Short-term rolling stock rostering and maintenance scheduling optimization

Giovanni Luca Giacco\textsuperscript{1,2}, Donato Carillo\textsuperscript{1}, Andrea D’Ariano\textsuperscript{2}, Dario Pacciarelli\textsuperscript{2}

RT-DIA-209-2014 Febbraio 2014

(1) Direzione Pianificazione Industriale, Trenitalia
Piazza della Croce Rossa, 1
00161 Roma, Italy

(2) Università degli Studi Roma Tre,
Via della Vasca Navale, 79
00146 Roma, Italy

This work was supported by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB “Advanced tracking system in intermodal freight transportation”. We thank Trenitalia managers for their suggestions and support with data collection and analysis. Corresponding author: Giovanni Luca Giacco
Tel: +39 3351510508 E-mail: g.giacco@trenitalia.it
ABSTRACT

This paper describes an optimization framework for rolling stock rostering and maintenance scheduling at a busy workshop. A new mixed-integer linear-programming formulation is provided for the maintenance scheduling problem faced by Trenitalia (train operating company) managers, with input data taken from the rolling stock rostering plan. The computational results are carried out on a Trenitalia’s maintenance site located in Naples. The solutions computed via a commercial MILP solver are compared with practical solutions. A relevant cost reduction is possible by using the proposed framework, involving both rostering and maintenance scheduling. We also show how the proposed method can be used as an effective tool to absorb real-time timetable perturbations while respecting the agreed level of service.
1 Introduction

The railway industry witnesses a variety of the problems that can be modelled and solved by using Operations Research techniques. Rolling Stock Management (RSM) is the main cost factor for Rail Undertakings. In order to reduce the costs due to railway operations, every company should address the joint problem of rolling stock rostering and maintenance scheduling since they are strongly related parts of the same problem. Maintenance optimization can be a key factor to increase the productivity of railway companies. At the same time, in a competitive globalized and multimodal market, RSM is one of the competitiveness key factors because services quality level depends on it. The strategic relevance of RSM, in particular of maintenance scheduling, is thus due to the reduction of needs (such as platforms) and to the enhancement of quality standards (such as vehicle reliability and cleaning). From our point of view the literature is focused on the study of individual sub-problems (such as vehicle scheduling and routing [1, 4, 5, 9], rolling stock rostering [7, 8], railway capacity management [11], workforce scheduling [15] and vehicle maintenance in order to reduce the occurrence of a failure [19]), while the coordination of maintenance and rolling stock scheduling is still underinvestigated.

This work studies how to manage the maintenance operations in order to increase availability and reliability of railway services. In view to the literature reviews related to maintenance operations [6, 10, 17, 20], we limit ourselves to review the following recent papers. Penicka et al. [14] provide a formal model of the train maintenance routing problem. However, the rostering problem with maintenance constraints and the maintenance routing are not addressed in this paper. Furthermore, in their case study rostering and rolling stock maintenance are considered as an appropriate method for long-distance trains only.

Màroti and Kroon [13] describe a mixed integer formulation for the maintenance routing problem in which the shunting process is considered the process bottleneck.

Budai et al. [2] discuss the preventive maintenance scheduling problem and the minimization of the time required for performing maintenance operations. Heuristic algorithms compute nearly-optimal solutions by combining maintenance activities on each track.

Wang et al. [18] propose a multiple criteria decision-making problem and evaluated maintenance strategies for different equipments.

Caprara et al. [3] study a train timetabling problem in which maintenance operations are modeled as fixed constraints. An integer linear programming formulation is proposed and solved by a Lagrangian heuristic. Tests on an Italian test bed with various train types show that maintenance constraints may seriously affect the quality of the timetabling process.

Marcos et al. [12] integrate maintenance routing requirements in a mathematical model for scheduling rolling stock. They observe that such requirements can easily be modeled but are quite difficult to consider exactly in a multi-commodity flow model. Based on this observation, they choose to deal with that problem in a heuristic manner.

Sato and Fukumura [16] present integer programming formulations for real-time freight locomotive rescheduling. The problem is to assign locomotives to the trains with the objective of minimizing the costs of the selected paths. Maintenance works are not modelled but inspection checks only.

This paper presents an original point of view on the RSM problem with respect to the state of the art that combines maintenance and rolling stock scheduling. We present a mathematical formulation for rolling stock maintenance with the aim to reduce the overall maintenance costs. This formulation is integrated in a framework for maintenance scheduling, routing and rolling stock rostering. The new formulation outperforms the one proposed in Giacco et al. [8], both in terms of solution quality and computation time.

The paper is organized as follows. Section 2 describes the problem and framework. Section 3 presents the new integer programming model for maintenance optimization. Section 4 shows the computational results. Conclusions and further research directions follow in Section 5.
2 Problem description and adopted framework

The Naples workshop is one of the key points of the Italian railway system for several maintenance services. The layout of the studied workshop consists of various indoor/outdoor tracks for light maintenance, wheel profiling, washing, cleaning, inspection and other services (see Figure 1).

![Naples workshop layout](image)

**Fig. 1 - Naples workshop layout**

Each row of Table 1 indicates a typical maintenance activity at the workshop with its identifier (Column 1), description (Column 2), expiry (Column 3) and mean duration (Column 4). The activities named I, II, and III are similar in terms of maintenance operations and duration, but have three different expiries. The activities named IV, V and VI have expiries that are multiples of 60,000 km. Specifically, Activity V incorporates Activity IV and extends it with an addition of ultrasonic flaw detection. Similarly, Activity VI extends Activity V by adding the turning of wheel flanges.

<table>
<thead>
<tr>
<th>ID</th>
<th>Short Description</th>
<th>Expiry</th>
<th>Duration [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Pantograph and Bogies Check</td>
<td>Each return to workshop</td>
<td>8</td>
</tr>
<tr>
<td>II</td>
<td>Pantograph and Bogies Check</td>
<td>7.500 Km (+/-10%)</td>
<td>8</td>
</tr>
<tr>
<td>III</td>
<td>Pantograph and Bogies Check</td>
<td>30.000 Km (+/-10%)</td>
<td>8</td>
</tr>
<tr>
<td>IV</td>
<td>Pantograph, Bogies and Wheel Flanges</td>
<td>60.000 Km (+/-10%)</td>
<td>24</td>
</tr>
<tr>
<td>V</td>
<td>Check Ultrasonic Flaw Detection</td>
<td>180.000 Km (+/-10%)</td>
<td>72</td>
</tr>
<tr>
<td>VI</td>
<td>Wheel Truing</td>
<td>360.000 Km (+/-10%)</td>
<td>108</td>
</tr>
</tbody>
</table>
Figure 2 shows two different routes in the workshop, named Routes #1 and #2. Both routes visit parts of the workshop where it is possible to provide maintenance operations (indoor terminal tracks for both routes), washing (washing tunnel) and cleaning (outdoor terminal tracks for Route #1 and terminal Gianturco tracks in Route #2). Note that Route #1 has two movements between workshop sites, while Route #2 has three movements.

In practice, the high complexity of the workshop problem requires to divide it into smaller sub-problems. Since rail companies tend to schedule first the resources with a long time of acquisition (e.g. rail tracks or asset units) and then the resources with higher degree of flexibility (e.g. human resources), this work is focused on the former type of resources.

![Figure 2 - Typical routes in the workshop](image)

The general problem addressed here can be defined as follows: Given timetables, rolling stock assets, maintenance operations, a rolling stock circulation solution has to be computed with minimum costs, that are expressed in terms of the number of used asset units, empty runs, and train movements between station platforms and the workshop.

Figure 3 shows the framework proposed to solve the problem. Each coloured module (i.e. roster, station and workshop) represents a specific sub-problem and is divided in input and outputs data, that are exchanged with the other modules. A sequential method is developed in order to integrate the solutions provided by each module. Our approach is to first solve the rostering problem for each asset unit type and then solve the other two modules involving passenger stations and workshop operations. An automatic procedure is under development to manage the interaction between the modules and a feedback information is returned in case an infeasibility is provided by some modules. In the following, we briefly describe each module and the models we developed to solve it.
2.1 Rolling stock rostering

The rostering module computes a rolling stock roster that covers a set of commercial services and minimizes the costs related to the asset units, including the empty runs. Specifically, this module optimizes the distance run by asset units of various types between consecutive maintenance operations. We suppose the timetable is cyclic and the assignment of commercial services per asset kind follows brand promise principles. As described in Giacco et al. [7], the rolling stock rostering problem corresponds to find a Hamiltonian path on a graph made by commercial services (nodes) and by feasible pairs (edges), representing service or maintenance activities to be provided. This problem is treated as a kind of traveling salesman problem with additional constraints and variables to guarantee the respect of maintenance expiry and to guarantee maintenance efficiency. The output is a cyclic roster including the schedule of maintenance activities, as time windows to provide the required maintenance, and an assigned workshop location.

Figure 3 - Interaction between the roster, station and workshop modules
2.2 Station and workshop scheduling

Maintenance operations have to be scheduled in passenger stations and workshops. There are constraints on the maintenance activities that have to be performed by each train and on the time windows that are defined in the rolling stock rostering plan. A workshop must manage some asset units simultaneously and must be able to absorb small perturbations of the circulation. A time window for the maintenance operations between two commercial services is thus considered. Generally, the time windows are bigger than the sum of all activities and the workshop has some recovery time. Also, the workshop could be considered as a space buffer to store trains. An objective function of practical interest is the minimization of the number of train movements in the workshop area.

The following section will present a new formulation for the workshop maintenance problem with minimization of train movements within the workshop. The system robustness will be investigated, i.e. the feasibility of the workshop schedule in presence of disturbances.

3 Formulation of the maintenance problem at a workshop

This section describes a big $M$ formulation for the workshop scheduling module. We call job a sequence of scheduled waiting times and maintenance operations. The problem variables are the initial and final times for each maintenance activity. We also model alternative jobs as additional variables.

3.1 Notation

- $A=[a_1, a_2, ..., a_a]$ is the activities set that the workshop can provide,
- $a$ is the number of the activities belonging to $A$,
- $U=[u_1, u_2, ..., u_u]$ is the set of asset units we have to analyze,
- $u$ is the number of asset units belonging to $U$,
- $R=[r_1, r_2, ..., r_r]$ is a feasible route (or sequence) in the workshop,
- $r$ is the number of feasible routes in the workshop,
- $t_{u_{ini}}$ is the time in which the rolling stock $u$ arrives at the workshop,
- $t_{u_{fin}}$ is the time in which the rolling stock $u$ should leave the workshop,
- $j_{ru}=[r, (t_{u_{ini}}, t_{u_{fin}})]$ is the job on route $r$ for the asset unit $u$ with time window $(t_{u_{ini}}, t_{u_{fin}})$,
- $J=[j_1, j_2, ..., j_n]$ is the set of feasible jobs,
- $n$ is the number of feasible jobs,
- $B=[b_1, b_2, ..., b_b]$ is the set of workshop resources (platforms),
- $b$ is the number of resources of the workshop,
- $x_{ruob}$ is a binary variable that assumes value 1 if the $o$-th task in route $r$ of asset unit $u$ is assigned to resource $b$,
- $o_r^{max}$ is the maximum number of elements in the sequence of route $r$,
- $y_{ru}$ is a binary variable that assumes value 1 if route $r$ is chosen for asset unit $u$,
• $M$ is a BIG integer (i.e. a sufficiently large number),
• $s_{ac}$ is a binary variable that assumes value 1 if activity $a$ if scheduled before $c$, in which $a$ and $c$ do not belong to the same asset unit,
• $w_{ac}$ is the binary variable that assumes value 1 if the end of activity $a$ precedes the start of activity $c$ by more than 1440 minutes (i.e. 1 day),
• $w_{ca}$ is the binary variable that assumes value 1 if the end of activity $c$ precedes the start of activity $a$ by more than 1440 minutes (i.e. 1 day),
• $kr$ is the number of movements within the route $r$,
• $t_{ruo}^l$ ($t_{ruo}^f$) is a non-negative variable that represents the start (end) time of the $o$-th task in route $r$ of unit $u$ begins (ends),
• $ar(o)$ is the $o$-th task of route $r$,
• $pr(a)$ is the $o$-th task of activity $a$ of route $r$,
• $dar(o)$ is the maximum allowed duration of activity $ar(o)$.

3.2 Constraints
Model constraints are of four types:
• one and only one alternative job is chosen for each asset unit,
• no overlap between activities is allowed on each resource,
• only one assignment is possible for each maintenance activity,
• time windows and maximum allowed duration of activities are respected.

We next show the inequalities that model the various types of constraints.

(1) \[ \sum_{r \in R} y_{ru} = 1 \quad \forall u \in U \]
Inequality (1) imposes that one and only one route must be chosen for each train.

(2) \[ \sum_{b \in B} x_{ruob} = y_{ru} \quad \forall (r, u, o) \in R \times U \times O \]
Inequality (2) constrains all the activities of a chosen route to be assigned to workshop resources.

\[ t_{ruo}^f - t_{ruo}^l \geq dar(o) \sum_{b \in B} x_{ruob} \quad \text{with} \quad dar(o) = \max(d_j) \quad \forall (r, u, o) \in R \times U \times O \quad \text{with} \quad j \text{ parallel tasks on the route } r \text{ and in the position } o \]
Inequality (3) is to check whether a sufficient processing time is given to each maintenance activity. The duration of activities $t_{ru_0}^r - t_{ru_0}^f$ is 0 if route $r$ is not chosen for asset unit $u$, otherwise it represents the processing time for unit $u$ on the platform $b$.

$$t_{ru_0}^i = t_{u_i}^{ini} \text{ with } o = 1$$

In Inequality (4), workshop must manage the rolling stock constraints delivered by the rostering optimizer on the time window start, including scheduled waiting times.

$$t_{ru_0}^f = t_{u_i}^{fin} \text{ with } o = 0^{\text{max}}_r$$

In Inequality (5), the workshop must manage the rolling stock constraints delivered by the rostering optimizer on the time window end.

$$t_{r_2u_2}^i \geq t_{r_2u_2a}^f - M (1 - s_{ac}) - M (1 - x_{r_2u_2pr(a)b}) -$$
$$M \left( 1 - x_{r_1u_1pr(c)b} \right) \forall b, r_1, r_2, u_1, u_2 \text{ with } u_1 \neq u_2$$

$$t_{r_1u_1c}^i \geq t_{r_2u_2a}^f - M s_{ac} - M (1 - x_{r_2u_2pr(a)b}) -$$
$$M \left( 1 - x_{r_1u_1pr(c)b} \right) \forall b, r_1, r_2, u_1, u_2 \text{ with } u_1 \neq u_2$$

Inequalities (6) and (7) are used to avoid overlaps on each resource. If two activities are scheduled on the same resource and activity $a$ precedes activity $c$, the start of $c$ must be scheduled after the end of $a$. For this reason, given a resource $b$, Constraint (6) guarantees that $t_{r_1u_1c}^i \geq t_{r_2u_2a}^f$ if $a$ is scheduled before $c$ ($s_{ac} = 1$), $a$ is assigned to resource $b$ ($x_{r_2u_2pr(a)b} = 1$) and $c$ is assigned to resource $b$ ($x_{r_1u_1pr(c)b} = 1$). Similarly, Constraint (7) guarantees that $t_{r_2u_2a}^i \geq t_{r_1u_1c}^f$ if $c$ is scheduled before $a$ ($s_{ac} = 0$), $a$ is assigned to resource $b$ ($x_{r_2u_2pr(a)b} = 1$) and $c$ is assigned to resource $b$ ($x_{r_1u_1pr(c)b} = 1$).

Let’s now suppose that overlaps of activities beyond the end of a work day should be avoided. For modelling this situation, Inequalities (8) and (9) are necessary:
If activity \( a \) precedes \( c \), both activities are scheduled on resource \( b \) and \( c \) ends after midnight (\( t^f_{r_1u_1c} \geq 1440 \)), we have to check that the end of \( c \) and the start of \( a \) do not overlap. To this aim, Inequality (8) checks if the completion of \( c \) minus 1440 is scheduled before the start of \( a \). This constraints set is valid even if \( t^f_{r_1u_1c} < 1440 \).

Inequalities (9) are used when activity \( c \) is scheduled before activity \( a \). These constraints only work if two consecutive activities on the same resource are separated by less than 1440 minutes. Alternatively, Inequalities (8) and (9) have to be replaced by Inequalities (10) – (15):

\[
(10) \quad w_{ac} > (t^f_{r_1u_1c} - t^f_{r_2u_2a} - 1440)/2880 \quad \forall r_1, r_2, u_2, a, c
\]

\[
(11) \quad w_{ac} \leq 1 + (t^f_{r_1u_1c} - t^f_{r_2u_2a} - 1440)/2880 \quad \forall r_1, r_2, u_2, a, c
\]

With Inequalities (10) and (11), the variable \( w_{ac} \) is set to 1 if the end of activity \( a \) and the start of activity \( c \) are separated by more than 1440 minutes.

\[
(12) \quad w_{ca} > (t^f_{r_2u_2a} - t^f_{r_1u_1c} - 1440)/2880 \quad \forall r_1, r_2, u_2, a, c
\]

\[
(13) \quad w_{ca} \leq 1 + (t^f_{r_2u_2a} - t^f_{r_1u_1c} - 1440)/2880 \quad \forall r_1, r_2, u_2, a, c
\]

Similarly, with Inequalities (12) and (13), the variables \( w_{ac} \) are set to 1 if the end of activity \( c \) and the begin of activity \( a \) are separated by more than 1440 minutes.
To avoid overlaps of activities on a same resource, we use Inequalities (14) and (15). If activities $a$ and $c$ are both scheduled on resource $b$, $a$ is scheduled before $c$ and there are more than 1440 minutes between the end of $a$ and the start of $c$, then the end of $a$-1440 must be previous of the start of $c$.

3.3 Objective functions

The objective function is expressed in terms of the minimization of the number of train movements. The number of train movements is computed as the number of changes of resources during maintenance routing in the workshop. Since this objective function can be formulated by using either the variables $x_{ruob}$ or the variables $y_{ru}$, in our test experiments both options will be evaluated:

- **Approach1** Min: $\sum x_{ruob}$ where $ar(o)$ is a movement
- **Approach2** Min: $\sum k_r y_{ru}$

4 Computational experiments

The results are based on real instances taken from a Trenitalia’s maintenance site located in Naples. We analyze 100 working days and we compare the programs used in real life with our model solutions. We consider all asset units types and all maintenance types. Waiting time longer than one day is considered not as a variable but as a constraint on resources availability. A pre-processing was also performed in order to reduce the problem size to a sequence of one-day problems, so we generated 100 one-day cases. For each train, we consider all possible routes in the workshop. These instances are solved by CPLEX 12 on a PC with 2.27 GHz and 4 GB of RAM. Ten minutes of computation are allowed for the branch and bound code of CPLEX.

4.1 Practical versus model solutions

This section compares the solutions found by our maintenance scheduling formulation with the practical solutions at Naples workshop. Table 2 reports the average results on the 100 instances obtained for the two approaches (i.e. the two definitions of the same objective function) of Section 3.3: the percentage of solutions that have been improved via optimization compared to the practical solutions, the percentage of optimal solutions, the average saving in terms of train movements reduction and the total computation time of CPLEX (in seconds).

From the results of Table 2, **Approach1** offers the best percentage of improved solutions with a large computation time. However, **Approach2** guarantees the best average gain, good percentage of improved solutions and a considerable smaller computation time. So, the former approach gives the largest % improvement while the latter approach can be better used for real-time maintenance control applications.
Table 2 - Comparison between models

<table>
<thead>
<tr>
<th>Approach</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved Solutions</td>
<td>60</td>
<td>43</td>
</tr>
<tr>
<td>Optimal Solutions</td>
<td>65</td>
<td>48</td>
</tr>
<tr>
<td>Average Train Movements Saving</td>
<td>1.8</td>
<td>2.6</td>
</tr>
<tr>
<td>Total Computation Time (sec)</td>
<td>531.2</td>
<td>152.9</td>
</tr>
</tbody>
</table>

The remaining experiments of this paper are dedicated to a further evaluation of the potential impact of Approach 1. We use the same set of instances but with a reduced set of routes available for each train. Figure 4 shows a deeper average comparison between the real world solutions and our model solutions in terms of the percentage of saving between 0 and 5 train movements. In the 95.8% of the 100 one-day cases a better solution is obtained by our model, while in 4.2% of cases the practical solutions are identical to the model solutions. This result can be considered as remarkable since a better solution is very often achieved via optimization. Specifically, the number of savings is often between 2 and 3 train movements. We observe that this analysis is done only for unscheduled train movements between workshop and passenger stations, that are required for the computation of a feasible train schedule.

Figure 4 - Savings of our model solutions compared to the practical solutions

4.2 Assessment of the computation time

Figure 5 shows the CPLEX performance when varying the number of jobs. The time to compute the CPLEX solution is up to around 1 minute, so we can conclude that the formulation can be quickly solved by CPLEX for real-world instances. Practical use of the proposed method can be envisaged both during maintenance planning and operational phases.
4.3 Robustness analysis

A robustness analysis is provided in order to evaluate the schedule quality in case of short-term traffic disturbances that alter the off-line plan of operations. The objective is to re-balance the use of workshop resources to compute of a feasible schedule, i.e. a schedule of all workshop operations within the time windows given by the rolling stock rostering planner and perturbed by traffic disturbances.

Figure 6 and 7 represent a sample job, made by three activities, to be scheduled in the workshop. We distinguish between a not compressed schedule, i.e. a schedule with some recovery time, and compressed schedule, i.e. a schedule with less recovery time or without recovery time. Specifically, we analyze the case in which the time window start is delayed by unforeseen events and the schedule needs to be compressed accordingly, without reducing the duration of the maintenance activities, in order to schedule all activities within the given time window.

**Figure 5** - Computation time for an increasing number of jobs in the workshop

**Figure 6** - Time window not compressed

**Figure 7** - Time window compressed
Let's now define a compression factor

\[ CF = \frac{\text{Recovery Time to be Compressed}}{\text{Total Available Recovery Time}} \]

where total available recovery time = \( t_{\text{fin}} - t_{\text{ini}} - \sum d_i \) (30 minutes in the case of Figure 6) and recovery time to be compressed is a reduction of time at workshop disposal due to traffic disturbances (20 minutes in the example case of Figure 7).

Figure 8 reports results on a subset of 63 instances for 9 different values of compressed time windows \((0.1, 0.2, \ldots, 0.9)\), obtained by delaying the start time of the time windows. For compression values equal to 0.1 – 0.4 we always obtain a feasible solution, while for 0.5 and upper values the number of feasible solutions decreases with the increase of the compression factor. However, we did not find, as we expected, an exact correlation between the computation time and the compression factor, i.e. a systematic increase of computation time when increasing the compression factor. Surprisingly, for some practical cases we have that compressed instances are easier to solve by CPLEX than uncompressed instances.

![Figure 8 - Time window compressed](image)
5 Conclusions

This paper described a new framework for rolling stock and maintenance scheduling problems. A new formulation is proposed for the workshop maintenance sub-problem, that is to find feasible schedules with a minimum number of movements within the workshop. A commercial solver is able to solve practical-size instances of this problem in a few minutes, so the proposed formulation can also be adopted to compute good quality solutions in real-time. Future research will be dedicated to the improvement of the framework, the analysis of its sensitivity to uncertainty, and the investigation of global formulations and algorithmic approaches.

References


