



**HAL**  
open science

## Solver based heuristics for rolling stocks corrective maintenance scheduling

Tom Ray, Ronan Bocquillon, Vincent t'Kindt

► **To cite this version:**

Tom Ray, Ronan Bocquillon, Vincent t'Kindt. Solver based heuristics for rolling stocks corrective maintenance scheduling. PMS 2024 - 19th International Workshop on Project Management and Scheduling, Feb 2024, Berne (Suisse), Switzerland. hal-04696728

**HAL Id: hal-04696728**

**<https://hal.science/hal-04696728v1>**

Submitted on 13 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Solver based heuristics for rolling stocks corrective maintenance scheduling

Tom Ray<sup>1,2</sup>, Ronan Bocquillon<sup>1</sup> and Vincent T'kindt<sup>1</sup>

<sup>1</sup> Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT), France

<sup>2</sup> SNCF Voyageurs, ingénierie du matériel cluster ouest, France

**Keywords:** Rolling Stock maintenance, Mathematical Programming, Constraint Programming, Heuristics.

## 1 Introduction

Passenger trains have a very precise schedule due to the transportation demand and railway systems aim to exploit rolling stocks to their maximum capacity. Maintaining a healthy network of rolling stocks can be really difficult because it must rely on an effective maintenance schedule that does not impact the transportation plan. But while some maintenance operations are known beforehand, some repairing that could not have been predicted still needs to be done. These jobs are brought to our knowledge through the train itself. The time allowed to fix these malfunctions is relatively short (from a few hours to a few days). It is allowed to schedule a complete repair, or a partial repair named diagnosis that ensures that the train can be used in normal condition even if the operation is not completely done. The aim of our study is to find an efficient way to schedule the starting times of the maintenance jobs, completely or not, so that their due dates are met.

Section 2 defines the considered problem, next presents a Mixed Integer Linear Programming (MILP) model and a Constraint Programming (CP) model. Section 3 introduces two local search heuristics based on these models and Section 4 provides an overview of the computational results.

## 2 Problem definition and associated models

### 2.1 Problem definition

Let  $I = \{1, \dots, n\}$  be the set of jobs to schedule and  $J = \{1, \dots, m\}$  be the set of tracks available for the maintenance. Each job  $i$  has a repair duration  $p_i$ , a diagnosis duration  $p_i^d$ , a due date  $d_i$ , a tardiness cost  $w_i$ , a diagnosis cost  $u_i$  and a need for a specific infrastructure. Each track  $j$  has one or more infrastructures. We define  $V_i$  as the vector of tracks  $j$  on which job  $i$  can be assigned. Starting from the compatibility between the infrastructure requirements of the jobs, those available on the tracks and the availabilities of trains and tracks, we define  $\mathcal{T}_{ij}^d$  as the set of time intervals at which job  $i$  can start on track  $j$ . The starting times values are in  $[0, H]$  with  $H$  being the planning horizon. We define  $T_i$  as the tardiness of job  $i$  based on its starting time. Let  $S_i$  be the starting time of job  $i$  in a given schedule, then we have:  $T_i = S_i - d_i$ .

In this problem, we aim to minimize the sum of the weighted tardiness of jobs while limiting the number of performed diagnosis, especially on highly important repairs. We denote by  $\varepsilon$  the total cost allowed for the performed diagnosis. This problem is noted  $P|\mathcal{T}_{ij}^d|\sum w_i T_i$  and is strongly NP-Hard.

## 2.2 A Mixed Integer Linear Programming Model

We present a time-indexed model based on binary variables representing the times  $t$  at which jobs start. We have:

- $x_{ijt}$  : 1 if job  $i$  starts on track  $j$  at the time  $t$ , 0 otherwise,  $\forall i \in I, \forall j \in V_i, \forall t \in \mathcal{T}_{ij}^d$  ;
- $y_i$  : 1 if job  $i$  is scheduled in diagnosis mode, 0 if it is scheduled in repair mode,  $\forall i \in I$ .

We also introduce integer variables to model the starting times and tardiness that are used to compute the objective function:

- $S_i$  : the starting time of job  $i$ ,  $\forall i \in I$  ;
- $T_i$  : the tardiness of job  $i$  based on its due date,  $\forall i \in I$ .

The model is given as follows :

$$\text{Minimize } z_1 = \sum_{i=1}^n w_i \times T_i \quad (1)$$

$$\sum_{i=1}^n u_i \times y_i \leq \varepsilon \quad (2)$$

$$\sum_{j=1}^m \sum_{[t^b, t^e] \in \mathcal{T}_{ij}^d} \sum_{t=t^b}^{t^e - p_i^d} x_{ijt} \leq 1, \forall i \in I \quad (3)$$

$$\sum_{j=1}^m \sum_{[t^b, t^e] \in \mathcal{T}_{ij}^d} \sum_{t=t^b}^{t^e - p_i} x_{ijt} \geq 1 - y_i, \forall i \in I \quad (4)$$

$$\sum_{j=1}^m \sum_{[t^b, t^e] \in \mathcal{T}_{ij}^d} \sum_{t=t^b}^{t^e - p_i^d} x_{ijt} \geq y_i, \forall i \in I \quad (5)$$

$$\sum_{\substack{i'=1 \\ i' \neq i}}^n \sum_{t'=t}^{t+p_i^d} x_{i'jt'} \leq (1 - x_{ijt}) \times n, \forall j \in V_i, \forall t \in \mathcal{T}_{ij}^d \quad (6)$$

$$\sum_{\substack{i'=1 \\ i' \neq i}}^n \sum_{t'=t+p_i^d}^{t+p_i-1} x_{i'jt'} \leq (y_i - x_{ijt} + 1) \times n, \forall j \in V_i, \forall t \in \mathcal{T}_{ij}^d \quad (7)$$

$$S_i \geq \sum_{j=1}^m \sum_{[t^b, t^e] \in \mathcal{T}_{ij}^d} \sum_{t=t^b}^{t^e - p_i^d} x_{ijt} \times t, \forall i \in I \quad (8)$$

$$\begin{aligned} T_i &\geq 0 \\ T_i &\geq S_i - d_i, \forall i \in I \end{aligned} \quad (9)$$

The objective function (1) represents the total weighted tardiness that has to be minimized. The constraint (2) gives a limit on the cost induced by the performed diagnosis. Constraints (3), (4) and (5) are used to schedule each job at a single time point, taking into account the possibility of a diagnosis. Constraints (6) and (7) guarantee that a track can host only one job at a time, whether it is a diagnosis or a complete repair. Constraints (8) and (9) are used to compute the starting times of jobs and their tardiness.

### 2.3 A Constraint Programming Model

In this section, we present a constraint programming (CP) model based on interval variables. For this model we define  $J_i$  as the interval variable associated with job  $i$ ,  $J_{ij}$  as the interval variable representing the possibility of job  $i$  being scheduled on track  $j$  and  $J_{ij}^c$  (resp.  $J_{ij}^d$ ) as the interval variables representing the possibility of job  $i$  being scheduled on track  $j$  in complete repair mode (resp. in diagnosis mode).

To guarantee that exactly one track and one mode (complete repair or diagnosis) is selected for each job, we use two levels of alternatives: the first one ensures that exactly one track is selected for each job, while the second ensures that exactly one mode is selected for each job. For each track, we use a disjunctive constraint to ensure that the jobs do not overlap. Due to a lack of space, the complete model is not reported in the paper but it will be presented during the conference.

## 3 Heuristics

In this section, we introduce two local search heuristics. We chose to use matheuristics because they proved to be efficient for solving scheduling problems (see e.g. (T'kindt 2023)). We chose to use two different matheuristic frameworks to leverage at best these two models.

The concept used is the same for both as they are local search heuristics: we define a neighbourhood of solutions to explore and we try to iteratively improve our current solution, step by step, until we are stuck into a local optimum or we have reach a given time limit. Each heuristic exploits two procedures: the first one, called intensification, explores the neighbourhood of a solution. The second one, called diversification, is used in case we are stuck into a local optimum to try to jump to another neighbourhood that may be more interesting.

### 3.1 Local Branching

In this section we describe a Local Branching (LB) heuristic (Fischetti and Lodi 2003) that exploits the MILP formulation given in section 2.2. For this heuristic, we use a Hamming distance constraint to define the neighbourhood of the current solution. This distance counts every change between two consecutive iterations. Let  $s^b$  be the solution at the current iteration, and let  $x^b$  and  $y^b$  be its associated variables. We define the following sets:

$$X_0^b = \{x_{ijt} | \forall i \in I, j \in V_i, t \in \mathcal{T}_{ij}^d \text{ and } x_{ijt}^b = 0\} \quad Y_0^b = \{y_i | \forall i \in I \text{ and } y_i^b = 0\}$$

$$X_1^b = \{x_{ijt} | \forall i \in I, j \in V_i, t \in \mathcal{T}_{ij}^d \text{ and } x_{ijt}^b = 1\} \quad Y_1^b = \{y_i | \forall i \in I \text{ and } y_i^b = 1\}$$

and then the Hamming distances:

$$D^x(x, x^b) = \sum_{x_{ijt} \in X_0^b} x_{ijt} + \sum_{x_{ijt} \in X_1^b} (1 - x_{ijt})$$

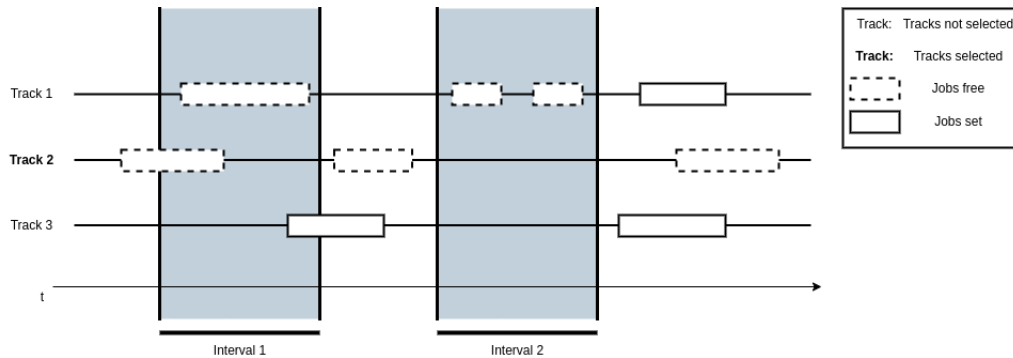
$$D^y(y, y^b) = \sum_{y_i \in Y_0^b} y_i + \sum_{y_i \in Y_1^b} (1 - y_i)$$

Then, the intensification and the diversification processes limit the number of changes in the current solution by using these distances and a parameter  $K_x$ .

### 3.2 Variable Partitioning Local Search

In this section we describe a variable partitioning local search (VPLS) heuristic (Della Croce *et. al.* 2013) that exploits the constraint programming formulation sketched in section 2.3. The neighbourhood is defined by randomly selecting multiple non-overlapping intervals and a set of tracks. At each iteration, we free everything that has been scheduled in the selected intervals and tracks. Everything else is set exactly as it is in the current solution (Figure 1). The corresponding subproblem is solved by CP. Then, we repeat as many iterations as possible within a given time limit.

**Fig. 1.** Exemple of neighborhood for the intensification process in the VPLS heuristic



## 4 Experiments

We conducted experiments on a set of randomly generated instances. This set contains 105 different types of instances defined by a triplet  $(s, m, n)$ . The parameter  $s$  determines the way availabilities between trains and tracks are distributed through simulating different maintenance sites. All the generated instances follow a structure similar to the case of rolling stocks fleets and maintenance sites located in Paris, more specifically in the northern part of the city, and represent scenarios that the planners may encounter during their work.

After comparing the two exact models, we see that for difficult instances the CP model is, on the average, worse than the MILP model. In some cases both models find an optimal solution but the MILP model struggles to prove its optimality. The use of the CP model is very time efficient for infeasible and easy instances but for big and difficult instances, while both model reach the given time limit, the MILP shows better deviations. After evaluating the two heuristics, we see that LB and VPLS improve the results of their respective parent model alone but also that on the average LB improves the results of both models. The VPLS heuristic is faster than LB in most cases but not necessarily more efficient. Therefore, VPLS is more interesting to use with a reduced time budget. But as long as efficiency is considered, LB outperforms VPLS. More detailed results will be discussed during the conference.

## References

- Fischetti M. and Lodi A., 2003, “Local branching“, *Mathematical Programming*, vol. 98, pp. 23-47.
- Della Croce, F. and Grosso, AC. and Salassa, F. and others, 2013, “Matheuristics: embedding MILP solvers into heuristic algorithms for combinatorial optimization problems“, *Heuristics: theory and applications*, pp. 31-52
- T’kindt, V., 2023, “The marriage of Matheuristics and Scheduling“, *Scheduling seminar*, [https://schedulingseminar.com/presentations/SchedulingSeminar\\_VincentTkindt.pdf](https://schedulingseminar.com/presentations/SchedulingSeminar_VincentTkindt.pdf)