# Iterative Two-Stage Stochastic Programming Approach for Real-Time Rolling Stock Rescheduling Under Uncertainty

Jia Hui Zhu[1], Twan Dollevoet[1], Dennis Huisman[1,2],

[1] Econometric Institute and Erasmus Center for Optimization in Public Transport (ECOPT), Erasmus University Rotterdam, 3000 DR Rotterdam, Netherlands;
[2] Digitalization of Operations, Netherlands Railways, 3500 HA Utrecht, the Netherlands
E-mail addresses: j.h.zhu@ese.eur.nl (J.H. Zhu), dollevoet@ese.eur.nl (T. Dollevoet), huisman@ese.eur.nl (D. Huisman)

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

January 23, 2025

Keywords: Rolling Stock Rescheduling, Railway Optimization, Stochastic Programming

## 1 INTRODUCTION

With over a million passenger train trips during an average day, the Dutch railway network plays a big role in the mobility system of the Netherlands. To operate these trips, the major railway operator, Netherlands Railways (NS) is tasked with solving several complex planning problems, including the planning of individual train units, or rolling stock. On the day of operation, disruptions that range from a missing rolling stock unit to a complete blockage of a railway track can occur. As a result, rolling stock units may not be able to complete the remaining trips that they are scheduled for. Unavailability of rolling stock influences passenger satisfaction, as passengers possibly have to stand or even wait for the next train in case not enough seats are available. Restoring the rolling stock schedule can be expensive, as it can lead to additional carriage kilometers and additional shunting movements. It is therefore essential to have efficient procedures regarding the rescheduling of rolling stock in case of disruptions.

The rescheduling of rolling stock is a well-researched topic in operations research. However, many existing models and methods are only able to deal with isolated and well-defined disruptions. Typically, it is assumed that there is one single disruption such as a partial or complete blockage of railway tracks, of which the exact location and duration are known and given as input for the models. In reality, disruption information becomes available dynamically as some time is required for railway dispatchers to uncover the circumstances of the disruption. Furthermore, the duration of a disruption can be uncertain, since it often depends on the time that is required for repairing railway infrastructure or resolving system failures, which can take longer or shorter than expected. Additionally, the precise location and severity of a disruption can change if the disruption turns out to be different than expected.

In practice, rolling stock rescheduling is often performed myopically, by only considering the affected train trips that take place within, e.g., the next half an hour. The schedule is then gradually adjusted as the disruption progresses. A possible drawback of this approach is that irreversible decisions can be made which are suitable for the initially expected disruption, but may lead to unnecessary cancellations or decreased passenger satisfaction if the disruption information changes. Some existing literature has looked at uncertainty in the disruption duration, such as Nielsen *et al.* (2012) and Wagenaar *et al.* (2023). However, these papers only consider situations where disruptions last longer than originally expected, whilst disruptions with shorter disruption durations are also vulnerable to irreversible decisions. Furthermore, uncertainty in the accuracy of the disruption information is not considered, as the disruption may be smaller or larger than initially projected. To the best of our knowledge, no previous research has focused on these sources of uncertainty in the context of rolling stock rescheduling. In this paper, we propose an iterative stochastic programming approach for real-time rolling stock rescheduling under uncertainty in disruption information.

# 2 METHODOLOGY

## 2.1 Problem Setting

To describe the uncertainty that arises in rolling stock rescheduling in practice, we assume that the disruption starts at time $\tau_0$. We define $L_0$ as the set of trips which is affected by the current diagnosis of the disruption. This set naturally depends on the duration of the disruption $d_0$, as a longer duration leads to more affected trips. Other aspects that can affect the set of affected trips relate to the location and severity of the disruption. For example, it is possible that the initial diagnosis of the disruption states that one track on a double-track section of the railway network is unavailable, which means that most of the trips that utilize the functioning track can still be operated. However, after a more detailed inspection, it may turn out that both tracks are broken, thereby affecting more trips than initially expected.

Based on $d_0$ and $L_0$, an updated timetable $\mathcal{T}_0$ is created, in which the trips which are directly affected by the disruption are canceled and short-turnings are created which connect train services that reach stations adjacent to the disruption to train services going in the opposite direction. We assume that the expected end time of the disruption, the set of affected trips and the updated timetable all become available at the same time and we define the information update as $i_0 := (\tau_0, d_0, L_0, \mathcal{T}_0)$. As time progresses, more information updates $i_1, \ldots, i_n$ about the disruption become available at times $\tau_1 < \tau_2 < \cdots < \tau_n$, of which $\tau_n$ denotes the time of the final information update at which the disruption information is certain. This can happen when we observe at time $\tau_n$ that the disruption is already over, or when sufficient information about the disruption is available to determine the disruption measures and end time with certainty. It is common that earlier information updates turn out to be incomplete or inaccurate descriptions of the disruption. Hence, the expected end time of the disruption and the size of the set of affected trips can both increase and decrease, as the disruption evolves and more accurate information becomes available.

## 2.2 Iterative Two-Stage Stochastic Programming Approach

Our method relates to the works of Cacchiani *et al.* (2012) and Nielsen *et al.* (2012) and revolves around iteratively re-optimizing the rolling stock schedule whilst taking into account different disruption scenarios. First, we define the timing with which the rolling stock schedule is re-optimized. In principle, an updated rolling stock schedule should be created with each information update. However, if the time between two subsequent information updates is big, a large number of decisions will be made, which may be disadvantageous if the disruption information changes. We therefore define a period $p$, which denotes how often the rolling stock schedule should be updated in case no information update arises. Hence, if no new information has been released for $p$ time units after the start time of the disruption $\tau_0$, the rolling stock schedule is updated at times $\tau_0 + p, \tau_0 + 2p, \ldots$ until the next information update arrives at time $\tau_1$. We define the set of points in time at which the rolling stock schedule is updated as $U$. Note that at update time $u \in U$, rolling stock that is assigned to trips which depart before time $u$ is fixed.

To account for the uncertainty in the accuracy and completeness of the information updates, we define a set of possible disruption scenarios $S_u$ for each update time $u \in U$. Each disruption scenario $s \in S_u$ corresponds to a (slightly) different disruption. In particular, for each $u \in U$, the set of disruption scenarios contains one scenario which assumes that the disruption occurs exactly as specified in the most recent information update. The remaining scenarios deviate by containing shorter and longer disruption durations, as well as smaller and larger sets of affected trips.

To model the problem as a two-stage optimization problem (TSOP), we define a recovery algorithm $R$ which takes as input a rolling stock schedule without disruption $x_0$, an update time $u$ and a disruption scenario $s \in S_u$, and outputs a recovered schedule $(y, x_s^u) = R(x_0, u, s) \in F_s^u$, where $F_s^u$ denotes the feasible region of the considered disruption scenario. Here, $y$ represents the first-stage decisions that are made in the recovered schedule within the next $p$ time units, and $x_s^u$ represents the second-stage decisions of scenario $s$ for the remainder of the day. To create a schedule that can at least be used until the next update time, we enforce that the decisions which are made within the next $p$ time units are identical across all scenarios. We denote the vector of second-stage decisions across all scenarios for update time $u$ as $x^u = (x_{s_1}^u, x_{s_2}^u, \ldots, x_{s_n}^u)$. Furthermore, we define recovery cost functions $c(y, x_0)$ and $d(x^u, x_0)$ for the first-stage and second-stage decisions, respectively. The TSOP that is solved at each update time $u$ is then as follows:

$$\text{TSOP}_u = \min\{c(y, x_0) + d(x^u, x_0) \mid (y, x_s^u) = R(x_0, u, s) \in F_s^u \ \forall s \in S_u\}. \tag{1}$$

In this paper, we define $d(x^u, x_0) = \frac{1}{n}\sum_{i=1}^{n} d(x_{s_i}^u, x_0)$. Previous literature on railway optimization under uncertainty has often chosen to minimize the worst-case deviation. However, in our application, the worst-case scenario would always be the scenario with the longest disruption duration and the largest set of affected and canceled trips, whilst this scenario is not necessarily most likely to occur. Hence, we choose to take into account all disruption scenarios simultaneously, thereby minimizing the average deviation in a stochastic programming approach. Algorithm 1 presents our iterative two-stage stochastic programming approach.

**Algorithm 1:** The iterative two-stage stochastic programming approach.

```
Initialization i ← 0, j ← 0, u_i ← τ_j, create scenarios S_{u_i}, initialize x_0
while Disruption is not over do
    Solve TSOP_{u_i} (1)
    if (u_i + p < τ_{j+1}) then (u_{i+1} ← u_i + p) else (u_{i+1} ← τ_{j+1}, j ← j + 1)
    create scenarios S_{u_{i+1}}, update x_0, i ← i + 1
end
```

The approach starts at time $u_0 = \tau_0$ and initializes the set of scenarios $S_{u_0}$ and the rolling stock schedule $x_0$. The TSOP is then solved iteratively while the disruption is not over. As the duration and severity of the disruption are not known with certainty, we create a rolling stock schedule that can at least be used until the next update time. The next update time corresponds either to the time of the next information update, or, if no new information update arrives for $p$ time units, the current update time plus $p$. Hence, the rolling stock schedule should be usable until at least $u_i + p$. The set of disruption scenarios is then updated by removing the scenarios in which the disruption ended before the current update time. Furthermore, depending on the information that arrives at the next update time, new scenarios with shorter and/or longer disruption durations and a smaller and/or larger set of affected trips may be added. Since the disruption information becomes more certain and accurate as time passes, fewer scenarios are included in each subsequent iteration. Finally, the rolling stock schedule $x_0$ is updated by creating a new rolling stock schedule without disruption from time $u_{i+1}$ until the end of the day, using the state of the rolling stock at time $u_{i+1}$ as specified by $\text{TSOP}_{u_i}$. The proposed approach iteratively combines pieces of the created schedules into one working rolling stock schedule and is therefore a heuristic problem decomposition.

## 2.3 Modeling and Solving the Two-Stage Optimization Problem

To formulate the rolling stock rescheduling problem, a deterministic model from the literature can be used, which takes as input a timetable that has been updated to incorporate the disruption measures and outputs a recovered rolling stock schedule. A general formulation of the TSOP at update time $u$ is then as follows:

$$\min \ c(y, x_0) + \frac{1}{n}\sum_{i=1}^{n} d(x_{s_i}^u, x_0) \tag{2}$$

$$
\begin{aligned}
\text{s.t.}\quad & Ay \geq b && && (3)\\
& y \in \mathbb{Z}^+ && && (4)\\
& A_s^u x_s^u + \bar{A}_s^u y \geq b_s^u && \forall s \in S_u && (5)\\
& x_s^u \in \mathbb{Z}^+ && \forall s \in S_u && (6)
\end{aligned}
$$

We define the feasible region of the first-stage rescheduling problem as $F = \{y \mid Ay \geq b, y \in \mathbb{Z}^+\}$ and the feasible region of the second-stage rescheduling problem of scenario $s$ at update time $u$ as $F_s^u = \{(x_s^u, y) \mid A_s^u x_s^u + \bar{A}_s^u y \geq b_s^u, x_s^u \in \mathbb{Z}^+, y \in \mathbb{Z}^+\}$. The TSOP can become difficult to solve as the number of scenarios becomes large. Initially, we will solve the TSOP with a commercial solver. In case this is computationally restrictive, we will propose other techniques for solving the TSOP. Due to the block-diagonal structure of the formulation, a Benders Decomposition approach in line with Cacchiani *et al.* (2012) seems fruitful, in which the first-stage problem corresponds to the Benders master problem and subproblems are constructed for each disruption scenario. Cuts are then derived based on the solutions to the subproblems, which are fed back to the master problem, thereby iteratively guiding the master problem to an optimal solution.

# 3 RESULTS AND OUTLOOK

We consider instances for the Dutch railway network, provided by NS. All disruptions start at 8:00 on a Tuesday morning and have a variety of different durations, circumstances, and information updates. In our instances, the railway operator has access to six subtypes of Intercity units, which amount to 321 units in total. The timetable contains 9,056 trips which are operated with Intercity units. Table 1 presents a comparison between a naive algorithm (NA), which periodically updates the rolling stock schedule under the assumption that the disruption information is always correct, and an all-knowing algorithm (AKA), which knows beforehand how the disruption will play out. It is typical for rolling stock rescheduling models to assign very large penalties to additional cancelations, i.e., canceled trips on top of the trips which are directly affected and canceled by the disruption. Hence, for both approaches, we report the remaining recovery costs separately from the number of additional cancelations. A notable gap can be observed between these two approaches. We expect that our proposed approach will significantly improve upon the naive approach and will output schedules which lie in between these two approaches. We will benchmark our approach against the rolling horizon approach of Nielsen *et al.* (2012) and present detailed computational results at the conference.

Table 1 – *Results of the naive and all-knowing rolling stock rescheduling algorithms.*

| Instance | 1 | | 2 | | 3 | | 4 | |
|---|---|---|---|---|---|---|---|---|
| | NA | AKA | NA | AKA | NA | AKA | NA | AKA |
| Remaining recovery costs (thousands) | 837 | 818 | 694 | 650 | 627 | 590 | 592 | 635 |
| #Additional cancelations | 0 | 0 | 0 | 0 | 2 | 0 | 4 | 0 |

# References

Cacchiani, Valentina, Caprara, Alberto, Galli, Laura, Kroon, Leo, Maróti, Gábor, & Toth, Paolo. 2012. Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, **46**(2), 217–232.

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.

Wagenaar, Joris, van der Hurk, Evelien, Lusby, Richard Martin, & Schmidt, Marie. 2023. Robust Rolling Stock Rescheduling. *Available at SSRN 4622539.*