Version of Record: https://www.sciencedirect.com/science/article/pii/S0377221720302484
Manuscript_b3c3f07e063bc48ffd632094e0f86846

A multi-objective optimization-simulation approach for real time rescheduling in dense railway systems

Estelle Altazin[1,2] Stéphane Dauzère-Pérès[2,3] François Ramond[1]
Sabine Tréfond[1]

[1]SNCF, Innovation and Research Direction
Paris, France
Email: estelle.altazin@grenoble-inp.org, {francois.ramond,
sabine.trefond}@sncf.fr

[2]Mines Saint-Etienne, Univ Clermont Auvergne
CNRS, UMR 6158 LIMOS
CMP, Department of Manufacturing Sciences and Logistics
Gardanne, France
E-mail: dauzere-peres@emse.fr

[3]Department of Accounting, Auditing and Business Analytics
BI Norwegian Business School
Oslo, Norway

Abstract

Rescheduling trains in dense railway systems to cope in real time with limited disturbances is a challenging problem with multiple conflicting objectives and various types of decisions. Based on the French railway system in the Paris region, this paper proposes an approach combining multi-objective optimization, to select rescheduling decisions, and macroscopic simulation, to compute the objectives associated to these decisions. Possible decisions include canceling or short-turning trains and skipping or adding stops. Three main objectives are optimized to propose multiple solutions to the decision makers: The recovery time, the quality of service for passengers and the number of decisions. Two greedy heuristics are presented whose results on actual data are compared with a full enumeration method. The multi-objective feature of the approach is also analyzed. The implementation and successful validation in real life of a decision-support tool, that is now implemented, is discussed.

Keywords: Transportation, Real time rescheduling, Optimization and simulation, Multi-objective, Dense railway system

1. Introduction

The Paris area covers only 2% of the French territory, but is home to 18% of the French population. More than 10 million public transportation trips are performed each day in the Paris region. SNCF Transilien is a major operator of Paris suburban trains, carrying daily more than 3.5 million commuters. As the number of passengers is increasing by 3% each year since 2000, trains have been added, leading to a congested situation with up to 32 trains per hour on the busiest part of the infrastructure. Daily operations of the Transilien network are difficult, and a minor incident, such as a longer stop at a station when many passengers want to board a train, can cause a delay. Due to the congested network, buffer times are short and a small delay during running, dwell or turning times rapidly propagates along the line, and to other lines via shared resources. Various actions are applied by dispatchers, in real time, to adjust plans to the traffic.

Transilien is a railway system: Tracks can be shared with other railway operators, drivers and rolling-stock are shared between Transilien lines and different services (trains with different stopping patterns) exist on each line. Yet, Transilien is a mass transit system, with characteristics close to subways: The frequency is very high (up to one train every two minutes on some lines) and commuters do not aim for a certain train but board the first train serving their destination station. Operational actions such as canceling a train or skipping a stop can thus be taken without severely deteriorating the quality of service offered to passengers, as long as the anticipation of an action allows operators and passengers to be informed of the associated changes.

Many stakeholders are involved in the Transilien transit system, making the decision-making process complex. The Paris region transportation authority defines a number of performance criteria, the most important one being the number of delayed passengers. The infrastructure manager goal is that each train respects its scheduled path without delay. Passengers expect frequent and comfortable trains with a reliable service, whatever the train path, the rolling-stock unit or the driver. Transilien must combine these different objectives while maximizing its operational efficiency and reducing its costs.

During real time operations, Transilien decision makers need to quickly determine the best actions from a system-wide perspective. In Altazin et al. (2017), we propose an Integer Linear Programming model to optimize both the recovery time and a relatively simple passenger criterion with only type of actions: Stop skipping. The model was validated on industrial data and was useful for showing the potential of a rescheduling tool to decision makers. Endorsed by Transilien, the research was pursued to overcome the limitations of the first model by developing the novel approach presented in this paper and to support the development of a real time decision support tool. In particular, more actions had to be integrated, the approach had to be fast enough to propose actions in real time, and the passenger criterion had to consider the reaction of passengers to the operational actions. It was also essential to tackle the problem with a multi-objective approach, as decision makers need to have multiple solutions to choose from, associated to different trade-offs on the evaluation criteria. This paper proposes an optimization-simulation approach that tries to answer these requirements, presents numerical experiments on actual data and discusses the successful real-life experiments conducted with decision makers using the decision support tool, that led to the actual implementation integration of the tool.

A recent review covering the literature related to our problem, i.e. rescheduling approaches both in railway systems and in public transportation systems, can be found in Altazin et al. (2017). We summarize here how our work is positioned, provide some of the references in Altazin et al. (2017) and discuss some newer papers. Compared to research on the train dispatching problem, also known as train path rescheduling (see for example Meng and Zhou (2014) and Samà et al. (2016)), trains cannot be rerouted or reordered in our problem because overtaking is not allowed with the track layout. We are considering reservicing actions that modify the traffic plan (adding or skipping stops, canceling trains, short-turning trains) that are included in some train dispatching approaches, such as Sato et al. (2013), Veelenturf et al. (2016) and Ghaemi et al. (2018), but not in the context of dense railway systems. Compared to research on the delay management problem (see for example Dollevoet et al. (2015), Corman et al. (2017) and Schön and König (2018)), our approach does not consider passenger connections since we focus on rapid transit systems with high frequencies. As in the research on the rolling-stock rescheduling problem (see for example Kroon et al. (2015) and Cadarso et al. (2013)), we are considering rolling-stock that is a critical resource because of the short turning times in rapid transit systems.

However, we only deal with small disturbances, usually causing less than 10-minute delays and no blockage, and other types of decisions and more objectives are considered. Compared to research on the crew rescheduling problem, we assumes that drivers operate the same rolling-stock unit during the rescheduling horizon, that is limited to a few hours. The above mentioned problems are on conventional railway systems. Because the Transilien system has a high frequency of trains, short distances between stations, short turning times for rolling-stock and overtaking or rerouting trains is not allowed, reservicing decisions commonly used in subway or in bus traffic are considered. Rescheduling in public transport systems is often referred to as real time control strategies (see for example Eberlein et al. (1999), Eberlein et al. (1998), Gao et al. (2016) and Nesheli and Ceder (2015)). However, our problem is different, because of the additional constraints associated to the infrastructure and the rolling-stock and because multiple objectives have to be considered, whereas the objective is usually only focusing on passengers in public transit. More recently, Veelenturf et al. (2017), Wagenaar et al. (2017) and Zhu and Goverde (2019), as in Kroon et al. (2015), also consider passenger objectives by modeling the dynamic flows of passengers. Wagenaar et al. (2017) propose a rolling-stock rescheduling model that allows for dead-heading trips to limit the number of deleted trains for relatively large disruptions. Veelenturf et al. (2017) introduce a rolling-stock rescheduling approach where stops can be added in some stations to minimize the impact on passengers. Gao et al. (2017) propose a retiming model, and thus the stopping and travel times, for small disturbances (at most 5 minutes). Their goal is to minimize the gap between the planned timetable and the actual timetable. Dollevoet et al. (2017) propose an iterative framework allowing to reschedule the timetable as well as rolling-stock and crew assignments in case of large disruptions. The number of services that have to be canceled or delayed is minimized, thus minimizing the gap with the original schedule. Canca et al. (2016) use short-turning to increase the frequency in congested area when a disruption causes a passenger overload in trains and on platforms. Their goal is to minimize the passenger waiting time. Zhu and Goverde (2019) propose various rescheduling actions, including changing the stopping pattern, short-turning and cancelling trains, and adjusting the rolling-stock circulation in case of large disruptions. Their approach aims at minimizing passenger delays. Jiang et al. (2019) describe an approach allowing stop-skipping and passenger inflow control (i.e. restraining passenger access to the platform) to minimize the discomfort of passengers in an over-

crowded urban line. In terms of schedule design, Shang et al. (2018) propose a stop-skipping model to ensure passenger equity in an oversaturated urban rail transit network, minimizing the number of passengers who suffer the maximum number of missed trains because the train is fully occupied when serving their origin station. Nielsen et al. (2012) propose a rolling-stock rescheduling approach in case of disruptions, with a rolling horizon to cope with the evolution of the situation in real time. Recently, van der Hurk et al. (2018) consider large disruptions, and propose an approach combining optimization and simulation to reschedule the rolling stock and to propose route advice to passengers.

To summarize and to our knowledge, our problem is the only one that combines various types of rescheduling decisions in dense railway systems that have to be taken in real time while considering rolling-stock constraints and various objectives. Another difference of our work with most of the previous literature is that we explicitly consider a multi-objective analysis to provide multiple solutions to the decision makers. In terms of approach, Corman et al. (2017), Dollevoet et al. (2017), van der Hurk et al. (2018) and Zhu and Goverde (2019) are probably the closest to the one we propose, but with significant differences in terms of the possible decisions, the objectives and the constraints. Indeed, we believe our work has interesting new characteristics, such as a multi-objective process and analysis, different operational actions and the implementation of a real-time decision support system with experiments during traffic on an important line in the Paris area.

The reservicing actions we are considering are canceling trains, adding and skipping stops and short-turning trains. Various objectives are optimized, the two main ones being the recovery time and the quality of service for passengers. The passenger flows are modeled using Origin-Destination data, and both the waiting time and the in-vehicle time are computed. To model the impact of all these actions and the flows of passengers, we chose an approach combining an optimization module and a simulation module. The simulation module is used to evaluate the operational actions selected by the optimization module, in particular to estimate the passenger flows through the railway system.

The paper is structured as follows. The considered rescheduling problem is described in Section 2. In Section 3 the optimization-simulation approach is presented. In Section 4, computational results of the approach on actual instances of SNCF Transilien are discussed. Section 5 first briefly

presents the characteristics of the decision support tool, and then some qualitative and quantitative feedback from real-life experiments conducted on a Transilien line. Finally, conclusions and directions for further research are provided in Section 6.

2. Problem description and modeling

2.1. General description

The real time rescheduling problem considered in this work is an extended and more realistic version of the problem described in Altazin et al. (2017). The modeling we propose is rather generic since it was defined based on the Transilien lines, and thus should fit many lines in other large cities. The network consists of one track for each direction. There is one platform in most served stations, the capacity of terminal stations and stations with more than one platform is considered to be infinite. Trains cannot overtake each other, and some lines have several branches. The frequency of trains on each line is between 4 and 20 trains per hour during peak hours, and can be lower during off-peak periods.

A limited rescheduling horizon of 1.5 to 2.5 hours is considered. In case of small disturbances, typically causing less than 20 minutes of delay, various reservicing actions can be taken: Cancelling trains, short-turning trains in stations with the necessary infrastructure, skipping or adding stops. Note that only stop skipping was allowed in Altazin et al. (2017). The types of decisions considered in our approach can rather easily be modified, by adding or removing possible decisions. In this paper, we call a "decision" the implementation of a given action on the service of trains, compared with the reference situation of merely keeping the (possibly delayed) theoretical service. Hence, the reference situation corresponds to the "no decision" case, even if it is a well-considered choice. A time margin can be added for each type of decisions, for instance stop-skipping is usually only allowed for trains departing more than 10 minutes after the rescheduling time, so that passengers can be warned on time and change their itinerary. The running and dwell times of trains in the theoretical schedule are considered as minimal values for train operations. Headway constraints between trains running consecutively on the same track (i.e. in the same direction), are considered as well.

The rolling-stock schedule is considered through minimal turning time constraints at terminal stations between trains using the same rolling-stock

unit. The capacity of rolling-stock units are considered in the simulation module of our approach. We assume that drivers operate the same rolling-stock units during the rescheduling horizon.

2.2. Problem modeling

We model railway operations at macroscopic level as a directed graph, where $\mathcal{N}$ is the set of nodes corresponding to events of trains: Departures from a station, arrivals at a station or transits through a station. Arcs of the graph on set $\mathcal{A}$ can correspond to two types of links between events: (1) Operations, particularly running between stations, dwelling at a station or turning at a terminal station, or (2) Headway constraints between two trains running consecutively on the same infrastructure. A duration is associated with each arc, corresponding to the minimum time between the two linked events: Running or dwell time, minimum headway between consecutive trains or minimum turning time.

An illustrative example is provided in Figure 1. Trains t_1 and t_3 are consecutively running from station A to station D, passing through station B and stopping at station C. Train t_2 runs in the opposite direction with the same stopping pattern. Trains t_1 and t_2 are using the same rolling-stock unit. Plain arcs correspond to run and dwell time constraints between two consecutive events e and e' of a train. Thus, $d_{e_1e_2}$, $d_{e_2e_3}$, and $d_{e_4e_5}$, are minimal running times between events e_1 and e_2, e_2 and e_3, and e_4 and e_5 respectively. Similarly, $d_{e_3e_4}$ is the minimal dwelling time at station C between events e_3 and e_4. The dotted arc represents the turning time of the rolling-stock unit between arrival of train t_1 at its terminal Station D, represented by event e_5, and the departure of t_2 from Station D, represented by event e_6. Hence, $d_{e_5e_6}$ corresponds to the minimal turning time between e_5 and e_6 required to turn the rolling-stock unit. The dashed arcs represent headway constraints between trains t_1 and t_3 that run in the same direction. As there is only one track per direction, and one platform per direction in each station, only one train can dwell at a time. Thus, train t_3 cannot arrive at Station C (event e_{13}) before train t_1 has left the station (event e_4) since there is a minimal headway time $h_{e_4e_{13}}$.

The operational actions that are considered change the graph structure. If the stop at station C is skipped for train t_2, events e_7 and e_8 are merged into a single passage event and the arc durations from and to this new event are modified to consider the saved breaking and acceleration times. Conversely, adding a stop in station B would result in splitting e_9 into two

events for arrival and departure and lengthening arc durations to model the extra time for breaking and acceleration. Canceling a train means removing its nodes, running and dwelling arcs, and adjusting the headway arcs, while short-turning train t_1 in station C means deleting events e_4, e_5, e_6 and e_7, and connecting e_3 and e_8 with a dotted arc for the rolling-stock turning operation.

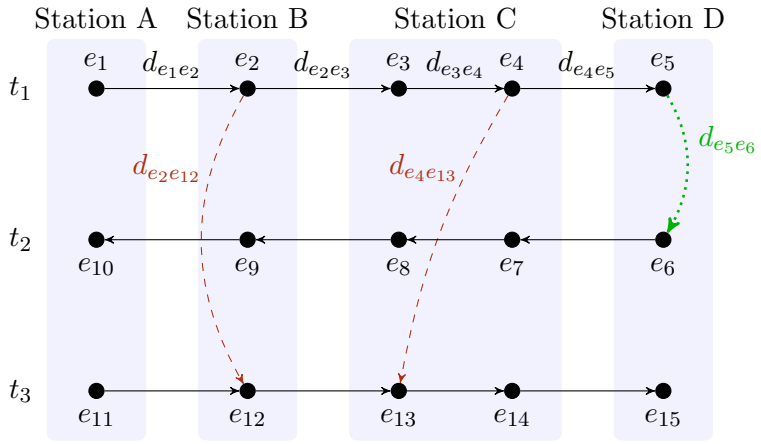

Figure 1: Event-activity graph with 3 trains running on a line serving 4 stations

The notations used in this paper for parameters and variables are detailed below.

Parameters:

$\mathcal{N}$	:	Set of nodes modeling events e of a train at a location in the original schedule (departure, arrival or passing),
$\mathcal{A}$	:	Set of arcs,
s_e	:	Time at which event e is originally scheduled,
$d_{e'e}$	:	Minimal duration of the running, dwelling or turning operation between events e' and e such that $(e', e) \in \mathcal{A}$,
m_{RT}	:	Tolerance margin to define if an event is delayed,
α	:	Weight of the waiting time of passengers compared to their in-vehicle time in the calculation of the Quality of Service QS.

Variables:

P_e : Rescheduled time of event e,
D_e : 1, if event $e \in \mathcal{N}$ is late compared to its original schedule (i.e. $P_e > s_e$), and 0 otherwise,
RT : Recovery time, i.e. maximum time before which rescheduled times of events are different from scheduled times,
QS : Quality of Service for passengers, detailed below.

2.3. Evaluation criteria

This problem has three main objectives:

1. Minimize the recovery time, i.e. recover the original schedule as fast as possible to ensure the performance of the railway system,
2. Minimize the inconvenience for passengers, by minimizing the waiting time and in-vehicle time of passengers,
3. Minimize the number of proposed actions. Limiting the number of actions also ensure that the solution can be implemented in real time.

An important goal is to allow the decision makers to choose from various rescheduling solutions with trade-offs on the criteria. All non dominated solutions are thus stored in order to select one or more solutions to present to the decision makers. We use Pareto (Pareto and Bonnet (1963)) dominance to build the set of non dominated solutions (see Section 3.3).

The recovery time is determined by the rescheduled time of the latest delayed event. An event is delayed if the difference between its planned time and its rescheduled time is larger than a given tolerance margin m_{RT}:

$$\forall e \in \mathcal{N} \quad D_e \geq (P_e - s_e - m_{RT})/M \tag{1}$$
$$RT \geq P_e D_e \tag{2}$$

where M is a sufficiently large coefficient so that the right-hand side of (1) is always smaller than 1 when P_e is larger than $s_e + m_{RT}$. In our experiments, m_{RT} was set to to 2 minutes, although this value can be adjusted to the railway line and the period in the day.

The quality of service QS for passengers is characterized by two durations: The waiting time at the origin station to board a train serving the destination station and which is not already full, and the in-vehicle time until the destination station. Data on Origin-Destination (OD) trips of passengers are used, and passengers are assumed to arrive regularly and continuously at stations since the frequency of trains is important. An arrival rate of passengers in each station for each destination at each time slot is

derived from passenger counting data. The capacity of trains is considered, i.e. some passengers may have to wait for another train if the first train to serve their destination station is full. The quality of service QS, which must be minimized to maximize the quality of service, is calculated as follows:

$$QS = \sum_{\text{All passengers}} \alpha * \text{Waiting time} + \text{In-vehicle time} \tag{3}$$

Note that a much simpler modeling of the quality of service was used in Altazin et al. (2017). Based on the research on waiting time perceptions (see recently Fan et al. (2016)), we set α to 2 in our experiments, i.e. one minute of waiting time is perceived as 2 minutes of in-vehicle time.

The number of proposed operational decisions is also minimized, to limit their impact for passengers and to ensure that the solution can be implemented by the decision makers. Finally, as secondary criteria, the number of delayed events and sum of delays are minimized to ensure the consistency of the solution. These criteria ensure that, if two solutions have the same values for RT, QS and the same number of proposed decisions, the solution with the smallest sum of delays and the smallest number of delayed events is the returned solution.

Our approach aims at proposing Pareto-optimal solutions. Yet, a guiding function, which is a weighted combination of the five criteria, is used in the heuristics described in Section 3 to generate non-dominated solutions.

3. An optimization-simulation approach

3.1. Overview of the approach

In order to integrate more operational actions and better model passenger flows than in Altazin et al. (2017), but also to determine multiple rescheduling scenarios to propose to decision makers, we chose to develop a multi-objective optimization-simulation approach. The approach is a two-step process, as shown on Figure 2: (1) Generation of a set of non-dominated solutions and (2) Selection of a limited number of solutions in this set.

The generation of the set of non-dominated solutions is an iterative process between an algorithm enumerating decision scenarios and the simulation module. The simulation module propagates both delays and passengers through the event-activity graph, to compute the various criteria used to evaluate the decision scenarios.

Because the approach is modular, new heuristics could be added to generate solutions, or another simulation module could be used to evaluate the decision scenarios.

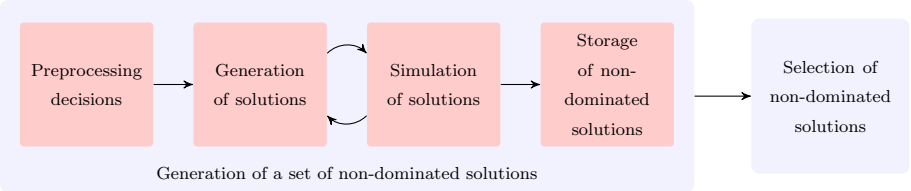

Figure 2: Optimization steps

3.2. Generation of a set of non-dominated solutions

3.2.1. Preprocessing decisions

Preprocessing the possible decisions allows the number of decisions that can actually be applied to be significantly reduced. We assume that decisions concerning trains and events that will not be affected by delays that already occurred at the rescheduling date should not help to recover faster. A simulation with no operational decision applied is run to determine the events affected by delays. Following this simulation, some decisions are removed from the set of possible decisions. Cancelling, short-turning and stop-skipping decisions for trains or events not impacted by the initial delays is no longer possible. Yet, stop-adding decisions remain possible as adding a stop at a station aims at improving the quality of service for passengers boarding or alighting at this station and not at improving the recovery time (it can even cause a delay because of the dwell time in station and the breaking and acceleration times before and after the stop). The schedule of trains not impacted by initial delays is thus not fixed and can be impacted by decisions made on other trains and by stop-adding decisions.

This preprocessing significantly reduces the number of possible decisions. On the test instances used in Section 4, the number of decisions is reduced by 78% on average, and up to 84%. The generation and simulation of decision scenarios are thus facilitated and accelerated.

3.2.2. Generation of decision scenarios

Each solution is a decision scenario with one or several proposed decisions, and a set of indicators to evaluate the scenario. Evaluation indicators

are computed through simulation, but decision scenarios must be build beforehand. We implemented two greedy heuristics, to quickly and efficiently determine some scenarios in order to generate a set of non-dominated solutions, and a full enumeration method enumerating all possible solutions, to analyze the quality of solutions provided by the heuristics.

The heuristics have been designed following an analysis of the results obtained with the Full Enumeration method (FE). The logic of the heuristics is based on trying to find, at each step, the decision that improves the criteria the most. Their performance, which is compared to the one of the full enumeration method, is presented in Section 4. The heuristics are guided by a weighted sum of the five criteria presented in Section 2.3. The weights of the different criteria were set following a series of tests. A proposed decision must guarantee a significant gain on the recovery time. In the case of an added stop, the recovery time must not be degraded and the quality of service for passengers should be sufficiently improved.

Full Enumeration method (FE). It implicitly enumerates all possible decision scenarios through a depth-first exploration of the decision tree. Each level of the decision tree corresponds to all solutions with the same number of decisions applied: Nodes at the first level include all solutions with only one decision applied, and so on. All feasible solutions are evaluated. This exact method is computationally expensive and thus impossible to use in real time in most cases, yet it helps to estimate the performance of heuristics through numerical experiments.

Heuristic Add. This greedy heuristic determines which decision, applied alone, minimizes the objective function. This decision is then activated, and the second decision that minimize the objective function is determined, and so on. Two stopping conditions are used: (1) The maximal number of proposed decisions is reached, or (2) the objective function is not decreased by applying one more decision.

This heuristic, that provides a local optimum very quickly, is described in Algorithm 1. The set of authorized decisions following the preprocessing is designated $\mathcal{D}$. Function *activate(d)* sets decision d to 1 when function *deactivate(d)* set it to 0. Function *feasible(sol)* checks that the solution *sol* is feasible, as explained later in this section. Function *addNonDominated(sol)* compares solution *sol* with saved non-dominated solutions and updates the set of non-dominated solutions accordingly (see at the end of this section).

Algorithm 1 Heuristic Add(initSol)

```
D: Set of possible decisions
sol ← initSol
bestSol ← sol
betterSol ← true
objectiveMin ← +∞
while sol.nbActiveDecisions <nbDecisionsMax and betterSol is true do
  betterSol ← false
  for Decision d ∈ D do
    sol.activate(d)
    if feasible(sol) is true then
      sol.simulate
      addNonDominated(sol)
      if sol.objective <objectiveMin then
        objectiveMin ← sol.objective
        bestSol ← sol
        betterSol ← true
      end if
    end if
    sol.deactivate(d)
  end for
  sol ← bestSol
  sol.nbActiveDecisions ← sol.nbActiveDecisions + 1
end while
return bestObj
```

Heuristic Add-3-Branches. This heuristic is an extension of Heuristic *Add*. At the first step, the three best decisions applied alone are kept, and three different search branches, i.e. three instances of Heuristic *Add*, are launched, as shown in Algorithm 2. Three local optima are thus determined.

Feasibility of a scenario. The feasibility of a scenario built by the full enumeration method or by a heuristic must be checked before its simulation. When applying a decision, others become inapplicable. No decision can be taken on a canceled train. Trains can only be short-turned once, so if the beginning of a train has been skipped due to a previous short-turning, the train has to reach its terminal station. The service of a short-turned train cannot be changed. These measures are necessary to maintain an adequate quality of service for passengers. They can be deactivated through a parameter. Another parameter sets the maximal number of simultaneously proposed decisions for a scenario.

3.2.3. Simulation module

Each feasible decision scenario generated by the optimization module is evaluated through simulation. For each scenario, the event-activity graph corresponding to the proposed decisions is built. The time of each event is first computed, and passengers are then assigned to the train they should take. The times of the events prior to the rescheduling date are known and the times of the events subsequent to this date have to be fixed. Delays existing at the rescheduling date are propagated through the graph to compute the times of all events after the rescheduling date.

To set the time of an event, all its incoming arcs are considered. If the times of the first events of these arcs are not yet computed, they are computed. Once first events of all incoming arcs are set, the time of the event is computed according to the most stringent arc: If event e has two incoming arcs, say a running arc from the same train at the previous station and a headway arc from the previous train at the same station, both previous event times and arc durations (here, running time and headway) are taken into account to compute the earliest time at which event e could occur, while respecting the infrastructure constraints. The following calculation is thus done for each event $e \in \mathcal{N}$:

$$P_e = \max_{\forall e' \in \mathcal{N};\ (e',e) \in \mathcal{A}} (P_{e'} + d_{e'e})$$

Once the simulated time for each event is computed, passengers are

Algorithm 2 Heuristic Add-3-Branches(initSol)

```
D: set of possible decisions
sol, bestSol1, bestSol2, bestSol3 ← initSol
bestObj1, bestObj2, bestObj3 ← initSol.objective
for Decision d ∈ D do
  sol.activate(d)
  if feasible(sol) is true then
    sol.simulate
    addNonDominated(sol)
    if sol.objective <bestObj3 then
      if sol.objective <bestObj2 then
        if sol.objective <bestObj1 then
          bestSol3 ← bestSol2
          bestObj3 ← bestObj2
          bestSol2 ← bestSol1
          bestObj2 ← bestObj1
          bestSol1 ← sol
          bestObj1 ← sol.objective
        else
          bestSol3 ← bestSol2
          bestObj3 ← bestObj2
          bestSol2 ← sol
          bestObj2 ← sol.objective
        end if
      else
        bestSol3 ← sol
        bestObj3 ← sol.objective
      end if
    end if
  end if
  sol.deactivate(d)
end for
bestSol1 ← Add(bestSol1)
bestSol2 ← Add(bestSol2)
bestSol3 ← Add(bestSol3)
return bestSol1, bestSol2, bestSol3
```

assigned to the trains they should board to calculate their waiting time at the origin station and their in-vehicle time. Passengers are gathered by OD (Origin-Destination). The number of passengers in each OD is known for each one-hour time slot. We consider a uniform and constant arrival of passengers at the station in a time slot.

At each stop e of a train t at a station, the concerned ODs are determined according to the next stops of train t. For each of these ODs, the number of passengers arriving at the station since the last stop of a train t_{OD}^{prec} serving the OD is computed, and this number is added to the number of passengers, if there are any, who could not board train t_{OD}^{prec} because it was already full.

Figure 3 illustrates the number of passengers wishing to reach a given destination in the origin station over time. The arrival rate at the station varies according to the time slot (characterized by slopes τ_{7-8}, τ_{8-9} and τ_{9-10}). Since train t_2 is already full when arriving at the station, not all waiting passengers can board. Each group of passengers wishing to join the same destination is then divided: Some passengers board the train while the others wait for the following train which stops at their destination. The number of passengers that can board depends on the ratio between the number of passengers in each OD group and the total number of passengers wishing to board. For example, passengers on train t_3 are passengers that arrived between t_1 and t_2 who could not get in t_2 and passengers who arrived since t_2.

Once the assignment of passengers is completed, it is possible to compute both the waiting times and the in-vehicle times.

Storage of non-dominated solutions. Each feasible generated scenario is simulated, and its criteria are computed. If a solution is not Pareto-dominated by solutions of the set of non-dominated solutions, it is stored in the set. If stored solutions are dominated by the new solution, they are removed from the set. The set of non-dominated solutions is thus updated after each simulation.

3.3. Selection of non-dominated solutions

Once the set of non-dominated solutions is completed, one or several solutions has to be picked from this set. Selection criteria are first recalled, then the selection process is described.

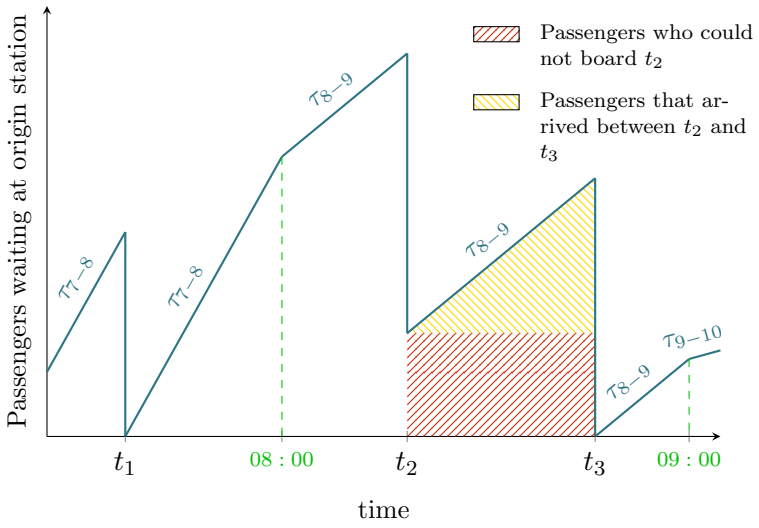

Figure 3: Waiting time of passengers for a specific OD

3.3.1. Selection criteria

Based on the three main objectives introduced in Section 2.3, three criteria are used to evaluate a solution:

- The recovery time,
- The quality of service for passengers,
- The number of proposed decisions.

Let us recall that the goal is to present multiple solutions with trade-offs on these criteria to Transilien decision makers.

3.3.2. Selection process

Between one and three rescheduling solutions are proposed as follows:

1. The solution minimizing the objective function as a weighted sum of the recovery time, the quality of service for passengers, the number and type of proposed decisions and the number and sum of delays. The recovery time and the quality of service for passengers are dominating in the objective function, thus this function minimizes the recovery time RT_{min}.

2. The solution that minimizes the impact on the quality of service for passengers with a minimal recovery time RT_{min}.
3. The solution that minimizes the impact on the quality of service for passengers with a degradation threshold on the minimal recovery time RT_{min}. This allows a trade-off between the recovery time and the impact of decisions for passengers.

This selection process quickly determines interesting solutions to show to the decision makers. It has been used for the industrial experiments at the end of 2017 (see Section 5).

4. Numerical experiments

4.1. Test instances

The Transilien network consists of 13 train lines. Five of them are crossing Paris (RER lines), along a north-south axis or an east-west axis. The other eight lines start from a large train station in Paris or close to Paris, and travel towards the suburbs. Most lines run on an infrastructure dedicated to Transilien trains, but some lines share the infrastructure with other SNCF long distance trains (mostly regional and intercity trains). Most of the lines consist of several branches, and each line has its own characteristics (platform height, length, rolling stock, train protection system, etc.). Lines consist of double tracks, and tracks are operated in a single direction. As lines are double-tracked, re-routing is not possible, except in specific cases like large disruptions or maintenance works. Some stations are equipped with a third central track, allowing trains to turn. These tracks are rarely used to change the order of trains, except in case of large disruptions.

Frequencies widely vary depending on the type of lines (crossing Paris or not) and the distance from Paris. Furthest stations have the lowest frequency (two trains per hour) while, in central Paris stations, the frequency can reach one train every two minutes during peak hours. Thus, the impact of an operational action depends on its location: Skipping a stop in Paris has almost no impact on the quality of service, while skipping a stop further from Paris can result in passengers waiting for one hour.

In terms of rolling stock, Transilien trains are most of the time composed of a locomotive and a set of coaches or two coupled units that operate as a pair for the entire day. Here, each train is considered as one rolling-stock unit.

4.2. Comparing with Altazin et al. (2017)

The first experiments with the optimization-simulation approach have been carried out on the same instances as in Altazin et al. (2017). A delay of a few minutes is created on one train running on a line serving 8 stations during morning or evening peak hours, with a frequency of one train every 10 minutes in both directions. The instances cover between 1.5 and 2.5 hours of circulation. The approach can tackle any number of delays, as it is the case in Section 5, but only one delay is created here in order to study the performance of the approach.

The minimal turning time is set to 8 minutes, and the stop-skipping strategy to *after10*: Stops can be skipped only for trains departing at least 10 minutes after the rescheduling time.

First, only stop skipping was allowed to compare the optimization-simulation approach and the Integer Linear Program (ILP) of Altazin et al. (2017). The optimization-simulation approach provides a finer modeling of passenger flows, and more realistic criteria for evaluating the quality of service for passengers: The ILP only computes the maximum waiting time between two consecutive trains. The Full Enumeration method (FE) is used in the optimization-simulation approach, to ensure that the optimal solution is found. Parameters of the optimization-simulation approach are set to the same values as for the ILP, or similar values for new or different parameters.

Initial delay	Solution	Recovery time (min.)		Degradation of passengers' objective		Skipped stops		Total delay (min.)	
		ILP	FE	ILP	FE	ILP	FE	ILP	FE
3	S_0	16	16			0	0	43	37
	S^*	16	16	0.0%	0.0%	0	0	43	37
5	S_0	30	28			0	0	84	69
	S^*	23	21	1.0%	0.6%	0.8	0.4	74	64
7	S_0	43	38			0	0	152	137
	S^*	32	25	3.0%	1.0%	1.6	1	132	123
10	S_0	59	52			0	0	296	277
	S^*	42	34	8.0%	1.6%	4	2.2	249	245

Table 1: Comparison of the average results obtained with ILP and FE

Table 1 compares the average results obtained on five instance with the ILP and the optimization-simulation approach that uses the Full Enumeration method (FE). For each initial delay, average values of indicators are

shown for solution S_0 with no authorized decisions and optimal solution S^*. Note that the results are similar without being identical. The models are slightly different. For instance, the recovery time is computed from the rescheduling date given by the user to the theoretical time of the last delayed event in the ILP, while it is computed from the delayed time of the event where the delay is created, to the simulated time of the last delayed event, as constraints do not need to be linear.

The optimization-simulation approach proposes to skip less stops. Skipping a stop costs the same in the objective function but does not allow the same to be gained in the two approaches. The ILP considers a fixed gained duration for a skipped stop, when the time gained in the optimization-simulation approach depends on the theoretical dwell time at the station. Less stops needs to be skipped to recover the same delay with the optimization-simulation approach.

Impact on passengers seems to be greater with the ILP, though the values follow the same trend for the two approaches. The ILP considers the maximal waiting time between two consecutive trains, weighted by the number of passengers wanting to board or alight at the concerned station. Passengers are continuously arriving at stations in the optimization-simulation approach, and the destination station is considered. The modeling of passengers trips is more precise and realistic in the optimization-simulation approach.

4.3. Comparing only stop-skipping decisions with all decisions

The second series of tests aims at analyzing the interest of the new decisions compared to only allowing only stop skipping: Adding stops, short-turning trains and canceling trains. Heuristic *Add-3-Branches* (3B) is used with the guiding objective function discussed in Section 2.3.

First, only stop skipping is activated, and then all decisions are allowed. Parameters are set to similar values as before, 8-minute minimal turning time and a time period to observe before applying a decision: (1) Before the train departure for canceling the train or modifying its stopping pattern, or (2) Before the arrival event at the concerned station in case of a short-turning. This time period is set to 5 minutes in these tests.

Table 2 summarizes the results for both tests: Only stop-skipping decisions (SS) and all decisions (AD). The results are slightly different than those from Table 1: More stops can be skipped because of the 5-minute time period before applying decisions, and heuristic *Add-3-Branches* is used.

Initial delay (min.)	Solution	Recovery time (min.)		Degradation of passengers' objective		Proposed decisions		Total delay (min.)	
		SS	AD	SS	AD	SS	AD	SS	AD
3	S_0	16	16			0	0	37	37
	S^*	16	13	0.0%	1%	0	0.2	37	32
5	S_0	28	28			0	0	69	69
	S^*	21	14	0.9%	4%	0.4	0.6	64	56
7	S_0	38	38			0	0	137	137
	S^*	25	19	1.7%	13%	1.4	1.0	119	102
10	S_0	52	52			0	0	277	277
	S^*	33	22	0.4%	8%	1.6	2.0	230	193

Table 2: Comparison of only stop-skipping decisions with all decisions

The recovery time is shorter when all decisions can be applied, which is coherent. Indeed, short-turning a train consists in skipping several consecutive stops, which allows more time to be gained than skipping one stop. Short-turning can also be applied on running trains, when stops can be skipped only on trains departing at least 5 minutes after the rescheduling date.

Yet, the degradation of the quality of service for passengers is larger. Short-turning or canceling a train affects more passengers than a skipped stop. Note that the impact for a 10-minute initial delay is smaller than with a 7-minute delay. In case of a large delay, many passengers are affected in S_0 and short-turning a train will impact some passengers but shorten the waiting time for others. The lengthening of waiting times and in-vehicle times is thus smaller for a larger initial delay.

4.4. Analysis of heuristic performance

These last tests aim at evaluating the quality of the optimal solution S^* proposed by the heuristics. Same instances have been ran with both heuristics, *Add* (ADD) and *Add-3-Branches* (3B), with the optimal solution found with the Full Enumeration method (FE). Table 3 presents the same indicators as in previous tests, and Table 4 sums up the number of simulations and number of non dominated solutions when using each algorithm.

Table 3 shows that the two heuristics have identical results. The local optima reached by heuristic *Add-3-Branches* are thus identical to the local optimum of heuristic *Add* or worse in terms of the objective function.

Initial delay (min.)	Sol.	Recovery time (min.)			Degradation of passengers' objective			Proposed decisions			Total delay (min.)		
		FE	ADD	3B	FE	ADD	3B	FE	ADD	3B	FE	ADD	3B
3	S_0	16	16	16				0	0	0	37	37	37
	S^*	13	13	13	0.9%	0.9%	0.9%	0.2	0.2	0.2	32	32	32
5	S_0	28	28	28				0	0	0	69	69	69
	S^*	14	14	14	3.7%	3.7%	3.7%	0.6	0.6	0.6	56	56	56
7	S_0	38	38	38				0	0	0	137	137	137
	S^*	19	19	19	13.4%	13.4%	13.4%	1.0	1.0	1.0	102	102	102
10	S_0	52	52	52				0	0	0	277	277	277
	S^*	22	22	22	8.3%	8.4%	8.4%	1.6	2.0	2.0	194	194	194

Table 3: Comparison of heuristics (ADD and 3B) with FE

Solutions minimizing the objective function S^* are the same with the heuristics and the full enumeration method, except for one 10-minute initial delay instance. For this instance, the recovery times are identical, but the solution found by FE reaches this result by proposing less decisions, and the impact on passengers is smaller. These tests confirm the performance of both heuristics in terms of the evaluation criteria, with a much shorter computational time. As shown in Table 4, the full enumeration method requires several million simulations when the heuristics need at most a few hundreds. Some test instances require several hours of computational time, when heuristic *Add-3-Branches* solves them in less than one minute.

In terms of computational time, 3 to 4 seconds are required for 1,000 simulations on a machine with an Intel Core i3-6100U processor running at 2.3GHz with 1GB of RAM allocated to the rescheduling application. This time is reduced to 1.5 seconds on a machine with an i7-3930K processor clocked at 4GHz, when 8GB of RAM is allocated for rescheduling.

Heuristic *Add-3-Branches* does not require significantly more simulations than heuristic *Add*. Because more solutions are evaluated, more non dominated solutions might be found with heuristic *Add-3-Branches*, offering more choice for the selection process. The full enumeration method generates more non dominated solutions, yet the difference is not significant, compared to the very large number of additional simulations that are required.

To conclude, both heuristics provides good results, for a very strongly reduced computational time, that allows the rescheduling approach to be used in real time.

Initial delay (min.)	Number of simulations			Number of non dominated solutions		
	FE	ADD	3B	FE	ADD	3B
3	1,537,090	27	27	1.0	1.0	1.0
5	1,440,709	34	49	3.6	2.8	2.8
7	3,012,378	43	94	6.6	3.8	5.6
10	6,664,050	60	117	8.5	5.8	7.3

Table 4: Number of simulations and non dominated solutions generated by each algorithm

4.5. Multi-objective analysis

To assess the quality of the sets of non dominated solutions generated by the heuristics, indicators from Jaszkiewicz (2004) and Zitzler (1999) have been calculated on the 13 test instances generating rescheduling solutions (the 7 other instances proposed no operational decisions).

Table 5 shows the number of solutions in the Pareto front generated by FE and the two heuristics. As specified in Section 2.3, three criteria are used to establish the dominance: The recovery time RT, the quality of service for passengers QS and the number of proposed actions. As expected, heuristic *Add-3-Branches* generates more non dominated solutions than heuristic *Add*. FE proposes more non dominated solutions than the heuristics in most cases, except when some of the generated solution are better and the set is smaller.

Instance	Initial delay	Number of non dominated solutions		
		FE	ADD	3B
1	3	5	4	4
2	5	4	4	4
3		4	3	3
4		6	6	6
5	7	4	4	4
6		8	6	9
7		4	3	3
8		12	5	11
9	10	5	3	3
10		13	5	10
11		5	3	4
12		18	10	11
13		2	2	2

Table 5: Number of non dominated solutions generated for 13 test instances

Net Front Contribution (NFC)

The Net Front Contribution (NFC) is expressed as the percentage of solutions of each set of non dominated solutions that are in the reference front $\mathcal{S}$. The reference front $\mathcal{S}$ is the set of non dominated solutions when combining all sets. This indicator measures the contribution of each set to the global set.

Table 6 presents the values of NFC for all instances, for the full enumeration method and both heuristics. As expected, FE enumerates all solutions, thus its NFC is always 100%. Heuristic *Add-3-Branches* has a slightly better NFC than heuristic *Add*, as it browses more solutions. These results show that the heuristics are not as good as FE, but they still provide an interesting set of non dominated solutions. Note that the NFC of heuristics is smaller when the initial delay is larger, and more decisions can thus be proposed.

Instance	Initial delay	NFC FE	NFC ADD	NFC 3B
1	3	100%	80%	80%
2	5	100%	100%	100%
3		100%	75%	75%
4		100%	100%	100%
5	7	100%	100%	100%
6		100%	75%	75%
7		100%	75%	75%
8		100%	33%	75%
9	10	100%	60%	60%
10		100%	38%	69%
11		100%	60%	80%
12		100%	22%	22%
13		100%	100%	100%

Table 6: Net Front Contribution (NFC) for 13 test instances

Weak-Out Performance (WOP)

The Weak-Out Performance (WOP), that evaluates the dominance between two sets of non dominated solutions, has also been calculated. The WOP between two sets A and B is equal to 1 if A weakly dominates B, -1 if B weakly dominates A and 0 if no set weakly dominates the other. A set A of non dominated solutions is said to weakly dominate another set of non dominated solutions B if no solution in A is dominated by a solution in B, and if at least one solution of A dominates a solution of B. The results

are shown in Table 7. The full enumeration method weakly outperforms both heuristics or no heuristic outperforms the other, and heuristic *Add-3-Branches* weakly dominates heuristic *Add* in two cases. These results are expected, and confirm that the sets of non dominated solutions generated by the heuristics are about the same quality as those generated by FE.

Instance	Initial delay	WOP		
		FE-ADD	FE-3B	3B-ADD
1	3	0	0	0
2	5	0	0	0
3		0	0	0
4		0	0	0
5	7	0	0	0
6		0	1	0
7		0	0	0
8		1	1	1
9	10	0	0	0
10		0	1	0
11		0	0	0
12		1	1	1
13		0	0	0

Table 7: Weak-Out Performance (WOP) for 13 test instances

Spacing

This indicator measures the average distance between consecutive solutions in a set of non dominated solutions. The smaller the spacing, the more evenly the solutions are distributed across the set.

Table 8 presents spacing values for the full enumeration method and both heuristics. Spacing is smaller for FE on some instances, as there are more non dominated solutions. Note that some instances have a smaller spacing for heuristics, such as instance 11, for which the set of non dominated solutions obtained with heuristic *Add* is small and not very diverse.

The maximum spread of the sets has also been calculated, showing that the set of non dominated solutions of FE is almost always more spread than those of the heuristics, except for instance 12, for which the optimal solution found by FE is not found by the heuristics. In this case, the set of non dominated solutions is less spread for FE as it is narrowed around this optimal solution.

This multi-objective analysis has pointed out that, although the heuristics provide a smaller number of non dominated solutions, the quality of

Instance	Initial delay	Spacing FE	ADD	3B
1	3	1.33	0.92	0.92
2	5	0.88	0.88	0.88
3		1.93	1.43	1.43
4		1.01	1.01	1.01
5	7	0.90	0.90	0.90
6		1.22	1.58	1.96
7		1.93	1.42	1.42
8		2.23	2.06	2.01
9	10	1.74	1.46	1.46
10		1.00	0.91	1.19
11		1.58	0.86	1.26
12		1.25	2.21	2.24
13		2.96	2.96	2.96

Table 8: Spacing of the set of non dominated solutions

their sets of non dominated solutions is good enough. We thus chose to implement this real time rescheduling approach in an operational tool.

5. Industrial validation

5.1. Presentation of the experiments

A rescheduling tool has been implemented to test the optimization-simulation approach in real time operations and evaluate the consistency of the proposed solutions. Decision makers from the Transilien operational center took part in the tool design, particularly for the graphical interface and the parameters. The home screen of the tool is a map of the considered line, with real time alerts. The tool is online, and thus connected to real-time data on trains. Hence, when the rescheduling process is launched, the current situation, with all trains running on the line and the current delay of each train, is taken into account. Note that the previously applied decisions are also taken into account. The solution S_0 without actions and the proposed rescheduling solutions to recover the multiple delays can be compared through their recovery time and quality of service for passengers. A time-space diagram of each solution can then be plotted, showing the proposed decisions and their impact on the traffic.

This tool is connected to Transilien data flows. Experiments have been conducted during three weeks at the end of 2017, on a line with two branches: The two blue branches of line L on Figure 4. Five trains per hour are running

on each branch during peak hours, thus 10 trains per hour are running in the common section. Trains from another Transilien line are running on the same tracks on a portion of the line (red line U on Figure 4). These trains are considered because of the headways, but no actions can be taken on them. Tests have been carried out during the end of the morning peak hours.

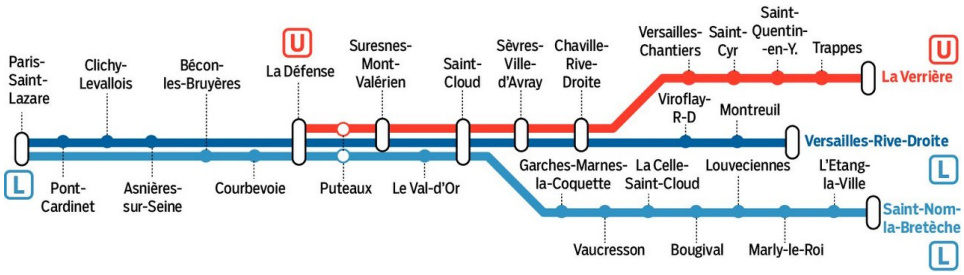

Figure 4: Map of the L line on which the experiments have been conducted

5.2. Results

The rescheduling approach was launched 302 times during the three weeks of experiments. About half of the time, no solution was proposed as the situation was not disturbed. When solutions were found, only one solution was proposed in 75% of the cases and two solutions in the remaining cases.

On average, the recovery time was reduced by 40% (35 minutes) compared to solution S_0. 79% of the proposed solutions had a negative impact for passengers. The larger the delays, the more the number of proposed solutions and decisions, and the smaller the impact was for passengers, as noted in Section 4.3.

A total of 43 rescheduling proposals have been discussed in details with the decision makers. Figure 5 shows the distribution of these 43 proposals. Most of them are consistent, meaning that the tool behaved as expected, some minors bugs have been corrected in the first days. This result is very satisfying for a first operational implementation. Consistent solutions might not be relevant, mainly because of characteristics not expected to be managed by the tool. For instance, a short-turning requiring a long turning time in a station, and thus occupying a track for too long. Decisions could be proposed for a train that has a small delay or no delay, so decision makers would wait to see the evolution of the situation before taking an action.

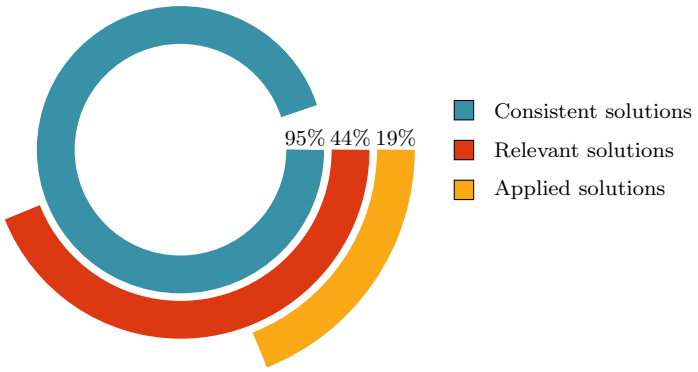

Figure 5: Distribution of discussed rescheduling solutions

These cases are still interesting as they drag the decision maker's attention on a train whose delays could increase in the near future.

Relevant solutions are meaningful solutions for the decision makers that could be applied. Solutions that are relevant but not applied are lacking data, for instance on operational actions that have already been taken, or about driver assignments, making the proposed decisions infeasible.

8 solutions have been applied by the decision makers, meaning that at least one of the proposed decisions has been applied, which was a pleasant surprise. Since these experiments initially aimed at validating the consistency of the rescheduling solutions, and the decision makers were not expected to actually implement the proposed rescheduling solutions. 44% relevant solutions and 19% applied solutions confirmed the industrial interest of our optimization-simulation rescheduling approach.

Solutions	Number of solutions	Reduction of recovery time		Degradation of passengers' objective	Number of decisions
		min.	ratio		
Consistent	41	27	36%	3.4%	1.5
Relevant	19	30	35%	3.1%	1.5
Applied	8	32	36%	1.2%	1.3

Table 9: Average results for discussed rescheduling proposals

Table 9 shows average values of indicators for the consistent, relevant and applied solutions. Note that applied solutions are the most efficient: Reducing the most the recovery time while reducing the least the quality

of service for passengers, with a few less decisions applied. These results confirm that the evaluation criteria of our rescheduling approach are representing the criteria used by the decision makers to reschedule train traffic.

Decision makers reckoned that the graphical interface was clear and easy-to-use and that the tool helped them in their everyday work. They appreciated the alerts and the time savings brought by the rescheduling proposals, especially when the situation was disturbed on another line they had to manage. Most of the solutions were deemed interesting and consistent, and they noted the interest in the long projection horizon that they could not have without a tool. Their manager is also very enthusiastic about the tool and would also like to use it to train new decision makers.

6. Conclusion and perspectives

Following the diagnosis of the situation in the dense railway system of the Paris area and supported by the preliminary results in Altazin et al. (2017), this paper proposes a multi-objective rescheduling approach for limited disturbances. In order to minimize the recovery time and the impact of disturbances on passengers' waiting and travel times, various operational decisions are considered: Canceling and short-turning trains, adding and skipping stops. An approach combining an optimization module and a simulation module is developed to generate multiple non-dominated solutions, that are proposed to the decision makers. Real timetables, rolling-stock assignments and origin-destination data of passengers have been used to conduct numerical experiments, showing that the approach allows the recovery time to be significantly reduced in case of disturbances. A multi-objective analysis has also been carried out to study the quality and diversity of the proposed solutions. Based on these positive results, an operational rescheduling tool connected to real time data flows has been developed in collaboration with Transilien decision makers, and first real-life experiments were conducted in the fall of 2017. These experiments were successful, with most of the solutions proposed by the approach being judged as relevant, and even with some solutions been actually applied.

This work has different perspectives. First, the data collection and visualization functions of the decision support tool have been improved and extended to the six railway lines of the Paris Saint-Lazare region, for which new experiments have been conducted in 2018. In the near future, the rescheduling approach will actually be included in the Transilien traffic management

tool, and thus be generalized to all Transilien lines. The performances of the approach will have to be closely monitored when scaling to a larger network, as well as the computational time. Various research avenues are also being investigated. We are analyzing the relevance of extending our approach to consider the assignments of drivers and to reschedule the rolling stock. More operational actions could also be included if needed on different networks, for instance re-ordering trains, and the capacity of terminal stations could be refined to be more realistic. Another interesting perspective is to rethink how transportation plans are designed, and in particular how robustness should be ensured, based on the fact that our approach is now available to deal with disturbances.

Acknowledgements

This work has been partially financed by the ANRT (Association Nationale de la Recherche et de la Technologie) through the PhD number 2014/1195 with CIFRE funds and a cooperation contract between SNCF and ARMINES.

References

Altazin, E., Dauzère-Pérès, S., Ramond, F., Tréfond, S., 2017. Rescheduling through stop-skipping in dense railway systems. Transportation Research Part C: Emerging Technologies 79, 73–84.

Cadarso, L., Marín, A., Maróti, G., 2013. Recovery of disruptions in rapid transit networks. Transportation Research Part E: Logistics and Transportation Review 53, 15–33.

Canca, D., Barrena, E., Laporte, G., Ortega, F.A., 2016. A short-turning policy for the management of demand disruptions in rapid transit systems. Annals of Operations Research 246, 145–166.

Corman, F., D'Ariano, A., Marra, A.D., Pacciarelli, D., Samà, M., 2017. Integrating train scheduling and delay management in real-time railway traffic control. Transportation Research Part E: Logistics and Transportation Review 105, 213–239.

Dollevoet, T., Huisman, D., Kroon, L.G., Schmidt, M., Schöbel, A., 2015. Delay management including capacities of stations. Transportation Science 49, 185–203.

Dollevoet, T., Huisman, D., Kroon, L.G., Veelenturf, L.P., Wagenaar, J.C., 2017. Application of an iterative framework for real-time railway rescheduling. Computers & Operations Research 78, 203–217.

Eberlein, X.J., Wilson, N.H., Barnhart, C., Bernstein, D., 1998. The real-time deadheading problem in transit operations control. Transportation Research Part B: Methodological 32, 77–100.

Eberlein, X.J., Wilson, N.H., Bernstein, D., 1999. Modeling real-time control strategies in public transit operations, in: Computer-aided transit scheduling. Springer, pp. 325–346.

Fan, Y., Guthrie, A., Levinson, D., 2016. Waiting time perceptions at transit stops and stations: Effects of basic amenities, gender, and security. Transportation Research Part A: Policy and Practice 88, 251–264.

Gao, Y., Kroon, L.G., Schmidt, M., Yang, L., 2016. Rescheduling a metro line in an over-crowded situation after disruptions. Transportation Research Part B: Methodological 93, Part A, 425–449.

Gao, Y., Yang, L., Gao, Z., 2017. Real-time automatic rescheduling strategy for an urban rail line by integrating the information of fault handling. Transportation Research Part C: Emerging Technologies 81, 246–267.

Ghaemi, N., Cats, O., Goverde, R.M., 2018. Macroscopic multiple-station short-turning model in case of complete railway blockages. Transportation Research Part C: Emerging Technologies 89, 113–132.

van der Hurk, E., Kroon, L., Maróti, G., 2018. Passenger advice and rolling stock rescheduling under uncertainty for disruption management. Transportation Science 52, 1391–1411.

Jaszkiewicz, A., 2004. Evaluation of Multiple Objective Metaheuristics. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 65–89.

Jiang, Z., Gu, J., Fan, W., Liu, W., Zhu, B., 2019. Q-learning approach to coordinated optimization of passenger inflow control with train skip-stopping on a urban rail transit line. Computers & Industrial Engineering 127, 1131–1142.

Kroon, L.G., Maróti, G., Nielsen, L., 2015. Rescheduling of railway rolling stock with dynamic passenger flows. Transportation Science 49, 165–184.

Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. Transportation Research Part B: Methodological 67, 208–234.

Nesheli, M.M., Ceder, A.A., 2015. A robust, tactic-based, real-time framework for public-transport transfer synchronization. Transportation Research Part C: Emerging Technologies 60, 105–123.

Nielsen, L., Kroon, L.G., Maróti, G., 2012. A rolling horizon approach for disruption management of railway rolling stock. European Journal of Operational Research 220, 496–509.

Pareto, V., Bonnet, A., 1963. Manuel d'économie politique. Number vol.1 in Bibliothèque internationale d'économie politique, Librairie Générale de Droit et de Jurisprudence.

Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. Transportation Research Part B: Methodological 85, 89–108.

Sato, K., Tamura, K., Tomii, N., 2013. A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. Journal of Rail Transport Planning & Management 3, 38–53.

Schön, C., König, E., 2018. A stochastic dynamic programming approach for delay management of a single train line. European Journal of Operational Research 271, 501–518.

Shang, P., Li, R., Liu, Z., Yang, L., Wang, Y., 2018. Equity-oriented skip-stopping schedule optimization in an oversaturated urban rail transit network. Transportation Research Part C: Emerging Technologies 89, 321–343.

Veelenturf, L., Kidd, M., Cacchiani, V., Kroon, L.G., Toth, P., 2016. A railway timetable rescheduling approach for handling large-scale disruptions. Transportation Science 50, 841–862.

Veelenturf, L., Kroon, L.G., Maróti, G., 2017. Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. Transportation Research Part C: Emerging Technologies 80, 133–147.

Wagenaar, J., Kroon, L.G., Fragkos, I., 2017. Rolling stock rescheduling in passenger railway transportation using dead-heading trips and adjusted passenger demand. Transportation Research Part B: Methodological 101, 140–161.

Zhu, Y., Goverde, R.M., 2019. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. Transportation Research Part B: Methodological 123, 149–181.

Zitzler, E., 1999. Evolutionary algorithms for multiobjective optimization: Methods and applications. volume 63. Citeseer.