granularity.




(b) 3-second granularity.




(c) 2-second granularity.

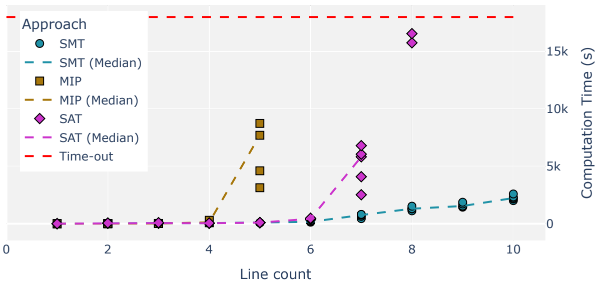



(d) 1-second granularity.

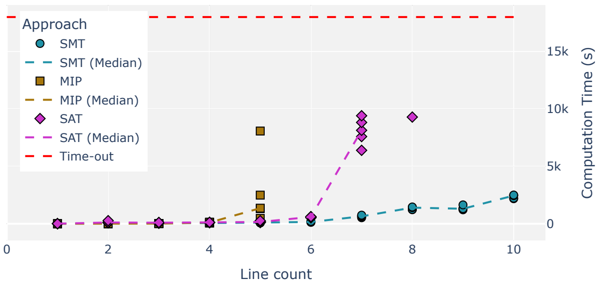
Figure F8: Computation times for `MIP`, `SAT`, and `SMT` under `No-Sharing` strategy across four discretisation levels.

The results under the `No-Sharing` policy in Fig. F8 clearly demonstrate the superior scalability of the `SMT` formulation. While `SAT` and `MIP` exhibit acceptable performance at coarse resolutions (6 and 3 seconds), both degrade significantly as the temporal resolution increases. `SAT` fails to solve many instances beyond six lines at 1-second resolution, and `MIP` times out already at intermediate sizes. In contrast, SMT maintains stable runtime across all tested resolutions and solves all instances up to ten lines without reaching the time limit. This advantage in scalability becomes more pronounced as instance complexity grows, underlining the practical advantage of the SMT approach for large-scale, high-resolution periodic timetable optimization.

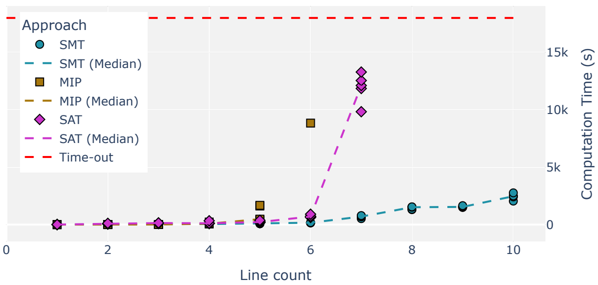
Under the `No-Sharing` policy in Fig. F8, we observe clear tipping points beyond which solvers fail to compute solutions within the time limit. At coarser time steps (6 and 3 seconds), SAT performs comparably well and outperforms MIP, solving all instances quickly. As the granularity increases, its performance degrades sharply. At the 1-second level, it frequently times out beyond six lines. MIP only handles networks up to 4–5 lines reliably across all resolutions and for these counts offers competitive runtime. However, for instances of larger sizes, MIP is no longer suitable. In contrast, SMT remains the most stable, showing consistent performance at high resolution and with larger networks. Notably, the difference in behavior between solvers is already visible at intermediate sizes, suggesting a gradual rather than sudden breakdown.



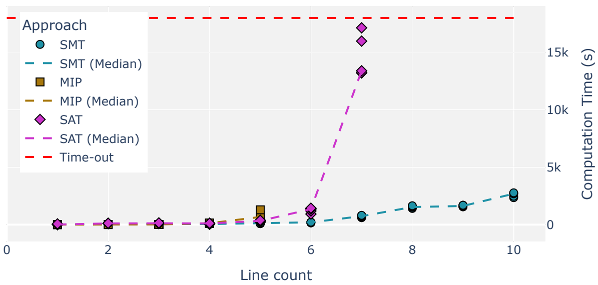

(a) 6-second granularity.




(b) 3-second granularity.




(c) 2-second granularity.

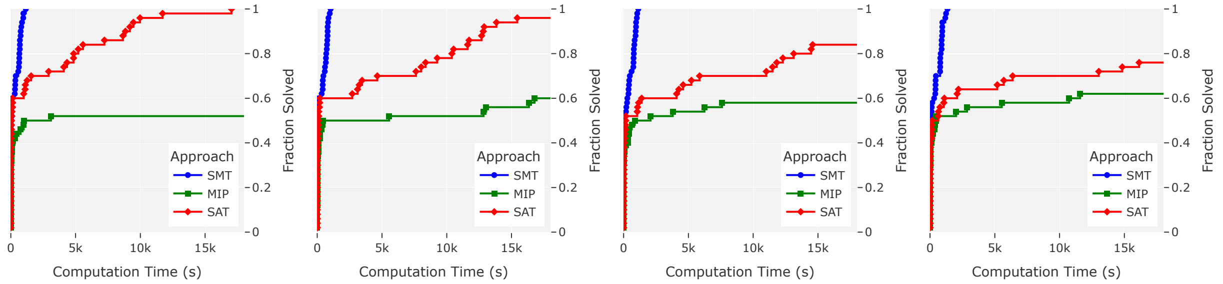



(d) 1-second granularity.

Figure F9: Computation times for MIP, SAT, and SMT under `Full-Sharing` strategy across four discretisation levels.

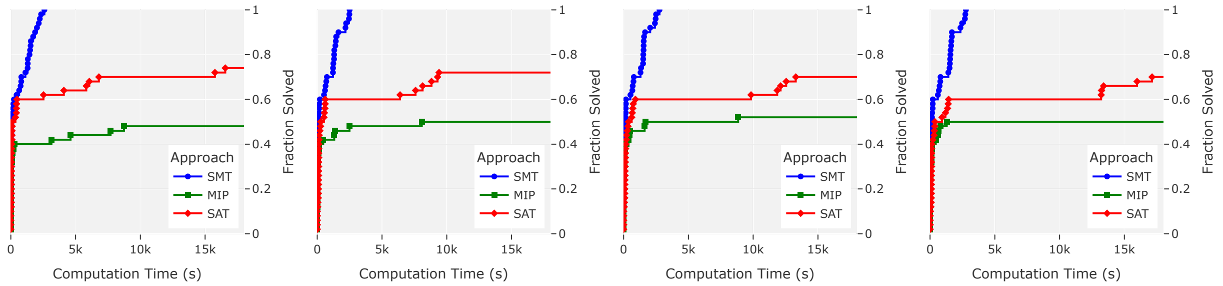
The patterns under `Full-Sharing` in Fig. F9 are qualitatively similar but more pronounced than the ones with `No-Sharing` in Fig. F8. While SAT again performs well at coarse resolutions, it fails even earlier at finer ones. For example, at 1-second resolution, partial timeouts already appear from 5 lines onward. MIP behaves comparably to the `No-Sharing` case, while SMT continues to scale reliably. `Full-Sharing` appears to amplify the runtime demands of solvers, likely because solution space flexibility introduces additional combinatorial complexity.

To complement the runtime analysis, we report the cumulative success rate of each solver. We track whether a solver found a feasible solution for every instance and time limit and plot the fraction of solved instances over time. These plots provide a comprehensive overview of how quickly and reliably each solver performs across different instance sizes.




(a) 6-second granularity (b) 3-second granularity (c) 2-second granularity (d) 1-second granularity

Figure F10: Cumulative success rate for MIP, SAT, and SMT under No-Sharing strategy.




(a) 6-second granularity (b) 3-second granularity (c) 2-second granularity (d) 1-second granularity

Figure F11: Cumulative success rate for MIP, SAT, and SMT under Full-Sharing strategy.

The cumulative success plots in Fig. F10 (No-Sharing) and Fig. F11 (Full-Sharing) provide a more detailed view of solver performance across instances. SMT consistently solves all configurations within two hours, often well before the limit. SAT performs well at coarse granularities but exhibits a clear tipping point, beyond which runtime increases steeply and success rate drops. MIP solves the fewest instances overall, typically succeeding quickly or not at all.

The relative impact of sharing strategies is also visible. Under Full-Sharing, solution space complexity increases, and solvers generally take longer, particularly for SAT. Conversely, SMT 's completion profile remains identical, mainly, confirming its suitability for large and complex configurations.

These observations are consistent with the trends seen in computation times and are further reinforced by the success rate analysis. Across both circulation strategies, SMT emerges as the most reliable and scalable solver, maintaining low runtime and solving all instances. SAT performs well in simpler settings but scales poorly. MIP is only effective on small networks.

Time discretisation and vehicle sharing influence solver performance, with finer time steps and full-sharing policies introducing additional computational load.

### 4.4 Effect of Time Discretisation and Vehicle Sharing Strategies on Vehicle Count

This section analyses the impact of time discretisation on vehicle requirements, comparing the five different granularities used in Section 4.3. While the baseline resolution is without discretisation, we apply four different levels of discretisation (1, 2, 3, and 6 seconds) to assess the impact of discretisation on the number of vehicles needed. Furthermore, we evaluate the potential benefits of increased flexibility through vehicle sharing by comparing both vehicle circulation strategies (`No-Sharing` and `Full-Sharing`).

When solving each scenario, we distinguish between two conflict configurations to isolate the effects of time discretization. This distinction helps separate the impact of rounding durations from the additional restrictions imposed by infrastructure conflicts. In real-world applications, rounding up activity durations is required for operational safety, but it can reduce schedule flexibility and increase travel times, potentially requiring more vehicles.

We therefore analyse:

1. `No Conflicts`: Only activity durations (e.g., running and dwell times) are rounded up to match the discretisation step, while headway and other conflict-related constraints are omitted. This setup quantifies the isolated impact of rounding.

2. `Conflicts`: All activities, including those modeling headways and resource usage, are adjusted for discretisation and fully included in the model. This setup reflects the full impact of time discretisation under realistic operational constraints.

Table T3 reports the required vehicles for each configuration. Parentheses indicate `No Conflicts` cases. We include the relative percentage increase in square brackets compared to the corresponding non-discretized baseline.

Table T3: Vehicle requirements under varying time discretisation steps for increasing network sizes (given by the number of lines from 1 to 10). Values show absolute fleet size and the relative increase over the non-discretised baseline.

| **Strategy** | **Step** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| `No-Sharing` No Conflicts | – | 2 (0%) | 4 (0%) | 5 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 1 sec | 2 (0%) | 4 (0%) | 5 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 2 sec | 2 (0%) | 4 (0%) | 5 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 3 sec | 3 (+50.0%) | 5 (+25.0%) | 6 (+20.0%) | 9 (+28.6%) | 12 (+20.0%) | 16 (+14.3%) | 19 (+11.8%) | 22 (+10.0%) | 24 (+14.3%) | 27 (+12.5%) |
| | 6 sec | 3 (+50.0%) | 5 (+25.0%) | 6 (+20.0%) | 9 (+28.6%) | 12 (+20.0%) | 16 (+14.3%) | 19 (+11.8%) | 22 (+10.0%) | 24 (+14.3%) | 27 (+12.5%) |
| `No-Sharing Conflicts` | – | 3 (0%) | 5 (0%) | 6 (0%) | 9 (0%) | 12 (0%) | 16 (0%) | 19 (0%) | 22 (0%) | 23 (0%) | 26 (0%) |
| | 1 sec | 3 (0%) | 5 (0%) | 6 (0%) | 9 (0%) | 12 (0%) | 16 (0%) | 19 (0%) | 22 (0%) | 23 (0%) | 26 (0%) |
| | 2 sec | 3 (0%) | 5 (0%) | 6 (0%) | 9 (0%) | 12 (0%) | 16 (0%) | 19 (0%) | 22 (0%) | 23 (0%) | 26 (0%) |
| | 3 sec | 3 (0%) | 5 (0%) | 6 (0%) | 9 (0%) | 12 (0%) | 16 (0%) | 19 (0%) | 22 (0%) | 24 (+4.3%) | 27 (+3.8%) |
| | 6 sec | 3 (0%) | 5 (0%) | 6 (0%) | 9 (0%) | 12 (0%) | 17 (+6.3%) | 20 (+5.3%) | 23 (+4.5%) | 25 (+8.7%) | 28 (+7.7%) |
| `Full-Sharing No-Conflicts` | – | 2 (0%) | 4 (0%) | 4 (0%) | 6 (0%) | 9 (0%) | 12 (0%) | 15 (0%) | 17 (0%) | 18 (0%) | 21 (0%) |
| | 1 sec | 2 (0%) | 4 (0%) | 4 (0%) | 6 (0%) | 9 (0%) | 13 (+8.3%) | 15 (0%) | 17 (0%) | 18 (0%) | 21 (0%) |
| | 2 sec | 2 (0%) | 4 (0%) | 4 (0%) | 6 (0%) | 9 (0%) | 13 (+8.3%) | 15 (0%) | 18 (+5.9%) | 19 (+5.6%) | 21 (0%) |
| | 3 sec | 3 (+50.0%) | 4 (0%) | 4 (0%) | 7 (+16.7%) | 10 (+11.1%) | 14 (+16.7%) | 16 (+6.7%) | 19 (+11.8%) | 20 (+11.1%) | 22 (+4.8%) |
| | 6 sec | 3 (+50.0%) | 4 (0%) | 5 (+25.0%) | 8 (+33.3%) | 10 (+11.1%) | 14 (+16.7%) | 16 (+6.7%) | 19 (+11.8%) | 20 (+11.1%) | 23 (+9.5%) |
| `Full-Sharing Conflicts` | – | 3 (0%) | 4 (0%) | 4 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 1 sec | 3 (0%) | 4 (0%) | 4 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 2 sec | 3 (0%) | 4 (0%) | 4 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 3 sec | 3 (0%) | 4 (0%) | 4 (0%) | 7 (0%) | 10 (0%) | 14 (0%) | 17 (0%) | 20 (0%) | 21 (0%) | 24 (0%) |
| | 6 sec | 3 (0%) | 4 (0%) | 5 (+25.0%) | 8 (+14.3%) | 11 (+10.0%) | 15 (+7.1%) | 18 (+5.9%) | 21 (+5.0%) | 22 (+4.8%) | 25 (+4.2%) |

The results in Table T3 confirm several previous trends. The `Full-Sharing` strategy consistently results in lower fleet requirements than `No-Sharing`, as the additional flexibility enables more efficient vehicle reuse. This effect is evident across all line counts and discretisation levels. As expected, vehicle requirements increase with network size due to higher service numbers and more interactions due to shared infrastructure. On average, across all configurations, adopting `Full-Sharing` instead of `No-Sharing` reduces the vehicle count by 12.9% in the `No Conflicts` setting and by 12.1% when headway constraints are included.

Discretisation has a visible and cumulative impact. Under the `No Conflicts` setting, coarser time steps lead to systematic increases in fleet size. For example, under `No-Sharing`, shifting from the continuous-time baseline to 6-second discretisation increases total vehicle needs by 19 vehicles, corresponding to an average increase of 15.3%. Under `Full-Sharing`, the increase is 14 vehicles or 13.0%. These differences arise solely from reduced scheduling precision, without any infrastructure constraints.

When headway constraints are introduced, the relative increases are smaller: seven vehicles (+5.0%) for `No-Sharing` and eight vehicles (+6.5%) for `Full-Sharing`. In this case, constraints already restrict the solution space, making it less sensitive to discretisation. Nevertheless, finer time steps remain beneficial in minimizing the required fleet.

Notably, switching the vehicle circulation strategy cannot compensate for the increase due to coarser steps. Although `Full-Sharing` reduces vehicle needs overall, it does not eliminate the structural overhead introduced by time rounding. The cost of discretisation is additive to the cost of constraint-induced rigidity.

These findings underscore the importance of modeling both precise timing and flexible circulation. Coarse discretisation inflates vehicle requirements significantly, even without conflicts. The impact is somewhat dampened but still measurable when infrastructure constraints are present. Overall, modeling accuracy and resource flexibility are essential for efficient railway operations.

### 4.5 Impact of Fixed Routes on Vehicle Requirements

This section evaluates the effect of fixing train routes on vehicle requirements. While previous sections assume routing flexibility, we explore a constrained variant where routes are pre-assigned. To ensure feasibility, the fixed routes are extracted from solutions previously computed with full routing flexibility. Specifically, we solve the original problem under flexible routing and enforce the obtained paths in a second run. This enables us to assess the cost of eliminating routing freedom while avoiding infeasibility.

The experiments are run using the SMT-based solver with a one-second discretisation step under `No-Sharing` and `Full-Sharing` strategies. Table T4 reports the number of vehicles required in the fixed-routing case and the percentage increase compared to the flexible-routing baseline.

Table T4: Vehicle requirements with fixed train routing for increasing network sizes (1–10 lines). Values show absolute fleet size, with relative increase over the flexible-routing baseline.

| Strategy | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| `No-Sharing` | 3 (+0.0%) | 5 (+0.0%) | 6 (+0.0%) | 9 (+0.0%) | 13 (+8.3%) | 17 (+6.3%) | 19 (+0.0%) | 22 (+0.0%) | 24 (+4.3%) | 27 (+3.8%) |
| `Full-Sharing` | 3 (+0.0%) | 4 (+0.0%) | 5 (+25.0%) | 8 (+14.3%) | 11 (+10.0%) | 14 (+0.0%) | 17 (+0.0%) | 21 (+5.0%) | 22 (+4.8%) | 24 (+0.0%) |

The results in Table T4 confirm that fixing routes generally increases the required fleet size. For the `No-Sharing` strategy, the average increase is 2.3%, while for `Full-Sharing` it is slightly higher at 5.9%. This difference highlights that, while full-sharing provides more efficient vehicle utilisation under flexible routing, it is also more sensitive to restrictions once flexibility is removed. Nevertheless, even under fixed routing, in most cases, the configuration of the `Full-Sharing` configuration still requires fewer vehicles overall than the `No-Sharing` baseline, confirming that the inherent benefits of vehicle sharing persist despite routing constraints.

The impact of fixed routing remains negligible for small networks, where route alternatives are limited and the feasible space is less constrained. However, from medium-sized networks onward, the effect becomes visible. In the `Full-Sharing` case, pre-assigned paths require up to 25% more vehicles than their flexible counterparts (e.g., instances with three lines). Interestingly, increases are not uniformly distributed. While some configurations (e.g., instances with 6 or 10 lines) show no penalty, others, such as with 4 or 5 lines, exhibit sharp increases. This suggests that the availability of alternative paths plays a key role: where the flexible model can exploit routing options to compress turnaround times or vehicle transitions, the fixed model is more likely to induce schedule fragmentation and idle time.

Overall, the findings reinforce the role of routing flexibility in reducing rolling stock requirements. While fixing paths may be operationally convenient, it should be applied with care in larger systems, as it can eliminate optimization potential and increase operational cost through larger fleet needs. The variability in impact across network sizes suggests that routing decisions should be informed by network structure and available topological alternatives, rather than treated as static inputs.

### 4.6 Benefit of Integration

This section investigates the benefit of jointly optimizing train routing, periodic timetabling, and vehicle circulation, rather than solving these components sequentially. Traditional planning pipelines typically approach these problems in isolation: routes are fixed first, then a timetable is computed—often optimizing travel times—and finally a feasible vehicle circulation is derived. While intuitive, this neglects interdependencies that can impact resource efficiency.

An assumption sometimes made is that minimizing travel times indirectly reduces fleet size. For example, Liebchen and Möhring (2004) report that minimizing passenger travel time on the Berlin Underground led to fewer required vehicles, despite vehicle count not being part of the objective. Our study offers another perspective by explicitly targeting vehicle minimization.

Lieshout (2021) previously demonstrated the benefit of integrating periodic timetabling and vehicle circulation regarding travel time and operational costs. In contrast, our work focuses exclusively on the required fleet size and extends the integrated formulation to include routing decisions. This allows us to isolate the impact of full integration while avoiding the need to model passenger travel times directly.

To quantify the effect, we compare two approaches:

- **Integrated Solve:** Uses our model, where routing, timetabling, and circulation are jointly optimized to minimize the number of vehicles.
- **Sequential Solve:** Routing is fixed first, then timetabling is solved with an objective of minimal travel time, followed by vehicle circulation. Multiple runs with different initial seeds ensure robustness of the results.

The results are shown in Table T5, which reports the required fleet size for the sequential solve and its relative increase over the integrated baseline.

Table T5: Vehicle requirements under sequential optimization for increasing network sizes (1–10 lines). Values show absolute fleet size and relative increase over the integrated baseline.

| Strategy | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| No-Sharing | 3 (+0.0%) | 5 (+0.0%) | 7 (+16.7%) | 9 (+0.0%) | 13 (+8.3%) | 19 (+18.8%) | 21 (+10.5%) | 25 (+13.6%) | 29 (+26.1%) | 32 (+23.1%) |
| Full-Sharing | 3 (+0.0%) | 4 (+0.0%) | 6 (+50.0%) | 8 (+14.3%) | 11 (+10.0%) | 19 (+35.7%) | 21 (+17.6%) | 26 (+30.0%) | 25 (+19.0%) | 30 (+25.0%) |

The results show that the integrated approach consistently reduces fleet size compared to the sequential strategy. Under the `No-Sharing` policy, the average reduction is 11.7%, increasing with the number of lines. For `Full-Sharing`, the improvement reaches up to 35.7%, highlighting how routing flexibility and vehicle sharing jointly benefit from integration. Notably, savings persist even in smaller configurations, with a marked increase in larger networks, where decisions across planning stages interact more tightly.

The increased vehicle count in the sequential solve stems from misalignments between routing, scheduling, and circulation. Minimizing travel time can yield structurally inefficient timetables, such as tight or incompatible turns, requiring additional vehicles to preserve feasibility. In contrast, the integrated model internalizes such constraints early and identifies globally efficient schedules.

While sequential approaches are easier to implement and align with traditional planning practice, they leave significant optimization potential untapped. Our results reinforce that jointly solving routing, scheduling, and circulation is essential to minimize operational cost and fleet size, especially as system complexity grows.

## 5. Conclusion

This paper presents a novel, integrated optimization model for periodic timetabling, train routing, and vehicle circulation in railway systems. We propose the VCR-PESP formulation and introduce the first Satisfiability Modulo Theories (SMT)-based solution method tailored to this problem. In contrast to traditional approaches using Mixed Integer Programming (MIP) or Boolean Satisfiability (SAT) (Großmann 2016, Kümmling et al. 2015), our SMT formulation enables continuous-time modeling and supports scalable solving without time discretisation (Armando et al. 2004, Leutwiler and Corman 2022).

Using real-world data from the Swiss narrow-gauge network of RhB, we conduct an extensive computational study to assess the impact of time discretisation, routing flexibility, and vehicle sharing strategies. Our results show that coarser discretisations significantly increase fleet requirements due to rounding, and this effect persists even in conflict-free configurations. While vehicle sharing across lines can mitigate some of this overhead, only continuous-time models—as enabled by SMT—consistently avoid these losses without compromising resolution.

In all tested scenarios, our SMT solver outperforms SAT and MIP, solving large-scale instances with higher precision and shorter runtimes. We also show that restricting routing or solving planning stages sequentially leads to unnecessary vehicle use. In contrast, jointly optimizing routing, timetabling, and vehicle circulation within a single model yields more efficient and implementable solutions—confirming insights from earlier integration studies (Lieshout 2021, Goossens et al. 2006) and extending them to include routing.

Future work could extend our model to incorporate passenger-centric objectives such as travel time and transfer quality (Polinder et al. 2021, Gattermann et al. 2016, Kroon et al. 2014), or to address robustness and real-time re-optimization. Our approach lays a strong foundation for high-resolution, scalable, and fully integrated railway planning using modern constraint solving techniques.

# Author Contributions

**Florian Fuchs**: Conceptualization, Methodology, Software, Investigation, Formal analysis, Visualization, Writing – original draft. **Bernardo Martin-Iradi**: Conceptualization, Methodology, Writing – review & editing. **Francesco Corman**: Supervision, Project administration, Writing – review & editing.

**Declaration of Generative AI and AI-assisted technologies in the writing process** Statement: During the preparation of this work the author(s) used ChatGPT in order to occasionally help with readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

# References


A. Armando, C. Castellini, E. Giunchiglia, and M. Maratea. A sat-based decision procedure for the boolean combination of difference constraints. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 16–29. Springer, 2004.

G. Audemard and L. Simon. On the glucose sat solver. *International Journal on Artificial Intelligence Tools*, 27:1840001, 02 2018.

T. Balyo, P. Sanders, and C. Sinz. Hordesat: A massively parallel portfolio sat solver. In M. Heule and S. Weaver, editors, *Theory and Applications of Satisfiability Testing – SAT 2015*, pages 156–172, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24318-4.

A. Biere, M. Heule, and H. van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

R. Borndörfer, M. Karbstein, C. Liebchen, and N. Lindner. A Simple Way to Compute the Number of Vehicles That Are Required to Operate a Periodic Timetable. In R. Borndörfer and S. Storandt, editors, *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, volume 65 of *OpenAccess Series in Informatics (OASIcs)*, pages 16:1–16:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-096-5.

R. Borndörfer, N. Lindner, and S. Roth. A concurrent approach to the periodic event scheduling problem. *Journal of Rail Transport Planning and Management*, 15:140–159, 2020. ISSN 22109706.

E. Bortoletto, N. Lindner, and B. Masing. Periodic Timetabling with Cyclic Order Constraints. In D. Frigioni and P. Schiewe, 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, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-302-7. doi: 10.4230/OASIcs.ATMOS.2023.7.

E. Bortoletto, R. N. van Lieshout, B. Masing, and N. Lindner. Periodic Event Scheduling with Flexible Infrastructure Assignment. In P. C. Bouman and S. C. Kontogiannis, 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, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-350-8. doi: 10.4230/OASIcs.ATMOS.2024.4.

M. Bussieck, T. Winter, and U. Zimmermann. Discrete optimization in public rail transport. *Mathematical Programming*, 79(1-3):415–444, 1997. ISSN 14364646, 00255610. doi: 10.1007/BF02614327.

G. Caimi, L. Kroon, and C. Liebchen. Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4):285–312, 2017. ISSN 22109706.

J. P. Fonseca, E. van der Hurk, R. Roberti, and A. Larsen. A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. *Transportation Research Part B: Methodological*, 109:128–149, 2018. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2018.01.012.

F. Fuchs and F. Corman. 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, 2019.

F. Fuchs, A. Trivella, and F. Corman. Enhancing the interaction of railway timetabling and line planning with infrastructure awareness. *Transportation Research Part C: Emerging Technologies*, 142:103805, 2022.

P. Gattermann, P. Großmann, K. Nachtigall, and A. Schöbel. Integrating passengers' routes in periodic timetabling: a sat approach. In *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

M. Goerigk and C. Liebchen. An improved algorithm for the periodic timetabling problem. In *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, page 1, 2017.

J. W. Goossens, S. Van Hoesel, and L. Kroon. On solving multi-type railway line planning problems. *European Journal of Operational Research*, 168(2 SPEC. ISS.):403–424, 2006. ISSN 03772217.

P. Großmann. *Satisfiability and Optimization in Periodic Traffic Flow Problems*. PhD thesis, Technische Universität Dresden, 2016.

P. Großmann. Polynomial reduction from pesp to sat. Technical report, Knowledge Representation and Reasoning Group, TU Dresden, 2011.

P. Großmann, S. Hölldobler, N. Manthey, K. Nachtigall, J. Opitz, and P. Steinke. *Solving periodic event scheduling problems with SAT*. Springer Berlin Heidelberg, 2012. ISBN 3642310869, 3642310877, 9783642310867, 9783642310874. doi: 10.1007/978-3-642-31087-4_18.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2025. URL https://www.gurobi.com.

S. Herrigel, M. Laumanns, J. Szabo, and U. Weidmann. Periodic railway timetabling with sequential decomposition in the PESP model. *Journal of Rail Transport Planning and Management*, 8(3-4): 167–183, 2018.

M. Heydar, M. E. Petering, and D. R. Bergmann. 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, 2013. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2013.06.003.

O. J. Ibarra-Rojas, R. Giesen, and Y. A. Rios-Solis. An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. *Transportation Research Part B: Methodological*, 70(C):35–46, 2014. doi: 10.1016/j.trb.2014.08.010.

A. Ignatiev, A. Morgado, and J. Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*, pages 428–437, 2018.

L. G. Kroon, L. W. Peeters, J. C. Wagenaar, and R. A. Zuidwijk. Flexible connections in PESP models for cyclic passenger railway timetabling. *Transportation Science*, 48(1):136–154, 2014. ISSN 15265447.

M. Kümmling, P. Großmann, K. Nachtigall, J. Opitz, and R. Weiß. A state-of-the-art realization of cyclic railway timetable computation. *Public Transport*, 7(3):281–293, 2015. ISSN 16137159.

F. Leutwiler and F. Corman. A logic-based benders decomposition for microscopic railway timetable planning. *European Journal of Operational Research*, 303(2):525–540, 2022.

C. Liebchen. Symmetry for periodic railway timetables. *Electronic Notes in Theoretical Computer Science*, 92(1):34–51, 2004. ISSN 15710661.

C. Liebchen and R. H. Möhring. The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables — and Beyond. In *Computer-aided Systems in Public Transport*, pages 117–150. Springer Berlin Heidelberg, 2004.

R. N. V. Lieshout. Integrated periodic timetabling and vehicle circulation scheduling. *Transportation Science*, 55:768–790, 2021. ISSN 15265447.

R. M. Lusby, J. Larsen, and S. Bull. A survey on robustness in railway planning. *European Journal of Operational Research*, 266(1):1–15, 2018. ISSN 0377-2217.

B. Martin-Iradi and S. Ropke. A column-generation-based matheuristic for periodic and symmetric train timetabling with integrated passenger routing. *European Journal of Operational Research*, 297(2): 511–531, 2022. ISSN 0377-2217.

B. Masing, N. Lindner, and C. Liebchen. Periodic timetabling with integrated track choice for railway construction sites. *Journal of Rail Transport Planning and Management*, 28:100416, 2023. ISSN 2210-9706. doi: https://doi.org/10.1016/j.jrtpm.2023.100416.

G. P. Matos, L. M. Albino, R. L. Saldanha, and E. M. Morgado. Solving periodic timetabling problems with sat and machine learning. *Public Transport*, 13(3):625–648, 2020. ISSN 16137159, 1866749x. doi: 10.1007/s12469-020-00244-y.

L. W. P. Peeters. *Cyclic railway timetable optimization*, volume 22. Erasmus University Rotterdam, Roterdam, 2003.

G. J. Polinder, M. Schmidt, and D. Huisman. Timetabling for strategic passenger railway planning. *Transportation Research Part B: Methodological*, 146:111–135, 2021. ISSN 01912615.

RhB. *Streckennetz*. Rhätische Bahn (RhB), 2023. URL https://www.rhb.ch/de/service-souvenirs/streckennetz.

T. Robenek, Y. Maknoon, S. S. Azadeh, J. Chen, and M. Bierlaire. Passenger centric train timetabling problem. *Transportation Research Part B: Methodological*, 89:107–126, 2016a. ISSN 01912615.

T. Robenek, S. Sharif Azadeh, Y. Maknoon, and M. Bierlaire. Hybrid cyclicity: Combining the benefits of cyclic and non-cyclic timetables. *Transportation Research Part C*, 75:228–253, 2016b. ISSN 0968090x, 18792359. doi: 10.1016/j.trc.2016.12.015.

P. Schiewe. *Integrated optimization in public transport planning*, volume 160. Springer, 2020. ISBN 9783030462697.

V. Schmid and J. F. Ehmke. Integrated timetabling and vehicle scheduling with balanced departure times. *OR spectrum*, 37:903–928, 2015.

P. Serafini and W. Ukovich. A Mathematical Model for Periodic Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

C. Sinz. Towards an optimal cnf encoding of boolean cardinality constraints. In P. van Beek, editor, *Principles and Practice of Constraint Programming - CP 2005*, pages 827–831, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-32050-0.

R. Wüst, S. Bütikofer, S. Ess, C. Gomez, A. Steiner, M. Laumanns, and J. Szabo. Periodic timetabling with 'track choice'-pesp based on given line concepts and mesoscopic infrastructure. In *Operations Research Proceedings 2018*, pages 571–578. Springer, 2019.