DTU Library

Susceptibility of optimal train schedules to stochastic disturbances of process times

Larsen, Rune; Pranzo, Marco; D'Ariano, Andrea; Gorman, Francesco; Pacciarelli, Dario

Robustness of Optimal Train Schedules
to Stochastic Disturbances of Process Times

**Larsen Rune · Pranzo Marco · D'Ariano
Andrea · Corman Francesco · Pacciarelli
Dario**

Abstract This work focuses on the stochastic evaluation of train schedules
computed by a microscopic scheduler of railway operations based on determin-
istic information. The research question is to assess the degree of sensitivity
of various scheduling algorithms to variations in process times (running and
dwell times). In fact, the objective of railway traffic management is to reduce
delay propagation and to increase disturbance robustness of train schedules
at a network scale. We present a quantitative study of traffic disturbances
and their effects on the schedules computed by simple and advanced schedul-
ing algorithms. Computational results are based on a complex and densely
occupied Dutch railway area; train delays are computed based on accepted
statistical distributions, and dwell and running times of trains are subject
to additional stochastic variations. From the results obtained on a real case
study, an advanced branch and bound algorithm, on average, outperforms a
First In First Out scheduling rule both in deterministic and stochastic traf-
fic scenarios. However, the characteristic of the stochastic processes and the

R. Larsen
Department of Mathematics and Computer Science - University of Southern Denmark, Den-
mark.
E-mail: rular@imada.sdu.dk

M. Pranzo
Dipartimento di Ingegneria dell'Informazione - Università di Siena, Italy.
E-mail: pranzo@dii.unisi.it

A. D'Ariano
Dipartimento di Informatica e Automazione - Università degli Studi Roma Tre, Italy
E-mail: a.dariano@dia.uniroma3.it

F. Corman
Centre for Industrial Management - Katholieke Universiteit Leuven, Belgium
E-mail: francesco.corman@cib.kuleuven.be

D. Pacciarelli
Dipartimento di Informatica e Automazione - Università degli Studi Roma Tre, Italy
E-mail: pacciarelli@dia.uniroma3.it

way a stochastic instance is handled turn out to have a serious impact on the scheduler performance.

Keywords Railway Traffic Optimization · Delay Minimization · Simulation · Uncertainty.

1 Introduction

Railway services are normally managed on the basis of a timetable (detailed plan of operations) that has been planned in advance with suitable time margins to absorb minor delays.

When planning the traffic over a railway network, it is usually assumed the presence of deterministic running times. Uncertainties are normally dealt with by insertion of appropriate buffer times in the timetable to absorb delays. However, during operations major disturbances cause traffic deviation from the planned timetable and thus may influence the timetable feasibility. In such cases, real-time adjustments of train timing and orders are required to assure compatibility with the real traffic situation and to limit delay propagation. This task is currently performed manually by dispatchers. There is thus a need for advanced decision support tools to reduce the dispatchers' workload and to satisfy the requirements of infrastructure managers, train operators and passengers [2]. Recently, advanced models and algorithms have been developed to reschedule trains in complicated railway areas with dense traffic and multiple delayed trains (see, e.g., the literature reviews in [5,14,18]).

In rescheduling of railway operations, the need to quickly find solutions has directed most efforts to develop advanced heuristic methods, that find a good solutions with a limited computation effort. This is in line with a general trend for scheduling and managing systems under uncertainty for complex real-world scheduling problems, that use solving procedures based on fully deterministic information [20,15]. Such approaches simplify the problems by feeding the scheduling routines with only deterministic and static data and neglecting further uncertainties. However, most real-life scheduling problems, and especially railway traffic scheduling, are inherently dynamic and prone to uncertainties. A key question is therefore whether the quality of the optimized solutions is partially or completely lost when dealing with uncertainty.

Assuming deterministic and static data may cause discrepancies between what is expected to happen in the model and what actually happens when the plan is applied in practice. In fact, the gap between the offline published plan and the actual schedule put into operations affects the operations' performance and railway passengers' comfort heavily.

Research aiming to increase the robustness of railway systems is generally directed at addressing uncertainty to fill this gap, mostly following two directions:

(i) computing schedules with some sort of slack to recover from disturbances,
(ii) assessing the effects of uncertainties of a pre-defined schedule before implementation.

In (i), optimal schedules can be computed using detailed models and dedicated algorithms, able to take the uncertainties and the dynamic nature of the problem into account. However such methods require the development of complex methods such as robust optimization [1] or stochastic programming [22] approaches that are generally harder to solve. The approach in (ii) is to use simple models able to compute efficient schedules based on deterministic information and to evaluate the effects of uncertainty, e.g., with the use of Monte Carlo simulations. This enables assessing the effects of stochastic disturbances on a schedule by running a simulation of the future evolution of the railway system.

This paper follows the (ii) approach. Our aim is to develop a tool able to assess the effects of unpredictable disturbances on a schedule enabling a more informed evaluation of different rescheduling algorithms. In fact when dealing with real life problems one should also include the possibility that the system might evolve in directions unknown at the moment of optimization, and not considered in the schedule generated by the solver. This concern is especially relevant when comparing solutions obtained by simple dispatching rules with solutions obtained by sophisticated optimization methods, as tighter packed schedules might be more susceptible to disruptions. This work studies the variability of optimized train schedules in presence of uncertainty. A simulation setup is proposed, in which train schedules are computed by minimizing delays in case of deterministically known perturbed operations. Such train schedules are computed by a microscopic optimization-based train scheduler, working at the level of block signals. We compute the process times in seconds. This detail is required to assess the propagation of delay between following trains precisely. The resulting solutions are then evaluated under small stochastic variations in process times, that simulate errors in input data due to uncertainty in train tracking, or further unpredictable events. This is done within a detailed microsimulation tool, that allows quantifying the effects of such disturbances precisely.

The paper is organized as follows: Section 2 briefly reviews the train scheduling and robustness analysis literature. Section 3 presents the proposed framework for the robustness evaluation while Section 4 describes the computational results by evaluating the robustness of three rescheduling algorithms on a complex dispatching area of the Dutch railway network. Finally conclusions and future research directions are highlighted in Section 5.

2 Literature on robust railway systems

A growing stream of scientific literature focuses on addressing how to generate robust timetables. Even if there is still no clear consensus on the exact definition of robustness, as also discussed in Dewilde et al. [10]. In fact several definitions have been proposed in the literature [3,11,16,23].

Among the papers dealing with robust timetabling problems we distinguish between approaches developing optimization-based timetables with some de-

gree of disturbance robustness, i.e., point (i) of the Introduction, and papers dealing with the assessment of the robustness of a timetable by means of simulation tools, i.e., point (ii) of the Introduction.

Liebchen et al. [17] use an integer programming approach to generate a delay resistant periodic timetable on a part of the German railway network. They sequentially evaluate it according to a delay management strategy. The problem of managing passenger connections is to decide which connections should be kept in the actual plan and which should be dropped [6,13].

Fischetti et al. [12] propose a light robustness approach to take robustness of a solution into account and at the same time maintain low computational times. On a medium sized network, the proposed approach results in two order of magnitude improvements for CPU time while maintaining almost the same solution quality.

Schöbel and Kratz [24] present a bi-objective approach for generating robust and efficient timetables. Given a level of uncertainty, the problem of finding a Pareto optimal timetable is solved as efficiently as the original single objective problem. However, it is generally difficult to predict the impact of the level of stochasticity of railway systems.

Shafia et al. [25] describe a mathematical model for train timetabling with some degree of robustness. As described by the same authors in their remarks, the model makes strong assumptions on safety issues related to train movements in the network. Based on this model and on two methods for the computation of buffer times, a branch and bound algorithm and a beam search-based heuristic are proposed by the authors for the computation of timetables. A real railway line is used for testing the algorithms while fictitious data are assumed for the travelling trains. The results show some improvements obtained by the two dedicated algorithms compared with a commercial software.

Vansteenwegen and Van Oudheusden [27] focus on the development of a robust timetable. They propose a two phase approach, in which, first, ideal buffer times with respect to train connections are computed, and then LP is used to build a timetable minimizing the waiting costs. Finally a discrete event simulation is used to compare the original timetables with the new one. The authors reports substantial improvements.

Simulation is a common approach to evaluate the robustness of train schedule, as in point (ii) of the Introduction. And we comment on some railway applications of simulation tools to evaluate timetable robustness:

Carey and Carville [4] compare the reliability of proposed schedules by simulating detailed train traffic in a busy complex station. The simulator uses a rescheduling algorithm to solve arising conflicts. Their approach can also be used to evaluate different reliability measures on a given timetable.

Salido et al. [23] propose two analytical measures of robustness of a timetable and use a macroscopic simulator to validate the analytical and the observed empirical robustness. They use their approach on a case study with about 30 trains to compare two timetables with different buffer times.

Takeuchi et al. [26] propose a robustness index, measuring the passenger disutility introduced by a disturbance. They simulate the traffic on a line with
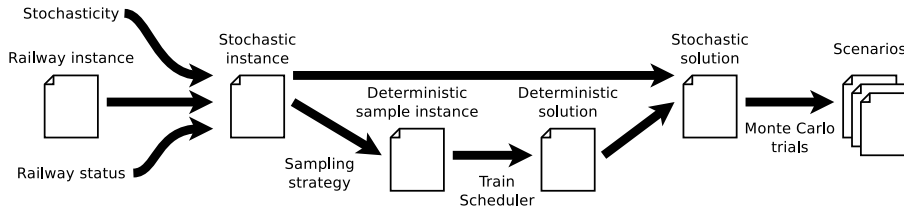
Fig. 1 Flow of information in the framework.

39 stations and 63 trains, to show the differences between two timetables by changing time margins in the timetable.

To the best of our knowledge, very limited research has addressed the topic of robustness during rescheduling of railway networks, with a microscopic detail, and solutions based on optimization algorithms. Only recently Meng and Zhou [19] developed a stochastic programming approach with recourse to incorporate different probabilistic scenarios in the rolling horizon decision. The approach reschedules trains in a perturbed situation, in order to minimize delays under different forecasted scenarios. Experiments on a single track line located in China with 18 stations and 50 daily circulating trains are carried out using a rolling horizon approach.

However there is no study trying to assess the performance of different rescheduling actions in real-time. Different train rescheduling plans can be obtained by using advanced conflict resolution methods, by solving train conflicts and by predicting the delay propagation in the overall studied area or by adopting local and myopic rescheduling rules. On the one hand, an optimization algorithm based on deterministic information may produce a more effective but fragile solution, i.e., an apparently good solution at the time it is computed, that is highly sensitive to uncertain factors, and that might result in unattractive plans for infrastructure managers and train operating companies. On the other hand, a rule-based algorithm may produce a seemingly less optimized but more robust solution, i.e., it may be able to absorb disturbances without a relevant degradation of the quality of the solution. However, in both cases, the impact of stochastic disturbances needs to be carefully evaluated before schedule implementation.

The main motivation of our study is thus to quantify the improvement that can be achieved in real time, if any, achievable by using optimization-based rescheduling methods under stochastic disturbances.

3 Robustness evaluation framework

Figure 1 presents the flow of data in the framework developed to evaluate the quality of train schedules when dealing with uncertain information.

The framework takes as input:

A railway instance. A deterministic railway instance contains a detailed description (at block section level) of the railway network under study, the

set of trains running in the network over the considered time horizon and the timetable they should follow on a railway network.

Railway network status. The current status of the network, i.e., train position and speed of all trains at time t_0, is assumed to be known. Based on this information it is possible to generate the actual entrance delays in the network that must be provided to the solver. The entrance delays are generated as deterministic values of train positions and speeds, that represent the current status of the network and the expected evolution in the short term.

Stochasticity information. The simulator needs to know the probability functions of the duration of all operations subject to uncertainty. If an operation is not deterministic the probability function should be known and specified. In our tests we considered uncertain duration for all running and dwell times.

All this input defines a *stochastic instance*. Since it is assumed that the train scheduling solver is not able to directly tackle a stochastic train scheduling instance the framework must transform it into a *deterministic instance*.[1] This is done as follows: The probability function associated to each operation (running times and dwell times) is converted into a deterministic value by means of a *sampling strategy*. As a sampling strategy one could consider taking the median of the probability function, the expected value, or a given quantile.

Next the train scheduler is invoked on the deterministic instance and it is expected to produce a *deterministic solution* to the problem. The scheduler uses a train scheduling algorithm in order to detect and solve potential conflicts, and outputs a microscopic plan of operations, with a level of detail of single block sections and a precision of seconds.

As solver we use ROMA [8,9], a computerized system that can assist train operators in their tasks. ROMA is able to estimate and control the future evolution of the railway traffic by considering deterministic information on track occupation and safety constraints. This may prevent the decision maker from taking less effective decisions, such as causing a deadlock situation or unsatisfactory throughput. The resulting solution represents a new plan of arrival and departure times that minimizes train delays with respect to an original timetable. ROMA also enables the optimization of railway traffic if the actual timetable is not conflict-free and/or deadlock-free, or during severe traffic disturbances, such as when emergency timetables are required and dispatchers need extensive support to solve train conflicts and to handle disruptions (i.e., unexpected blockage of some tracks).

After the computation of a new schedule by the solver, the framework replaces deterministic process times in the deterministic solution with their original probability functions from the stochastic instance, creating a *stochastic solution*.

---

[1] Clearly, if the solver is able to deal with a stochastic instance there is no need of removing stochastic data.
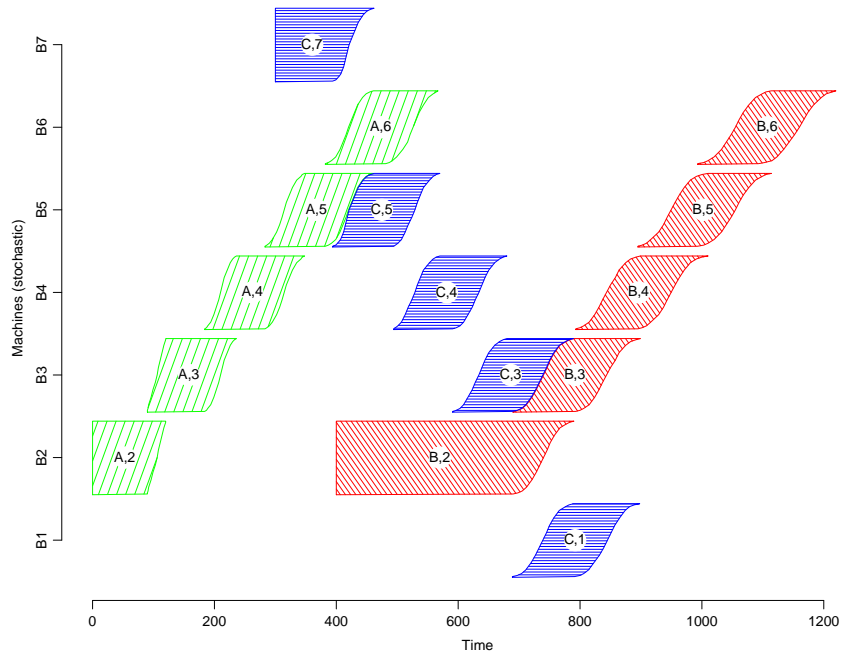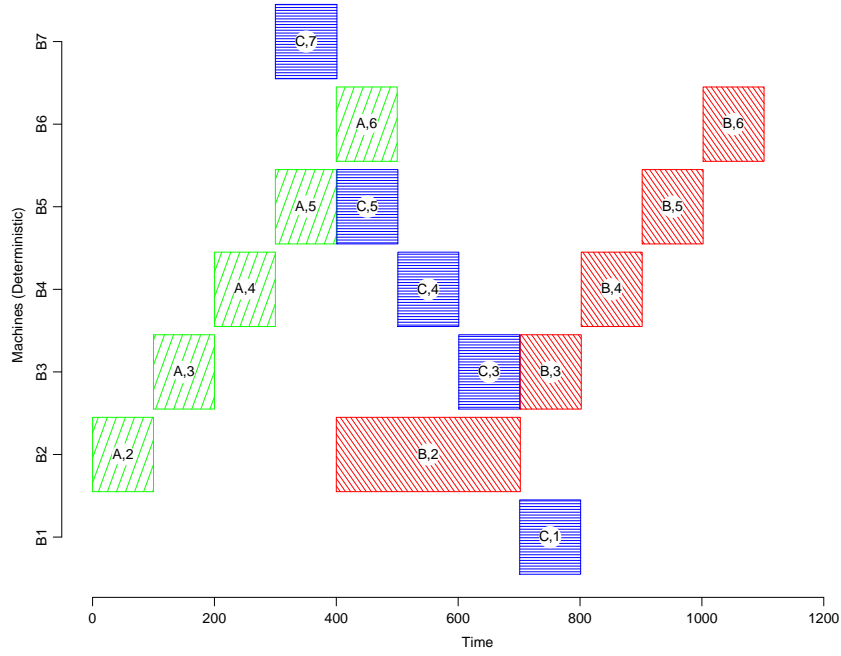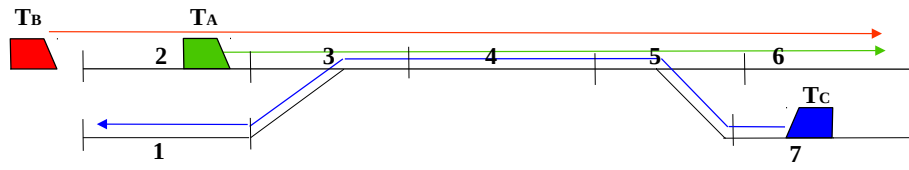
Fig. 2 The layout of the network (top). A deterministic solution (middle) and the corresponding stochastic solution (bottom), for a small instance with three trains. Each train passes a section in 100 time units and is subject to $[-10\%, 20\%]$ variability.

A set of Monte Carlo trials are then performed on the stochastic solution creating a set of sample solutions (*scenarios*). This transformation of the solution is depicted in Figure 2: The top plot represents the small railway network with three trains. The middle plot is a Gantt chart, in which all operations are deterministic, and shows two trains progressing along a track with another train traversing part of it in the opposite direction. The bottom plot is a stochastic Gantt chart, and instead of square block representing operations, its shapes height corresponds to the likelihood of the operation running at the current time. On the basis of the generated scenarios, train delay indicators are computed. We observe that computing the train delay indicators does not add excessive computational burden on the solver. In fact, the more costly operation in the framework is to evaluate the robustness of a solution. For large scale instances (as the one we consider in our tests) this can be done quickly. Furthermore, different solvers can be used as long as they are able to handle a stochastic or a deterministic train scheduling instance and provide a solution with the desired level of modeling detail.

4 Case study

In this section we first introduce the experimental setup of the tests and then we evaluate the robustness of different train scheduling algorithms.

4.1 The railway network

The test case is a complex and densely occupied area of the Dutch railway network, as in [7]. We study the central station of Utrecht area, which is a major hub of the Dutch railway network, at the crossroads of the five main lines criss-crossing the Netherlands. About 200.000 passengers use the station every day. Specifically, there are:

- a double track line to the west, towards Rotterdam and The Hague, served by 12 trains per hour per direction;
- a four-track line to the north-west, towards Amsterdam, served by 11 train per hour per direction;
- a four-track line to the north, towards Amersfoort, served by 12 trains per hour per direction;
- a double track line to the east, in the direction of Arnhem and Germany, served by 6 trains per hour per direction;
- and finally a double track line to the south, towards Den Bosch, served by 9 trains per hour per direction.

The train network under consideration is shown in Figure 3. The lines are interconnected by two complex interlocking areas at both sides of the station with a total of about 100 switches, and 20 platform tracks at the station itself. The area under consideration stretches along a few kilometres of the lines, for a diameter of about 20 km.
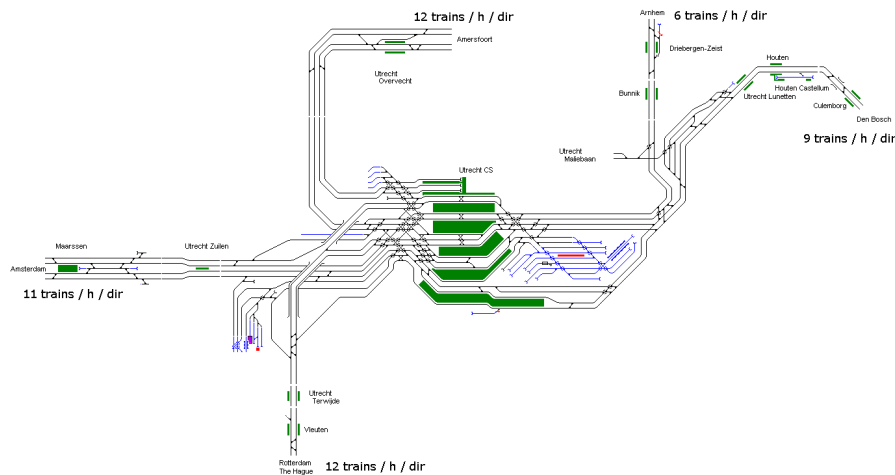
Fig. 3 Map of the Utrecht area.

4.2 Timetable and deviations

The 2008 timetable is considered, which is very dense, featuring 80 trains per hour on a regular-interval timetable. The traffic is mixed, divided approximately evenly between intercity trains and commuter trains, with a few long-distance, high-speed trains. We consider two train scheduling instances, with a time horizon of traffic prediction of 3600 and 7200 seconds. Those two instances define a set of 1962 and 3924 stochastic process times respectively, corresponding to travel and dwell times subject to delays.

For each of the two instances we assess 40 cases of entrance delays from [7], representing delays that have already happened before the time window under consideration in the instances, or that are expected in the short term. Each train has a *planned arrival time* defined by the timetable and associated with each scheduled stop, giving rise to different delay measures.

The difference between plan and realization is the *total delay* of a train at a scheduled stop (equivalent to a *tardiness* measure), and can be further divided in two parts: *Primary delays* that cannot be avoided even by letting train running at maximum allowed speed and with no hindrances, and *consecutive delays* (or *secondary delays*) that are caused by domino effects and usually can be partially avoided through a careful planning of the operations [5,7]. Since primary delays are by definition unavoidable, we use the minimization of consecutive delays as performance measures (that is the output delay due to potential train conflicts in the area under study). In the computational experiments, we show the minimization of the maximum and average consecutive delays.

The schedule robustness is evaluated by the analysis of small stochastic variations for the running and dwell times of train traffic in the network. Those

variations can be the result of e.g., measurements errors, coarse grained train detection data (for instance based on block section occupancy, rather than a precise positioning system), additional unexpected disturbances, overcrowding or additional delays at station platforms.

We generate 1000 random instantiations of the associated probability functions and use them to evaluate schedule robustness. In other words, each stochastic solution is evaluated over 1000 Monte Carlo trials (scenarios). In order to obtain a fair comparison, when evaluating the robustness of the solutions provided by different algorithms on the same instance, we apply exactly the same scenarios in the Monte Carlo trials phase.

Since the scheduling solvers are deterministic all durations must be supplied as deterministic values and not as probability functions. The proposed robustness evaluation framework has a single parameter that has to be set. Namely we have to specify how the sampling strategy works, i.e., how a stochastic instance is transformed into a deterministic instance. To this aim we use a sampling strategy parametrized on:

$$g \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$$

Those values represent the quantiles of the probability functions given in the stochastic instance, ranging from $g = 0.0$ to $g = 1.0$. Setting $g = 0$ means that the scheduling solver is considering an instance where all the times are set to the shortest possible duration allowed by the distribution of that operation. When $g = 0.5$ the sampling strategy provides the median times, and when $g = 1$ the maximum of each distribution is taken. As we use unbounded Weibull distributions (explained later), we treat $g \geq 0.99$ as $g = 0.99$ for them to avoid infinities. Note that for all the distributions used, the expected values of the distribution corresponds to using $g = 0.5$ (Uniform distributions) or $g \approx 0.56$ (the used Weibull distributions).

4.3 Performance indicators

Recall that consecutive delays were defined as the part of the total delays (tardiness) that were unavoidable regardless of scheduling decisions, and note that operations are indexed by j. Arrival times t_j^a are computed as longest paths in the temporal network representing the solution $G = (N, F \cup S)$, where F is a set of fixed edges that must be present in any solution, and S is a set chosen by a scheduling algorithm.

Given a planned arrival time t_j^p, the tardiness can be calculated as $t_j^t = \max\{t_j^a - t_j^p, 0\}$. Let $l(0, j)$ be the longest path in the temporal network without any non-fixed edges selected ($G = (N, F)$). The primary delay $t_j^u = \max\{l(0, j) - t_j^p, 0\}$ and the consecutive delay can be computed as $t_j^c = t_j^t - t_j^u$ which is guaranteed to be positive or zero.

In the train scheduling problem we consider both maximum consecutive delay:

$$\max_{j \in planned} (t_j^c) \tag{1}$$

average consecutive delay:

$$\frac{\sum_{i \in planned}(t_j^c)}{|planned|} \tag{2}$$

maximum tardiness:

$$\max_{j \in planned}(t_j^t) \tag{3}$$

and finally average tardiness:

$$\frac{\sum_{i \in planned}(t_j^t)}{|planned|} \tag{4}$$

where $planned$ is the set of operations with a planned arrival time and $|planned|$ is the cardinality of the set.

4.4 Solver settings

The scheduler works with deterministic information based on an alternative graph formulation of the train scheduling problem. Three scheduling algorithms are considered: the simple dispatching rule First In First Out (FIFO), a greedy heuristic (AMCC) [21] and a truncated Branch and Bound (B&B) algorithm [5]. The FIFO rule simply states that the train arriving first at a track junction passes first. The AMCC heuristic is based on the alternative graph formulation of the train scheduling problem [8] and it basically takes sequential decisions trying to avoid the worst possible decisions at each time. The B&B is an advanced branch and bound algorithm using both FIFO and AMCC as initial solutions and a Jackson Preemptive Schedule lower bound. In order to guarantee reasonable computing times the B&B is truncated after 120 seconds.

The deterministic scheduling solution computed by the three algorithms is transformed into a stochastic solution using predefined probability functions as explained in the following, and then evaluated using 1000 instantiations of the stochasticity in the process times. From our tests it turns out that, given a solution, the analysis of the 1000 scenarios can be done in about 10 seconds for the largest instance and all objectives. This time is affected by the measures of consecutive delays, as they require an extra longest path calculation to be made to determine primary delays. The process is easily parallelized if faster computation is needed.

4.5 Distribution of process times

We consider probability distributions for the running and dwell times as follows. Weibull distributions are used to characterize the variability of relevant process times, as in [7, 28]. Specifically, the same dataset of events is used as in [7], that contains about 33000 records of arrival and departure events at Utrecht station, corresponding to a month of operations. When studying

On peak and off peak Weibull distributions
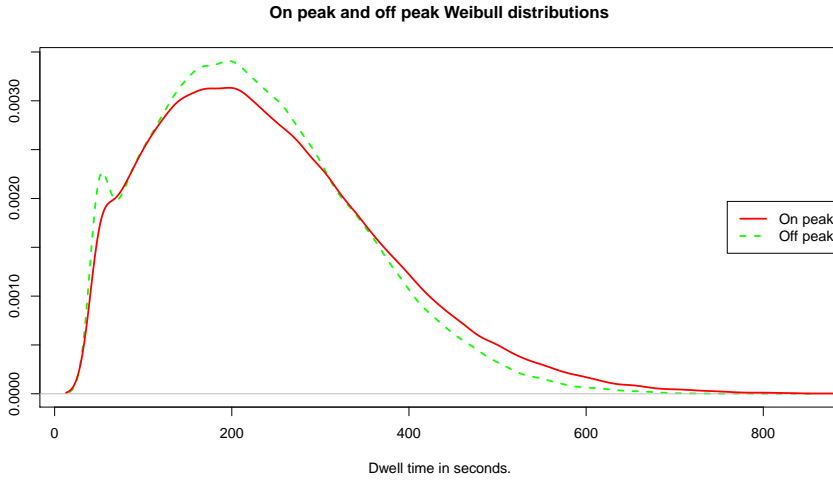
Fig. 4 Densities of dwell times for trains at stations. The spike at time 48 represents the enforced minimum. The height of the curve indicates the likelihood of the probability distribution assuming that delay.

dwell times, we divided the events into *on peak* and *off peak* events. A standard Maximum Likelihood fitting procedure is used to derive the three parameters describing the two Weibull distributions. Distributions of peak dwell times have a shape parameter $\kappa = 1.7914$, a scale parameter $\lambda = 255$ and a shift of -16 seconds towards shorter durations. The off peak dwell times have $\kappa = 2.0824$, $\lambda = 261$ and a shift of 4.

Densities for the two probability distributions (on- and off-peak dwell times) are plotted in Figure 4. As expected, during peak hours the larger amount of passengers results in longer dwell times. Both probability functions are bound on the lower values by a minimum technical stopping time of 48 seconds, resulting in a small peak on the probability density graph.

Another source of uncertainty is related to running times. This is modeled as a variation of the running time per every single operation (i.e. the passage of a train over a block section); the variation is considered normalized with regard to the planned duration, and follows uniform distributions.

The probability functions related to different trains and different block sections have no correlation in time or space. Unfortunately, little research has been focused on the assessment of the variability of individual running times at a microscopic level. Further research would be required to have a comprehensive description of the statistical characteristics of running time variations, based on recorded data.

We consider four different levels of distributions of running time variations: the intervals describing the uniform distributions are respectively $v = [-2.5\%, +5\%]$, $[-5\%, +10\%]$, $[-7.5\%, +15\%]$ and $[-10\%, +20\%]$, i.e. there is a slight bias towards longer durations.

4.6 Robustness evaluation

Using two time horizons of traffic prediction, four sets of process time stochasticity, on/off peak behavior, 40 entrance delay cases and 10 sampling strategies, we have 6400 runs per algorithm. Adding 1000 scenarios and a deterministic case, three algorithms (FIFO, B&B and Greedy) and four performance indicators, 76.876.800 objectives are evaluated. This was completed with 12 machines (Intel Core i5 CPU at 3.20GHz), with the last one finishing after 36 hours.

The plots in Figure 5 show the effect of the sampling strategy. The plots are averaged over all cases and on peak/ off peak dwell times. The right plots show the average and maximum consecutive delays obtained by the solvers while solving the deterministic problems, and the left plots are a result of applying these solutions to the 1000 scenarios. The right plots show a gradual increase in objective values, which is expected as all process and dwell times increase as g grows. Note also that this growth does not correctly represent what would happen with added random perturbations, as the plots on the left show. In fact, if solvers are optimistic (i.e., assume low values of g) when executing the schedule with added random perturbations the observed performance indicators are greater than expected by the solvers. On the other hand, when assuming high values for g, the objective function as seen by solvers is larger than what it happens when actually executing that plan with random durations.

The missing values in Figure 5 indicate that the greedy heuristic failed to find a feasible solution for $g \in \{0.4, 0.7, 0.8, 0.9, 1.0\}$. It does however produce competitive results for $g \in \{0.5, 0.6\}$.

The spread between the scheduling algorithms is in general limited, unless a large value of g is considered, representing a more conservative expectation of the process times. The closeness of results between algorithms for the same value of g indicates that the g parameter is a stronger predictor of the quality of the stochastic solution than the selected algorithm. Note that underestimating delays when solving the deterministic instances corresponds to low values of g ($g \leq 0.33\ldots$), and thus resulting (on average) in worse solutions. Using expected values for the probability distributions (corresponding to $g = 0.5$ for the uniformly distributed running times, and $g \approx 0.56$ for the dwell times following Weibull distributions) yields much better results (roughly 100 seconds less average consecutive delay), and if the distributions of the delays are unknown, overestimation is better than underestimation, as shown in Figure 5.

Let's observe how assuming $g = 1$ leads to poor solutions. This is due to the presence of unbounded distributions, i.e. the Weibull used to model dwell times. Specifically, assuming $g = 1$ ($g = 0.99$) corresponds to sample extremely large dwell time thus shadowing the travel times and producing poor quality solutions when stochasticity has been applied.

A value of $g = 0.6$ yields the best solutions for all three algorithms, and will thus be used in subsequent comparisons between the three algorithms.
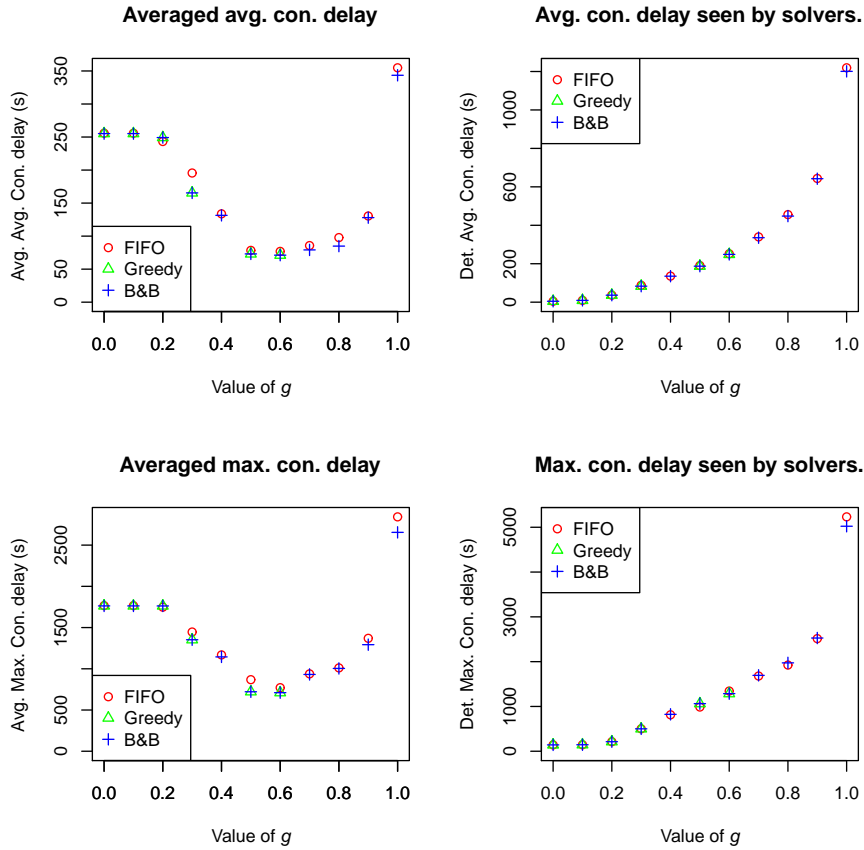
Fig. 5 Average values of consecutive delays as a function of the sampling strategy (g). The two plots to the right are the solution quality produced by the solvers, and the two plots to the left are the quality after the Monte Carlo samplings are applied to the same solutions.

Table 1 reports a quantitative comparison between B&B and FIFO, for the deterministic (first two columns) and stochastic (last two columns) solutions, in terms of the four delay indicators (reported in seconds). Namely, we consider average and maximum tardiness and average and maximum consecutive delays. The greedy heuristic has been omitted since we obtained the same solutions as B&B for $g = 0.6$. The presented results are average over the off-peak on-peak dwell time distributions, respectively. Overall, the B&B slightly outperforms FIFO. However, in case of high stochasticity B&B produces significantly better solutions compared to FIFO in terms of the minimization of maximum consecutive delays.

When comparing deterministic and stochastic instances in Table 1, it must be noted that the train delays are much larger when stochasticity is considered, on average 10% for total delay, and up to $\approx 250\%$ for average consecutive

Table 1 Average results on deterministic and stochastic instances for $g = 0.6$, split by off/on peak, process time stochasticity and objective function. The greedy heuristic has been omitted since we obtained the same solutions as B&B for $g = 0.6$. The % column indicates how many % FIFO was outperformed by B&B on average.

Off peak Stoch.	Obj. func.	Deterministic FIFO	B&B	%	Stochastic FIFO	B&B	%
$[-2.5\%, 5\%]$	Avg. Tard.	223.02	219.1	1.76	241.01	235.76	2.18
	Max. Tard.	1160.00	1160.00	0.00	1456.41	1460.19	-0.26
	Avg. Con.	27.36	23.44	14.34	63.75	58.50	8.23
	Max. Con.	256.50	253.00	1.36	667.89	658.09	1.47
$[-5\%, 10\%]$	Avg. Tard.	233.54	229.74	1.63	247.53	242.09	2.20
	Max. Tard.	1208.00	1208.00	0.00	1482.45	1482.15	0.02
	Avg. Con.	29.14	25.33	13.05	65.38	59.94	8.32
	Max. Con.	280.50	280.50	0.00	670.63	653.20	2.60
$[-7.5\%, 15\%]$	Avg. Tard.	242.40	240.76	0.68	254.76	249.91	1.90
	Max. Tard.	1252.00	1252.00	0.00	1508.07	1503.40	0.31
	Avg. Con.	30.46	28.81	**5.39**	67.79	62.94	**7.15**
	Max. Con.	305.50	303.00	0.82	676.21	652.42	3.52
$[-10\%, 20\%]$	Avg. Tard.	255.78	254.48	0.51	261.58	257.39	1.60
	Max. Tard.	1456.50	1304.00	10.47	1603.30	1531.23	4.50
	Avg. Con.	33.10	31.81	**3.91**	69.39	65.20	**6.04**
	Max. Con.	361.50	331.00	8.44	736.55	659.19	10.50

On peak Stoch.	Obj. func.	Deterministic FIFO	B&B	%	Stochastic FIFO	B&B	%
$[-2.5\%, 5\%]$	Avg. Tard.	247.05	244.63	0.98	282.09	277.27	1.71
	Max. Tard.	1238.00	1238.00	0.00	1640.27	1640.12	0.01
	Avg. Con.	32.89	30.47	7.35	82.74	77.91	5.83
	Max. Con.	325.50	297.00	8.76	777.00	754.66	2.87
$[-5\%, 10\%]$	Avg. Tard.	257.88	256.05	0.71	288.70	284.28	1.53
	Max. Tard.	1453.00	1286.00	11.49	1712.81	1664.68	2.81
	Avg. Con.	34.66	32.83	5.28	84.24	79.81	5.25
	Max. Con.	376.50	323.50	14.08	816.48	754.47	7.59
$[-7.5\%, 15\%]$	Avg. Tard.	266.92	265.55	0.51	295.37	291.05	1.46
	Max. Tard.	1477.50	1330	9.98	1735.24	1690.04	2.60
	Avg. Con.	36.13	34.76	**3.79**	86.08	81.77	**5.01**
	Max. Con.	362.00	348.50	3.73	817.38	760.29	6.98
$[-10\%, 20\%]$	Avg. Tard.	287.29	277.97	3.25	310.29	298.01	3.96
	Max. Tard.	1507.00	1507.00	0.00	1754.92	1753.14	0.10
	Avg. Con.	45.73	36.40	20.39	95.55	83.27	12.85
	Max. Con.	722.50	339.50	53.01	1006.18	789.95	21.49

delay. This reflects a high sensitivity of the train movements to delay propagation, and justifies the usage of microscopic models that can compute delay propagation between subsequent trains precisely. The ratio (% columns) between average delays and maximum delays (both consecutive delay and tardiness/total delay) produced by B&B and FIFO remains comparable across the different stochastic levels. B&B generates consistently better solutions com-
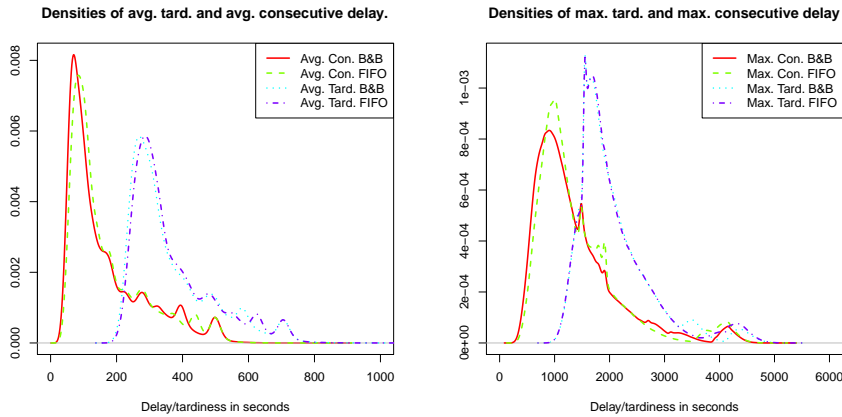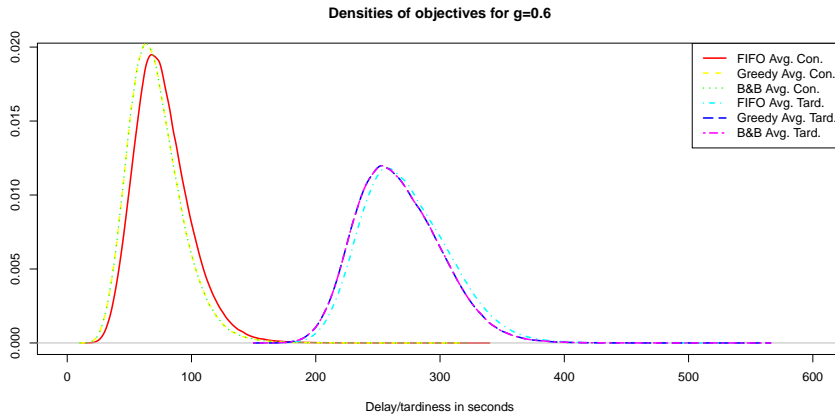
Fig. 6 Densities of the average delays and consecutive delays (left) and maximum delays and consecutive delays (right) for FIFO and B&B in seconds, for all g, for all v.

pared to FIFO in terms of maximum and average consecutive delay, both in the deterministic and the stochastic case.

Surprisingly the relative gap in average consecutive delay between B&B and FIFO increases for the stochasticity level $[-7.5\%, 15\%]$ (and other bold entries in the table), when stochasticity is added. It was hypothesized before running the tests that the tighter schedule would make more delays propagate. We also observe that the maximum consecutive delays produced by the B&B grow much slower than the maximum consecutive delays produced by FIFO as the stochasticity v increases. The difference between algorithms increases slightly with the amount of stochasticity v, due to both a greater expected value of the process times, and a greater chance of delay propagation due to the larger variability.

The plots in Figure 6 give further details on the statistic trends, aggregating over all values of g, all stochasticity levels v and on/off peak. Also in this case the greedy heuristic has been omitted from the plots, as it fails to give a feasible solution in all cases thus skewing any density plots. The plots on the left report the distribution of the averaged average[2] tardiness and the averaged average consecutive delay incurred by FIFO and B&B. Solutions computed by B&B are in general better, as observed in Table 1. This can be seen by a shift towards lower delays for both performance indicators. Another interesting feature of B&B is the reduced spread compared to FIFO, i.e., its distributions has a sharper peak. This means that B&B is able to come up with good solutions more consistently and suffers from uncertainty to a lesser extent. Similar trends arise for the average tardiness. Finally, the ripple pattern on the right tail of the figure is due to the aggregation of multiple values of g into a single plot. This highlights the role of the sampling strategy, that has a relatively large

---

[2] When two aggregators are specified, the inner aggregates over the operations in the solution.

Fig. 7 Densities of average consecutive delay and average tardiness for FIFO, greedy and B&B when $g = 0.6$, for all stochastic levels v.

influence on the final solutions. The plot on the right represents the densities of the average maximum tardiness and the average maximum consecutive delay. B&B outperforms FIFO on average consecutive delays as seen in Table 1, while having similar averaged maximum tardiness values. In general, the same patterns arise as for the average delays, but due to the high non-linearity of the maximum delay as indicator, the plots are more erratic, and trends are more difficult to visualize.

Figure 7 shows the contribution to left plot in figure 6 that comes from the case where $g = 0.6$, for all stochasticity levels v. The right tail shows a very neat reduction towards zero; the other trends that were discussed in Figure 6 are more easily seen. In this plot, the greedy heuristic and B&B are indistinguishable.

Figure 8 illustrates the difference in average consecutive delays between solutions produced by FIFO and B&B for all Monte Carlo samples for $g = 0.6$, for all stochastic levels v. Since the plot is skewed to the right is clear that B&B is the best algorithm, on average. On the other hand, the tails of the distributions are quite large, so there exist samples for which the solution provided by the FIFO performs better than the one produced by the B&B on the scenarios. However, the B&B solutions happen to be better than or at least as good as after stochasticity is added in the 77.47% of the cases for $v = [-2.5\%, 5\%]$, in 76.65% for $v = [-5\%, 10\%]$, in 73.97% for $v = [-7.5\%, 15\%]$ and finally in the 82.23% of the cases for the higher stochastic level $v = [-10\%, 20\%]$. The gap between B&B and FIFO in terms of average consecutive delays increases as the travel times become more uncertain.

Two scenarios were considered for dwell time durations: on peak and off peak. Figure 9 demonstrates a shift in average solution quality: when dwell time increases in the peak hours (Figure 4), the average consecutive delay increases significantly, with about 15 seconds deviation on average. Thus, a

FIFO – B&B Average consecutive delay in seconds for g=0.6.
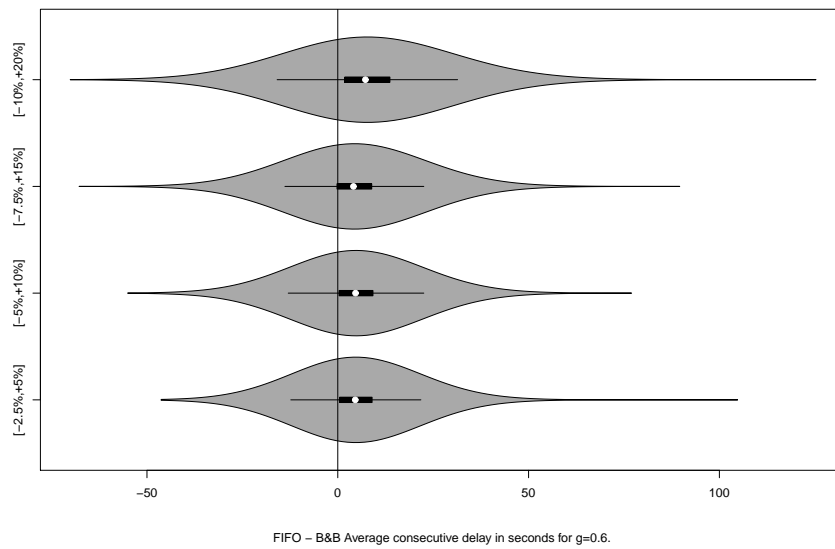
Fig. 8 Difference between FIFO and B&B with respect to average consecutive delay when $g = 0.6$, for all v. The area of shape left of the vertical line indicates the cases where FIFO outperformed B&B after stochasticity is considered, and vice versa. The corresponding box plots are drawn in the shape.
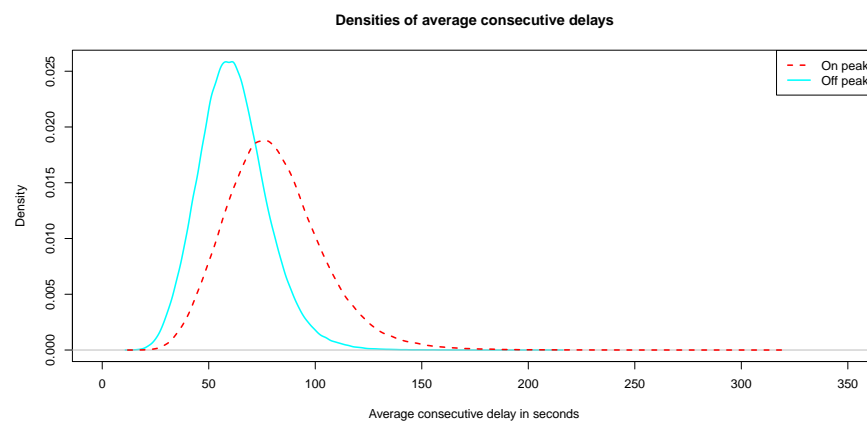
Fig. 9 Densities of average consecutive delay produced by B&B for $g = 0.6$, for all v. The steeper shape of the density plot for the off peak case indicates that the average consecutive delay tends to be smaller as expected.

relatively limited difference in dwell time probability may result in remarkable differences in output. This confirms the general idea that stations are the main bottleneck of railway systems, specially for stations with multiple platforms and tracks such as the one considered in this study, that have many conflicting inbound and outbound routes, and serve large amounts of passengers.

4.7 Needs for rescheduling

It can be observed that in practice, when the delays exceed a given threshold the human dispatcher may execute the rescheduling algorithm to produce a new plan more adherent to the current network status. Therefore an algorithm able to produce a robust schedule may lead to a reduced rescheduling frequency, which is desirable. The proposed robustness evaluation framework can then be used to evaluate how often the given rescheduling threshold is exceeded and the rescheduling process is triggered. To determine in what percentage of cases a rescheduling is required, we must first decide on the threshold level to consider. We used the average consecutive delay as objective function indicator, because it allows comparisons between instances with different primary delays. It is also clear that the tardiness measures are a poor basis on which to decide if a rescheduling is needed, as all of them might be unavoidable, i.e., caused by entrance delays an travel times. The percentage of simulated outcomes requiring rescheduling is plotted against the chosen threshold level based on the average consecutive delay in Figure 10. The figure shows the behavior of FIFO and B&B for a low stochasticity ($v = [-2.5\%, 5\%]$) and high stochasticity ($v = [-10\%, 20\%]$) situations when $g = 0.6$ is considered. From the plot it is clear that average values below 50 seconds are rare, while values greater than 150 seconds seldom occur. For values in between, there is a consistent gap between B&B and FIFO, indicating that choosing acceptable average consecutive delays in this interval ($[50, 150]$) would force more frequent reschedules for FIFO than for B&B.

4.8 Analysing the influence of a single dwell or process time

Another value of interest is the influence on the overall schedule of the variation of a single dwell operation. To this aim we have to identify the operation with the greatest potential impact on the average consecutive delay of all trains.

Longest path calculations done on the graph are used to obtain starting times for each operation, and allows for identification of critical operations. By selecting an operation o_i to analyze, changes can be made to that operations process times for all the sample graphs[3], and the resulting sets of objectives can be analyzed.

---

[3] By evaluating estimates for o_i in increasing order, longest path calculations can be reduced to propagate changes on paths rooted in the o_i in the temporal networks.

Percentage accepted solutions as a function of threshold values
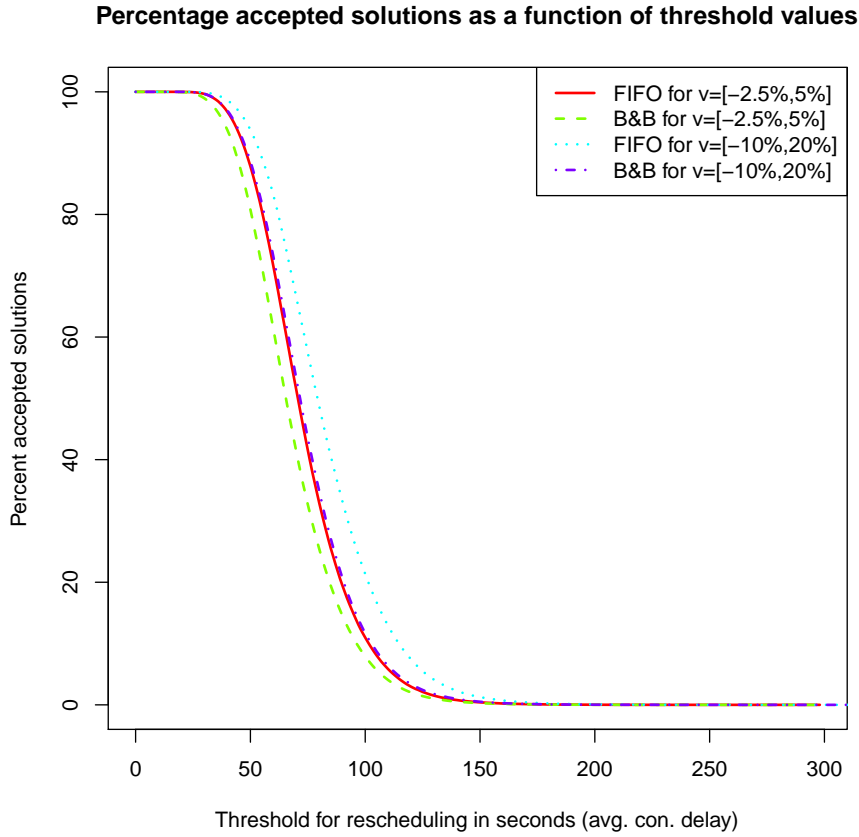
Fig. 10 If an acceptable level of average consecutive delay is defined, the resulting percentage of scenarios requiring rescheduling can be determined. These values are computed for $g = 0.6$.

The operation with the greatest potential impact on the objective is identified by applying the sampling strategies $g = 0.0$ and $g = 1.0$. For each of these two cases, the average consecutive delay is computed on and averaged over all Monte Carlo samplings. The difference in the two computed means represents an estimate on the averaged impact of changing that operations duration from the minimum to the maximum possible value.

Figure 11 shows the distribution of the average consecutive delay for different values of g for the operation changing the average objective the most. This operation corresponds to a dwell time as expected due to dwell times having a larger range of possible values. On the contrary, the corresponding plot for trains travel times shows little discernible difference for $g = 0$ and $g = 1$, even with $v = [-10\%, +20\%]$. This again confirms that while process

Maximum effect of dwell time
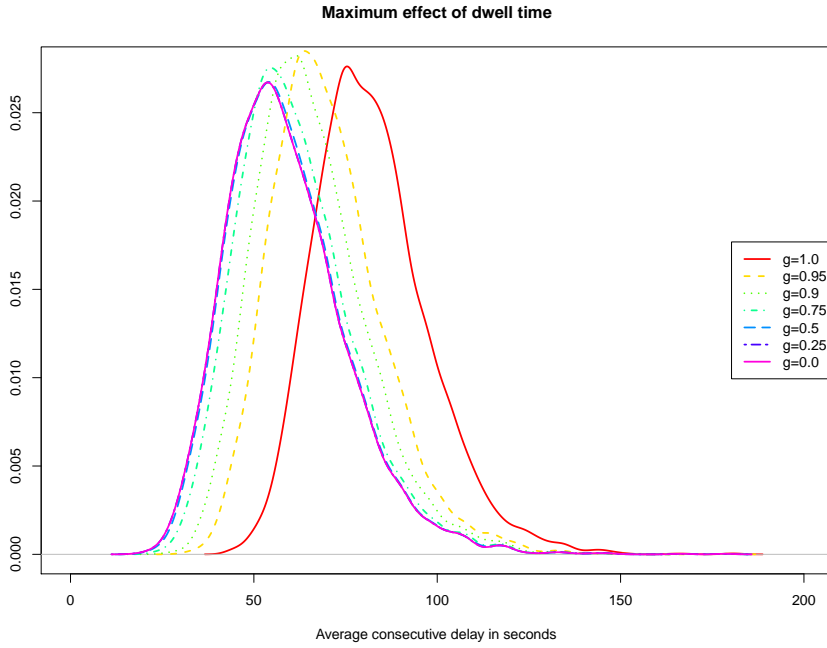
Fig. 11 Average consecutive delays sensitivity to changes in a single dwell time on a 3600 seconds instance.

times for trains travel summed up matters, their individual influence is easily drowned out by a single dwell time.

5 Conclusions and further research

This paper proposed a framework for evaluating the robustness of a given solution to the train rescheduling problem, where robustness under disturbances is the ability to handle small stochastic variations in running and dwell times of trains in the network, without increasing output delays. The robustness analysis is carried out with a very small computational cost, thus it is suitable for the real-time evaluation of different alternative solutions.

The behavior of three deterministic scheduling algorithms has been evaluated in a stochastic environment. The B&B solutions are only slightly less robust to variability of input parameters than the myopic FIFO rule that relies on local information only, as demonstrated in Table 1. The greedy heuristic performs well when it delivers results, but it must be deemed too unstable to use outside a framework such as B&B.

The influence of algorithms, stochasticity and sampling strategy was also assessed. From the results, it turned out that the sampling strategy, used to generate the deterministic instances, has significant effect on the solution

quality. Sampling strategies when ignoring delays (i.e. $g \approx 0.33$) performed significantly worse than strategies corresponding to using expected values for each distribution ($g \approx 0.5$). Furthermore it was shown that slightly overestimating the expected value of the distributions performs better than underestimation. This observation becomes important when the exact distributions are unknown.

B&B was originally feared to be more sensitive to stochasticity than FIFO, being the schedule produced by B&B tighter. We have shown, however, that B&B often outperforms FIFO after taking stochasticity into account.

Future research should be directed to:

- Exploring more sophisticated sampling strategies, and further determining the statistical properties of the recorded distributions of process times.
- Applying closed loop rescheduling based on the real time analysis of solutions robustness.
- Maintaining pools of candidate solutions, and guiding (as real time events occurs) execution of schedules by the plan exhibiting the best performance indicators.

References

1. Ben-Tal, A., Nemirovski, A.: Robust optimization - methodology and applications. Mathematical Programming 92, 453–480 (2002)
2. Biederbick, C., Suhl, L.: Decision support tools for customer-oriented dispatching. Lecture Notes in Computer Science 4359, 171–183 (2007)
3. Carey, M.: Ex ante heuristic measures of schedule reliability. Transportation Research Part B: Methodological 33(7), 473–494 (1999)
4. Carey, M., Carville, S.: Testing schedule performance and reliability for train stations. Journal of the Operational Research Society 51(6), 666–682 (2000)
5. Corman, F., DAriano, A., Pacciarelli, D., Pranzo, M.: Dispatching and coordination in multi-area railway traffic management. Tech. Rep. 2011-190, Dip. Inform. Autom. - Roma Tre University (2011)
6. Corman, F., DAriano, A., Pacciarelli, D., Pranzo, M.: Bi-objective conflict detection and resolution in railway traffic management. Transportation Research Part C: Emerging Technologies 20(1), 79–94 (2012)
7. Corman, F., DAriano, A., Pranzo, M., Hansen, I.: Effectiveness of dynamic re- ordering and rerouting of trains in a complicated and densely occupied station area. Transportation Planning and Technology 34(4), 341-362 (2011)
8. DAriano, A., Corman, C., Pacciarelli, D., Pranzo, M.: Modeling reordering and local rerouting strategies to solve train conflicts during rail operations. Transportation Science 42, 405–419 (2008)
9. DAriano, A., Pranzo, M.: An Advanced Real-Time Train Dispatching System for Minimizing the Propagation of Delays in a Dispatching Area Under Severe Disturbances. Networks and Spatial Economics 9(1), 63–84 (2009)
10. Dewilde, T., Sels, P., Cattrysse, D., Vansteenwegen, P.: Defining robustness of a railway timetable. Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis 2011, IAROR 2011 pp. 1–20 (2011)
11. Fischetti, M., Monaci, M.: Light robustness. Lecture Notes in Computer Science 5868, 61–84 (2009)
12. Fischetti, M., Salvagnin, D., Zanette, A.: Fast approaches to improve the robustness of a railway timetable. Transportation Science 43(3), 321–335 (2009)
13. Ginkel,A., Schöbel,A.: To wait or not to wait? the bicriteria delay management problem in public transportation. Transportation Science 41(4), 527–538 (2007)

14. Hansen, I., Pachl, J.: Railway Timetable & Traffic: Analysis - Modelling - Simulation. Eurailpress (2008)
15. Larsen, R., Pranzo, M.: A framework for dynamic rescheduling problems. Tech. Rep. 2012-01, Dip. Ingegneria dellInformazione - University of Siena (2012)
16. Liebchen, C., Lübbecke, M., Möhring, R., Stiller, S.: The concept of recoverable robustness, linear programming recovery, and railway applications. Lecture Notes in Computer Science 5868, 1–27 (2009)
17. Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S., Prigge, A.: Computing delay resistant railway timetables. Computers & Operations Research 37, 857–868 (2010)
18. Lusby, R., Larsen, J., Ehrgott, M., Ryan, D.: Railway track allocation: models and methods. OR Spectrum 33, 843–883 (2011)
19. Meng, L., Zhou, X.: Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. Transportation Research Part B: Methodological 45, 1080–1102 (2011)
20. Ouelhadj, D., Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling 12(4), 417–431 (2009)
21. Pranzo, M., Meloni, C., Pacciarelli, D.: A new class of greedy heuristics for job shop scheduling problems. Lecture Notes in Computer Science 2647, 223–236 (2003)
22. Ruszczynski, A., Shapiro, A.: Stochastic Programming. In: Handbooks in Operations Research and Management Science. Elsevier, Amsterdam (2003)
23. Salido, M., Barber, F., Ingolotti, L.: Robustness in railway transportation scheduling. Proceedings of the World Congress on Intelligent Control and Automation (WCICA) pp. 2833–2837 (2008)
24. Schöbel, A., Kratz, A.: A bicriteria approach for robust timetabling. Lecture Notes in Computer Science 5868, 119–144 (2009)
25. Shafia, M., Pourseyed Aghaee, M., Sadjadi, S., Jamili, A.: Robust train timetabling problem: Mathematical model and branch and bound algorithm. IEEE Transactions on Intelligent Transportation Systems 13, 307–317 (2012)
26. Takeuchi, Y., Tomii, N., Hirai, C.: Evaluation method of robustness for train schedules. Quarterly Report of RTRI (Railway Technical Research Institute) (Japan) 48(4), 197–201 (2007)
27. Vansteenwegen, P., Van Oudheusden, D.: Developing railway timetables which guarantee a better service. European Journal of Operational Research 173, 337–350 (2006)
28. Yuan,J.: Stochastic modelling of train delays and delay propagation in stations. TRAIL Thesis Series T2006/6, The Netherlands (2006)