Optimization of railway resource planning process in a multi-level scale

Estia Maliqari[a,b,*], Dritan Nace[a], Antoine Jouglet[a], Giuliana Barbarino[b], Loïc Hamelin[b]

[a] Laboratoire Heudiasyc, Compiègne, France
{estia.maliqari, dritan.nace, antoine.jouglet}@hds.utc.fr
[b] SNCF Réseau, DGEX DIGIT, Saint-Denis, France
{estia.maliqari, giuliana.barbarino, loic.hamelin}@reseau.sncf.fr
[*] Corresponding author

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

February 15, 2025

Keywords: slot, railway timetabling, rolling stock scheduling, column generation, maximal clique

1 INTRODUCTION

As the Infrastructure Manager, SNCF Réseau plays a key role in the annual timetable planning process, where trains are strategically scheduled based on major routes or geographical sectors to meet the mobility needs set by transport authorities. These needs are expressed through time slots, space-time reservations of capacity, which allow a specific train to circulate through an infrastructure. These slots have unique characteristics depending on the demands they handle, whether for freight or passenger trains, high-speed or regional services (SNCF Réseau (2024)).

In contrast, railway undertakers are responsible for managing the rostering of rolling stocks, ensuring that train compositions are aligned with the predefined timetable slots.

In both academic literature and industry practices, the issues of train timetabling and rolling stock rostering are often treated independently (Borndörfer *et al.* (2006), Caprara *et al.* (2002)), with only a few studies addressing both problems simultaneously (Yue *et al.* (2017)). Typically, the current railway planning process focuses on maximizing the number of trains in the timetable before assigning rolling stock according to these schedules. However, this separation can create conflicts that prevent the overall optimization of the railway system.

While many existing tools adopt a macroscopic approach with a lower level of detail in infrastructure planning (Borndörfer *et al.* (2006), Caprara *et al.* (2002), Serafini & Ukovich (1989)), our study leverages a microscopic simulation tool. This method aims to digitize the railway planning process, producing more accurate and detailed timetables that can better address the complexities of the entire railway network.

2 PROBLEM DESCRIPTION

The objective is to create a 24-hour operational timetable for trains, ensuring conflict-free movement across the rail network while meeting various constraints. Each train mission follows a unique path, specifying details like departure and arrival stations, stops, and departure times with some tolerance for flexibility. Each mission is linked to specific rolling stock, requiring compatible train sets.

The infrastructure is modeled at a microscopic level, with detailed information on track sections, signal systems, and train detection, allowing accurate conflict identification and resolution at a granular scale.

A key challenge is optimizing rolling stock rostering to balance available train sets and garage capacities, ensuring trains meet demand while minimizing unnecessary movements and empty trips. The ultimate goal is to maximize network efficiency, optimize track usage, avoid overloading, and ensure train availability, all while controlling operational costs related to train movements and parking.

3 METHODOLOGY

To produce a feasible timetable from the slot graph, several steps are taken. As illustrated in Figure 1, a maximal clique algorithm is first applied to the slot graph to find the optimal combination of candidates. If each mission has a selected candidate, a feasible timetable covering all requested slots is created. If some slots lack candidates, a column generation approach is used to propose new options as well as assign the rolling stocks to the already selected slots. They are then added to the original slot graph and the maximal clique algorithm is reapplied.

To obtain more robust results, we can use an existing algorithm as a pre-processing phase before building the slot graph. This algorithm generates mesoscopic timetables (Joubert *et al.* (2022)). The mesoscopic scale provides more infrastructure detail than the macroscopic level, while maintaining some abstraction from the microscopic scale. It incorporates elements like line tracks, station tracks and routes. By taking a series of missions as input, the algorithm creates a conflict-free timetable with assigned track time slots.

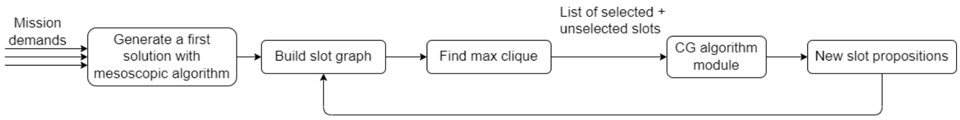

Figure 1 – *Logic diagram of the model workflow*

3.1 Slot graph

The problem is modeled as a **slot graph**, where M = {1, 2, ..., n} represents the missions to be planned, and N_s denotes the set of candidate slots for each mission. These candidate slots are generated on a microscopic scale using the Digital Twin module of Open Source Railway Designer (OSRD (2024)), which enables Infrastructure Managers to prepare timetables through detailed operational studies. Candidates form graph nodes, grouped by mission. Candidates of the same mission follow similar paths through the same stations, differing in track occupancy and departure times. To meet client's departure tolerances, two parameters are introduced: *frequency*, the time interval between nodes, and *num_ nodes*, the total nodes generated per mission.

After nodes are constructed, potential conflicts between candidate slots of different missions are identified. An edge is added between two nodes from different missions if there are no conflicts between them.

3.2 Maximal Clique model

By identifying a maximal clique within the constructed graph, we develop an initial timetable that maximizes the number of conflict-free time slots. This clique represents mutually compatible allocations, ensuring that no two selected nodes (candidate slots) are in conflict.

We use a greedy heuristic algorithm, starting with the most connected node as the clique's foundation. This node's group, which represents the set of nodes from the same slot, is recorded to avoid duplicates. The algorithm then iteratively expands the clique by adding nodes from remaining groups not yet included. For each group, it selects nodes connected to all current clique members, preserving the clique property. This continues until no suitable nodes remain, efficiently building a large, constraint-compliant clique.

3.3 Column Generation model

After the maximal clique module provides an initial timetable, a post-treatment module suggests new candidates using a Column Generation approach. This macroscopic model refines the initial solution by proposing a revised timetable with slight adjustments and new slot options for non-selected missions. Additionally, it allocates available rolling stock for each selected slot.

Let $\mathcal{S}$ be the set of slots to be added to a timetable, containing the already selected slots as well as the candidate ones. Each slot has a set of potential paths $\mathcal{P}_s$, defined by sequences of node-time pairs, and must operate within a specified time window. Paths of the same slot have the same nodes, differing only by the passing times. We model the paths of slots as a dynamic graph with timed nodes. The edges between the physical nodes all have values for capacities, representing the number of one or bi-directional tracks between these two points in the real infrastructure. We consider $\mathcal{P}$ to be the set of all paths and $\mathcal{E}$ the set of all edges.

$$\max \sum_{p\in\mathcal{P}} x_p - \lambda \sum_{p\in\mathcal{P}} z_p \tag{1}$$

$$\sum_{p\in P_s} x_p \leq 1 \quad \forall s \in \mathcal{S} \qquad (\sigma_s) \tag{2}$$

$$\sum_{p\in P,((i,j),t)\in p} x_p \leq C_{ij}, \quad \forall (i,j) \in \mathcal{E}, \forall t \in \mathcal{T} \qquad (\pi_{ijt}) \tag{3}$$

$$\sum_{p\in P,((j,i),t)\in p} x_p \leq C_{ji}, \quad \forall (j,i) \in \mathcal{E}, \forall t \in \mathcal{T} \qquad (\pi_{jit}) \tag{4}$$

$$\sum_{p\in P,((i,j),t)\in p} x_p + \sum_{p\in P,((j,i),t)\in p} x_p \leq C_l, \quad \forall l \in Link[(i,j),(j,i)], \forall t \in \mathcal{T} \qquad (\gamma_{ijt}) \tag{5}$$

The objective function (1) aims to maximize the total number of selected paths across all slots while respecting track capacity constraints and minimizing rolling stock usage. The binary variable x_p indicates whether path p is selected for slot s, while z_p denotes if path p requires new rolling stock from the depot. The parameter λ penalizes the use of new rolling stock when reuse is possible. Constraints (2) enforce slot assignment, ensuring only one path per slot. Constraints (3) and (4) limit the number of slots using a track in a given direction. Constraints (5) restrict the total number of slots using a track in both directions simultaneously, expressed by $Link[(i,j),(j,i)]$.

$$x_p \leq \sum_{q\in \text{prec}(p)} y_{pq} + z_p \quad \forall p \in \mathcal{P}_s, \forall s \in \mathcal{S} \tag{6}$$

$$\sum_{p\in\mathcal{P}_s} z_p \leq \text{Stock}_{d,r} \quad \forall d \in \mathcal{D}, \forall s \in \mathcal{S}_r, \forall r \in \text{Rolling Stock Types} \tag{7}$$

$$\sum_{q\in \text{prec}(p)} y_{pq} \leq 1 \quad \forall q \in \mathcal{P} \qquad \sum_{p\in \text{succ}(q)} y_{pq} \leq 1 \quad \forall p \in \mathcal{P} \tag{8}$$

Constraints (6-8) handle the allocation of rolling stocks for each slot. We introduce a new binary variable y_{pq} expressing whether the stock of path q can be reused by p. In order to reuse its stock, path p must depart after the arrival of q, in the same station. Constraints (6) ensure that a path x_p can only be selected if it either reuses the stock of a preceding path q or uses new rolling stock (indicated by z_p). Constraints (7) ensure that the number of rolling stock units leaving a depot does not exceed its available capacity for each rolling stock type. We note $\mathcal{S}_r$ the set of slots using rolling stock type r. Finally, constraints (8) limit the reuse of a rolling stock unit to only one path at a time.

The approach alternates between solving the master problem and a sub-problem (pricing problem) to generate feasible paths, based only on the slot selection constraints. Dual variables as specified in constraints (2-5) assign costs to potential paths, reflecting their marginal value in the current solution. Paths with minimal reduced costs (where the reduced cost is expressed as $RC = \sigma_s + \sum_{((i,j),t)\in\mathcal{P}}(\pi_{ijt} + \pi_{jit} + \gamma_{ijt})$) are iteratively added to the master problem, refining the solution efficiently.

4 RESULTS

Initial tests of this approach have been conducted, focusing both on the timetable planning problem and rolling stock rostering.

Table 1 – *Some results from a small test infrastructure with 5 stations, planning 8 slots highly in conflict with each other and requiring the same type of rolling stock.*

Slot graph		Max clique	Column Generation	
frequency (min)	num_nodes	selected slots	generated slots	rolling stocks per depot
1	10	5/8	8/8	6
1	10	5/8	7/8	3
2	30	8/8	8/8	6
2	30	8/8	7/8	3

With an increasing number of nodes in the graph, the maximal clique module delivers more easily a viable solution. In parallel, the column generation module successfully identifies macroscopic paths for any unselected slots when the quantity of rolling stocks in a depot is enough. In cases of shortages at certain depots, previously selected paths may need to be excluded from the solution. Further tests and case studies are ongoing to evaluate the robustness and practical applicability of these methods.

5 DISCUSSION & FUTURE WORK

Future research will enhance the column generation framework by adding constraints to tackle operational challenges, such as limited rolling stock at depots, leading to empty trips, and ensuring maintenance compliance for the infrastructure.

References

Borndörfer, R., Grötschel, M., Lukac, S. G., Mitusch, K., Schlechte, T., Schultz, S., & Tanner, A. 2006. An Auctioning Approach to Railway Slot Allocation. *Competition and Regulation in Network Industries*, **1**, 163–196.

Caprara, A., Fischetti, M., & Toth, P. 2002. Modeling and Solving the Train Timetabling Problem. *Operations Research*, **50**(5), 851–861.

Joubert, G., Nace, D., Jouglet, A., & Postec, M. 2022 (June). An adaptive procedure for railway periodic timetabling and tracks assignment. *In: TRISTAN XI Proceedings*.

OSRD. 2024. *Open Source Railway Designer*.

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

SNCF Réseau. 2024. *Network Statement of the National Rail Network. 2025 Timetable*.

Yue, Y., Han, J., Wang, S., & Liu, X. 2017. Integrated Train Timetabling and Rolling Stock Scheduling Model Based on Time-Dependent Demand for Urban Rail Transit. *Computer-Aided Civil and Infrastructure Engineering*, **32**, 856–873.