A robust-planning methodology for railway rolling-stock

S. Tréfond[1], H. Djellab[1], E. Escobar[1], A. Billionnet[2] & S. Elloumi[2]
[1]*SNCF-Innovation & Research department, France*
[2]*CEDRIC-ENSIIE, France*

Abstract

This paper deals with an investigation of combinatorial and robust optimization models to solve rolling-stock planning problems for passenger trains. Here robustness means that rolling-stock can better deal with some disruptions of the railway system. The proposed method is based on optimization and simulation techniques to handle the problem with imperfect information on data. The goal of the optimization module is to capture the combinatorial complexity, whereas simulation is used to evaluate and validate solution quality. The first results obtained on regional and intercity lines are promising.
Keywords: railway, disturbance, scenario, robustness, rolling-stock, optimization, simulation.

1 Introduction

The SNCF railway operator transports millions of passengers every working day. Providing a high level of punctuality and reliability to its passengers is of the highest importance for SNCF. But disruptions such as infrastructure failures or rolling-stock breakdowns can occur. Consequently, optimal solutions to the nominal transportation planning problem may turn out to be infeasible; indeed, primary delays caused by disruptions can propagate throughout the transportation plan, degrading punctuality and reliability. Therefore, planned timetable, rolling-stock (RS) roster, and crew schedule must be adapted. This can increase the target cost, and decrease the target service quality. Robustness in rolling-stock planning could make it possible to anticipate disruptions and limit delays propagation in the transportation plan.

WIT Transactions on The Built Environment, Vol 127, © 2012 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR120291

Our research focuses on this issue. We propose an optimization method to build robust rolling-stock rosters, and we evaluate their robustness with a delay propagation simulation tool.

This paper is structured as follows. In Section 2, we first describe the rolling-stock planning problem and its characteristics. Section 3 contains a review of previous works carried out about rolling-stock rosters computations and robustness issues. In Section 4, we propose a methodology to build robust RS rosters. Finally, Section 5 contains our first results. Conclusion and perspective for our future work are then presented.

2 Robust rolling-stock rostering

This section is devoted to the presentation of the real-world case study on which we focus. Roughly speaking, the railway management problem includes three major tasks:

- Train timetabling
- Rolling-stock rostering to cover timetable.
- Crew planning to operate the rolling-stock.

These tasks are interdependent, but solved separately. In France, the first one is carried out by RFF (the infrastructure manager) and SNCF (as a delegated infrastructure manager), while the two other tasks are managed by SNCF as a railway operator. Our research deals with robustness issue in rolling-stock rostering, but we point out that building a robust transportation plan requires a collective work of all the actors.

We define a task as a trip characterized by departure and arrival stations and times. For each task, we know the demand (e.g. the number of passengers to transport), and for each rolling-stock unit, we know the number of seats. To cover tasks, it is possible to provide one or more rolling-stock units, depending on the demand (e.g. number of seats). In case of multiple units (MU), the train composition has to be specified. A duty is a sequence of tasks including for each task the position, e.g. front or rear. It represents the workload of a single rolling-stock unit during a week. A rolling-stock roster is then a set of duties.

In addition, rolling-stock units must respect maintenance requirements. Based on the travelled distance or on the number of operating days, each unit needs to make periodic visits to specified maintenance depots.

For a set of tasks, the rolling-stock rostering problem consists in finding a rolling-stock roster such that:

- Each task is covered (if possible),
- Technical operating are respected
- Operating costs are minimal,
- Maintenance constraints are not violated.

To cover every task, empty rides (deadheading) can be added to the roster, in order to connect services. Although these rides cause an additional local cost for the company (e.g. related to additional energy, rolling-stock, and crew resources consumption), and increase the rail traffic in the network, they may help to find feasible solutions, and to reduce maintenance cost and roster length.

Furthermore, different types of rolling-stock units are operated by SNCF, which means that there should be as many rolling-stock rosters as unit types. Moreover, this generates compatibility constraints between different types.

In this paper, we focus on robustness in rolling-stock rostering. Throughout this paper, we shall consider that a robust roster should resist, limit delay propagation, or be easily recoverable, when a "weak disturbance" occurs. This refers to the definition we proposed:

- "*Resistance*": ability to absorb small delays immediately without any change, so that there is no impact on the transportation plan;
- "*Limitation of delay propagation*": ability to absorb small delays, to limit their propagation to the entire transportation system;
- "*Recoverability*": ability to be easily "repaired" by measures (or handling scenarios) that can solve or limit the delay propagation when facing a specific disturbance.

In this paper, we focus on "resistance" and "limitation of delay propagation" of the transportation plan. "Recoverability" will be part of our future work.

3 Literature review

Two main classes of models have been proposed in the literature to handle uncertainty: stochastic programming models [1], and robust optimization models [2, 3]. Stochastic programming requires records about production data, which may be hard to obtain. Furthermore, robust optimization may result in conservative solutions, since it aims at finding solutions that would be feasible under *all* disruptions scenarios.

In the railway context, robustness has been taken into account in timetabling problems [4, 5]. Cadarso and Marín [6] have focused on robustness in the rolling-stock planning problem, but in rapid transit networks; robustness issues are different because of high frequencies and short distances. Nielsen *et al.* [7] identified railway resource planning rules. They used them to build resource schedules, and measured their impact on robustness by simulating disturbances in the transportation plan. In addition, Liebchen *et al.* [8] studied recoverable robustness in timetabling problems. And Takeuchi and Tomii [9], and Veelenturf *et al.* [10] have studied robustness from the passengers' perspective.

At SNCF, the rolling-stock planning problem (deterministic case) has been studied and implemented for a few years [11]. In addition, Chandesris [12] gave a first definition of the robust timetabling problem, and proposed a stratified sampling method to generate disruption scenarios. Vianey [13] studied a method to limit delay propagation. It consists in adding buffers constraints to a Mixed Integer Program while solving the rolling-stock planning problem. Furthermore, a first simulation tool has been designed to simulate the propagation of disruptions in a rolling-stock roster [14].

In the studied case, we decided to model uncertainty by defining a set of scenarios. In addition, we identified indicators to construct and to evaluate robustness of transportation plans.

4 Methodology

In this section, we propose a five-stage methodology that solves the robust rolling-stock rostering problem by optimization, and then evaluates the obtained rolling-stock roster by simulation.

1. First of all, we have defined robustness indicators to evaluate rolling-stock robustness.
2. The second stage is a data modelling phase by a space-time graph.
3. Next, we generate scenarios that represent possible configurations of the system.
4. Then, a hybrid optimization procedure builds a rolling-stock roster that should be robust to the scenarios we generated.
5. Finally, a simulation procedure evaluates and validates robustness of the retained solutions.

The architecture of the target decision support systems is described in fig. 1.

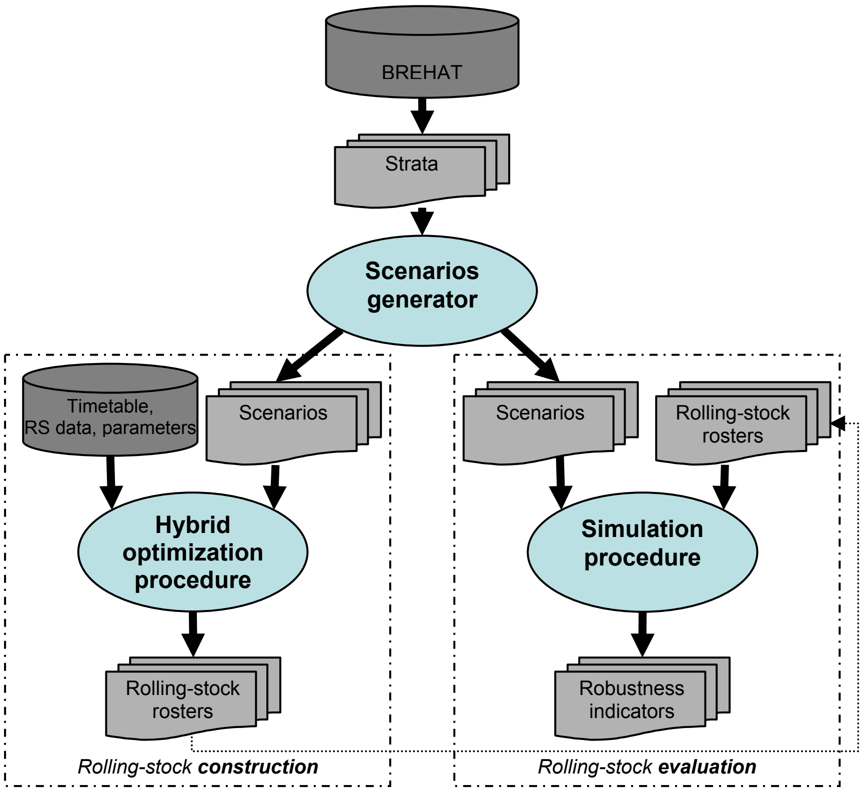

Figure 1: Architecture of the target decision support systems.

4.1 Robustness indicators

We need to quantify robustness, in order to compute robust rosters, and to evaluate them. Based on literature, and in agreement with internal experts on rolling-stock rostering, we have defined a set of robustness indicators:

- The cumulative propagated delay [13]: let a unit U cover 2 tasks T_1 with an initial delay d_1 and T_2 with an initial delay d_2. We note t_{1-2} the turning time between T_1 and T_2, so that $t_{1-2} < d_1$. After T_1, U will have a delay d_1. The turning time will absorb a part of it. Before T_2, the delay of U will be $d_1 - t_{1-2}$. We call it the propagated delay. After T_2, U will have a delay $d_1 - t_{1-2} + d_2$.
- Turning times [4]: homogeneous turning times should make the transportation plan more robust, the associated indicator is the sum of the inverses, and should be minimized.
- Number of composition changes [7]: we want to limit train units coupling and uncoupling.
- Regularity [5]: rate of on time departures or arrivals.

4.2 Data modelling phase

The problem is represented by a space-time graph $G = (V, A)$. Each node $v_i = (G_i, d_i)$ corresponds to a station G_i at time d_i. Each arc $a_i = ((G_j, d_j),(G_k, d_k))$ is associated to a task in the timetable, from station G_j at d_j to station G_k at d_k. Furthermore, conventional nodes are created: one source S_k per type k of units corresponds to the beginning of the week, and a sink T corresponds to the end.

There are several types of arcs between any two nodes (driving tasks, empty rides, waiting tasks, or maintenance tasks). The demand (e.g. number of passengers to be transported) and the maximal number of rolling-stock units are known for each arc.

4.3 Scenarios generation

The French railway system data base BREHAT contains theoretical and realised timetables. Based on the analysis of this database, we can use a stratification method to divide trains into subgroups according to their delay (no delay, a one- to a two-minute delay, a two- to a five-minute delay, etc.), their time period (e.g. peak period), and their zone (the railway network is divided into several parts).

From these strata, we can generate scenarios by stratified sampling. A single scenario consists in a set of initial delays. Each delay is represented by a weight (number of minutes) allocated to a task.

We tested two stratified sampling strategies, proportionate allocation and optimum allocation [12]. The proportionate allocation allows us to generate representative scenarios (to be realistic), while the optimum allocation generates "efficient" scenarios (to cover at best each possible disturbances configuration).

4.4 Hybrid optimization procedure

The proposed method is a combination of a MIP (Mixed Integer Programming) model that calculates the optimal number of units needed, and a local search method that constructs a rolling-stock roster by an iterative process, in order to improve robustness:

1. The MIP model corresponds to a multi-commodity flow problem. It is used to generate an optimal solution in terms of operating costs to the rolling-stock planning problem (the minimal number of units needed to cover the timetable). At this stage, we do not take into account robustness indicators or disturbances scenarios.
2. From the MIP solution, we can assign to each arc the exact number of required units of each type. In addition, we generate a scenario from the defined strata (§4.3), and allocate initial delays to the tasks in the graph. Then, a duty for a single unit corresponds to a path from the associated source to the sink in the space-time graph (§4.2). The delay associated to a path (a unit) is the sum of initial delays of the tasks minus the turning times.
3. A greedy algorithm is used to construct an initial feasible roster by generating paths. Nodes are treated chronologically. For each node, entering arcs are listed, and we choose the units' allocation that optimizes robustness indicators. Especially, for a task, we choose units with minimal propagated delays.
4. Then, a local search method is called to improve the existing solution. Paths are sorted according to their robustness indicators. We randomly choose one of the worst ones (in terms of robustness), and randomly forbid one of its tasks. The previous greedy algorithm is called, starts from the forbidden task, and builds a new solution. The best solution is saved. The local search method is called again, until the stopping criterion (e.g. a target computational time) is satisfied.
5. Finally, the best paths are linked together, in order to obtain a robust rolling-stock roster.

4.5 Simulation procedure

We implemented a simulation process so that changes of the transportation plan, and so far changes of robustness indicators, can be anticipated when disturbances occur. This helps us to evaluate a rolling-stock roster, and in case of several possible rosters to select the most robust one.

From a transportation plan and a delay model (strata), the simulation module calculates the robustness indicators by simulating delay propagation for several scenarios. The algorithm used is based on the same principle as the previous hybrid optimization method. We allocate to each task an initial delay, and further evaluate the delay of each unit at each node by propagating the initial delays.

This algorithm has been designed at a macroscopic level, and completed with basic repairing strategies, so that propagated delays can be limited.

5 Results

This section presents our first computational results obtained by applying the proposed methodology (optimization and simulation) for regional and intercity lines at SNCF. The data that were used for our experiments comprise 302 tasks corresponding to a selected week of the 2011 annual service. The numerical experiments have been performed on a 2.33GHz AMD Athlon PC with the memory of 2Go and running windows XP.

5.1 Results from the optimization point of view

Table 1 shows optimization results. It presents the robustness criteria in order of priority for each type of strata and method used. For each criterion, we indicate improvement or deterioration compared to the existing solution ("Gap").

Table 1: Optimization results.

Strata	Optimization methods	Robustness criteria							
		cumulative delay propagation	Gap (%)	rate of on time trains	Gap (%)	turning times	Gap (%)	number of composition changes	Gap (%)
Representative	Existing solution	00:15:36	-	99,12%	-	4,65	-	84	-
	Hybrid method (0 min)	00:01:00	**-93,59%**	99,33%	**0,21%**	4,85	4,30%	82	**-2,38%**
	Hybrid method (30 min)	00:00:00	**-100%**	99,33%	**0,21%**	4,72	1,51%	82	**-2,38%**
Efficient (1)	Existing solution	00:33:08	-	95,31%	-	4,71	-	84	-
	Hybrid method (0 min)	00:13:30	**-61,47%**	95,00%	-0,33%	4,85	2,97%	82	**-2,38%**
	Hybrid method (30 min)	00:08:00	**-75,86%**	95,37%	**0,06%**	4,73	0,42%	82	**-2,38%**
Efficient (2)	Existing solution	00:31:35	-	94,13	-	4,68	-	84	-
	Hybrid method (0 min)	00:00:00	**-100%**	95,37	**1,32%**	4,76	1,06%	82	**-2,38%**
	Hybrid method (30 min)	00:00:00	**-100%**	95,37	**1,32%**	4,55	**-3,40%**	80	**-4,76%**

The "existing solution" has been obtained by an optimization method without robustness. The "hybrid method (0 min)" refers to the hybrid optimization method without local search, and the "hybrid method (30 min)" corresponds to the same method with local search limited to thirty minutes.

In addition, compared solutions have the same cost, especially the same number of units.

Representative strata come from a proportionate allocation, and efficient strata have been created according to an optimum allocation strategy. It means that representative strata are supposed to generate realistic scenarios, while efficient strata should generate worse cases (with longer delays).

For each type of strata, the hybrid optimization method has significantly improved the cumulative propagated delay, which was the priority criterion. Furthermore, the regularity and the number of composition changes have been improved by the hybrid method with local search. Hence, the proposed method seems to dominate the existing method in terms of robustness criteria.

5.2 Results from the simulation point of view

Then, we want to test solutions by simulation. Solutions have been optimized by taking into account one scenario, and we would like to know if it remains robust in case of different scenarios. For each solution, 100 new scenarios are generated, and for each scenario, the simulator calculates the robustness indicators. Table 2 presents average values of the indicators for each solution.

Table 2: Simulation results.

Strata	Optimization methods	Robustness criteria							
		cumulative delay propagation	Gap (%)	regularity	Gap (%)	turning times	Gap (%)	number of composition changes	Gap (%)
Representative	Existing solution	00:15:36	-	99,12%	-	4,65	-	84	-
	Hybrid method (0 min)	00:16:34	5,84%	99,06%	-0,06%	4,95	6,45%	82	**-2,38%**
	Hybrid method (30 min)	00:14:53	**-4,59%**	99,07%	-0,05%	4,89	5,16%	82	**-2,38%**
Efficient (1)	Existing solution	00:33:08	-	95,31%	-	4,71	-	84	-
	Hybrid method (0 min)	00:14:42	**-58,55%**	95,42%	**0,11%**	5,02	6,58%	82	**-2,38%**
	Hybrid method (30 min)	00:31:23	**-6,79%**	95,37%	**0,06%**	4,98	5,73%	82	**-2,38%**
Efficient (2)	Existing solution	00:31:35	-	94,13%	-	4,68	-	84	-
	Hybrid method (0 min)	00:30:59	**-1,90%**	93,88%	-0,25%	4,98	6,41%	82	**-2,38%**
	Hybrid method (30 min)	00:27:48	**-14,56%**	94,02%	-0,11%	4,94	5,56%	80	**-4,76%**

The propagated delay of the hybrid method solutions with local search has been improved compared to the existing solutions. Furthermore, the gain seems to be better when the optimization method used efficient strata. It would mean that solutions are more robust when they have been optimized for “worse cases” scenarios.

However, robustness indicators are not systematically improved. Especially, we would expect better results with local search (30 min) than without (0 min), which is not always observed. It can be explained by the use of only one scenario during the optimization stage. It would be interesting to take into account several ones. Moreover, these tests were run with strata based on trains’ delays only. We would need accurate strata (also based on periods and zones, §4.3), to generate more realistic scenarios.

6 Conclusion and perspective

In this paper, we outlined a three-stage methodology to solve the rolling-stock planning problem, and to ensure its robustness. The first stage consists in generating delay scenarios. Then, the hybrid optimization method calculates the minimum number of rolling-stock units required, and aims at building a robust roster with a local search method. Finally, the simulation procedure relies on simulation techniques to evaluate and to validate robustness of the proposed rolling-stock rosters. The first results are promising: on real data, our methodology quickly builds a roster that minimizes operating costs to cover a given timetable, while improving robustness. Further numerical experiments are needed.

The proposed system is intended for a strategic decision making tool. It would help experts to reduce the time of the planning process. Meanwhile, it would allow them to increase the flexibility, and to react faster to changes in the environment.

Future research on the rolling-stock rostering problem could aim at defining an alternative neighbourhood system, and integrating new indicators (see (e.g. Robustness Indices based on Passengers' Utilities [9]). We also plan to investigate the use of column generation technique to integrate the whole set of constraints (for instance maintenance) at the same time, and compare both approaches (local search and column generation) in terms of robustness and computational time.

References

[1] Birge, J.R. and Louveaux, F., Introduction to Stochastic Programming, Springer-Verlag, New York, 1997.
[2] Mulvey, J., Vanderbei, R. and Zenios, S., Robust Optimization of Large Scale Systems, Operations Research, Vol. 43, No. 2, 1995.
[3] Bertsimas, D. and Sim, M., Robust Discrete Optimization and Network Flows, Mathematical Programming B(98), 49-71, 2003.
[4] Vromans, M. J. C. M., Dekker, Rommert and Kroon, L. G., Reliability and Heterogeneity of Railway Services, ERIM Report Series Reference No. ERS-2003-090-LIS, 2003.
[5] Hofman, M.A. and FrØlund Madsen, L., Robustness in train scheduling, Master thesis, IMM, DTU, September 2005.
[6] Cadarso, L. and Marín, A., Robust rolling stock in rapid transit networks, Computers & Operations Research 38(8), 1131-1142, 2011.
[7] Nielsen, L.K., Kroon, L.G. and Maroti, G., Absorption Robustness of Railway Resource Schedules, ARRIVAL-TR-0113, 2007.
[8] Liebchen, C., Lübbecke, M., Möhring, R. and Stiller, S., The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications, *Robust and Online Large-Scale Optimization*, *LNCS*, 5868, pp. 1-27, 2009.
[9] Takeuchi, Y. and Tomii, N., Robustness Indices based on Passengers' Utilities, WCRR2006, Montreal, 2006.
[10] Veelenturf, L.P., Nielsen, L.K., Maroti, G. and Kroon, L.K., Passenger Oriented Disruption Management by Adapting Stopping Patterns and Rolling Stock Schedules, 4th International Seminar on Railway Operations Modelling and Analysis (IAROR), Rome, February 2011.
[11] Marcos, N., Modélisation et Optimisation de la Gestion du Matériel Roulant à la SNCF, Thèse de doctorat, Université Paris 13, SNCF, 2006.
[12] Chandesris, M., Prise en compte de phénomènes stochastiques dans l'optimisation des plans de production: Application au domaine ferroviaire, Thèse de Doctorat, Laboratoire de statistique théorique et appliquée de l'université de Pierre et Marie Curie (Paris VI), SNCF, 2005.
[13] Vianey, S., Robustesse dans la planification de la production ferroviaire, SNCF internal report 2007.
[14] Rovetta, C., Évaluation de la robustesse dans la planification ferroviaire, SNCF internal report 2008.