A Multi-Neighborhood Search Approach to Rolling Stock Rescheduling

Roberto Maria Rosati[1], Valentina Cacchiani[2], Vera Hemmelmayr[1]
[1] WU Vienna University of Economics and Business, Austria
robertomaria.rosati@wu.ac.at, vera.hemmelmayr@wu.ac.at
[2] DEI, University of Bologna, Italy
valentina.cacchiani@unibo.it

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

Keywords: Rail transport systems, Rolling stock rescheduling, Multi-Neighborhood Search

1 Introduction

Railways are a backbone infrastructure for passenger and freight transport. Disruptions, however, happen frequently because of unexpected events. They can impact economies and supply chains and, ultimately, cause a loss of money and reputation to railway companies. In case of a disruption, rescheduling is applied to determine a new feasible plan according to the following steps: first, the timetable is updated, then the rolling stock and, finally, the crew are rescheduled (Cacchiani *et al.*, 2014). In this work, we focus on the Rolling Stock Rescheduling Problem (RSRP) that consists of assigning compatible railway vehicles (*rolling stocks*) to trips according to an updated timetable, with the main goal of minimizing trip cancellations.

For passenger transportation, RSRP was studied by Nielsen *et al.* (2012) who presented a multi-commodity flow model based on a graph defining unit composition changes. Later, several extensions to this model have been proposed, to include deadheading trips and adjusted demand (Wagenaar *et al.*, 2017) or combining timetable rescheduling and passenger flow with RSRP (Veelenturf *et al.*, 2017). A multi-commodity flow model on a hypergraph was proposed by Borndörfer *et al.* (2017), while a path-based model was proposed by Lusby *et al.* (2017). Recently, Wang *et al.* (2021) proposed a two-stage heuristic algorithm for the integration of RSRP with timetable rescheduling.

Fewer studies focused on freight transportation. In particular, a path-based model was proposed in Sato & Fukumura (2012) who studied the locomotive rescheduling problem under disruptions consisting of several delayed trips and a few canceled ones and solved it by a column generation algorithm. However, situations like the recent floods that affected several countries in Europe led to significant challenges for rail freight companies, including penalty payments and lawsuits. Ultimately, the foreseen shift towards low-carbon freight transportation will be slower if railways are seen as unreliable. Therefore, the design of effective and efficient rolling stock rescheduling methods that take into account severe disruptions in freight transportation is needed. We propose a Multi-Neighborhood Search (MNS) for the RSRP, which has not been previously used in this context and achieves good solution quality in low computation time.

2 Problem Description

We consider a realistic RSRP taking as a baseline the planning problem introduced by Frisch *et al.* (2021), that deals with a real-world freight transport case in Austria. In their approach, compatible *locomotives* from set L are assigned to *trips* in set T, which are characterized by departure and arrival stations and times. Sequencing rules have to be respected between pairs of trips that are executed by the same locomotives. Two trips can be in sequence even if their connecting stations are not the same, thus requiring a deadheading trip. They are computed on the railway network, which is represented as a graph composed by nodes (*stations*) and edges (*sections*, S). Each locomotive has a starting location and a maximum number of residual kilometers it can travel before maintenance has to be executed at specific stations. The goal is to minimize a weighted sum of the cost associated to the number of used locomotives, the number of deadhead kilometers, and the number of maintenance appointments, with the first term having a very large weight.

We consider the scenario in which, during operations, a disruption affects a portion of the railway network, causing trip cancellations and delays as well as unavailability of some deadheading trips. Let $\bar{T} \subset T$ be the set of trips that are not cancelled. A disruption is characterized by a time interval $[t_{d_s}, t_{d_e}]$ and by a subset of sections $\bar{S} \subset S$ that remain available for trips and deadheads during that time slot. Additionally, only the set $\bar{L} \subset L$ of locomotives used in the original plan are available.

The RSRP consists of rescheduling trips in the horizon comprised between t_{d_s} and the end of the original schedule, considering current locations of locomotives and the driven kilometers. A solution is an updated feasible schedule with the objective to minimize, in order of importance, the additionally canceled trips, the missed maintenance appointments, and the deviation from the original plan (i.e., different assignments of locomotives to trips). Therefore, the original plan is also an input to the problem.

We generate our instances based on those by Frisch *et al.* (2021), based on real-world data. They have a scheduling horizon of one week, use up to 113 locomotives and a number of trips ranging from 75 to 3178. Our disrupted instances consist of the original instance, a rolling stock plan, a list of disrupted sections, and a disruption start and end time. The *severity* of the disruption (δ) is the ratio $|S \setminus \bar{S}|/|S|$ of disrupted sections on total sections.

3 Multi-Neighborhood Search

The composition of multiple local search neighborhoods increases the connectivity in the search space and gives access to a wide range of search trajectories. Used within a stochastic framework, Multi-Neighborhood Search (MNS) has proven successful on a wide range of challenging combinatorial problems (see, e.g. Lü *et al.*, 2023, Rosati & Schaerf, 2024). We propose an MNS for the RSRP, and hereafter we describe its main components: the solution representation, the search space, the neighborhoods, the initial solution, and the metaheuristic that guides the search.

We represent a *solution* (i.e., a rescheduled plan) by two vectors Π and Φ, both with size $|\bar{T}|$. Given a sorting of trips in $\bar{T}$ by increasing departure time, the value $\pi(i)$ of Π is the locomotive $\ell \in \bar{L}$ assigned to trip $i \in \bar{T}$, or $\pi(i) = -1$ if a trip is canceled. Each entry $\phi(i)$ of Φ is a binary value which is set to 1 if locomotive $\ell = \pi(i)$ is undergoing maintenance *before* trip i (i.e., either at the arrival station of trip $i-1$ or at the departure station of trip i).

The core of our method is the composition of three *neighborhoods*: $\mathsf{Change}\langle \ell, i\rangle$ that assigns trip i to locomotive $\ell \in \bar{L} \cup \{-1\}$, $\mathsf{Swap}\langle i, j\rangle$ that swaps the locomotives currently assigned to trips i and j, and $\mathsf{MergeLocomotive}\langle \ell_1, \ell_2\rangle$ that assigns all trips currently assigned to locomotive ℓ_1 to a new locomotive $\ell_2 \in \bar{L} \cup \{-1\}$. After every move, a fast-forward greedy procedure verifies if maintenance appointments can be executed as in the plan. This procedure allows to keep the size of the search space manageable and the search efficient. The *search space*, therefore, is the set of all possible assignments of values from $\bar{L} \cup \{-1\}$ to members of vector Π, which includes both the feasible and the infeasible region.

The exploration criterion is stochastic: neighborhoods Change, Swap, and MergeLocomotive have probabilities σ_{C}, σ_{S}, and σ_{M}, respectively, such that $\sigma_{\mathsf{C}} + \sigma_{\mathsf{S}} + \sigma_{\mathsf{M}} = 1$. Neighborhoods Change and MergeLocomotive also have internal biases, called b_C and b_M, respectively. They define the ratio of trips cancellations ($\ell = -1$). The move choice is performed with a *two-step biased random move selection*. First, a neighborhood is chosen, randomly depending on the probabilities σ_{C}, σ_{S}, and σ_{M}. If the chosen neighborhood is Change or MergeLocomotive, a cancellation will be performed with probabilities $b_C \in [0, 1]$ or $b_M \in [0, 1]$, otherwise a reassignment to any locomotive $\ell \in \bar{L}$ will be executed. Finally, a move is uniformly drawn inside the neighborhood.

As *initial solution* we take the original plan, restricted to $\bar{T}$, which might be infeasible. To ensure a feasible solution, we apply a repair procedure that iteratively cancels the trips that are infeasible w.r.t. sequencing rules and maintenance appointments.

A Simulated Annealing (SA) *metaheuristic* guides the search, as it is suited to the stochastic exploration criterion of our MNS. We use a classic SA with a *cut-off* mechanism, used to speed up the early stages of the search (see Rosati & Schaerf, 2024). The parameters of our SA are: initial temperature T_0, final temperature T_f, cooling rate ρ, and the cut-off rate α.

In addition to the MNS, we re-implemented the mathematical model from Frisch *et al.* (2021), suitably adapted to take into account the specific constraints and objectives of the RSRP, and use it for a computational comparison.

4 Experimental Results

Our MNS is written in C++, compiled with g++ 11.4.0 in `-O3` mode on Ubuntu 22.04.4. The mathematical models are written for CPLEX using the C++ Concert interface. We run all the experiments on a machine equipped with 13th Gen Intel(R) Core(TM) i7-1355U CPU.

We generated a wide set of instances with different disruption durations and severities. Among them, we selected a subset of 20 instances for validation, leaving the remaining for training. Table 1 resumes their features: number of trips $|\bar{T}|$, number of locomotives $|\bar{L}|$, disruption severity δ and duration Δt_d. The weights for the cost components of cancellations (c), missed maintenance appointments (m), and deviations from plan (d), are 100, 10 and 1, respectively. These weights set clearly the relative relevance of the terms in the objective function.

For the tuning of the parameters we employed the IRACE tool, with a single tuning stage for both the SA and the MNS parameters on 10000 repetitions on 464 training instances. The resulting values for the SA parameters are $T_0 = 1120.5$, $T_f = 10.162$, $\alpha = 0.993$, and $\rho = 0.268$. The neighborhood probabilities are: $\sigma_C = 0.816$, $\sigma_S = 0.143$, and $\sigma_M = 0.041$, and their biases $b_C = 0.494$ and $b_M = 0.257$.

In Table 1 we report the results of the mathematical model and the MNS. The model is solved with a time limit of 3600 seconds on 4 threads, while the MNS runs with time limits of 30 and 120 seconds, on a single thread. For the MNS, we report the average on 10 repetitions per instance. In both cases, a procedure to recompute the deadheads during the disruption is run beforehand. It takes on average 52 seconds and is not counted in the table. The model solves four instances to optimality (# 01, 04, 05, 06), in the elapsed time reported in column $t(s)$. For all the other instances, the time limit is reached, and for # 19 no feasible solution is found. The MNS, on the other hand, always succeeds in finding a feasible solution for all instances. Solutions on small instances are similar to the ones of the model, but are obtained in shorter runtimes, while MNS outperforms the model by far on larger instances. These considerations apply for both the $30s$ and $120s$ time limits, making our method a suitable option for real-world applications.

5 Discussion and future work

We studied the RSRP for a freight transportation problem and proposed an MNS algorithm which always found feasible solutions for the considered instances in short computing time. In the future, we plan to consider additional aspects relevant for locomotive rescheduling (e.g., rescheduling maintenance appointments in a flexible way, using reserve locomotives). Regarding the solution method, we plan to include additional neighborhoods and to speed up the evaluation

Table 1 – *Results for the 20 instances of the validation set.*

instance					Model				MNS 30 s			MNS 120 s		
#	$\lvert\bar{T}\rvert$	$\lvert\bar{L}\rvert$	δ	Δt_d (h)	$t(s)$	c	m	d	c	m	d	c	m	d
01	65	7	0.16	10.1	0	0	0	6	0.0	0.0	6.0	0.0	0.0	6.0
02	122	10	0.14	19.2	-	2	0	13	2.0	0.0	11.0	2.0	0.0	11.0
03	57	11	0.10	9.1	-	0	0	9	0.0	0.0	11.4	0.0	0.0	11.4
04	110	11	0.19	10.9	3	2	0	14	2.1	0.0	13.2	2.0	0.0	13.2
05	143	12	0.11	6.7	9	2	0	8	2.0	1.0	2.9	2.0	1.0	2.9
06	210	13	0.11	8.1	283	0	0	12	0.0	0.0	13.7	0.0	0.0	13.7
07	176	15	0.19	7.3	-	3	0	31	4.4	0.0	27.2	4.0	0.0	27.2
08	280	18	0.09	11.2	-	25	1	70	9.1	1.0	28.5	8.3	1.0	28.5
09	314	23	0.17	12.5	-	4	0	27	4.0	0.0	31.6	3.1	0.0	31.6
10	546	28	0.07	19.2	-	4	0	31	1.6	1.5	45.5	0.7	1.5	45.5
11	918	34	0.18	6.3	-	918	3	918	15.3	0.5	190.6	8.3	0.1	146.1
12	1537	40	0.16	6.4	-	1536	30	1536	25.4	4.6	150.1	15.5	4.7	94.5
13	663	43	0.14	8.1	-	317	1	329	15.9	1.0	102.6	11.0	1.0	86.3
14	1283	46	0.15	7.1	-	1282	7	1283	36.9	2.0	182.6	24.4	2.2	109.2
15	1634	47	0.13	9.4	-	1634	2	1634	45.5	0.1	218.8	21.1	0.0	169.3
16	499	50	0.16	11.3	-	340	0	350	15.5	0.0	93.7	11.4	0.0	91.0
17	1853	56	0.06	9.6	-	1853	13	1853	49.4	6.3	230.3	28.4	5.7	205.2
18	2081	58	0.08	36.0	-	2081	4	2081	154.7	1.6	324.7	100.9	1.7	364.2
19	2639	60	0.12	5.7	-	-	-	-	51.5	7.0	125.9	30.7	7.9	190.0
20	1257	61	0.06	16.4	-	1257	0	1257	61.8	0.0	220.1	40.5	0.0	163.0

of differential costs. In addition, we want to solve the planning problem by Frisch *et al.* (2021), as MNS can be easily adapted to tackle it. Finally, we plan to generate new instances and to make them and our results publicly available to foster future research on the RSRP.

References

Borndörfer, Ralf, Grimm, Boris, Reuther, Markus, & Schlechte, Thomas. 2017. Template-based re-optimization of rolling stock rotations. *Public Transport*, **9**(1), 365–383.

Cacchiani, Valentina, Huisman, Dennis, Kidd, Martin, Kroon, Leo, Toth, Paolo, Veelenturf, Lucas, & Wagenaar, Joris. 2014. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, **63**, 15–37.

Frisch, Sarah, Hungerländer, Philipp, Jellen, Anna, Primas, Bernhard, Steininger, Sebastian, & Weinberger, Dominic. 2021. Solving a real-world locomotive scheduling problem with maintenance constraints. *Transportation Research Part B: Methodological*, **150**, 386–409.

Lü, Zhipeng, Fang, Yuan, Su, Zhouxing, Wang, Yang, Wu, Xinyun, & Glover, Fred. 2023. Dual-neighborhood iterated local search for routing and wavelength assignment. *Computers & Operations Research*, **160**, 106396.

Lusby, Richard M, Haahr, Jørgen Thorlund, Larsen, Jesper, & Pisinger, David. 2017. A branch-and-price algorithm for railway rolling stock rescheduling. *Transportation Research Part B: Methodological*, **99**, 228–250.

Nielsen, Lars Kjær, Kroon, Leo, & Maróti, Gábor. 2012. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, **220**(2), 496–509.

Rosati, Roberto Maria, & Schaerf, Andrea. 2024. Multi-Neighborhood Simulated Annealing for the Capacitated Dispersion Problem. *Expert Systems with Applications*, **255**, 124484.

Sato, Keisuke, & Fukumura, Naoto. 2012. Real-time freight locomotive rescheduling and uncovered train detection during disruption. *European Journal of Operational Research*, **221**(3), 636–648.

Veelenturf, Lucas P, Kroon, Leo G, & Maróti, Gábor. 2017. Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. *Transportation Research Part C: Emerging Technologies*, **80**, 133–147.

Wagenaar, Joris, Kroon, Leo, & Fragkos, Ioannis. 2017. Rolling stock rescheduling in passenger railway transportation using dead-heading trips and adjusted passenger demand. *Transportation Research Part B: Methodological*, **101**, 140–161.

Wang, Yihui, Zhao, Kangqi, D'Ariano, Andrea, Niu, Ru, Li, Shukai, & Luan, Xiaojie. 2021. Real-time integrated train rescheduling and rolling stock circulation planning for a metro line under disruptions. *Transportation Research Part B: Methodological*, **152**, 87–117.