Modelling Train Re-scheduling with Optimization and Operational Research Techniques: Results and Applications at SNCF.

L. Gély, G. Dessagne and C. Lérin ¹

¹SNCF Innovation and Research Department, Paris, France

Abstract

Train rescheduling deals with the real-time adaptation of train schedules in order to minimize the delay consequences of traffic perturbations. From a mathematical point of view, this is a difficult, combinatorial and strongly constrained problem. Nowadays, human operators are facing this problem using great expert knowledge.

We first studied a mathematical model formalizing the train re-scheduling problem and the railway operations. Given this robust and efficient model, several solving algorithms have been developed and evaluated. These algorithms make use of operational research techniques, such as linear programming and evolutionary algorithms. Hybrid techniques are also tested, which gives encouraging results. These algorithms are detailed. Their integration in a re-scheduling module is also explained. Another part of this article is dedicated to the SNCF current applications of these studies. The first range of applications consist in software environments for the evaluation of punctuality performance of new traffic management systems. The second range of applications deals with projects prototyping operational decision support tools for specific cases of incident management.

Introduction

Train re-scheduling deals with real-time adaptation of train timetables when incidents occur. The main objective is to minimize the delay consequences of traffic perturbations. To do so, efficient decisions must be taken as soon as possible. In case of a small perturbation, decisions mainly concern re-routing, and extra stops in order to pass fast trains before heavy freight trains. In case of major incidents, decisions such as re-routing or cancellation of train missions may be considered.

Experts are facing this challenge daily and already perform regulation for years using great expert knowledge. However we must now consider that traffic density is getting very high in several areas. If an incident occurs in these crowded areas, it becomes more and more uneasy to avoid large scale perturbations. Management areas also tend to grow, in order to improve the global management of main corridors. In consequence, traffic management complexity is rising. A major challenge today is to study efficient tools to help expert’s decisions in the re-scheduling process of tomorrow.

This document presents the first investigation of the problem, leading to its modeling and solving as a Mixed Integer Program. A software module has been built which allows to perform interesting studies on new concepts of traffic management in simulation with real-world data. But we still need to improve our tools to meet the requirements for real-time processing of large scale problems. Leading to this objective, we present another solving method based on stochastic optimization techniques, which is currently studied in collaboration with INRIA (the French national institute for research in computer science and control). Finally, we will point out innovative decision support tools that are currently being prototyped for sub-domains of the global re-scheduling problem: the first one intends to help experts in case of major incidents and the second one helps to improve the traffic flow.
Mathematical model for the train re-scheduling problem

From the mathematical point of view, management of network regulation is a NP hard problem. This refers to situations where we cannot rely neither on computers improvement nor on “raw force” to solve such problems, as the number of trains keeps growing. This is why operational research is the most relevant answer to our problem. Modeling is the first step.

The model used for the train rescheduling problem is an evolution of an existing model built at SNCF for DEMIURGE [3,5] which is a decision support tool allowing to make railway capacity studies. It is used daily by SNCF experts and has proven its reliability.

Network representation

As illustrated by Figure 1, the railway network is represented as a graph where nodes are special points of the network (such as a junction) and where edges hold tracks. Complex stations may be represented individually as a graph describing all potential combination of incoming, inside and outgoing tracks.

![Figure 1: Network as a graph](image)

Variables

We consider a mixed integer programming model, containing both numerical and binary variables to describe railway network operations (including re-scheduling decisions).

First we must consider, for each train, at each node, the times of arrival and departure. Using the second as a time unit for schedules, these variables will be numerical values in the model. In the following section, D will be the departure time of the train and A the arrival.

Second we must deal with regulation decisions, represented by pure 0-1 values. We consider three kinds of actions: re-ordering, track choice and extra stop:
  - re-ordering variable expresses whether a given train passes before another at a node of the network (a crossing for example),
  - track choice variable expresses whether the train uses this track or not,
  - extra stop variable expresses whether the train stops in the new schedule while it shouldn’t in the original one. This is mainly used to allow fast trains to pass slower trains stopped at loops or sidings.

From a mathematical point of view, decisions variables are the hard ones to get (and require most of the calculation time).
Constraints

This subsection enumerates constraints representing the traffic management process. The following constraints are, obviously not exhaustive, however we represent the main ones necessary to the construction of a new schedule (following an incident):

The following constraints are associated with each train (c) at each node (n) of the network.

1. **Original schedule:**
   A train cannot leave a station earlier than previously defined in the original schedule:
   \[ D(c,n) \geq Do(c,n) \]  \hspace{1cm} (1)

   Due to commercial and operating reasons (maintenance, for example) stopping times must be bounded:
   \[ \text{Min\_stop} \leq D(c,n) - A(c,n) \leq \text{Max\_stop} \]  \hspace{1cm} (2)

2. **Headways:**
   In order to prevent conflicts, trains must be spaced. Considering each type of potential conflict between each pair of trains we impose a specific separation time between departures and/or arrivals of the two trains:
   \[ \text{Min\_spacing} \leq A(c1,n) - A(c2,n) \quad \text{and} \quad \text{Min\_spacing} \leq D(c1,n) - D(c2,n) \]  \hspace{1cm} (3)

3. **Running times:**
   According to infrastructure and rolling stock characteristics, there are maximal speeds for each train on each track. As we are not allowing trains to slow under a minimal speed too, this is equivalent to consider a minimal and a maximal running time needed to reach one node from another:
   \[ \text{Min\_time} \leq A(c,n2) - D(c,n1) \leq \text{Max\_time} \]  \hspace{1cm} (4)

Of course, other specific constraints must be treated: connections between two trains, shuttles, ...

Finally, due to decision variables, most of the constraints above must be refined. For example, each spacing constraint must also take into account the order between the trains and the choice between many tracks.

**Objective function**

The objective function showed here, intends to minimize the total accumulated delays. We sum the delays between the original timetable (before the incident happens) and the new one (solution of the calculation), over all the trains and all the nodes:

\[ f = \sum \text{delay}(c,n), \text{ where } \text{delay}(c,n) = A(c,n) - Ao(c,n) \]  \hspace{1cm} (5)

However, the objective function can be easily modified or fine tuned as we will see further on.

**Size of the problems**

This gives a great amount of constraint. In order to give an idea of the number of variables and constraints, we consider a study on the railway between Tours and Bordeaux. Indeed, the end of the “LGV Atlantique” (144 km) handles an interesting mixture of trains, moreover that we choose the timetable corresponding to July 1st, 2004, between 15h00 and 23h00, in order to include the most crowded times of the day. The study consisted in injecting incidents in this original schedule (some of them really happened that day) and test our methods.

The entire dataset held 99 trains (39 TGV and 23 freight trains), 43 nodes and 42 tracks. For a specific incident, this gives a problem with up to 220,000 variables and 380,000 constraints. Obviously, trying all possibilities, looking for the best one is intractable. We will now have a look at the algorithms we can use successfully to solve the problem.
Solving method

Our first objective was to use an exact method. Indeed, we needed optimal solutions not only to assess maximal performances of new traffic management concepts but also to value the quality of other potential solving methods.

Linear Programming Algorithm

We use the Ilog Cplex tool to solve our problem. It uses linear programming and works as follows: Algorithms first check all constraints and simplify as much as possible the problem (from a mathematical point of view). Then the software looks for a first solution. When a first solution has been found it tries to refine it and find the best solution.

![Linear Programming Algorithm](image)

- With the example showed previously, the **pre-solve** phase gives a reduced problem of approximately 64,000 variables and 300,000 constraints (and this takes only 2 seconds of calculation).
- The search of a **first feasible solution** is different from the one looking for the best solution. This is the critical stage: while the software is searching no solution can be provided to the expert. However we can also directly feed the software with an already known feasible solution and bypass this step.
- For the final phase, Ilog Cplex uses branch & cut algorithm to find the **optimal solution**.

In addition, performances have been greatly enhanced since first experiments with: some advanced user settings of the solver, the initialisation with a first solution concerning decision variables and model improvements (two mathematical descriptions can be equivalent although corresponding calculation time may be very different).

On the one hand this is an excellent method for the “re-scheduling module” we developed (cf. next section), as this is an efficient way to find the **optimal solution** that allows to compare different re-scheduling strategies configurations. This method also gives a reference value for non optimal solving methods. On the other hand, for an operational decision support, the calculation time should be improved to meet requirements for larger scale dataset and real time processing. That's why improvements and other solving methods are still studied, as we will see further on.

The Re-scheduling module

A software module called “re-scheduling module” is already running with the model and the algorithm described above. It has proven the feasibility of this approach and allows to perform studies on traffic management with real world data.

The experiments can be done within the virtual world of the SISYFE train simulator [4]. This software simulates the train running in detail and shows how the system would evolves in real life. This operation is carried out by means of algorithms that takes into account the dynamic performances of trains, the signalling patterns, the distances to be covered and the tracks layout. Different kinds of driver’s behaviour can also be simulated.
The re-scheduling module is part of a larger traffic control simulation environment named LIPARI [3,6]. The LIPARI prototype contains three different modules. The first one seeks to detect abnormal situations, comparing original timetable with “reality” produced by the train simulator. As soon as an incident is detected, it sends real data to the re-scheduling module. This module works just as showed above. The solution given by this module is a new timetable which minimizes delays. Last module seeks to calculate new orders concerning trains whom schedule has been modified between original and estimated timetable (with a new speed or a new routing for example). Theses orders are then sent to the simulator.

The SARDAIGNE module [1,2] allows to randomly inject incidents in a given timetable and statistically processes the results. The distribution of incidents can be estimated from an incident database which records every incident already happened on the network.

This module has already helped to develop innovative projects linked with re-scheduling. For example it allowed us to study a new policy of traffic management called “pipe rule” [7].

A real-world case study example

Since the rise in complexity in re-scheduling problems, the management rules of the network concerning disturbed situations must be clearly defined. This led to test a new kind of re-scheduling rule. Its principle is to associate to each train a “pipe” which represents its original schedule added of a given maximal delay all along the path. The traffic manager must avoid that a train runs out of its pipe at the exit point of the managed area. Other priority rules apply when it cannot be avoided that trains exit from their pipes, or when there are equality of pipes.

We used the re-scheduling module to validate and analyze the impact of such rules on train punctuality, with real-world data, according to:
- the number and the size of the pipes,
- the allocation of pipes between the types of trains,
- different characteristics of infrastructures (passing tracks, two direction tracks, …)
- different kind of incidents.

To do so, the objective function has been tuned. Given this new objective function, the best solutions were found by the re-scheduling module and we were then able to figure the maximal gain due to these new rules.

The results have been approved by experts and the new policy of traffic management is to be used on the field soon. These new rules would also help to figure parameters in the objective function with a valuation of each train delay (economical consequences are different regarding the type of train).
Towards operational decision support

In this section, we present a new solving method which should help to improve performance of our re-scheduling module. Next, we point out innovative decision support tools that are currently prototyped for sub-domains of the global rescheduling problem.

The Evolutionary Algorithm

Evolutionary Algorithms [8] are part of stochastic optimization techniques, which has known many studies during the last decade. These techniques are usually used to browse large, irregular search spaces to provide quickly solutions (although usually not optimal ones).

Here, we consider a population of trains permutations. These permutations describe the order that will be used to insert each train iteratively in the schedule. However the set size of permutations can become huge when the number of trains is increasing. The key role of Evolutionary Algorithm would be to find quickly interesting subsets of permutations.

The algorithm proposed by INRIA takes place into a main loop divided into three steps. First, the Genetic Engine is to give a (new) subset of permutations (the genotypes of the Evolutionary Algorithm). These permutations are turned into proper schedules (the phenotypes) by the scheduler. The schedules are then evaluated for total accumulated delay, to provide fitness. This helps to value the quality of the permutations and then is used to generate a new subset. The population goes through the loop until it reaches a satisfactory solution.

![Evolutionary Algorithm](image)

**Figure 3: [Evolutionary Algorithm]**

- **Genetic Engine & permutations:**
  The genotype is an ordering set of trains (a permutation). A standard replacement scheme is borrowed from Evolution like strategies: a population of μ parents produces λ children using variation operators. Among the set of both parents and offspring, μ individuals are chosen as the new parents for the next generation (the selection pressure comes from the replacement procedure). Here, the mutation consists in swapping two elements of the permutation, which means two trains exchanging their rank (they are inserted) in the schedule. This swapping operation can be repeated T times. Drawing an analogy from Simulated Annealing, we set the trade-off between exploration and exploitation by controlling the number of times this swap is performed by each mutation operation. This parameter T (for Temperature) remains fixed during the first n0 generations, then decreases.

- **Semi greedy scheduler:**
  It seeks to create a schedule by iterative insertion of trains while respecting all above constraints. (these insertions are done by following the order given by the permutation). “Greedy” means that when a train is placed in the schedule, it uses the available resources left by previously scheduled trains, considering the best consequence only for this train at this step.
The train is inserted node by node, checking at each time all possible routes in between and choosing the one that permits the earliest departure. To do so the algorithm must detect potential conflicts (e.g. a spacing constraint). Two different techniques can be used to solve a conflict: “moving forward in time” and ”kicking obstacles”.

1. Moving forward in time: when a conflict occurs (e.g. a train uses a resource unreleased by its predecessor) the troubling variable is moved forward in time until the conflict disappears.
2. Kicking obstacles: if the previous technique cannot be applied (e.g. a train uses a resource too soon before another train and moving forward enough in time would imply taking over the other train) the obstacle is removed (i.e. the other train creating the conflict) from the schedule.

- Delay calculation:
The schedules obtained are then evaluated by comparing them to the original one and summing all the delays. This give us a way to value each permutation of the subset (corresponding to each schedule). This new information (on “parents”) can now be used by the Genetic Engine to find a new subset of permutations (the “children”).

The main loop is continued until a good enough solution is reached or until a maximum number of generations have been considered.

Used alone, this algorithm can get quickly a first feasible solution. All the more with some enhancements as we will explain later. But this first solution is suboptimal and then convergence is usually too slow. In addition it is impossible with this method to value the distance to the optimal solution.

Initialisation enhancement

We showed previously how the main loop is works but not how to initialize it. The most common way to do so is to generate randomly the first subset (generation) of permutations. Another interesting way is to provide problem-specific knowledge by modifying the algorithm’s initialization phase heuristically. Here, the most interesting way is to build the initial population around a previously known solution.

The key idea is that part of the problem can be solved beforehand because it does not depend on the particular incident instance being solved. One just needs to run the Evolutionary Algorithm once and for all with an empty perturbation. It results in an good starting point called “inoculant”. This solution (the best individual obtained after a large number of generations) will be used as a suitable starting point whenever a new instance (new incident) needs to be solved.

Hybrid Algorithm

The main interest is to combine the ability to get quickly a first solution with evolutionary algorithms and then use linear programming solver efficiency to reach the best solution.

This was the second part of collaboration with INRIA: as a first step, the evolutionary part of the algorithm is used to quickly obtain a good (but sub-optimal) solution. Then, the best individual in the population after K generations is fed to CPLEX as an initial solution (a first feasible solution). This is the starting point for its search for the global optimum.
This hybrid algorithm seems to outperform the exact method. We will now soon evaluate this approach on full studies set and may insert it into the final version of the re-scheduling module.

**Some current projects for operational decision support tools**

A first project deals with decision support for re-routing trains when a major incident occurs. The first objective is to identify residual capacities on pre-defined potential routes. The second objective is to schedule re-routed trains while minimizing delays. Solutions take into account many of the previous constraints but the infrastructure is less detailed. From this point of view, the model is close to a capacity tool one. Exploratory developments have led to a prototype that is currently evaluated on real-world cases of traffic disruption along main axes.

Another project seeks to manage closely a railway node to prevent the conflicts between trains in order to ensure fluidity of the traffic. This is done by means of re-scheduling of trains and transmission of speed orders to drivers. When a conflict occurs, one of the trains may be stopped or at least strongly slowed because of the signalling system. In order to avoid a time consuming sequence of deceleration, stop and re-acceleration, the potential conflict has to be detected beforehand and the train has to be slowed earlier so that it could go free of speed restriction through the potential conflicting point. Early calculations showed that the traffic flow can be improved this way. The first prototype on the field is intended to be tested on the “Rémilly-Baudrecourt” junction, between the eastern side of the “LGV Est” and the existing railway network. We work to simulate its behavior with the LIPARI tool.
Conclusion and perspectives

We developed a reliable model dedicated to formalize real railway operations. We used an exact method to solve realistic problems to the optimum and we developed a software module which allows to perform studies on re-scheduling rules and new concepts of traffic management with real-world data.

The collaboration with INRIA showed that we can even improve the performance of this existing method. This can be done with combining it with evolutionary algorithm to give us a first feasible solution more quickly.

Research projects currently being performed at SNCF on sub-domains of the global re-scheduling process lead to operational decision support systems in the short term. Previous exploratory works have indeed shown the feasibility and efficiency of such systems.

Further works deal with real-time simulation of incident consequences and development of new innovative models and solving methods which are intended to reach maximal efficiency in real life conditions. These works will lead to an operational decision support system for the global re-scheduling problem, connected with deployment of both computer based control centres and automated information systems.

Acknowledgements

We are grateful to INRIA for our fruitful collaboration and we would like to personally thank Y. Semet and M. Schoenauer.

We also acknowledge our gratitude to the SNCF Innovation and Research department staff involved in the projects referenced in this paper.

References