



Transportation Science

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Interactions and Equilibrium Between Rescheduling Train Traffic and Routing Passengers in Microscopic Delay Management: A Game Theoretical Study

Francesco Corman

To cite this article:

Francesco Corman (2020) Interactions and Equilibrium Between Rescheduling Train Traffic and Routing Passengers in Microscopic Delay Management: A Game Theoretical Study. *Transportation Science* 54(3):785–822. <https://doi.org/10.1287/trsc.2020.0979>

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to download this work and share with others, but cannot change in any way or use commercially without permission, and you must attribute this work as “*Transportation Science*. Copyright © 2020 The Author(s). <https://doi.org/10.1287/trsc.2020.0979>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.”

Copyright © 2020, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Interactions and Equilibrium Between Rescheduling Train Traffic and Routing Passengers in Microscopic Delay Management: A Game Theoretical Study

Francesco Corman^a

^a Institute for Transport Planning and Systems (IVT), ETH Zurich, Zurich 8093, Switzerland

Contact: francesco.corman@ivt.baug.ethz.ch,  <https://orcid.org/0000-0002-6036-5832> (FC)

Received: December 4, 2018

Revised: July 7, 2019; October 15, 2019

Accepted: December 6, 2019

Published Online in Articles in Advance:
May 4, 2020

<https://doi.org/10.1287/trsc.2020.0979>

Copyright: © 2020 INFORMS

Abstract. In the last decade, optimization models for railway traffic rescheduling mostly focused on incorporating an increasing detail of the infrastructure, with the goal of proving feasibility and quality from the point of view of the managers of the infrastructure (tracks and stations). Different approaches that manage only the passenger flows instead focus more explicitly on the quality of service perceived by the passengers. This paper investigates microscopic railway traffic optimization models and algorithms, merging these two streams of research. In particular, we analyze the characterization of an equilibrium point between the reordering choices of train dispatchers in railway traffic optimization and the route choice of passengers in the available services of the railway transport network. We describe how passenger choice at stations along the route intertwines deeply with the problem of rescheduling trains over tracks and station resources in a very complicated setting that might not exhibit equilibrium points in general. Delaying trains and/or dropping passenger connections and/or giving particular route advice to passengers might influence the behavior of traffic controllers and passengers, determining a trade-off between the delays of trains, weighted by the passenger load, and the travel time of passengers. We study this problem with a game theoretical approach, focusing on the solutions corresponding to Nash equilibria of a game involving passengers and infrastructure managers. The proposed game theoretical approach is able to easily consider information and interdependence of the actions of multiple stakeholders. Computational results based on a real-world Dutch railway network quantify the trade-off between the minimization of train delays and passenger travel times and the performance, stability, and convergence of the equilibrium point given different algorithms and information available. The final aim of this work is to study the impact of effective implementations of railway traffic management and dissemination of information to passengers and operators.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. You are free to download this work and share with others, but cannot change in any way or use commercially without permission, and you must attribute this work as "Transportation Science. Copyright © 2020 The Author(s). <https://doi.org/10.1287/trsc.2020.0979>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-nc-nd/4.0/>."

Keywords: delay management • train scheduling and routing • passenger routing and route choice • game theory • passenger assignment • route choice and information

1. Introduction

Optimization of railway service is a key factor to reduce congestion on multimodal networks, especially in densely populated areas, and to provide an eco-friendly and sustainable way of transport. To attract new customers from other transport modes, European countries defined challenging targets in terms of quality of service (QoS) that the railway companies should provide to their customers (Caprara et al. 2006, ERRAC 2012, Pellegrini et al. 2014).

Many stakeholders interplay in the railway transport perspective: passengers who want to move from their origin to their destination, train operating companies (TOCs) that are selling transport services to

passengers between stations, and the infrastructure manager (IM) who sells infrastructure capacity to the TOC along railway lines. The viewpoints of the different stakeholders are different and translate into different paradigms to determine a best solution when managing railway traffic when delays occur. This paper addresses the interactions, the equilibrium, and the trade-off between the objectives and the decisions of the above-mentioned stakeholders, IM, and passengers (which we assume have the same interest as the TOC). Although the IM objective and decisions relate to rescheduling, to decrease train delays (Lamorgese et al. 2018), we consider a more general case in which the objective of reordering trains is to reduce passenger

delays. The passengers take decisions determining their route choice in the network, with the objective of minimizing their travel time.

We show that those two slightly different objectives and decisions align sometimes, but not always, and the resulting interaction might result in gaps between the promised system performance and the actual performance. In heavily used railway networks, when microscopic capacity limitation is considered, any small delay of a few trains easily propagates to the other trains, and rescheduling trains in real time to minimize train delays also directly addresses the travel times of passengers. On the one hand, the IM can only take decisions about train orders, with the goal of minimizing train delays; taking into account the relative importance of different trains related to the number of passengers onboard, passengers delays can also be reduced (typically, this is established as a weight factor per train). On the other hand, changing train orders may result in extra delays for passengers who miss a connection at some station. We assume that passengers have free route choice in the services provided on the railway network, so they can choose another route. The problem at hand is to determine which control actions of the railway traffic control would actually result in a reduction of passenger travel times, given the free route choice of the passengers as a reaction to the train order decisions.

The literature on the subject of railway traffic control experienced a significant growth in recent years (Caprara et al. 2006; Cacchiani et al. 2014; Binder, Chen, and Bierlaire 2014; Binder, Maknoon, and Bierlaire 2017). The existing models can be classified according to two different criteria. *Microscopic* and *macroscopic* models differ on the level of detail of the railway infrastructure, the first case aiming at including in the model all the relevant details of the railway infrastructure having an impact on railway traffic rescheduling and the second limiting the description at a higher level. As a different classification, *operations-centric* models focus on the minimization of railway operations objectives, such as train delays, whereas *passenger-centric* models focus on maximization of the QoS perceived by the passengers.

The first key contribution of this paper is to bridge those two paradigms and consider the interaction between two stakeholders (IM and passengers), two possible actions (reordering and route choice), and two objective functions (delays and travel time) as a game between IM and passengers.

In the proposed game, the strategy of the IM consists of determining train orders such that, given some passenger loads, the passenger delay (modeled as train delay, weighted by the passenger load) is minimized. The passenger loads (number of passengers expected on a train) implicitly define the relative importance of the different circulating trains, for

example, by associating a weight for each train equal to the number of passengers expected on board the train at scheduled stops during the service. The number of passengers expected on a train depends on the route choice of passengers and can be computed by a passenger routing procedure exploiting common assumptions on rationality and information provision to passengers. In fact, the strategy of the passengers is to find the trains delivering the shortest travel time despite delays and reordering actions. We show that, in general, the scheduling solution computed by a train traffic control might not always be in equilibrium with the passenger assignment solution; that is, passengers can improve their travel times by taking a different route compared with the passenger loads assumed by the IM, minimizing passenger delays. This also gives rise to a trade-off between passenger-oriented performance and train-oriented performance and the issue of the convergence of algorithms related to game theoretical approaches.

To determine possible solutions of the game theoretical study tackling the microscopic delay management problem, we use the formulation of the microscopic delay management (MDM) problem studied in Corman et al. (2016). The peculiarity of the formulation proposed in this paper resides further in the combination of the microscopic level of detail in managing traffic flows inside the station areas and passenger information availability. This level of detail is required to cope with train conflicts occurring at or around busy station areas and at interlocking areas when dealing with delayed traffic. The MDM formulation is solved by the generalizations of heuristic algorithms proposed in Corman et al. (2016). These heuristics iteratively solve a train scheduling problem integrated with a passenger routing problem (corresponding to the IM and passenger problems). We show how the solutions delivered by these game theoretical approaches correspond to Nash equilibria of the strategic interaction between IM and passenger players. We also consider solutions to the microscopic delay management problem considering other approaches with different forms of interaction, which can be framed as results of Stackelberg games. Based on specific scheduling solutions, for instance, the timetable solution, and the solution to the exact train rescheduling algorithm in D'Ariano, Pacciarelli, and Pranzo (2007), one can derive an associated microscopic delay management solution. We here remark that game theoretical (or algorithmic) studies in the field of railway transport scheduling are quite rare, with a few works tackling the offline timetabling point of view, where the strategic and political effects of decisions are much larger (Klabes 2010). To move toward a Nash equilibrium, iterative traffic optimization models are considered that alternate a move of the IM, a microscopic train

scheduling and routing model with the minimization of passenger delays (modeled as total weighted train delays) and a move of the passengers, and a passenger assignment model with minimization of total passenger travel times for a given train schedule.

A second key contribution of this paper is the theoretical study of the existence of equilibrium points in such games when including infrastructure capacity limitations. An equilibrium point cannot be always found, even when no capacity of the vehicle is considered. We believe that one reason for this is the freedom of route choice by passengers, which is influenced by the information available to them, which is immediate and complete.

This leads to the third key contribution of this paper, namely exploiting the game theoretical approach to consider the different grades of information about the decisions expected from the other player. To this end, we consider the difference between (suggested) passenger route advice and (actual) passenger route choice. We relax the assumption in Corman et al. (2016) and in most of the literature that passengers have full information at any time of their trip and study the impact of rerouting of passengers in the case of various degrees of partial information on the current status of traffic flows, which introduces a difference between suggested route advice and actual route choice.

The paper is organized as follows. Section 2 presents a literature review of mathematical models for the support of TOC/passengers and IM. Section 3 introduces the overall problem studied in this paper. Section 4 describes the proposed methodology to compute efficient solutions to the overall problem of microscopic delay management. Section 5 shows the results of the computational experiments on a large and complex Dutch railway network from the point of view of the algorithmic development, convergence, and solution quality. Section 6 draws the conclusions of this work and highlights directions for further improvements of the methodology.

2. Literature Review

2.1. Operations-Centric Railway Traffic Models

The state of the art of the railway rescheduling problem (which in the literature has been named the *real-time railway traffic management problem*, *train scheduling or rescheduling*, *dispatching*, and *railway traffic control*) focused mostly on *train rescheduling algorithms*, designing in real time, given the initial positions of trains and a reference timetable, a microscopically feasible schedule, with the goal of train delay minimization, that is, minimizing the difference between the planned arrival times and the actual arrival times (Cacchiani et al. 2014).

These algorithms are based on mathematical formulations and usually consider infrastructure capacity

limitations at the level of block sections and compute solutions to the train scheduling problem with the precision of seconds, incorporating as many practical details as necessary to ensure that schedule feasibility and the objective functions typically focus on train delays. The complexity of the train rescheduling problem (which includes determining typical control actions such as retiming, reordering, and rerouting train traffic) is due to the limited available capacity of railway lines and stations plus the constraints required to model the operating and safety rules, including the behavior of the failsafe signaling system.

Considering a sufficient level of detail of traffic flows allows investigation of the competition among train operators for the available railway capacity; in particular, the operating and safety rules need to be modeled according to the blocking time theory (Hansen and Pachl 2014).

Because the problem of computing a feasible scheduling solution has been shown to be nondeterministic polynomial-time hard (NP hard; Mascis and Pacciarelli 2002), much academic effort tackled the complexity of this problem. Models have been developed that are to some extent using graph theoretical structures (based on generalization of disjunctive graphs and alternative graphs), general-purpose mixed-integer linear programming (MILP) formulations, and purely combinatorial approaches. For some works, graph theoretical approaches are used as models to determine a relevant MILP model. Among the most relevant works in the literature, we mention the alternative graph-based approaches reported in D'Ariano, Pacciarelli, and Pranzo (2007); Corman et al. (2011, 2012); Corman, D'Ariano, and Hansen (2014); Dollevoet et al. (2014); and Samà et al. (2016) and the optimization models introduced in Carey and Crawford (2007); Törnquist and Persson (2007); Vansteenwegen and Van Oudheusden (2007); Meng and Zhou (2014); Pellegrini, Marlière, and Rodriguez (2014); Lamorgese and Mannino (2015); and Van Thielen, Corman, and Vansteenwegen (2018).

Most of these contributions consider only railway operations, in which a single actor takes decisions about railway orders, times, and routes. Passengers are mostly neglected or included as weights in objective functions. Some contributions described some more complicated model of passenger reaction and/or related QoS. Corman et al. (2011) presented a train rescheduling framework based on priority classes. An iterated lexicographic optimization is proposed to minimize the delay of the trains in each class such that the delay of trains in higher-priority classes does not increase. A generalization of this framework could take into account the overall passenger QoS in the definition of priority classes. Corman et al. (2012) proposed a biobjective optimization model in which passenger connections, indirectly relating to passenger

travel times, are weighted depending on the QoS. A Pareto frontier is computed for the minimization of train delays and the total weight of broken connections.

In general, the proposed models and algorithms are often able to manage traffic flows in networks of limited size and/or under few disturbed operations. However, the scheduling solutions produced by these models and algorithms demonstrate remarkable improvement with respect to current practice and/or to a surrogate of the practical dispatching rules. Most of the train rescheduling models and algorithms have a serious weakness related to the limited consideration of the QoS perceived by passengers, passengers' reactions to train order decisions, and information about the latter.

2.2. Passenger-Centric Railway Traffic Models

Some works try to move a step forward in considering the QoS by measuring the effects of passenger broken transfer connections, platform changes, or routing alternatives. To this end, *delay management models* focus on macroscopically feasible schedules with passenger flow optimization; the broader focus allows consideration of routes between stations neglecting precise infrastructure capacity, a precision of minutes, and less stringent limits on computation time. In this stream of work, passengers are assumed to follow route advice computed by the IM, which typically takes decisions as reactions of the observed traffic and passenger loads. In particular, the problem of deciding whether to keep or not transfer connections during operations is a crucial decision that directly affects passenger QoS (Schöbel 2007).

From a purely regulatory perspective, the goal of the IM is often not directly pertaining to passengers. The inclusion of passengers' perspectives in timetabling and railway traffic control (see, respectively, Parbo, Nielsen, and Prato (2016) and Josyula and Törnquist Krasemann (2017) for a discussion of both; see Lamorgese et al. (2018) for a discussion of goals of the IM in railway traffic control) clashes with the policy objectives of the IM, who should provide nondiscriminatory access to infrastructure and not address the commercial competing objectives of the TOCs. The gap between a passenger perspective and operator perspective is larger when complex multimodal networks are considered, rich in transfers, and when (multiple operators with) different service levels share the same infrastructure. Both papers also state that the current legal framework makes it much easier (when not needed) to focus on operator performance and train delays rather than on passenger delays because of the liberalized market where TOCs are providing their services. They both conclude that a larger focus on QoS rather than quality of operations should be considered for timetabling and railway traffic control, respectively.

A first stream of research on the optimization of the QoS perceived by passengers in passenger-centric train dispatching is introduced by Schöbel (2001) and Suhl, Biederbick, and Klierer (2001). This focuses on the minimization of passenger dissatisfaction, which is often translated into some (weighted) travel time formula. For instance, Espinosa-Aranda and García Rodenas (2013) directly include expected demand as a weight in the railway traffic management problem. A custom-designed heuristic thus minimizes a weighted train delay problem. No action or reaction from the users is taken into account as a consequence of the choices made by the IM. The delay management problem has been gradually investigated under increasing realism (Schachtebeck and Schöbel 2010, Dollevoet et al. 2012, Dollevoet and Huisman 2014, Dollevoet et al. 2015), including choices to reroute passengers between their origin and destination, which is a second crucial decision for passenger flows.

The limited detail on infrastructure capacity and train separation resulting from the safety system lead to a gap between the QoS promised by the solutions delivered and the QoS that can be achieved when implementing the solutions within the actual limitations and rules of practice. This has been addressed by so-called microscopic delay management (MDM) models, which consider passenger flows integrated in microscopic train rescheduling. Tomii et al. (2005) introduced a first microscopic railway traffic rescheduling model with the minimization of passenger dissatisfaction. Sato, Tamura, and Tomii (2013) addressed the problem of minimizing passenger inconvenience on simple railway lines, taking into account disruptions that might require adjustment of vehicle schedules. Passenger inconvenience is divided into three components (i.e., the time spent onboard the trains, the waiting time at platforms, and the number of transfers). The formulation consists of a mixed-integer programming problem that is solved by commercial software. The models report in general that trains can decrease passenger inconvenience by increasing delays (i.e., for catching more passengers at a busy station). Control actions for railway lines without rerouting of passengers are studied. Passenger flows are routed along the path of minimum inconvenience and then considered fixed while the timetable is adjusted to a minor extent. Capacity at stations is approximated by means of headway times along the line.

Dollevoet and Huisman (2014) iterated between solving a macroscopic delay management model and solving a microscopic train rescheduling model. The procedure delivers good feasible solutions in few iterations. An MDM model with passenger routing is introduced in Corman et al. (2016), in which passengers are always assigned to a shortest path between

their origin and destination. The MDM problem is modeled as a MILP problem that can be solved to optimality by commercial solvers for small size instances. Various heuristic procedures are developed based on decomposing the problem into a train rescheduling problem and a passenger routing problem. However, that paper focused only on algorithmic developments addressing passenger travel time and not exploring trade-offs, equilibrium, or convergence when including the free route choice of the passengers and the effect of information.

Binder, Chen, and Bierlaire (2014) proposed the time spent by passengers in the system as the most important indicator for passenger discomfort and pointed out the need for further research to better evaluate the quality of the solutions provided by heuristics. To this end, the introduction of an overall optimization framework is needed. The follow-up (Binder, Maknoon, and Bierlaire 2017) directly focused on the problem of considering capacity limitations, albeit only in the vehicle (i.e., not in the infrastructure), as one critical aspect in the choice and actions of travelers.

Some works include the uncertainty on passenger number, which is inherent in the online version of the problem, and tackle the issue of determining competitive decisions under incomplete information. Gatto (2007) first introduced the theoretical problem of deciding if a train should wait or not based on the number of people actually encountered at stations. A similar setting led to the competitive analysis of Bender, Büttner, and Krumke (2013). In those contributions, the railway infrastructure is particularly simple (a line), and its dynamics are simplified (i.e., no detailed track sections or signal and no running-time buffer to be exploited), whereas the inclusion of uncertainty in the actions of a single decision maker (the train operator) is the focus of the problem. Few approaches have been tried successfully in real life, and this includes the Panda algorithm (Rückert et al. 2017), which classifies future transfer connections based on the risk of potential passenger delays, given some assumed passenger loads and the current train delays. Static passenger flows are rerouted according to a model resembling the ones presented in Nielsen, Landex Rasmus, and Frederiksen (2009). In this last work, a railway planning problem is solved, considering effects on passenger flows by expected delays, with an increasing degree of realism. The further reaction of reordering trains as a reaction to delays is not considered.

In conclusion, most of the microscopic train traffic management models and algorithms in the literature still neglect the impact of train rescheduling decisions on the QoS to passengers, whereas most of the models

and algorithms focusing on the passenger perspective miss a detailed description of the rail infrastructure.

2.3. Passenger Guidance and Route Choice

We now review works that deal with passenger guidance. This is the problem of computing and disseminating route advice to passengers (guidance) and evaluating the results as filtered by the compliance that passengers might have toward the advice (i.e., their actual route choices), either for behavioral reasons or because the advice is incomplete, imprecise, or impossible.

We remark that the study of passenger routing (or route advice) in railway networks is quite an established field, especially for timetable information purposes (see Goerigk et al. 2013). What has been much less studied is the interaction between route advice and route choice in normal or exceptional situations. Concerning normal delays, preliminary works on delay management (Schöbel 2007) assumed that passengers would wait a full period if a connection would have been canceled (i.e., the route advice was specified in terms of the same service) for the successive period of the timetable. Schmidt et al. (2017) addressed the problem of passenger routing when no information is available (i.e., for a disruption), such as an exceptional event that has an unknown duration and is blocking some railway line. In this case, the choices of each passenger are how long to wait and which service to take. Complementary works deal with information coming over time, that is, an online setting of the problem for which competitive analysis has been provided (see Bender, Büttner, and Krumke 2013) or heuristics can be practically implemented that consider passenger flows to a limited extent (Bauer and Schöbel 2014).

More research on the interaction between information and route choice has been done in the general transit literature. A consistent stream of research refers to frequency-based transit assignment (Meschini, Gentile, and Papola 2007), stemming from concepts such as hyperpaths (Spiess and Florian 1989). In those models, passengers moving on the network take decisions on the fly based on which events would arise first; thus, route advice is not a sequence of services but rather a set of possible strategies. To model the decision-making process of passengers with (possibly) limited information and (possibly) limited compliance, Cats et al. (2011) used mesoscopic agent-based models, where agents represent passengers reevaluating their decisions at any decision point, such as at stops, departures, and merging of lines or services.

Those two latter assignment models typically result in a simulator of observed behavior, which can be further used to study the impact of policy settings,

whereas they lend to a minor extent to optimization purposes. Analysis of real-life situations has become more frequent in very recent years. Böhmová et al. (2015) proposed routing advice based on predicted operations, determined based on realized (vehicle) operations. Nassir, Hickman, and Ma (2018) studied how passenger route choices can be inferred from realized data by matching choice models to observed trips in smart card data transactions. Van der Hurk et al. (2015) studied how to determine the realized path (i.e., the chosen route) of travelers given the entrance and exit points of a railway network, with the purpose of studying their behavior especially in disruption situations. Delling et al. (2014) investigated the similar problem of integrating vehicle (position) information in the search for better route advice in transit networks.

In conclusion, the availability of so many decisions makers makes deterministic models and algorithms and closed-form results difficult to obtain; the interaction between (vehicle) operations and passenger actions and utility is not easy, nor described in full, especially for networks with very limited infrastructure capacity and relatively high vehicle capacity such as railways.

2.4. Solutions to Equilibrium Problems

Here we discuss possible ways by which a solution to the joint problem of train traffic control and passenger assignment can be determined. In general, we call an *MDM solution*, or a *solution* without further specification, a solution to a microscopic delay management problem (i.e., a set of variables that satisfies the constraints of the problem), combining a train scheduling solution with a coherent associated passenger assignment solution. The latter two, when needed, are identified as a *train scheduling solution* and a *passenger assignment solution*. The procedure that computes a solution is called an *algorithm*. For example, the *timetable MDM solution* is associated with a *timetable scheduling solution* computed by a *timetable scheduling algorithm*; a *Nash solution* is computed by a *Nash algorithm* (which is actually a game). This assumes that the rescheduling decisions $x = f\{y, \dots\}$ of the IM are taken in the interest of minimizing passenger delays, computed assuming some passenger loads on the running trains (among other input variables; we identify them as y). The route choice decisions $y = P\{x, \dots\}$ of the passengers are to minimize their travel times given some train services (among other input variables; we identify them as x). The two decisions are interacting, and we are interested in characterizing solutions corresponding to fixed points such that given a passenger assignment $P\{x\}$, the rescheduling solution $f\{P\{x\}\}$ is associated with a schedule equivalent to x , with all other input variables the same. The

relevance of such a fixed-point solution is described (for anticipatory road traffic control) in Taale (2008, p. 5) as “taking into account not only the current, but also the future traffic conditions.” In other words, if the users are better off changing their decisions, compared with what the traffic controllers determine, there will be a gap between system optimum and user optimum, and the performance of the transport system might degrade, with loss for both decisions makers. A fixed point is such that the behavior of both decisions makers is stable and therefore can be modeled properly.

Different approaches have been identified in the literature to study this fixed point, its computation, and its sensitivity to a variety of aspects. The direct solution of the equation $x = f\{P\{x\}\}$ is possible only in the cases where functions f and P are both invertible and well defined over a domain. In the general case, those functions include nonlinearities, exogenous factors, and intractable terms. The growing research community on schedule-based dynamic transit modeling (see Wilson and Nuzzolo 2004) considered an equilibrium point as a result of a learning process between the players, sometimes with multiple dynamics, within the day and day to day. Implicit models, also based on random utility theory, can deal with simplified network operations and information of users, whereas explicit models of information to users and technical constraints to vehicles or infrastructure are rarely considered. One explicit solution, which ignored both infrastructure capacity and vehicle capacity, is presented in Nielsen, Landex Rasmus, and Frederiksen (2009).

A numerical solution can be found by approximated approaches such as the method of successive averages (MSA). This allows for richer behavioral models of passenger assignment over networks, symbolized by diachronic graphs (Nielsen and Frederiksen 2006). A longer discussion of different methods used for assignment of passengers in the case of delayed railway operations with variable levels of information is presented in Nielsen, Landex Rasmus, and Frederiksen (2009). The focus of that paper is the offline problem of planning of an urban railway system.

The branch of optimization pertaining to mathematical problems with equilibrium constraints also studies fixed points by including equilibrium constraints explicitly in the optimization process (Patriksson 2008). Relevant examples of the application of this method to passenger assignment in road or transit networks mostly focus on pricing for road users, with a more limited application to schedule-based modes (also, for a pricing problem, see Hamdouch and Lawphongpanich (2010)). Its applicability and generalization are relevant open problems (Duc Quynh and Thuan 2018).

Moreover, in general, bilevel optimization has been used in a few cases to mathematically separate the objectives and constraints of two decisions makers.

In the context of public transport systems design, Mesbah et al. (2011) studied the problem of determining public transport lanes, with the higher level deciding on the size and use of transit lanes and the lower level resolving modal split, traffic assignment, and transit assignment. Dell’Olio, Moura, and Ibeas (2006) tackled the problem of locating bus stops and bus frequencies; again, the higher-level problem determines the system design, whereas the lower-level problem addresses user reactions. In the context of railway traffic control, Corman et al. (2014) determined a high-level problem of controlling flows over multiple traffic control areas and a lower-level problem of determining the precise traffic control actions in a local area. Wang et al. (2014) considered the higher-level problem as a stop-skipping problem and the lower-level problem as a passenger assignment problem. Overall, the degree of interconnection of the decisions of the stakeholders involved might determine an optimal solution (often related to as a system optimum) or suboptimal local minima (often related to user optimum).

Game theoretical settings have been used already in transport sciences for relevant control cases. One of the earlier works is the notion of anticipatory road traffic control put forward by Taale (2008). The idea is that a fixed point in the control space and in the decision space is a point that should be sought by controllers so that the reaction of users is completely incorporated (anticipated) in the control scheme. From a societal point of view, this includes implicitly the learning behavior of people who experience the situation over day-to-day dynamics (see Wilson and Nuzzolo 2004).

Based on this input, we also focus in what follows on game theoretical settings, especially given the added value that those settings can determine the influence of decisions and information on performance, stability, and convergence, for which a closed-form solution is not known and without resorting to approximation schemes such as MSA.

2.5. Game Theoretical Studies in Transit or Transport Networks

We report game theoretical studies in railway networks in the setting of two or more actors competing for getting some utility out of a limited resource. A similar structure but broader focus is used, for instance, in Zhang et al. (2010) for general transport networks. We also direct the interested reader to Hollander and Prashker (2006) and Easley and Kleinberg (2010).

Klabes (2010) discussed the railway capacity allocation problem, where actors are trains competing for capacity in the planning phase or in real time and have utility by running with the best path or the lowest delay. He demonstrated the existence of a Nash

equilibrium as outcome of the best response strategy. A more general setup, also discussed in Klabes (2010), is the one where actors are not trains per se but train operators, by which the economic appeal of the entire service intention is the goal of the game. For instance, combinatorial auctions have been used to competitively assign freight paths among train operators (Harrod 2013) in a planning phase. Game theoretical studies, mostly of cooperative game theory, and transferable utility assumptions, to study this allocation, can be found, for instance, in Bablinski (2015) and Kuo and Miller Hooks (2012).

A nontransferable utility game theoretical study is proposed by Fragnelli and Sanguineti (2014), who examine the need for information exchange for cooperative timetabling among multiple train operators. Companies exchange preferences about their needs and get compensation under limited information disclosure.

Over the long term, passengers repeatedly play a game and can actually change the services used through market-like competition mechanisms. Brewer and Plott (1996) studied how decentralized track allocation can be used when competition in the railway network increases. They studied the impact of decentralized allocation and deregulation. Lalive and Schmutzler (2008) studied the impact of competition of different operators for a service, which can be run by a single operator (i.e., after the competition has taken place). In this sense, it discusses how competition can actually increase service frequency and performance to users in a setup where competitive bidding has to take place.

Luan, Meng, and Corman (2017) discussed combinatorial models exploring Pareto optimality of the different available choices when assigning online capacity to train operators in the decision process of the IM. When controlling large networks under serious disturbances, in an online perspective, coordination between IMs of different geographic areas can also be seen as a form of game, for which combinatorial and decentralized algorithms have been reported (see Corman, D’Ariano, and Hansen 2014, Corman et al. 2014).

There exists a research gap and a practice gap about a full game theoretical study of the convergence and transferability of utility, as well as the precise setting against which this process is cooperative or competitive. Actors could be both multiple operators and multiple IMs. This is probably due to the formal obligations (partially cooperative, partially competitive) within the nondiscriminatory framework and the stress under which decisions must be taken (see, e.g., Ghaemi, Cats, and Goverde (2017) for a description of the decision process).

When actors are travelers competing for the limited capacity in a vehicle or even the seat in a vehicle, the

problem is typically known as *assignment*. This problem has been studied from a large range of different angles. Bouman et al. (2017) reported on minority games, focusing on crowding dynamics, in railway assignment under disturbances. They reported on a study clarifying the impact that the availability of information has on crowding, as well as the impact of the optimization from the train operator, which can change the capacity by using different rolling stock. Both actions have a direct impact on passenger satisfaction, here seen as the ultimate goal of the actors. Binder, Maknoon, and Bierlaire (2017) also studied the existence of a fixed point, similar to a Nash equilibrium in a repeated game setting, for capacitated railway assignment under disturbed conditions. The importance of compliance with information is becoming crucial, especially when the setting of a repeated game does not perfectly match the reality of specific disturbances happening only once.

Similar problems, but considering capacity of transit services, in general assignment problems are reported, among others, by Hamdouch and Lawphongpanich (2008), Hamdouch et al. (2011), and Poon, Wong, and Tong (2004). Overall, the problem is how to stochastically distribute utility among different travelers, who will partially face a denied boarding because of the limited capacity of transit vehicles. Extensions include static or dynamic setups, as well as schedule- or frequency-based setups Spiess and Florian (1989; Meschini, Gentile, and Papola 2007).

In the case of general road transport networks, the multiplicity of independent actors (the travelers) frequently suggested game theoretical studies. We refer the interested reader to the applications of game theory in transport networks in Zhang et al. (2010), as well as Hollander and Prashker (2006) and Easley and Kleinberg (2010). Chen and Ben-Akiva (1998) and Taale (2008) introduced an integrated framework to combine dynamic control and assignment. In their model, the actors are traffic users and a single traffic authority in a noncooperative setting. A series of different models, including Stackelberg and inverse Stackelberg models, is presented in Staňková (2009), with particular focus on road transport networks. Multiple game theoretical settings are discussed between a single traveler and a single authority and between multiple travelers and a single authority. The game proposed by Bell (2000) considers travelers as actors but includes an evil entity to assess reactions to unplanned disturbances and therefore assesses the reliability of the choices made by users under optimistic or pessimistic conditions.

In conclusion, there are many studies using game theory to report on particular evaluations of the utility or outcomes of a shared decision process. No model, algorithm, or formulation is currently known that considers the available capacity of the infrastructure,

the infrastructure managers who have to assign it, and the passengers as actors competing/collaborating for satisfactory transport performance.

3. Microscopic Delay Management Problem

We address the problem of computing in real time a microscopically feasible disposition schedule for some disturbed train traffic, minimizing delays, and passenger route advice for each passenger, minimizing their discomfort. In general, discomfort can be related to the time spent in various conditions. Similar to Binder, Chen, and Bierlaire (2014), in this paper, we directly adopt passenger travel time as a surrogate for passenger discomfort.

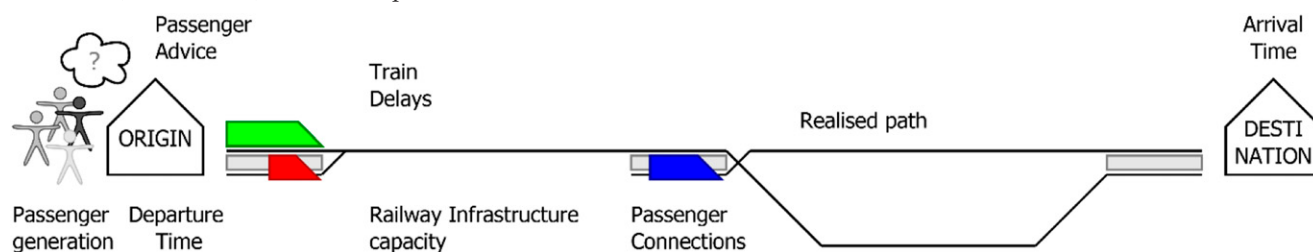
Figure 1 provides a graphical summary aimed at considering all aspects of the problem at hand. Passengers start from an *origin station* at a given departure time (which is the time they enter the station and not the time they board their first train) and want to reach a *destination station* as soon as possible.

We discretize the departure time of passengers and refer to a group of passengers departing at the same origin o at the same time w and having the same destination d as a demand odw . We call *passenger generation* the definition of all demands. Moreover, we assume that the trains have sufficient capacity to accommodate all passengers in each group for the only purpose of guaranteeing that all passengers in each demand can share the same route from origin to destination. Note that this assumption is not particularly restrictive because exceedingly large groups of passengers can be split into multiple demands. This has only to do with the possibility that any passenger groups can be transported in all possible vehicles connecting origin and destination and does not directly pertain to the competition effects of multiple demands odw (defining a set ODW) values on vehicles of limited capacity.

Passengers have available some information in the form of *route advice*. Basically, this is a strategy in terms of traveling along the network, involving a series of trains linking the departure station and the destination station through a series of feasible transfer connections. Typically, route advices are now available to passengers via online applications and websites or by personal construction of a plan when looking at printed timetables at stations.

In reality, this advice might not always be feasible, for instance, if information is not perfect or if some future event has not occurred yet. Based on their route advice, the passengers take actions, which determine a *realized path*, that is, the actual route choice for each demand from the associated origin to the associated destination station, possibly including transfers between pairs of connected trains at intermediate stations.

Figure 1. (Color online) Pictorial Representation of the Elements of the MDM Problem



Under these premises, three possible situations are given when relating route advice to a realized path.

3.1. Optimal Route Advice

The route advice corresponds to the best possible realized path; the information given to the passengers was enough to determine the optimal path.

3.2. Nonoptimal, Feasible Route Advice

The route advice corresponds to a feasible realized path but not to the best one; not having the right information prevents finding the optimal realized path. In other words, having full, perfect information would have routed the traveler to a different path with a shorter travel time.

3.3. Nonfeasible Route Advice

The route advice does not correspond to any feasible path (and therefore neither to the optimal one); not having the right information prevents finding a feasible realized path; that is, the passengers cannot reach their destinations if the route advice is followed precisely. This happens if the route advice specifies a transfer connection that is not kept in reality or if a train left too early for a passenger to take it. In the latter case, new feasible route advice must be issued to the passengers or the passengers cannot reach their destinations.

A *passenger connection* is the transfer of a demand from a feeder train to a connected train at an intermediate station along the demand route (either route advice or a realized path). The train rescheduling decisions determine the possibility to activate a passenger connection. In the given train schedule, the arrival time of the feeder train must be sufficiently in advance of the departure time of the connected train. Passenger connections between trains can be used whenever convenient to reach the destination.

In the presence of traffic disturbances, train rescheduling decisions are necessary to recover feasibility of operations and keep the fluidity of running. The plan of arrivals/departures described by the timetable is adjusted during operations while respecting the constraints on the limited capacity of the railway infrastructure.

The latter constraints limit the possibility of rescheduling train movements because the railway safety regulations must be respected while considering the signaling system, the speed of each train, and its relative position with respect to the other trains in the network. According to the commonly used fixed-block signaling system (Hansen and Pachl, 2014), the railway network is partitioned in block sections separated by signals. When a block section is occupied by a train, the signaling system forces other incoming trains to stop before the signal. In fact, each block section can host at most one train at a time. The precise modeling and consideration of the limited infrastructure capacity and its interplay in the decisions of the passengers are key contributions of this paper to the state of the art of delay management (usually neglecting microscopic detail) and train rescheduling (usually neglecting passenger decisions and consequent routing).

A train cannot depart from a station before its scheduled departure time, and its departure from a station can be constrained to be sufficiently larger than the arrival of another (feeder) train so that the passengers can transfer from the latter to the former. Moreover, a train is late when its arrival time at a station is larger than the scheduled arrival time. The delay of a train can be caused either by external disturbances related to that specific train or by the propagation of delays from other trains because of the operational constraints of the railway system. Specifically, *initial (entrance) delays* are due to disturbances that can be recovered only to a certain extent by exploiting running-time supplements of the timetable. *Consecutive delays* are determined by rescheduling decisions in response to initial delays and are the result of a train delay propagation resulting from some conflicting situations. In this sense, they are consecutive because they derive from some initial delays by means of interactions of trains in the limited railway infrastructure capacity. A *conflict* occurs whenever two trains require the same block section at the same time. Such a conflict is typically solved by adjusting departure times from stations and passing times along the network (retiming) or by specifying a passing order for the trains at the block section (reordering). We do not consider rerouting actions at

the current stage and leave this extension for future research. Determination of the retiming and reordering decisions to reduce delays (possibly weighted by the importance of trains or the number of people onboard to determine passenger delays) is the objective of the IM. More detailed descriptions of the problem, including formal definitions and formulations, can be found in D'Ariano, Pacciarelli, and Pranzo (2007) and D'Ariano (2007). The actions of the IM relate to train order and cannot include the decisions of passengers about their routes; they can only anticipate those later, given some information about their choice models.

In the computational experiments, we deal with timetable disturbances, that is, with entrance delays that can be managed without the need for train cancelations. As is common in the literature, we assume that those entrance delays are known at the beginning of the optimization (possibly resulting from a prediction model; see, e.g., Corman and Kocman (2018)). Microscopic train rescheduling models and algorithms dealing with traffic disruptions, arising in the case of huge train delays and/or network failures, are addressed in Corman, D'Ariano, and Hansen (2014) and Samà et al. (2016).

4. Methodology

This section briefly introduces the mathematical models of the two subproblems related to the MDM problem with passenger route advice: a train rescheduling model and a passenger assignment model. Then an example is proposed to illustrate and comment on the preceding optimization models for different passenger route advice. We present the algorithmic framework and some approaches proposed to determine a solution to the MDM problem in the case of strategic interactions between the key players. We finally examine the existence of equilibrium points under some general considerations.

4.1. Train Rescheduling Model

This section recalls the train rescheduling model of D'Ariano, Pacciarelli, and Pranzo (2007). The problem of controlling railway traffic with microscopic detail corresponds to a job-shop scheduling problem with blocking no-swap constraints (Mascis and Pacciarelli 2002). The blocking no-swap constraints model is the so-called fixed-block railway regulation that a train on a given block section cannot move forward if the block section ahead is not available or if it is occupied by another train.

This working of the safety system forces the occupation of a single block section by at most one train at a time. Consequently, a train on a previous block section cannot move forward if the block section ahead is occupied by another train. Detailed information on the modeling of the occupation time of

block sections can be found in Hansen and Pachl (2014). Normally, train operations are planned according to a timetable specifying passing times for all trains at relevant points. In the presence of delays, the goal of a train rescheduling model is to compute new times of operation.

We next recall very briefly the alternative graph model (Mascis and Pacciarelli 2002), which is the basis of the train rescheduling model of D'Ariano, Pacciarelli, and Pranzo (2007). An alternative graph G is a triple (N, F, A) . Nodes in N correspond to operations, each associated with the occupation of a block section by a train, representing either the traversing of a block section or the dwell in a station where a train has a planned stop. For each operation i belonging to N , a continuous variable h_i is associated with its starting time. A dummy operation 0 with $h_0 = 0$ is used to represent a common reference point in time for all operations. The graph is built for operations in a scheduling horizon in the future, starting from reference time t_0 , corresponding to now; for operations in the past before t_0 and for starting operations of trains in the area under control, their times can be known with certainty, and they are typically equal to the times planned in the timetable $h_{i, \text{timetable}}$ plus a possible entrance delay d_i . For future operations beyond t_0 , the times are optimization variables.

Arcs in the set F model time relations between the starting times of some pairs of operations. For example, if i and j are associated with the traversing of two consecutive block sections by the same train, then the directed arc (i, j) models the fact that h_j must be greater than or equal to h_i plus the minimum running time of the train on the block section associated with i . In this case, the minimum running time p_{ij} is a weight on the arc (i, j) . Fixed arcs can also model a minimum departure time of a train from a block section, which in this case is modeled with an arc $(0, j)$ having weight p_{0j} equal to the minimum departure time. Other examples will be discussed later in this section. In all cases, however, an arc (i, j) corresponds to the constraint $h_j \geq h_i + p_{ij}$.

Arcs in the set A model potential conflicts between trains on shared resources. Whenever two trains claim the same infrastructure element (block section, platform, etc.) at the same time, a conflict arises, and a decision on the order of the two trains on the infrastructure element must be then taken to resolve the conflict. To take into account signal status, a minimum time separation between the starting times of the conflicting operations is needed to ensure that the second train enters the infrastructure element after the first train has left it (i.e., the first train is occupying the next infrastructure element). Formally, let i and j be consecutive operations of a train, let k and l be consecutive operations of the other train, and let i and

k be the two conflicting operations associated with the entrance of the two trains in the same block section. Then the ordering decision is modeled with a pair of alternative arcs $((j, k), (l, i))$ representing the alternative constraints $h_k \geq h_j + s_{jk}$ and $h_i \geq h_l + s_{li}$, where s_{jk} is the minimum time separation when i precedes k and l , and s_{li} is the minimum time separation when k precedes i and j . The set A thus contains a pair of alternative arcs for each pair of potentially conflicting operations. Finding a feasible scheduling solution for a given alternative graph G consists of selecting one arc for each pair in A (which corresponds to defining an order to each pair of conflicting operations), thus obtaining a set S of selected arcs from A ; this defines all train orders. The selection S is feasible if and only if the graph $(N, F \cup S)$ representing the railway operations, their time relations, and the choices of train orders contains no positive length cycles. A decision on the arcs S determines the times h_i of operations based on a longest path computation. In other words, knowing the current and past states (i.e., the times at which current and past operations have happened plus the expected entrance delays), the future states are a function only of the ordering decisions.

We discuss two relevant scheduling solutions for this model. The timetable order $S_{\text{timetable}}$ is the optimal decision, assuming that delays are null. The same order can still be applied in the case of delays; trains will suffer some delays, whereas the general structure of precedence of operations will be preserved. Another scheduling solution is the one found by an algorithm that minimizes some performance function of the times h_i , that is, minimizing some delay function. This second solution can be expressed as a function $f\{x\}$ of some input x , which includes entrance delays, operator rules, and possible passenger flows. More details on job-shop scheduling-based models and optimization algorithms for railway traffic can be found in D'Ariano, Pacciarelli, and Pranzo (2007), Corman et al. (2011), and Lamorgese et al. (2018).

4.2. Passenger Assignment Model

The passenger routing problem (or passenger assignment problem) studies the distribution of passengers on the railway network. We consider a time discrete model for passenger arrivals at each station. Hence, we assume that we know the number of passengers willing to reach the same destination d (out of a set D) from the same origin o (out of a set O) starting their journey at the same time w for a discrete set of arrival times W . The discrete model is justified by the observation that all passengers with the same destination arriving at a station between two consecutive train departures will move together in the network as a group, because each passenger aims at

reaching his or her destination in the minimum time. In this paper, we do not consider explicit vehicle capacity restrictions; however, in theory, an appropriate choice of group sizing can allow an explicit modeling of vehicle capacity restrictions at the expense of a larger set ODW . In this paper, we refer to a group of passengers going from o to d and arriving in o at time w as a triple odw , hereinafter denoted as *demand*, and let ODW be the set of all demands odw . The passenger routing aspect of the MDM problem is a multicommodity flow problem on a graph derived from $(N, F \cup S)$, in which a commodity is associated with each odw triple.

Note that once the train schedule is fixed, assuming infinite vehicle capacity, each demand odw moves in the network independently of the other triples; that is, the choice of a particular routing for a given odw does not influence the routing of any other odw . Moreover, we assume that all passengers in a demand odw will follow the same origin–destination (OD) path (we assume this path to be unique, possibly breaking ties arbitrarily). The only difference is that passengers may change trains only at scheduled stops if a connection exists, that is, only if the connected train departs from the station sufficiently later that the arrival of the passengers. To take into account this difference, we introduce a set of active connection arcs C . Each active connection is associated with a pair (i, j) of operations, where i is the operation associated with the arrival of the feeder train at the station, and j is the operation associated with the departure of the connecting train from the station. Each arc in C has a weight c_{ij} equal to the minimum time for transferring passengers from the feeder train to the second train. Of all possible arcs in (i, j) , the active connections are always associated with the arcs for which $h_j \geq h_i + c_{ij}$ holds. A suitable choice of times allows for a connection to become active, that is, usable for passengers, or not. The active arcs in C are introduced in the graph only if a passenger transfer connection exists, that is, only if the passengers arrive at a station with a train and the connecting train departs from the same station sufficiently later than the arrival of the passengers plus the time required for their transfer.

We would thus refer to a train scheduling solution as the set of times h_i associated with nodes N in a graph of structure $(N, F \cup S \cup C)$. Therefore, the passenger routing aspect of the MDM problem is modeled as a multicommodity flow problem where passengers may flow only through the arcs of C and through the active arcs of C . In the case of infinite vehicle capacity, this simplifies into a series of shortest paths. In general, we can consider the passenger assignment as a function $P\{x\}$, which determines passenger

flows, based on some input x , which includes delays and a rescheduling solution.

Here we make the distinction between *route advice* and a *realized path*. The former corresponds to a particular sequence of services connecting origin to destination, which can be suggested, memorized, and possibly followed if all services and connections are kept (in other words, it refers to one feasible solution to a timetable information problem). A realized path represents instead one sequence of services that, in the particular instance chosen (i.e., depending on possible delays and changes), allows the destination to be reached from the origin and has been chosen by a passenger for this purpose. In short, routing advice represents a potential path, and a realized path is a feasible path in the time–distance graph connecting origin and destination, ultimately chosen by a passenger.

A typical assumption in the literature is to consider full rationality of passengers and perfect information, which (together with the assumption of infinite capacity) results in a complete overlap of the two concepts of route advice and realized path; that is, nothing prevents a passenger from actually knowing and choosing the route advice. Normally, the route advice corresponds to the shortest path for each demand, that is, the one minimizing the travel time.

We keep the assumption of rationality, but we drop the assumption of perfect information, considering various levels of passenger information based on different route advice, which translates into different solutions of the multicommodity flow problem/shortest-path problem (we keep the more general reference to the former expression in what follows). We identify the perception of the passengers by adding a prime to the name of the relevant variables; for instance, passengers assume that the time at which operation i occurs is not h_i but h'_i .

Under the assumption that the objective of the passengers is in any case to reach their destinations in the smallest travel time, routing advice corresponds to the optimal solution of the multicommodity flow on a graph $(N', F' \cup S' \cup C')$, which represents the information of the times of operations h'_i available to the passengers at time t_0 . Assuming THAT passengers are rational, their realized path will also be a solution TO the multicommodity flow problem but on a possibly different graph $(N, F \cup S \cup C)$, which represents the realized train paths, that is, the actual traffic.

The possible difference between the two graphs resides in the knowledge of the passengers in current and future operations. Given a start time of optimization t_0 , we can divide N into two disjoint sets N_{past} and N_{future} (in other words, $N = N_{past} \cup N_{future}$). In this distinction, nodes in N_{past} are associated with events with $h_i \leq t_0$, that is, at the current time (i.e., happening right now) or in the past (i.e., already happened). We

also include the entrance delays of trains in this set. Those times are not decision variables. Depending on the availability of the information, it is possible that the time known to passengers (h'_i) is exactly the time h_i at which the events actually happened, which is equal to a planned time $h_{i, \text{timetable}}$ plus possibly an entrance delay d_i .

The times at which nodes in N_{future} ($h_i > t_0$) will happen are instead dependent on some uncertainty dynamics (which we now neglect for the sake of clarity) and on the decisions taken by the actors. In fact, the time associated with any node in the future depends on the dispatching decision by some complex function

$$(h_i | h_i > t_0) = f\{(h_i | h_i \leq t_0), (N, F \cup S \cup C)\}. \quad (1)$$

The precise description of this function is related to longest paths in a graph, which has the structure given by the nodes and arcs in $(N, F \cup S \cup C)$ and based on the times of events $(h_i | h_i \leq t_0)$. We assume that all operations planned are still occurring, albeit possibly delayed (i.e., no cancellation); thus, the set N is the same for all cases. The graph gives a structure of the operations, with times and orders being a decision variable. More details are available in Mascis and Pacciarelli (2002).

To simplify the description, we especially focus on the two relevant classes of those graphs identified earlier, namely one where $F \cup S \cup C = F_{\text{timetable}} \cup S_{\text{timetable}} \cup C_{\text{timetable}}$ and the one where $F \cup S \cup C = F^* \cup S^* \cup C^*$ is the result of an optimization problem. We further distinguish the cases in which the delay values associated with $(h_i | h_i \leq t_0)$ are known to the passengers or not. This generates four possible combinations, namely the MDM solutions associated with the timetable solutions knowing the delay or not and the optimized solution knowing the delay or not. Because the optimized MDM solution assuming no delay corresponds to the timetable assuming no delay, three classes remain. Thus, in this paper, three levels of possible information available to the passengers are considered.

4.2.1. FULL INFO. This is an optimized MDM solution based on full perfect online information: $h'_i = h_i$, $(N', F' \cup S' \cup C') = (N, F^* \cup S^* \cup C^*)$. The passengers know exactly both the past and current events and the realized scheduling solution, which allows them to also compute the precise timing of future operations. In this case, the information available to determine the route advice is perfect and complete, and the route advice and realized path will coincide. If instead the information is partially wrong, incomplete, or unavailable, there will be in general a difference between the route advice and the realized path, as in the following cases.

4.2.2. First Scheduled, First Served (FSFS). This is an adjustment of the timetable scheduling solution based on the passengers having full information on entrance delay and times associated with N_{past} but not on train rescheduling decisions. Concerning the latter, passengers try to reconstruct them, based on the assumptions that no dispatching actions are taken, and the order of trains is as in the timetable. Formally, timing information for nodes $(h_i | h_i \leq t_0)$ in N_{past} is known perfectly; that is, $(h'_i | h_i \leq t_0) = (h_i | h_i \leq t_0)$ and includes the times at which past events are planned to happen $h_{i, \text{timetable}}$ plus the delays d_i . Moreover, the passenger perception is that $(N', F' \cup S' \cup C') = (N, F_{\text{timetable}} \cup S_{\text{timetable}} \cup C_{\text{timetable}})$; that is, passengers assume that the dispatchers apply the timetable order to the delayed operations. Based on this assumption, passengers can compute the times $(h'_i | h_i \geq t_0)$ they assume for events in N_{future} based on the graph structure determined by $(N, F_{\text{timetable}} \cup S_{\text{timetable}} \cup C_{\text{timetable}})$, that is, assuming that the scheduled plan is kept in the order the trains are served and all operations planned are still occurring, albeit possibly delayed (i.e., no cancellation). In fact, passengers can extrapolate the departure/arrival times of trains in the future and make their possible route advice based on that information. In any case, they have no information on what the dispatchers might actually decide; they therefore resort to a simple assumption about the train rescheduling solution, which corresponds to operations only in the case of no delays. In the absence of delays, *FSFS* corresponds to *FULL INFO*.

4.2.3. UNDELAYED. This is an adjustment of the timetable scheduling solution based on the passengers having no information on either entrance delay and times associated with N_{past} or train rescheduling decisions. Concerning the former, passengers assume that all operations occur at their planned time, ignoring the delays d_i : $h'_i = h_{i, \text{timetable}}$. Concerning the latter, passengers try to reconstruct them based on the assumptions that no dispatching actions are taken and the order of trains is as in the timetable; that is, $(N', F' \cup S' \cup C') = (N, F_{\text{timetable}} \cup S_{\text{timetable}} \cup C_{\text{timetable}})$. This represents the fact that passengers assume that the dispatchers apply the timetable order to the operations. Based on this assumption, passengers assume that all future operations are happening according to the plan, which corresponds to reality only in case of no delays. In the absence of delays, *UNDELAYED* corresponds to *FULL INFO*.

For the last two levels of information, there is a chance that the route advice does not correspond to reality, namely when there are delays, and connections might not be kept. Formally, if the delay of the feeder service minus the delay of the connecting service is larger than the connection time c_{ij} , the

connection is inactive (dropped), and passengers cannot benefit from it. It therefore makes sense to consider a more conservative value of the minimum connection time c_{ij} , which allows the route advice to still be valid in the case of delays. This is a well-known strategy to ensure robustness of the route advice at the cost of a longer travel time. This determines two configurations $\{FSFS, UNDELAYED\}$ with *LONG CONNECTIONS*: the passengers get information based on (respectively) *FSFS* or *UNDELAYED*, but the set C' is restricted to connections that are larger than a specified value, that is, including only the longer active connections in $C_{\text{timetable}}$. We consider, in such a case, a specified minimum value for connection equal to 600 seconds (i.e., only connections that are longer than 600 seconds, instead of the standard 180 seconds, are considered) and for both *FSFS* and *UNDELAYED*.

The combination of the three levels of information (*FULL INFO*, *FSFS*, *UNDELAYED*) plus the choice of standard or long connections for *FSFS* and *UNDELAYED* determines a total of five kinds of routing advice considered.

4.3. Illustrative Example

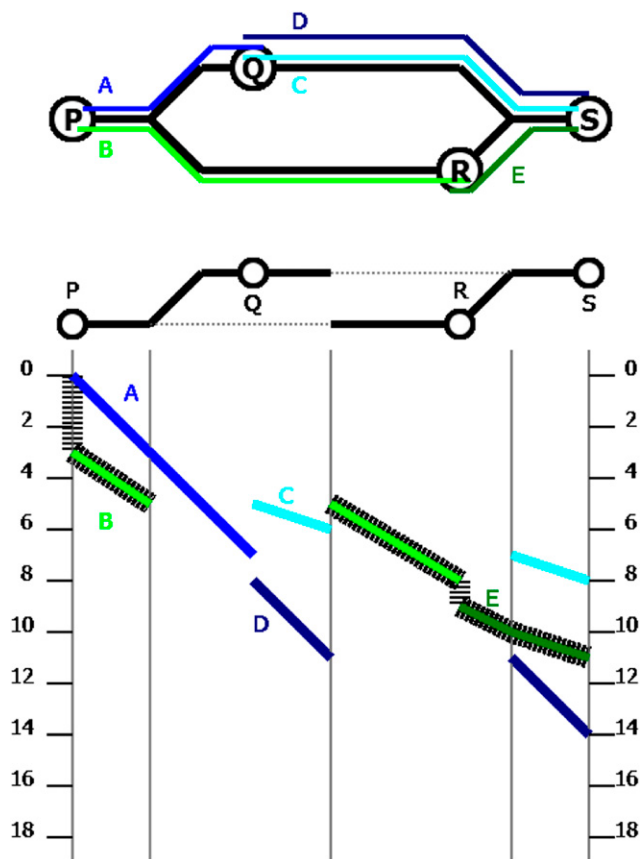
As an illustrative example, let us consider the following case, depicted in Figure 2. A small railway network is shown in Figure 2 (top); four stations exist in the network, namely P (left), Q and R, and S (right). Let us assume that all passengers want to go from P to S, and all depart at time zero. There is a series of services available, namely train A leaving P and ending in Q and train B leaving P and ending in R.

From those respective stations, there are services C and D connecting Q to S and service E connecting R to S. Thus, passengers must change in any case, either at Q along the upper branch or at R along the lower branch. All passengers must change exactly once. We report the timetable plan in Figure 2 (bottom), with space on the x -axis and time downward on the y -axis. We use the notation *station1*–(*train1*)–*station2*–(*train2*)–...–*stationN* to report a path in the network going over a sequence of trains and stations, which could be either route advice or a realized path.

In Figure 2 and the following figures, route advice or a realized path is represented as a larger dotted line along the services taken. For instance, for the timetable case, the advised route is P–(B)–R–(E)–S for a travel time of 11 time units. One can see that, in particular, the route advice P–(A)–D is very slow in the timetable, and train C leaves just before train A arrives at station Q, so passengers cannot make the connection.

Let us first consider the situation in which the traffic is delayed. In particular, the following delays are given: train B has two units of delay (departing at five instead of three); train C has four units of delay (departing at nine instead of five); train E has two

Figure 2. (Color online) Illustrative Example: Infrastructure (Top) and Timetable (Bottom)



units of delay (departing at eleven instead of seven). The question is: what are train traffic controllers going to do, and what are passengers going to do?

We first consider the routing advice related to no online information (*UNDELAYED*), reported in Figure 3 (left). With this route advice, the passengers keep following the path planned in the timetable, namely to take the trains in sequence: P–(B)–R–(E)–S. In this particular case, one can see that this path faces much delay, arriving at the destination only at time 17.

Let us now assume that the passengers would have access to full information (*FULL INFO*), reported in Figure 3 (right), and know both the delays and the schedule that will be followed. Their route advice would then differ from the original and would take this sequence of trains: P–(A)–Q–(D)–S, which arrives at the destination at time 14. In fact, this is the fastest way to reach the destination, given the schedule.

We next discuss the intermediate case in which some information is available to passengers. In particular, we refer to the *FSFS* case, in which passengers have an idea of the delay (because this is reported by some information source) but have no idea of the train rescheduling actions that will be implemented in the future and believe that the order specified in the timetable will be kept.

In this case, we discuss this case with the help of Figure 4. In Figure 4 (left), we report the information that is available to passengers, namely how they believe that the traffic will behave. We remark here that the information available for the *UNDELAYED* case corresponds exactly to the planned timetable; the *FULL INFO* corresponds exactly to reality, and thus we do not report it explicitly.

Given the delays and the timetable order, passengers would believe that the fast train C will be scheduled before the slow train D, as prescribed in the timetable. In fact, the decision to change order between those trains is totally left to the dispatchers, and passengers might not be aware of which decision will actually be taken in real life. Given this belief, the route advice that allows reaching the destination in the shortest travel time would be P–(A)–Q–(C)–S, which would arrive at the same time as in the timetable, namely 11. In fact, the delayed train C is now available for use by passengers, and the fast travel time of train C is able to make up for its delay.

Unfortunately for the passengers, the dispatchers choose to perform some rescheduling action and change the timetable order between C and D and let D go first. This translates to the situation in Figure 4 (right). Following their route advice, passengers at station Q board train C; in reality, train D would have been a better choice. This translates to an arrival time of 15, which is still faster than considering no information (case *UNDELAYED*).

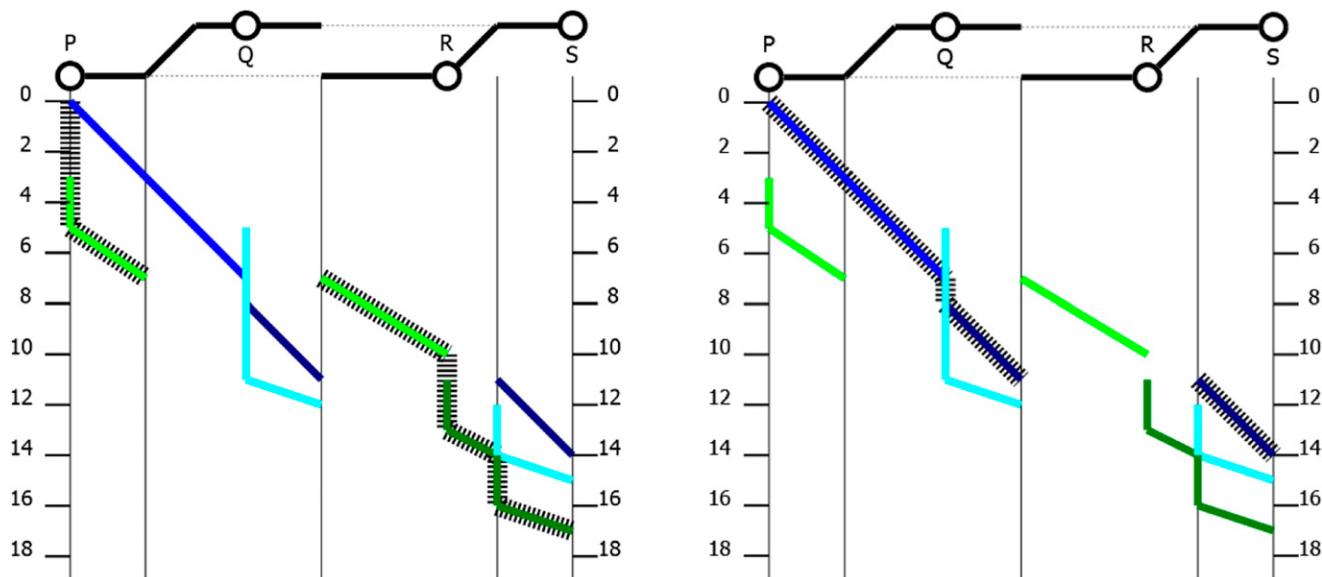
In both *UNDELAYED* and *FSFS*, the routing advice was not optimal but was still feasible; passengers could follow the advice to reach the destination. We comment briefly that a different choice of origin and destination might result in routing advice that turns out to be infeasible. In fact, if passengers need to go from train C to train D at station Q, this is impossible because the latter train leaves before the former train arrives. In these latter cases, different routing advice must be provided for those passengers.

4.4. Game Theoretical Setting

We next present a way to determine some possible interesting MDM solutions to the problem under study. We consider a game theoretical setting in which the decisions of the two players are successive and reactive on each other's decisions. In general, a game is a set of decisions by a set of players. In our case, multiple players exist, namely an IM player and a multitude of passenger players. Different from most algorithmic frameworks, in our case, their interests align, but only partially, and their decisions might change the objective/payoff of the other players.

The decision of a player is the set of all actions that a player will take while dealing with the problem at hand. In our case, the possible actions of a traveler are

Figure 3. (Color online) Solution of the Undelayed Routing (Left) and the *FULL INFO* Advice (Right)



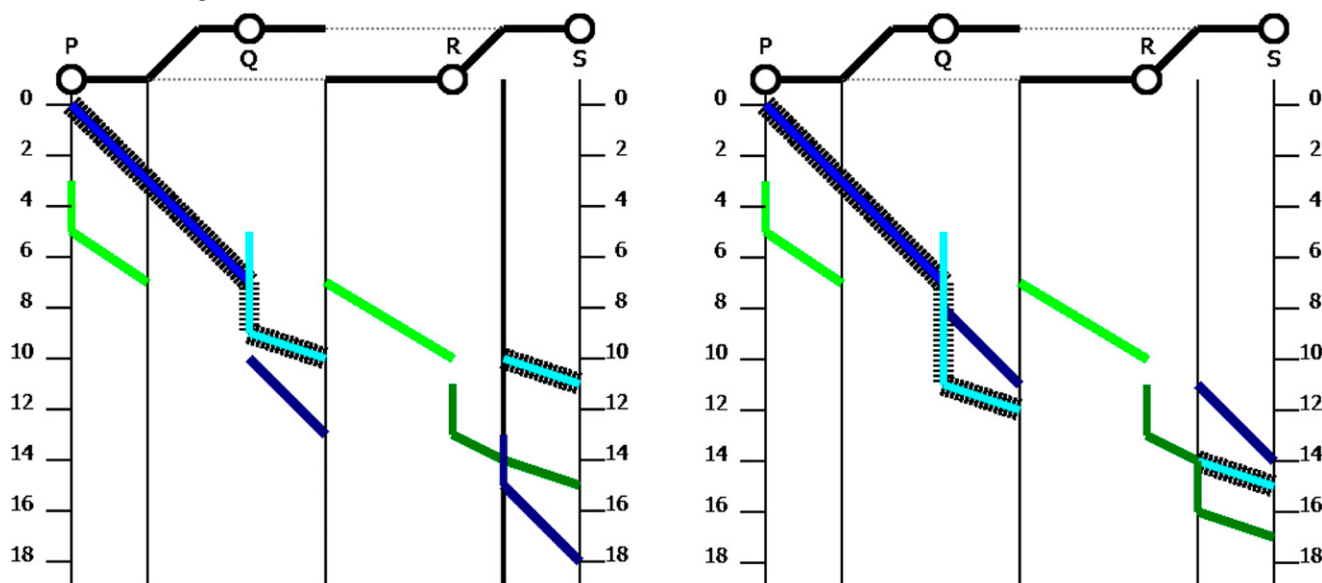
to wait at their origin station, board some trains, possibly get off somewhere, wait and board a new train, and possibly many times, and finally get off at the destination station. The objective of each traveler is always to minimize his or her travel time, that is, the total time spent in those actions. In our setup, passengers might not have complete information; thus, they might take decisions under incomplete or incorrect information. The function representing the route choices of the passengers, that is, their assignment, has been previously described by $P\{\cdot\}$.

The possible actions of an IM player are to change times and orders of trains, that is, the function $f\{\cdot\}$

introduced previously. Those actions can be changed to a fixed value or can be changed to a functional form; that is, the times and orders of trains would be such that a specific function is met. In our setup, IMs have complete information on their actions and also have complete information on the passenger choices, especially in terms of OD pairs, available information, and objective function.

Formally (see Staňková (2009) for a larger reference), we call D_i the set of all possible decisions of player i , where i can be, for instance, $\{IM, passenger_1, passenger_2, \dots, passenger_n\}$. The set of taken decisions is u_i . The combination $u = \{u_{IM}, u_1, u_2, \dots, u_n\}$ is a

Figure 4. (Color online) Passengers' Belief for Determining the Route Advice under the *FSFS* Policy (Left) and Corresponding *FSFS* Solution (Right)



decision profile that we associate with a solution to the MDM problem.

We call $J_i = J_i(u)$ the *objective function* of player i . Given a decision profile, it is possible to compute the objective function of all players involved. The objective function is individual and depends on the decision profile chosen by the same player and by the other players. We assume that a smaller objective function is preferred by each player (minimization).

A Nash equilibrium is a MDM solution $u^* = \{u_{IM}^*, u_1^*, u_2^*, \dots, u_n^*\}$ such that no unilateral deviation from it would result in an improvement in the objective function for any player; formally, $J_i(u_{IM}^*, u_1^*, \dots, u_{i-1}^*, u_i^*, u_{i+1}^*, \dots, u_n^*) \leq J_i(u_{IM}^*, u_1^*, \dots, u_{i-1}^*, u_i, u_{i+1}^*, \dots, u_n^*)$ for any $u_i \neq u_i^*$.

A Stackelberg game identifies explicitly two roles: a leader L and some followers F . It is natural to identify the leader in our case as the IM ($L = IM$) and the followers as the passengers $F = \{1, 2, \dots, n\}$.

In a multifollower Stackelberg setting, the leader announces the decision u_{IM} to the followers, which can then react choosing their own decisions, that is, $u_i^* = \operatorname{argmin} J_i(u_{IM}, u_1, u_2, \dots, u_{i-1}, u_i^*, u_{i+1}, \dots, u_N)$ or $\operatorname{argmin} J_{Fi}(u_{IM}, u_i^*)$ in the case that the objective function related to a decision is independent of the decisions of other followers (which we actually assume in our setting). The decision u_i^* could be expressed as a reaction function $l(\cdot)$ applied to the choice of the leader: $u_i^* = l(u_{IM})$. This can be a typical transit assignment problem.

In a multifollower inverse Stackelberg setting, the leader follows a strategy. A strategy χ_{IM} is a function (a specific case of the function $f\{\cdot\}$) able to determine a decision D_{IM} based on the set of all decisions from the followers. In general, this function might return a vector of values (i.e., a multidimensional decision involving multiple train times and train orders). The followers then minimize their objective function, conditional to the knowledge of this function. In other terms, $u_i^* = \operatorname{argmin} J_i(\chi_{IM}(u_1, u_2, \dots, u_N), u_1, u_2, \dots, u_N)$ for each follower i , given some decisions taken by the other followers. If the leader would desire some decision $u_F^* = \{u_1^*, u_2^*, \dots, u_N^*\}$ from the followers, a dominant strategy χ_{IM}^* is one that satisfies the following: $\operatorname{argmin} J_i(\chi_{IM}^*(u_1, u_2, \dots, u_N), u_1, u_2, \dots, u_N) = u^*$ for any arbitrary decisions from the followers $u_1, u_2, \dots, u_N$, for any follower i .

We assumed in the derivation of these formulas that the IM is the leader and the passengers are the followers, for the reason that the IM is a single role with a clear business objective, which does not pertain to directly controlling passengers flows. We therefore assume that the IM has a direct advantage of improving the utility of the passengers but cannot control their actions. In theory (and according to the strict

regulations; see the European Union (EU)–related directives, such as Single European Railway Directive 2012/24/EU), the utility of the IM does not relate to the passengers' utility. In practice, it does; if the IM makes decisions that are unnecessarily against the interests of the passengers and the TOC moving them, complaints and investigations might arise. We also assume that the passengers are many and do not have direct decision power over the IM and that the passengers do not directly influence others' decisions but do so only via a decision of the IM affecting them. This relates to the independence of travel and stopping (dwell) times on the number of passengers onboard, as well as the infinite capacity of the vehicles. Having finite capacity of the vehicles, a direct competition between travelers would arise, which would further complicate the scheme.

In theory, a Stackelberg setting in which the IM is a follower and the set of all passengers is the leader could be considered. In such a case, the passengers' objective function would be relatively easy to determine as minimization of total travel time, whereas the relation with the objective function of the IM would be relatively uneasy. In fact, assuming that vehicle capacity is infinite and that boarding and alighting processes do not depend on the number of people onboard or at the station, as well as assuming that train speed and acceleration rates do not change with the number of people inside, the objective of minimizing delays would then be completely independent of the decisions of the passengers. For this reason, we leave this direction for future studies.

In our setup, different various utility functions and reaction functions between the IM and passengers can still be considered. The objective function of the passengers and their decisions is discussed first. Every passenger decides his or her route advice based on the expected travel time. In other words, passengers would minimize their total travel time $\sum_{odw} n_{odw} (T_{odw} - \Pi_{odw})$ based on some information and route advice available. This problem has been already investigated in the literature as a transit assignment problem. The total travel time is defined as the number of passengers n_{odw} for each odw multiplied by the corresponding travel time. This latter is expressed as the arrival time at the destination T_{odw} minus the generation of the group of passengers at the origin station Π_{odw} . A passenger assignment solution is the sequence of trains to board and change for each passenger and, aggregating over all passengers, the number of passengers assigned per train between any two stations and sections. Related to this, one can compute the number of passengers disembarking the train at each stop, the number of passengers changing trains at each stop, and the number of passengers arriving at their final

destinations at each stop. These figures could be useful in determining the strategy of the IM (Figure 5).

The problem of the IM relates to a train rescheduling problem, which has been defined in the literature as the online updating of train paths by train retiming and reordering decisions with the minimization of a certain function of traffic performance related to delays. In the literature, no consensus is found on which exact delay expression can be used (see Cacchiani et al. (2014) and Samà et al. (2015)) when only the train-related operations are to be considered; enlarging the study to also include passenger decisions would not make the problem easier. We consider objective functions that are based on some form of a linear combination of a lot of factors $f_e z_e$, which refer to some events e related to a train arriving at a station or passing at relevant points in the network; f_e is the number of passengers related to event e , and z_e is the delay associated with event e compared with the published timetable. In other words, dispatchers are obliged to take decisions based on train flows and not directly on passenger desires.

We consider different ways (algorithms) to combine those $f_e z_e$ values as follows: a scheduling solution determined by an algorithm considering an objective function, where all $f_e = 1$, practically reflects the case where the IM does not directly care about the number of passengers on board the trains, nor about their travel times, but only about the delay of each train, further measured at all events e . If the event e is only one, the one associated with the maximum consecutive delay (i.e., the maximum delay associated with delay propagation resulting from conflicts for the limited infrastructure capacity), the objective function is then analogous to the model of D'Ariano, Pacciarelli, and Pranzo (2007). Otherwise, a total delay (which also includes the primary/entrance delay caused by some external disruptions and cannot be recovered by the running traffic) or average delay form can be determined. Because those decisions of the IM do not take into account the passengers, one can think of them as a Stackelberg game

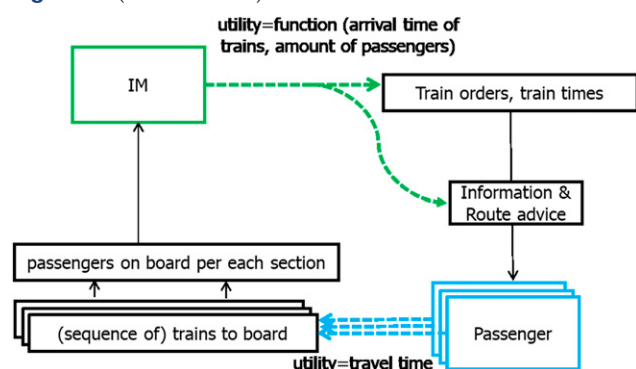
where the IMs publish their strategy, which depends on the delays of trains, and passengers take their decisions accordingly. Two cases (of solutions and algorithms) we consider in more detail are the following: the MDM solution associated with the scheduling solution computed by the algorithm minimizing the maximum consecutive delay, that is, the solution of D'Ariano, Pacciarelli, and Pranzo (2007), and the solution associated with the scheduling solution that keeps the timetable order, that is, a solution for which the train order is as in the timetable, and the train times are a direct consequence of it. The MDM solutions and algorithms show limited interaction between the players; in fact, one decision maker A takes a decision influencing the possible choices of the other decision maker B, but the specific choice of B does not feed back to a different decision of the former player A. In other words, there is no second-order effect. For this reason, the MDM solutions and algorithms also require limited communication between the players (i.e., only some information of one decision maker needs to be available to the other). In fact, if the IM decides to implement the train rescheduling solution based on factors other than passenger travel times, there is no need to change or update the scheduling solution for the passenger route advice and realized paths, and resulting passenger travel times are determined, and thus there is no need for the IM to know them.

4.4.1. Timetable. In this algorithm, the train schedule is simply obtained by keeping the same train sequence of the timetable and delaying each train by the minimum amount needed to achieve feasibility. Passengers then follow the path that would bring them as fast as possible to their respective destinations. This solution simulates the common practice of railway management in which the IM keeps the order of trains prescribed by the timetable while passengers react individually to delays by choosing the most convenient route in real time based on the information available to them.

4.4.2. Train-Based Rescheduling. This algorithm focuses on the problem of purely reducing train consecutive delays. We use, to this end, the optimization model introduced in D'Ariano, Pacciarelli, and Pranzo (2007); passenger travel time or passenger delay is neglected. Passengers follow the path that would bring them as fast as possible to their respective destinations given the train orders. Passengers would react individually to delays by choosing the most convenient route in real time based on the information available to them.

Other game theoretical solutions considering passenger flow are also studied (partially defined in Corman et al. 2016).

Figure 5. (Color online) Game Theoretical Interaction



4.4.3. Nash Integral (of Flows Onboard). This algorithm is able to determine an MDM solution that corresponds to a Nash equilibrium of a game seeking a compromise between passenger and IM objectives. In this game, the IM strategy is the minimization of a train total delay, where each train delay is weighted by the number of passengers on the train. Thus, the passengers are considered as a flow onboard the trains as the integral of passenger boarding and alighting along the travel route. The starting scheduling solution used to determine the equilibrium point is the one in which trains are rescheduled minimizing their delay regardless of passengers. Passengers react individually to the rescheduling actions by choosing the most convenient route in real time.

4.4.4. Nash Gradient (of Disembarking Flow). This algorithm is able to determine an MDM solution that corresponds to a Nash equilibrium of a slightly different game. In this game, the IM strategy also consists of the minimization of a weighted train total delay, but the weight of each train is equal to the number of passengers disembarking the train (similar to what was suggested by Schöbel (2009)). Thus, the passengers are considered to be the variation in flows entering and leaving the train along the travel route. The starting scheduling solution used to determine the equilibrium point is the one in which expected passenger flows are computed according to the timetable plan, that is, regardless of delays. The passengers move in the network as for Nash integral. In the case of perfect full information, the final solution corresponds numerically to the heuristic named H3 in Corman et al. (2016).

4.4.5. Nash Connect (Connections Optimization). This algorithm is able to determine an MDM solution that corresponds to a Nash equilibrium of a variant of the game determining the Nash-gradient solution. The differences reside in the selection of the active connections. This is based on the identification of promising connections to be enforced, that is, of the missed connections in which the arrival time of passengers is slightly later than the departure time of the connecting train. Connections are iteratively enforced as long as passenger travel time improves. In a sense, this mixes up the objective functions of the passengers and the IM. The IM and passenger strategies are the same as Nash gradient. The starting scheduling solution used to determine the equilibrium point is the one in which expected passenger flows are computed according to the timetable plan, that is, regardless of delays. In the case of perfect full information, the final solution corresponds numerically to the heuristic named H4 in Corman et al. (2016).

The MDM algorithms determine solutions that correspond to equilibrium points in Nash or Stackelberg games. The algorithms operate by starting from a

starting solution and alternating the train rescheduling phase, optimizing train orders and times for given passenger flows and network capacity, to the passenger assignment phase, in which passenger travel time is minimized and computed for the given train schedule based on some passenger routing strategy. For Nash games, the procedure iterates until convergence (i.e., until the current overall solution of train rescheduling and passenger assignment is equivalent to the one found in the preceding iteration) that would correspond to an equilibrium. We assume here that such an equilibrium can be found, even though in general this might not be the case (see Section 4.5). In case no equilibrium is found within a maximum number of iterations, the procedure is stopped, and the best solution is reported.

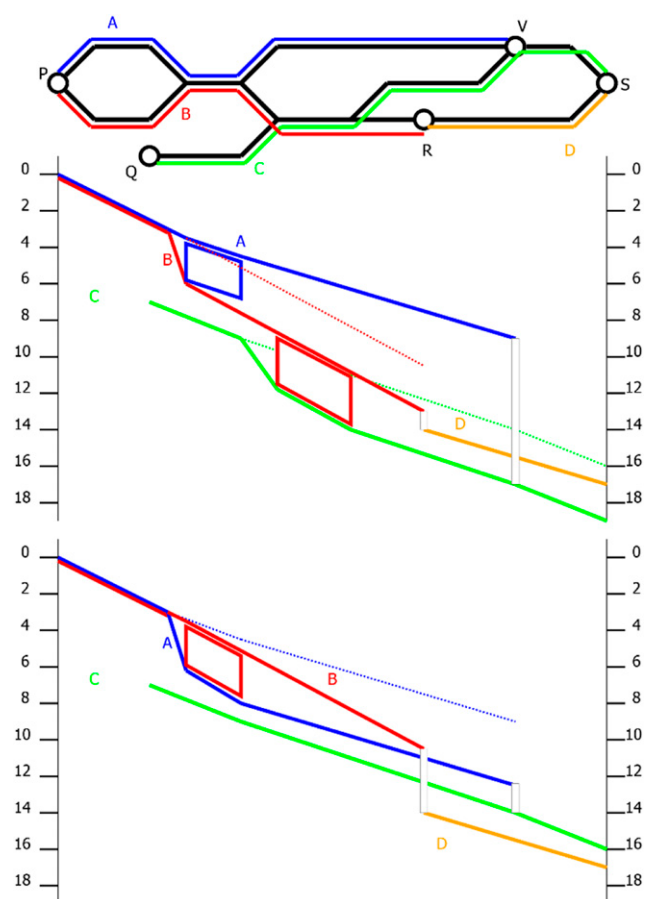
Note that this is a deterministic and static problem; that is, all actors are not expected to learn from the past or take different actions over time if they find themselves in the same situation again. This means that the compliance of actors is not explicitly considered (as suggested instead in Binder, Maknoon, and Bierlaire (2017)), that the conditions under which actors take their decisions do not change over time, and finally, that there is no dynamic learning process, as expressed, for instance, in the dynamic games of Staňková (2009) or day-to-day procedures for transit assignment (Wilson and Nuzzolo 2004).

4.5. Existence of an Equilibrium Point

We now discuss the possibility of having equilibrium under some restrictive conditions. In general, depending on the way the IM makes decisions based on the observed choices of passengers, some equilibrium point might exist or not. We show in this section that some reaction types might result in situations that do not converge to an equilibrium, and the game would keep oscillating. This is due to the discrete nature of the scheduling problem. Similar findings, although only in an empirical result of an algorithm and without a properly defined counterexample, have been reported in Binder, Maknoon, and Bierlaire (2017). Moreover, in general, the existence of an equilibrium point for continuous, convex, or noncontinuous, nonconvex domains and the potential to make games have an equilibrium point in general are discussed in Rosenthal (1973).

We refer to Figures 6 and 7 and Table 1, where we describe a delayed situation of train traffic running in a network. Four trains are running from left to right (labeled A, B, C, and D) across five stations (labeled P, Q, R, S, and V). Train A starts at P and ends at V; train B starts at P and ends in R; train C starts at Q and passes V before reaching S; and train D runs from R to V. Trains A and B share a part of the infrastructure, and so do trains B and C. We will see that these influences for the available capacity are the key mechanisms to inhibit reaching a fixed point. We assume that

Figure 6. (Color online) Nonconvergence to an Equilibrium Point



Note. (Top) Infrastructure and services; (middle) first case; and (bottom) second case.

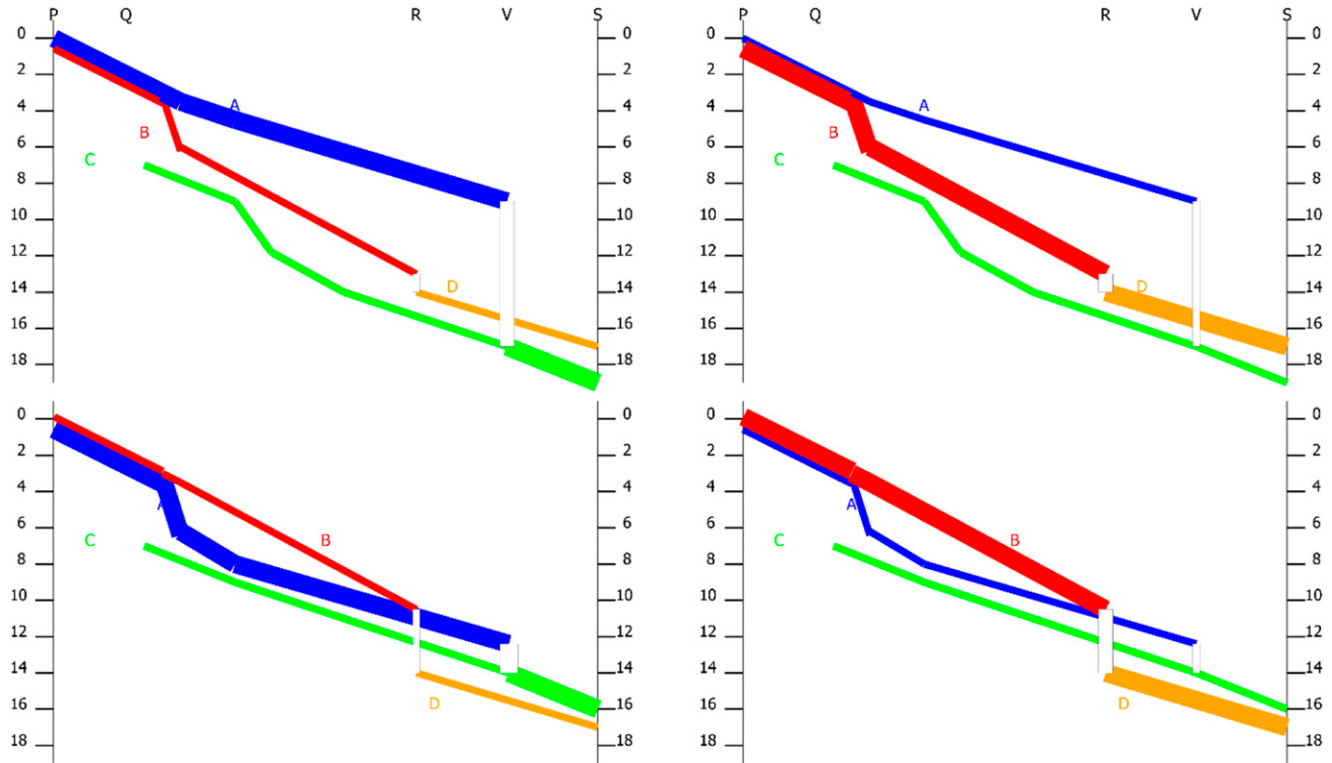
train A is delayed by an unspecified amount Y , and, as a consequence of this entrance delay, it departs at the same time as train B. Both trains arrive at the same time at the bottleneck section around time 4. This is the only entrance delay; thus, all other delays happening in the network are considered consecutive delays.

Figure 6 (middle and bottom) shows time–distance graphs. Time–distance graphs are a standard representation of movements of vehicles over time and space, especially used for railways, where the space coordinate is constrained to be on a line. Time (vertical, increasing downward) goes from 0 to 20, whereas distance (horizontal, matching the above-mentioned infrastructure) spans the four stations. Trains running on parallel stretches of the network (such as C and D) are reported on the same horizontal location, and conflicts, if they occur, are highlighted to keep the figure clearly arranged. We also report using the dotted line for the planned time–distance graph, that is, assuming no infrastructure conflict. We report using vertical white boxes the time spent at a station where people can change trains and continue. In particular, there is a

transfer between trains B and D at station R and a transfer time between trains A and C at station V.

In Figure 7, we report the passenger loadings for four situations, that is, the same time–distance graph as in Figure 6 but only focusing on where passengers are moving, symbolized by the thick lines. The top plots in Figure 7 represent two different passenger assignments on the top plot of Figure 6 and similarly for the bottom plots. The left (respectively, right) plot of Figure 7 represent two possible solutions of passenger assignment for each time–distance graph (train rescheduling solution). Table 1 represents, for the same four situations as in Figure 7, the (total and consecutive) delays of trains and the number of passengers on the train just before the stop (for Nash integral/Nash gradient). As for the delay, each train has only one stop, apart from train C, so five rows are reported. We also consider some initial delay Y for train A, which does not influence the reasoning. As for the passengers, we assume a (negligible) small amount ε_A , ε_B , ε_C , and ε_D for each train plus a single passenger (we would name him or her X) who is the focus of our study. One could think of this passenger as a larger passenger group and proportionally scale the ε conversely. The thick lines in Figure 7 report the passenger choice of passenger X. Trains running on parallel stretches of the network (such as C and D) are reported on the same horizontal location, and conflicts, if they occur, are highlighted to keep the figure visible.

Please note that the order between trains B and C is not relevant because train C cannot come before train B in any case. Also assume that no other train conflicts with no other train; that is, capacity at stations P, R, V, and S is enough to allow two trains to simultaneously stop. Two scheduling solutions are possible at the bottleneck. If train A is first, train B is held back and faces some delay, which we represent in the time–distance graph as a parallelogram (blue, stretching between times four and six; the delay corresponds to the vertical deviation between the dotted red line (planned) and the solid line (realized); i.e., the consecutive delay is 2.5 in this case), depicting the area in time where a train is blocking the infrastructure. For the sake of simplicity and clarity, we do not report formally blocking times or section boundaries, but all the reasoning applies. This delay results in a delay for train B, which is further propagated because train C goes after train B (another parallelogram, this time in red, between times 8 and 14; the delay corresponds to the vertical deviation between the dotted green line (planned) and the solid line (realized); i.e., the consecutive delay is three in this case). Overall, passenger X travels on train A until station V, where a transfer is possible to train C arriving at station P at time 19. This scheduling solution is reported in Figures 6 (middle) and 7 (top).

Figure 7. (Color online) Flows and the Two Possible Solutions

Otherwise, the opposite order is possible; namely train B goes first, and train A is delayed (red parallelogram; the delay corresponds to the vertical deviation between the dotted blue line (planned) and the solid one (realized), i.e., the consecutive delay is 3.5 in this case). This scheduling solution is reported in Figures 6 (bottom) and 7 (bottom). In this case, train C is not delayed any more by a delayed train B and thus arrives at its planned time, that is, 16, which is one time unit before train D. The summary of the (total/consecutive) delays of trains at the respective stations is reported in Table 1: For train C, we reported delays at both station V and at station S. The total delays can be computed by subtracting the planned arrival time from the actual arrival time. The consecutive delays would be equal to the total delays minus the entrance

delays; the latter exists only for train A and is of magnitude Y.

Let us look at how the passengers would react to these potential choices, starting from the former, that is, A goes first. This solution is the one minimizing the maximum consecutive delay, the average consecutive delay, and their weighted versions, given the passenger flows. We can also assume that delay Y is smaller than 2.5; this same solution also minimizes the maximum total delay and the average total delay. All trains have some passengers, which are assumed to be ϵ_A , ϵ_B , ϵ_C , and ϵ_D per train; these can be supposed, for instance, to be very small, and in any case, they are not influenced by the different train rescheduling actions. We assume that the origin and destination of each train correspond to those of the passengers

Table 1. Delays and Passenger Flows Corresponding to the Solutions of Figure 7

Train (station)	Total/ consecutive		Integral/gradient	Train (stat)	Total/cons		Integral/gradient
A	Y	0	$\epsilon_A + 1$	A	Y	0	ϵ_A
B	2.5	2.5	ϵ_B	B	2.5	2.5	$\epsilon_B + 1$
C (V)	3	3	ϵ_C	C (V)	3	3	ϵ_C
C (S)	3	3	$\epsilon_C + 1$	C (S)	3	3	ϵ_C
D	0	0	ϵ_D	D	0	0	$\epsilon_D + 1$
A	Y+3.5	3.5	$\epsilon_A + 1$	A	Y+3.5	3.5	ϵ_A
B	0	0	ϵ_B	B	0	0	$\epsilon_B + 1$
C (V)	0	0	ϵ_C	C (V)	0	0	ϵ_C
C (S)	0	0	$\epsilon_C + 1$	C (S)	0	0	ϵ_C
D	0	0	ϵ_D	D	0	0	$\epsilon_D + 1$

onboard; there are no two trains that are directly competing on the same OD pair. Moreover, because of infinite capacity, different passengers do not directly influence each other. The only relevant passenger is X. Confronted with the solution in Figure 6 (middle), the best choice for passenger X would be to board train B and then change to train D for an arrival time of 17 time units (Figure 7, top right). Instead, boarding train A would mean needing to change to delayed train C for an arrival time of 19 (Figure 7, top left).

Confronted by the solution in Figure 6 (bottom), that is, train B goes first, passenger X would instead board train A and then change to train C for an arrival time of 16 (Figure 7, bottom left). The opposite choice of the passenger, taking train B and then changing to train D, would have an arrival time at 17, which is one time unit later (Figure 7, bottom right).

In either case, the passengers would react to the choice of the dispatcher in a Nash fashion. We remark here that the three Nash setups would not differ: the (integral) number of people onboard the train or (gradient) disembarking (respectively, used by Nash integral and Nash gradient) are the same (in Table 1, the third column). Finally, Nash connect would focus on connections, which actually do not change between those two solutions; thus, no further action or difference would arise from Nash connect.

In fact, the optimal reaction of each player to the best move of the other player would result in not having a Nash equilibrium in pure strategies. This is summarized in the payoff matrix reported in Table 2, summarizing for the two possible actions of the passengers and the IM and the utility outcomes for the passengers and the IM. The utility for the reordering decisions of the IM is the passenger delay, which can be computed given the route choice of the passengers. The utility for the route choice decisions of the passengers is their travel time. Given the schedule of the train, passenger X can in fact use a different route that results in a shorter travel time. That is, passenger X can board train B, have a delay from train A, and change to train D at station R, which would arrive at 17, that is, two time units before the (delayed) train C. Thus, a different passenger assignment is considered, where passenger X takes trains B and D.

However, given this updated passenger assignment, it is better to invert the order of trains at the bottleneck. Given this schedule, passenger X then

changes route, taking trains A and C in succession, which is the starting solution reported in Figure 6 (middle) and discussed earlier.

We discuss a relaxed version of the problem where capacity constraints would not be enforced; that is, the travel time between two stations is a fixed number, independent of traffic. This is a common assumption in almost all studies in macroscopic delay management. In this case, the binary order variables are not used. Figure 8 reports a time–distance graph with both schedule and passenger flow. In fact, what is reported is the optimal solution for both train delays and passenger travel times; the variables associated with orders would in fact be integer and are actually numerically congruent with those of the solution in Figure 7 (top). In other words, solving the relaxed version of the problem, a coherent MDM solution exists, which is numerically equivalent to one that is found in the constrained version; however, this does not trigger any non-equilibrium behavior. This clarifies that it is not (just) the integrality required from the MDM solution to cause the switching behavior but actually the inclusion of capacity constraints.

5. Experimental Assessment

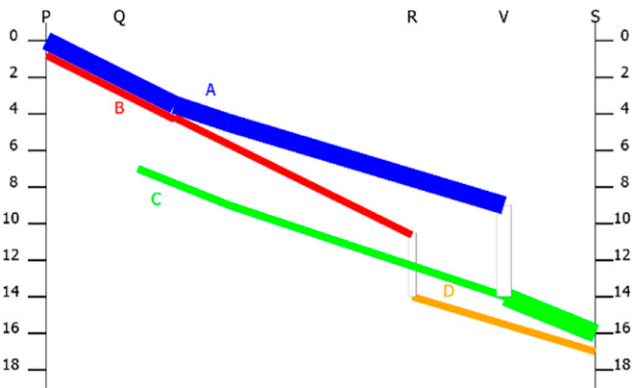
This section reports the results of a campaign of experiments on the performance evaluation of the five MDM algorithms described in Section 4 (Nash integral, Nash gradient, Nash connect, timetable, and rescheduling) in the case of five possible different kinds of route advice for passengers (UNDELAYED, FSFS, FULL INFO, UNDELAYED with long connections of 600 seconds, and FSFS with long connections of 600 seconds). Extensive computational experiments on a set of disturbed traffic situations demonstrate the potential to explore the trade-off between passenger satisfaction and the quality of railway service.

All experiments are executed on a normal computer equipped with an Intel i5 central processing unit at 2.70 GHz with 8 GB of memory. The commercial solver CPLEX 12.4 is used to solve the MILP formulations.

Table 2. Utility IM, Utility X

	X boards A	X boards B
IM A→B	$3 \cong (3 + 2.5\epsilon_B + 6\epsilon_C)$, 19	$2.5 \cong (2.5 + 2.5\epsilon_B + 6\epsilon_C)$, 17
IM B→A	$3.5 \cong (3.5 + 3.5\epsilon_A)$, 16	$0 \cong (3.5\epsilon_A)$, 17

Figure 8. (Color online) Macroscopic Solution



5.1. Instance Description

The test case studied is the Dutch railway network of Figure 9. The network is operated with mixed traffic according to a periodic timetable. Tens of thousands of passengers per hour travel between multiple origins and destinations. The generation of the instances is analogous to that in Corman et al. (2016) and is reported here briefly.

The network comprises a significant region of the Netherlands, including Amsterdam, Schiphol, and Utrecht. We report with larger labels the major stations. The traffic pattern we consider is the real timetable for the year 2010, which is presented schematically in Figure 10 (left), where every line is a service running twice per hour per direction. The network is highly interconnected, and there are several bottlenecks. As a result, there is a need for frequent rescheduling in peak hours in the case of disturbances because any small delay may propagate to other trains with a domino effect.

To reduce the instance complexity, we restrict the time horizon of traffic control to different lengths to study the impact of an increasing number of trains and more passengers traveling on the network. The time horizon is the time interval considered for optimization. In this section, we study time horizons of 30 and 60 minutes. Entrance delays for all trains in the network are defined based on a three-parameter Weibull distribution fitted to real data, computed as in Corman et al. (2011). In more detail, we generated different groups of instances by varying the train delay pattern generated according to a typical Monte Carlo scheme. Twenty delay cases are generated representing normal traffic conditions and are generated with the same Weibull distribution used in Corman et al. (2011). For each delay case, every train is subjected to a randomly generated delay according to

the relative Weibull distribution. Twenty more delay cases represent more perturbed traffic conditions with extended delays and are generated using a Weibull distribution with the same scale and shift parameters as the first 20 instances and a doubled shape parameter.

For the instances evaluated, we consider two time horizons of 30 and 60 minutes of traffic and two sets of delays, yielding a total of four instance groups with 20 delay cases per group.

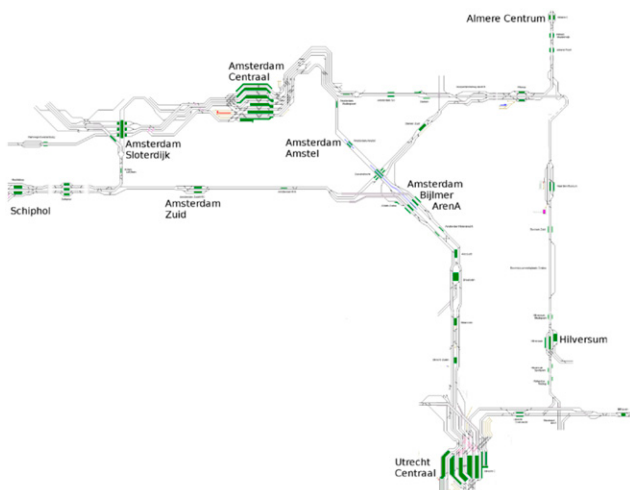
The OD pairs considered are as follows. Triples odw in ODW are generated by considering the largest 22 OD pairs in the network for different time windows. Because of the unavailability of detailed real data about passenger flows, we resort to realistic synthetic OD data based on the average flow of passengers at the considered stations, as published by the IM. Specifically, we first estimate the number of passengers traveling on the line, and we then assign those flows of passengers proportionally to the stations considered in proportion to the number of passengers entering/exiting each station. This is translated into an average rate of passenger generation per OD per time unit.

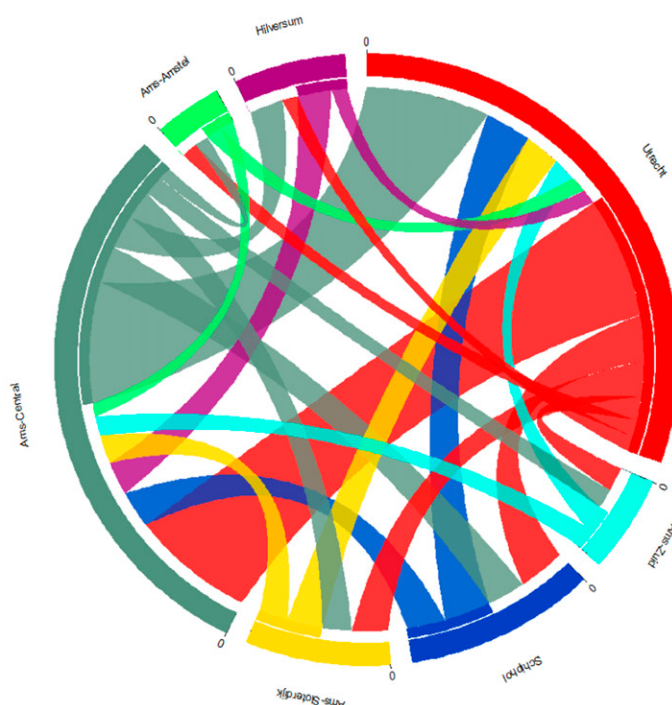
Figure 10 (right) gives a pictorial representation of the considered OD flows. The stations names are plotted along the circle from Utrecht (left) to Amsterdam to Schiphol, matching roughly the geographic locations of the stations along the network. Each station has a color (e.g., red for Utrecht, gray for Amsterdam Central Station). OD flows of passengers between pairs of stations are represented with a line connecting the stations with the same color of the departure station and a thickness corresponding to the number of passengers flowing on it. Based on these estimated OD flows, a demand odw is defined for each departure from the origin station, having a generation time p_{odw} equal to the scheduled departure time of the associated train. Given the values p_{odw} and the passenger arrival frequency at station o , the number of passengers n_{odw} in each demand is chosen equal to the interval between consecutive train departures multiplied by the passenger arrival frequency at station o .

Table 3 summarizes some relevant information for the four groups of instances. The first column reports the group under consideration in terms of delay distribution and time horizon; the average over all instance groups is also reported as a last row.

Columns (2)–(4) report on the traffic conditions, the average entrance delay per train (in seconds), the percentage of trains that have a positive entrance delay at the start of their trip, and the percentage of trains with an entrance delay larger than five minutes at the start of their trip (a typical punctuality measure in railways). Entrance delays are substantially higher for the extended instances, with almost

Figure 9. (Color online) Test Case Infrastructure Description (Elaboration from sporenplan.nl)





The last four columns describe the instances considered, divided into number of trains (column (5)), passenger travel time in the undelayed conditions (column (6)), count of the *ODW* considered (column (7)), and number of passengers (column (8)). The travel time increases when enlarging the time horizon to 60 minutes because the *ODW* covers longer distances. The number of passengers and *ODW* increases by approximately 30%. The results are reported aggregated in what follows because the different instance groups result in the same relative behavior and relationship across the five different MDM algorithms (two Stackelberg algorithms: timetable and train-based rescheduling; and three Nash algorithms, integral, gradient, and connect). The appendix provides a disaggregated description of the main results, focusing on the relative performances of the disaggregated instance groups.

Combining the five MDM algorithms with five instances of routing advice, each computed for 20

5.2. Final Solutions Found

We report in Figure 12 the trade-off between the interests of the two decision makers for the solutions computed. More precisely, we refer to a bidimensional surface, the x -axis being the average passenger travel time (i.e., the objective of the passengers) and the y -axis being the average consecutive train delay, (i.e., the objective of the IM). For both axes, we report the percentage gap regarding a normalization factor. The normalization is such that 100% passenger travel time corresponds to the passenger assignment solution without any delay, whereas 100% train delay corresponds to the best scheduling solution found, given the delays. Each solution (identified by a colored dot as in the legend) results in five points reported on the surface depending on the information associated and routing advice (indicated with a label

Table 3. Instance Data

Instance group	Average entrance delay (s)	Trains delayed at entrance (%)	Trains > 5-min entrance delay (%)	No. of trains	Undelayed passenger travel time (s)	No. of ODW	No. of passengers
30-min normal	37.0	52	6.4	101	1,778.9	136	12,098,010
30-min extended	211	87	45	101	1,778.9	136	12,098,010
60-min normal	39	53	6	202	1,800.8	164	16,975,784
60-min extended	221	88	46	202	1,800.8	164	16,975,784
Average	127	70	26	152	1,789.8	150	14,536,897

on the plot). Timetable is reported out of scale, having a value of approximately 1,300% regarding train delay. A disaggregated analysis of these solutions is reported and commented on in the appendix.

From the plot, many interesting conclusions are available. The solutions associated with the timetable algorithm perform badly in terms of train delay, being about one order of magnitude greater than the other solutions. The performance in terms of passenger travel times is also quite poor. By definition, the *FULL INFO* and *FSFS* route advice coincides for this solution. We also see that the route advice does not influence the delay; that is, the IM does not take any advantage or disadvantage from sharing (or not sharing) information with the travelers.

The solutions associated with the train-based rescheduling algorithm are also independent of route advice, for what concerns the average train delay. In this case, the relative order of the available information is different, with *FSFS* resulting in the largest passenger travel time and *UNDELAYED* being basically intermediate between *FSFS* and *FULL INFO*. In this case, it seems better to have no information about delays (i.e., *UNDELAYED*) than to have correct information

about current delays but the wrong assumption on the decisions of the dispatchers (i.e., *FSFS*).

Nash integral and Nash gradient perform in a very similar manner, having their *FULL INFO* and *UNDELAYED* points coinciding. For these cases, *FSFS* is ranked the worse, and *UNDELAYED* is better for both indicators. In fact, the different advice results in some interaction with the IM; that is, the different route advice leads to different train delays. For Nash integral, this is a rather linear relation: when passengers have better information and they can decrease their travel time, the delays of trains will be reduced. An exception to this is the very large delay and passenger travel time for *FSFS 600*. For Nash gradient, a similar trend is found, with slightly fewer train delays in case of *FSFS 600* and slightly more for *UNDELAYED 600*. We also remark on how both algorithms are associated with the best consecutive train delay. Nash connect performs remarkably differently. In this case, the influence of route advice on passenger travel times is negligible, whereas the influence toward train delays spans approximately 15% (i.e., from 110% to 125%). The quality of passenger travel time is only 4% larger than the situation without delays.

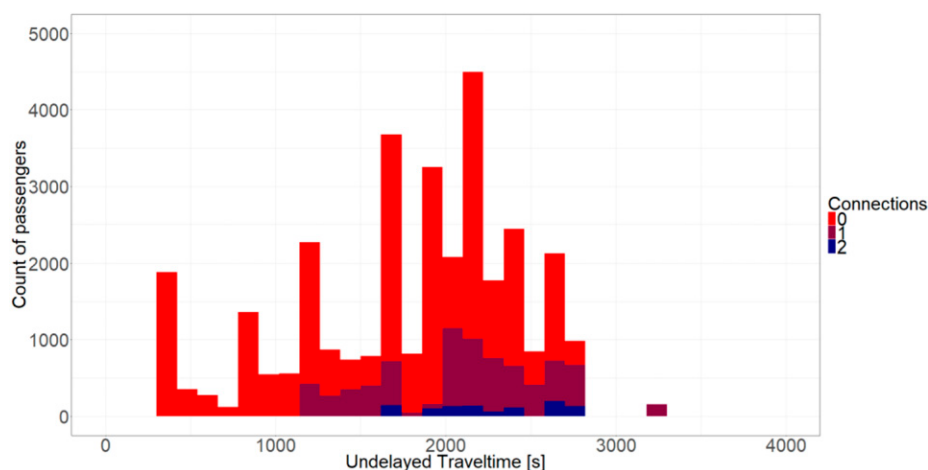
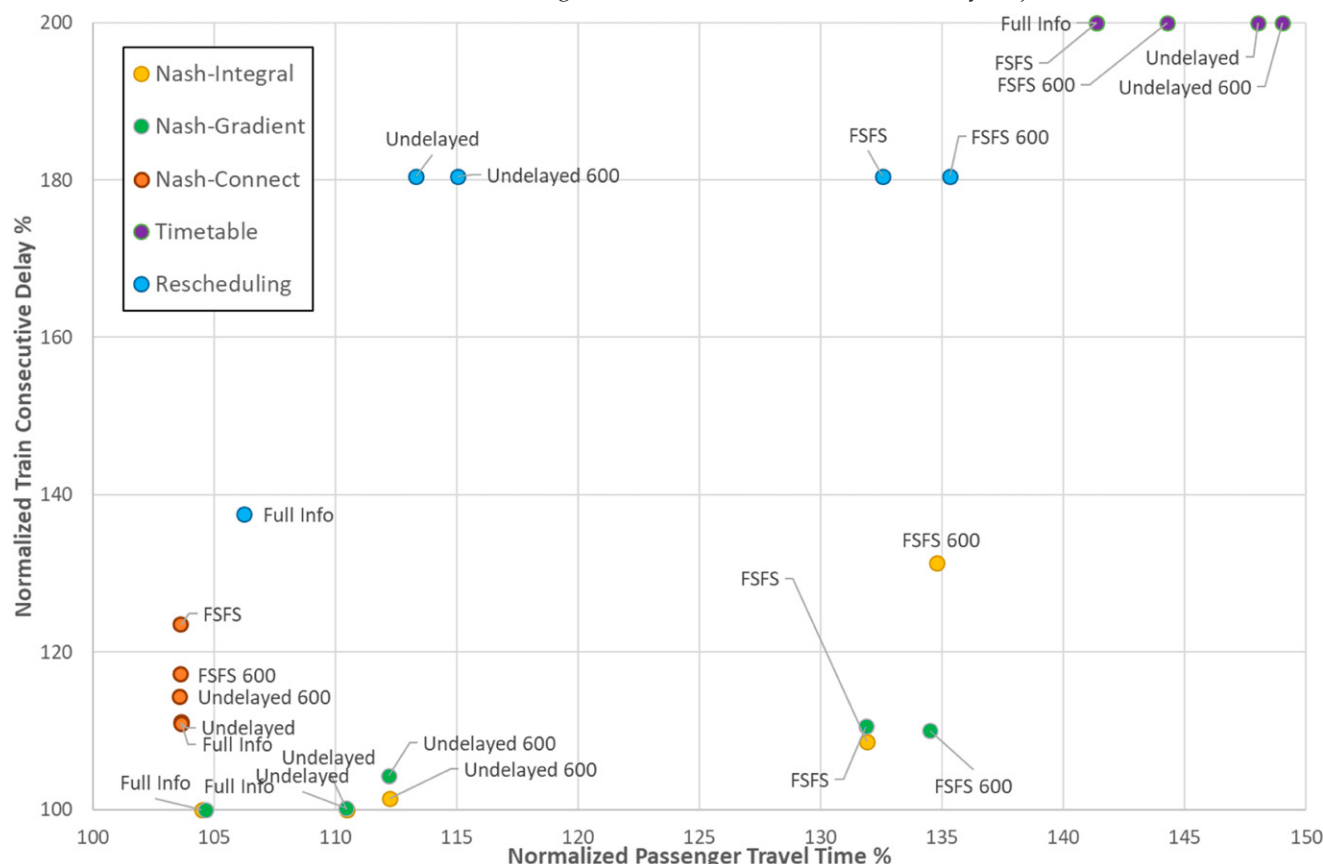
Figure 11. (Color online) Passengers in the Systems, Categorized by Their Travel Time and Amount of Transfer Connections under Undelayed Conditions

Figure 12. (Color online) Trade-off Between Passenger Travel Time and Consecutive Delay Objectives



The impact of long connections, that is, relating *FSFS* to *FSFS 600* and/or *UNDELAYED* to *UNDELAYED 600*, shows that having longer connections increases the travel time (shift to the right) and in the case of Nash integral and Nash gradient, the train delay (shift to the top). For Nash connect, such a trend is not occurring, with only minor variations in terms of train delays and no perceivable difference in terms of passenger travel times. For the solutions based on a Stackelberg principle, that is, timetable and rescheduling, giving long connections penalizes only the passengers. For solutions computed by Nash integral or Nash gradient, giving long connections also penalizes the train operator, which sees larger train delays. This can be associated with a lot of flow over trains that are not very well synchronized, which means some misalignment between the planned timetable and the offered connections to passengers. In this sense, information about future operations affect both the train operator (who would like to know where the people are) and the passengers, who risk having much longer travel times if they base their decision on wrong information. For Nash connect, providing longer connections or shorter connections does not affect the passengers, because the flows of people are considered in a more precise manner; having longer connections

might instead result in having trains being delayed to collect passengers. In this sense, the information on passengers is crucial for the train operator, whereas information is not crucial to the passengers; they do not need to know what the train operator will do, and they will still achieve good travel time.

We comment briefly on the number of iterations required for convergence for the Nash algorithms. Nash integral ranges from 3.1 to 3.4, Nash gradient ranges from 3.3 to 3.5, and Nash connect ranges from 3.1 to 3.3. Timetable and train rescheduling need a single iteration. Overall, the presence of information changes the number of iterations to a minor degree, with *FULL INFO* having the largest number of iterations and the algorithms including long connections having the minimum.

In Table 4, we report on the number of connections used by at least some passengers in the various algorithms (rows) and under different routing advice (columns). As a reference, in the case without delays, 52 connections are used by 2,257 passengers. The cases with the long connections, that is, *FSFS 600* and *UNDELAYED 600*, have a much smaller number of connections used in all approaches apart from Nash connect, which specifically targets keeping useful connections. The solution associated with the timetable

Table 4. Amount of Connections for the Different Cases

Routing advice algorithm	<i>FULL INFO</i>	<i>UNDELAYED</i>	<i>UNDELAYED 600</i>	<i>FSFS</i>	<i>FSFS 600</i>
Nash integral	51.8	49.0	40.6	52.3	44.2
Nash gradient	51.0	49.1	40.5	52.0	43.8
Nash connect	54.0	53.9	54.4	53.5	53.8
Timetable	58.7	55.0	47.2	58.8	46.7
Rescheduling	51.8	49.8	41.1	52.0	43.5

algorithm, under *FULL INFO*, *DELAYED*, and *FSFS*, enables the largest number of connections, although from a comparison with Figure 12, this results in a very large travel time for passengers.

5.3. Impact of Information on Final Solutions

We discuss how different amounts of information affect specific algorithms and specific passenger groups. The total number of connections made available and used by passengers has been reported in Table 4; in Figure 13, we describe graphically how many passengers (size of the bubble) have to change how many times (y -axis and color of the bubble) and how this relates to their realized travel time (x -axis). Basically, we want to show that it is important not only to keep connections (for which the Stackelberg-like algorithm timetable does well) but also to keep the connections useful to a large number of passengers and those that are short enough to ensure a fast travel time. We also want to understand how the variable information affects certain specific groups of passengers, who might require better information than others. We focus in detail on two algorithms with interesting behavior owing to reaction to information, namely Nash gradient (where the number of connections is quite sensitive to information) and timetable (where the number of connections is the largest). Nash integral performs very similar to Nash gradient, whereas Nash connect has almost no variation depending on the routing advice. Each row reports different routing advice, ranging from *FULL INFO* (i.e., best information and routing), to *UNDELAYED*, and finally to *UNDELAYED 600*, identified as algorithms associated with less information given to passengers.

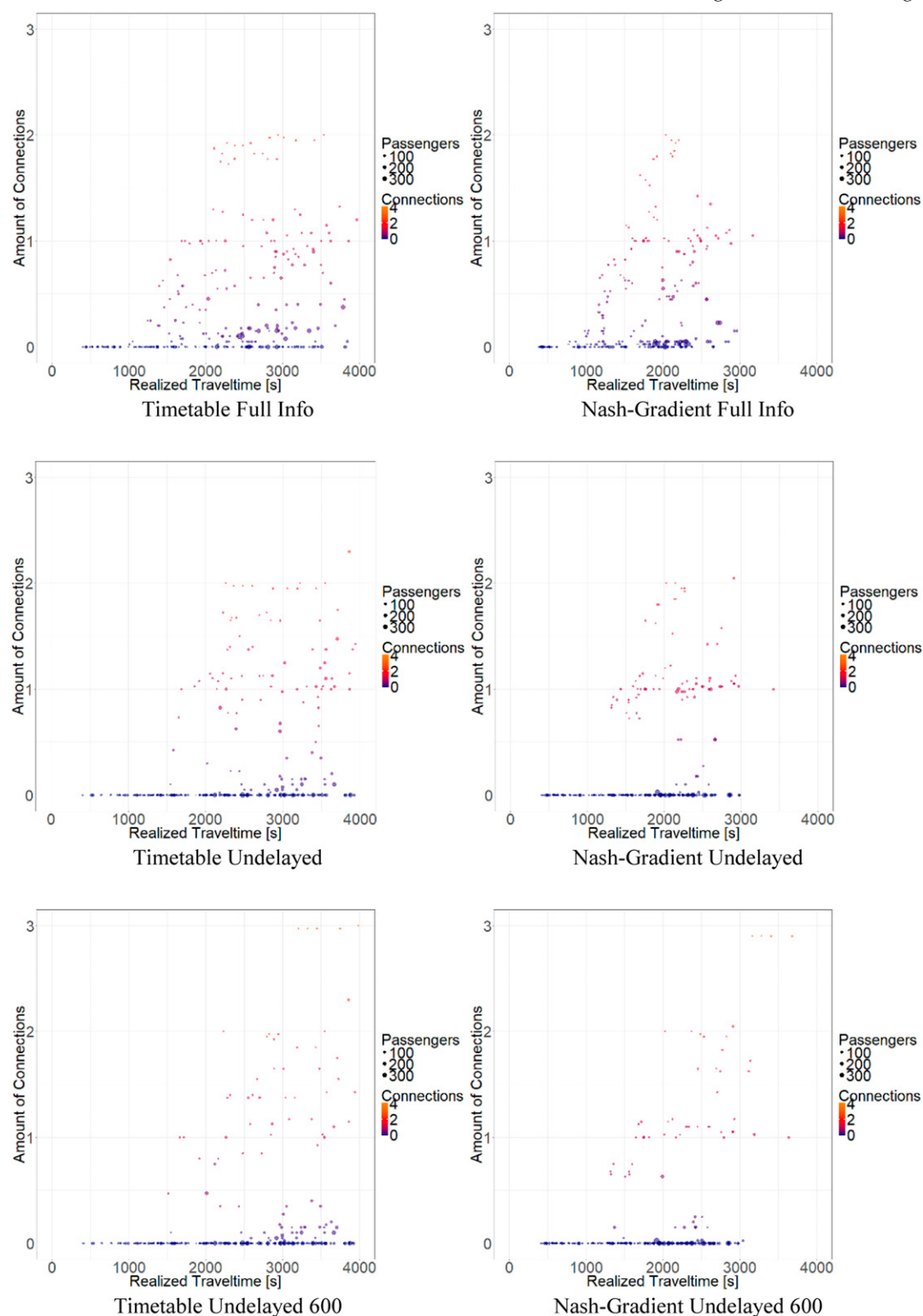
Looking at a single plot, we can understand how the number of connections is the largest for travel times of the upper-medium range and that many large passenger flows have a number of connections larger than zero; that is, under some random delays, they need to use a connection. Looking along the columns, we see a decrease in information and, for both timetable and Nash gradient, a reduction in the number of connections (i.e., the bubbles are pushed toward the x -axis) because they are less known to passengers and because they are too long to be attractive. Moreover, the average of the bubbles is moving to the right; that is,

travel times are increasing, with no clear distinction between trips with a small number of connections or trips with a large number of connections. Looking at the Nash-Gradient column, we see a polarization effect; that is, with decreasing information, the connections are getting closer to the integer numbers of zero, one, or two. In this sense, having less information and a long connection time helps in having very reliable connections, that is, connections that people know they will need in any case. This is good for passengers who do not have additional variability in their travel plans.

Figure 14 shows the variation between travel time and undelayed travel time, that is, the same as the normalized travel time in Figure 12, for the same two algorithms (we can call it *extra travel time* for simplicity). Similarly, here each figure reports graphically how many passengers (size of the bubble) have a large difference in the planned (i.e., without delays) versus realized travel times (y -axis) and how this relates to their realized travel times (x -axis) and number of connections (color of the bubble). The same algorithms and layout as in Figure 13 are used. In this case, it is easy to see that the colors are quite mixed; that is, trips with few connections might result in extra travel time as much as trips with many connections. Also in this case, we look for patterns along the columns. In the timetable case, the general shape of the difference between travel time and extra travel time is constant throughout the three types of routing advice, but it is noticeable how bubbles with many connections (pinkish) are shifting toward the top; that is, they are suffering the most, when information is less and/or the connections are long. For the Nash gradient, we see an even more evident trend that decreasing information and increasing connection length result in larger extra travel time, and this is particularly true for trips with many connections (pinkish bubble moves toward the top of the figure, going from the top row to the bottom row). In this case, having information is quite important.

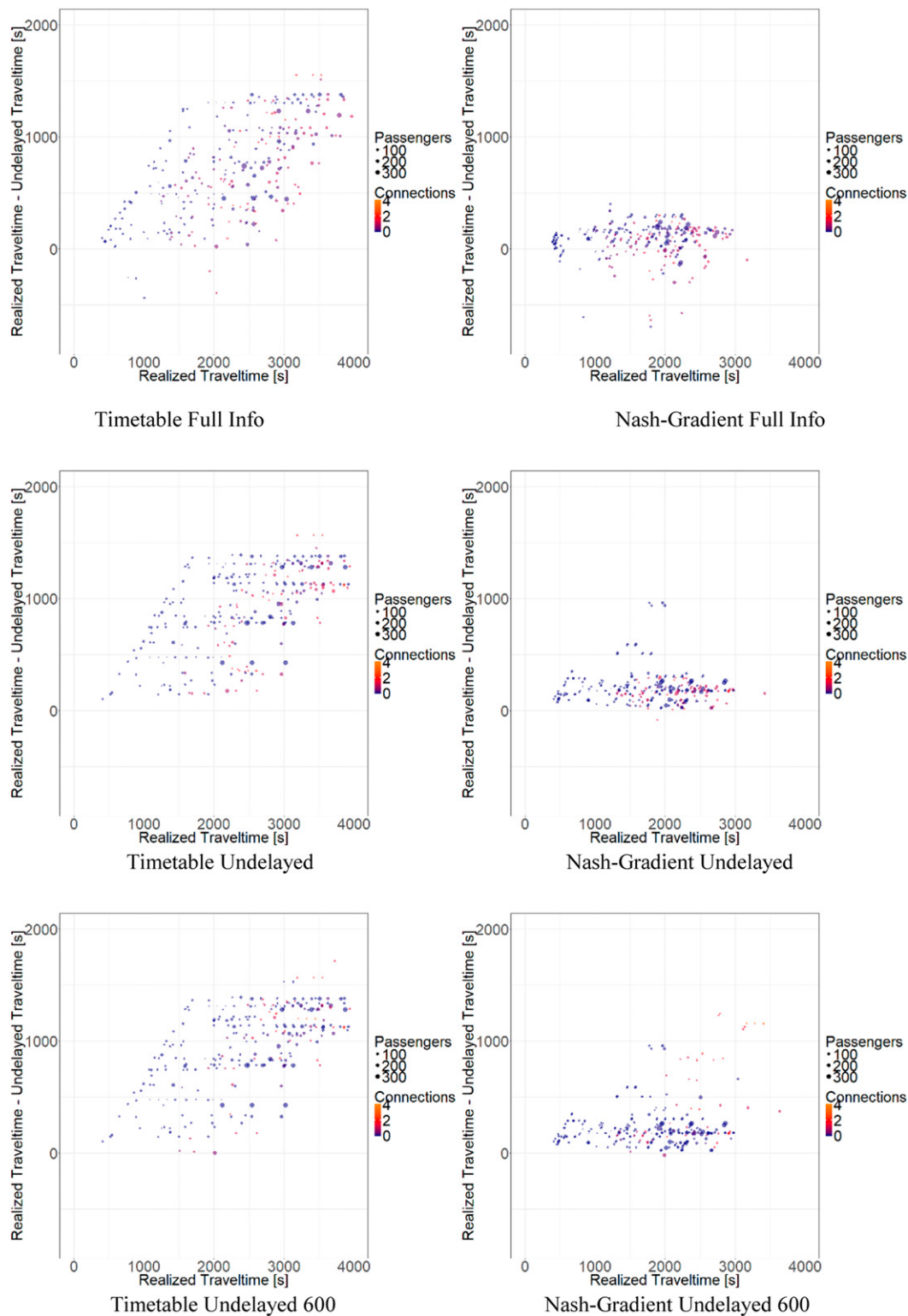
Note that it is possible that a delayed train results in a travel time that is shorter than in the undelayed case (e.g., points in Figure 14 with negative y -values). This happens, for instance, when a train departing late enables faster connections, which would be otherwise

Figure 13. (Color online) Amount of Connections vs. Realized Travel Time, Different Algorithms and Routing Advice



unavailable to passengers, or by enabling shorter waiting times at transfer stations. This is much more common in the *FULL INFO* routing strategy than in the other cases.

We next investigate the impact of different route advice on the travel times of passengers and the gaps between what the route advice suggests and what travel time is actually achieved. We do this in Table 5.

Figure 14. (Color online) Extra Travel Time vs. Realized Travel Times, Different Algorithms and Routing Advice

We divide this investigation for the three Nash algorithms, where people can actually react to the information (blocks of rows; see column (1)); for each of

them, we consider the four types of routing advice: *FSFS*, *FSFS 600*, *UNDELAYED*, and *UNDELAYED 600* (as individual rows; see column (2)). The *FULL*

Table 5. Influence of Routing Advice

Algorithm	Routing advice	Optimal advice		Suboptimal advice		Infeasible advice		Total: extra time (s)
		Percent pass.	Extra time (s)	Percent pass.	Extra time (s)	Percent pass.	Extra time (s)	
Nash integral	<i>FSFS</i>	61.6	−704.9	12.5	483.0	25.9	70.7	−355.8
	<i>FSFS 600</i>	62.5	−770.9	15.4	352.4	22.0	188.1	−386.1
	<i>UNDELAYED</i>	68.5	86.4	23.0	430.8	8.5	153.9	171.2
	<i>UNDELAYED 600</i>	62.1	91.3	29.7	387.0	8.1	481.7	211.0
Nash gradient	<i>FSFS</i>	61.8	−702.1	12.5	482.2	25.7	70.5	−355.7
	<i>FSFS 600</i>	62.5	−773.3	15.4	332.3	22.0	185.9	−391.2
	<i>UNDELAYED</i>	69.0	85.4	22.9	444.4	8.1	156.4	173.4
	<i>UNDELAYED 600</i>	62.1	90.6	30.0	386.8	7.9	485.9	210.8
Nash connect	<i>FSFS</i>	59.5	−655.9	14.7	432.5	25.9	78.0	−306.5
	<i>FSFS 600</i>	60.7	−721.5	18.1	346.0	21.2	191.6	−334.4
	<i>UNDELAYED</i>	65.6	120.0	24.8	455.5	9.6	165.5	207.6
	<i>UNDELAYED 600</i>	60.4	130.2	31.4	413.3	8.2	480.4	247.9

INFO routing advice is used as a benchmark and is not explicitly reported (having no gap between the routing advice and the realized travel time).

In columns (3), (5), and (7), we report how many passengers (pass.; in percentage of the total) have received, respectively, optimal route advice (i.e., the route advice under the particular belief of the passenger remains optimal in case of realized operations), feasible but suboptimal route advice (i.e., the route advice can be followed by the users, but there would be a better one available), and infeasible route advice (i.e., the passengers are told to follow some advice that cannot be done in reality, for instance, to take a transfer connection that is not kept). In columns (4), (6), and (8), we report the additional travel time incurred by the users. This is the difference between the predicted travel time (i.e., the one associated with the route advice that is believed by the passengers) and the realized path (i.e., the actual one). A positive difference is a longer realized travel time; a negative difference means that the passengers are better off than in the planned route, for instance, because they take a delayed train that leaves later. The difference in perception has been pointed out many times as a main driver for passenger satisfaction: when passengers are worse off than their prediction, they tend to feel more uncomfortable than in a complementary situation when they are better off than their prediction. The last column reports the total extra travel time, that

is, how much is the route advice believed by the passengers different compared with the realized one.

Many possible considerations refer to the degree to which all algorithms deliver optimal advice, which is somehow constant around 60%, typically decreasing when going to the long-connection version. The largest difference is between solutions computed using *FSFS* and *UNDELAYED* routing advice; the former has large negative extra travel times, which are driven mostly by the large negative extra travel time in the case of optimal advice; that is, the optimal advice turns out to be the same as given, but with a faster travel time than believed. In case of *UNDELAYED*, the belief of the system is very optimistic, being considered without any delay; that is why the extra travel time is positive and large. *FSFS* and *UNDELAYED* also differ in their ratio between suboptimal and infeasible advice, with *FSFS* giving a lot of infeasible advice, that is, tailor made to the scheduling actions supposed by the dispatchers, whereas *UNDELAYED* delivers mostly suboptimal advice, that is, connections that can exist but are not attractive. In a sense, these experiments show how having bad information (*FSFS*) is worse than having some extra information (i.e., *UNDELAYED*, which has no online information). This depends on the network, the changes, the planned connection times, and the degree to which the schedule is changed as a reaction to the delays. Nash gradient and Nash integral perform similarly,

Table 6. Percentage Variability of Connections Used Through the Iterations (Max–Min Connections Used, Normalized by the Final Value)

Routing strategy algorithm	<i>FULL INFO</i>	<i>UNDELAYED</i>	<i>UNDELAYED 600</i>	<i>FSFS</i>	<i>FSFS 600</i>
Nash integral	9.8	5.1	3.3	9.0	9.9
Nash gradient	12.0	7.3	6.1	10.8	12.2
Nash connect	20.8	41.8	60.5	40.1	53.6

Table 7. Percentage Variability of Travel Time Through the Iterations (Max–Min Travel Time, Normalized by the Final Value)

Routing strategy algorithm	<i>FULL INFO</i>	<i>UNDELAYED</i>	<i>UNDELAYED 600</i>	<i>FSFS</i>	<i>FSFS 600</i>
Nash integral	2.4	1.2	1.2	1.9	2.1
Nash gradient	5.0	3.6	3.4	3.2	2.9
Nash connect	6.4	12.2	14.2	34.5	37.8

with Nash connect having overall the main trends and slightly larger values for extra travel times.

5.4. Impact of Information on Convergence Behavior

We next study how the different information strategies affect the iterative behavior, that is, looking at possible faster convergence or instability. We remark that no instability is found in those small cases, but the extent to which the values of various solutions vary over the iterations gives an idea of the possible outcomes in the case of different information levels about the other players. Table 6 reports the variability (in percent of connections changing) in the number of connections per each OD group throughout all iterations for all delay cases. In a sense, it is a measure of the stability of the algorithm and the solution; a small number means that the convergence is easy; a large number means that large variations are observed along the search process. Similarly, Table 7 reports the variability that the travel time has along the search process (in percent of the final value). Again, this is interpreted as a stability of the final solution and the algorithm. Most relevant comments (Table 6) relate to the reduction in the variability of the number of connections for Nash integral and Nash gradient when using *UNDELAYED* or *UNDELAYED 600* and a small increase when using *FSFS* or *FSFS 600*. In the case of Nash connect, the values are much larger and increase very sharply when reducing the information (i.e., when going from *FULL INFO* to *UNDELAYED/FSFS* and to *UNDELAYED 600/FSFS 600*). This relates to the iterative Nash process, which enables most passengers to have good connections at the end (cf. Figure 12). In case of travel time (Table 7), similar considerations occur; the travel time variation is the highest for Nash connect and has a general trend comparable to the number of connections. For Nash gradient and Nash integral, the knowledge the passengers have of the system (i.e., which actions the IM is supposed to take) influences the convergence process the most in the cases of *FSFS* and *FSFS 600*, that is, a large variability in travel time and connections. This is due to the small number of useful connections.

In Figures 15 and 16, we look at a graphic representation of the variability for two selected algorithms and three selected routing advice types, namely Nash connect (which exhibits a strong increase with less information) and Nash gradient (which exhibits a very strong reduction with less information) and *FULL INFO*, *UNDELAYED*, and *UNDELAYED 600*. The two series of six pictures have similar structure; namely the *x*-axis is the realized travel time, color is the number of connections used, and the size of the bubble refers to how many passengers are affected. The two figures differ by their *y*-axes. Figure 15 shows targeting of the variation in connections throughout iterations, that is, the difference between the maximum and minimum number of connections that a single ODW group has for each single delay instance (i.e., the nonnormalized version of what is reported in Table 6). For Figure 16, this is the difference between the maximum and minimum travel times that a single ODW group has for each single delay instance (again, a nonnormalized version of what is reported in Table 7).

In Figure 15, along the left column from top to bottom, we can see how the realized travel time (position of the bubbles along the *x*-axis) is basically constant, whereas the variation in the *y*-axis increases largely along the column; that is, with less information or long connections, the number of connections varies and varies regardless of travel time. It varies the most (somehow, obviously) for the bubbles with many connections (i.e., the pinkish ones). In the right column, Nash gradient exhibits different dynamics; overall, the vertical spreading of bubbles decrease along the column. The bubbles associated with many connections move toward larger travel times; that is, less information means more stable, but worse, solutions and algorithms. This identified a trade-off for passenger information dissemination.

In Figure 16, we see a similar trend for Nash connect (increasing spread on the *y*-axis along the columns, with most pinkish bubbles moving up) and a very small variation in travel time for Nash gradient, which is further reduced with less information and longer connections. Again, for Nash gradient, the points with more connections (pinkish) move the most to the right; that is,

Figure 15. (Color online) Variation in Connections Across Iterations for Different Algorithms and Routing Advice

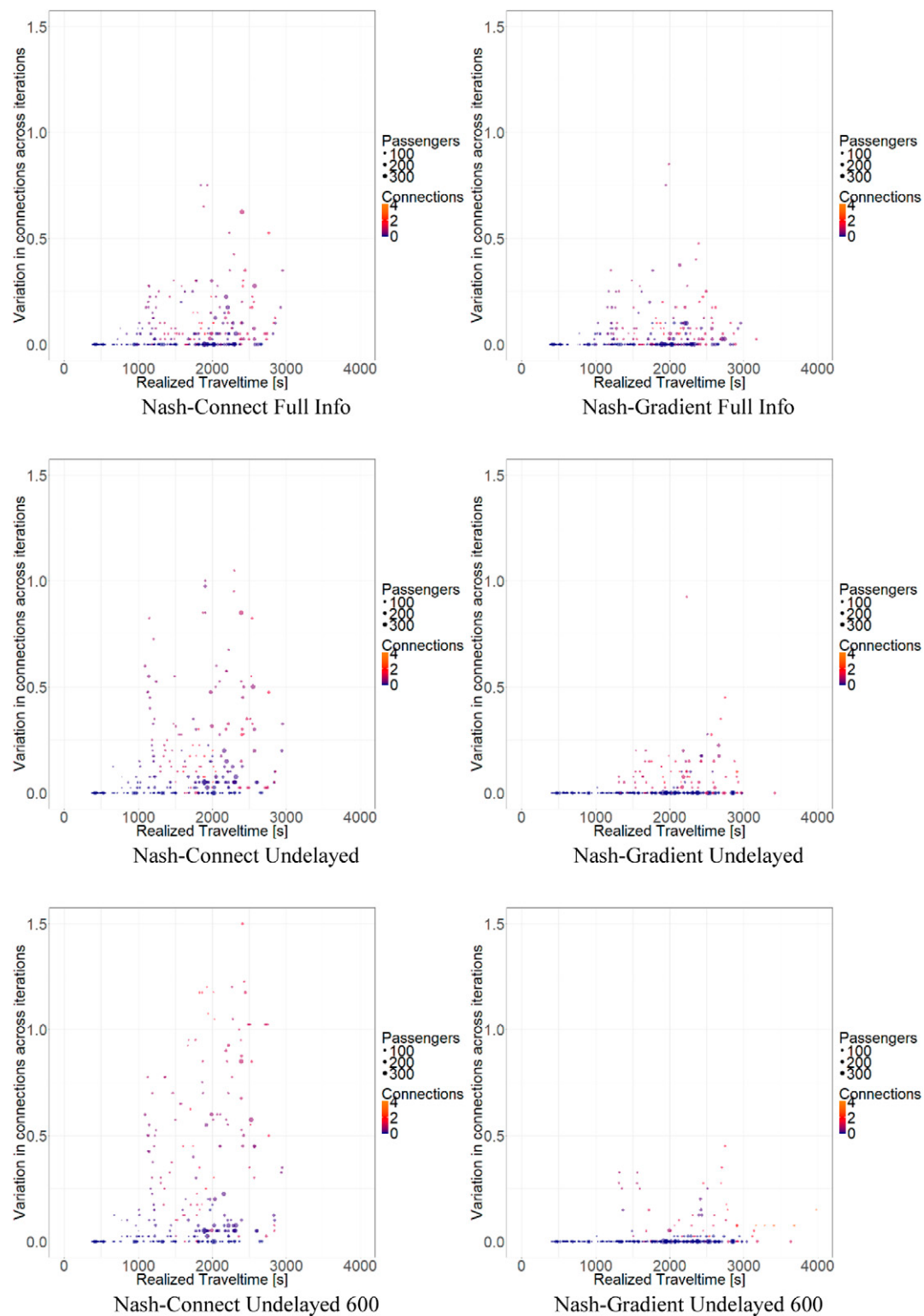
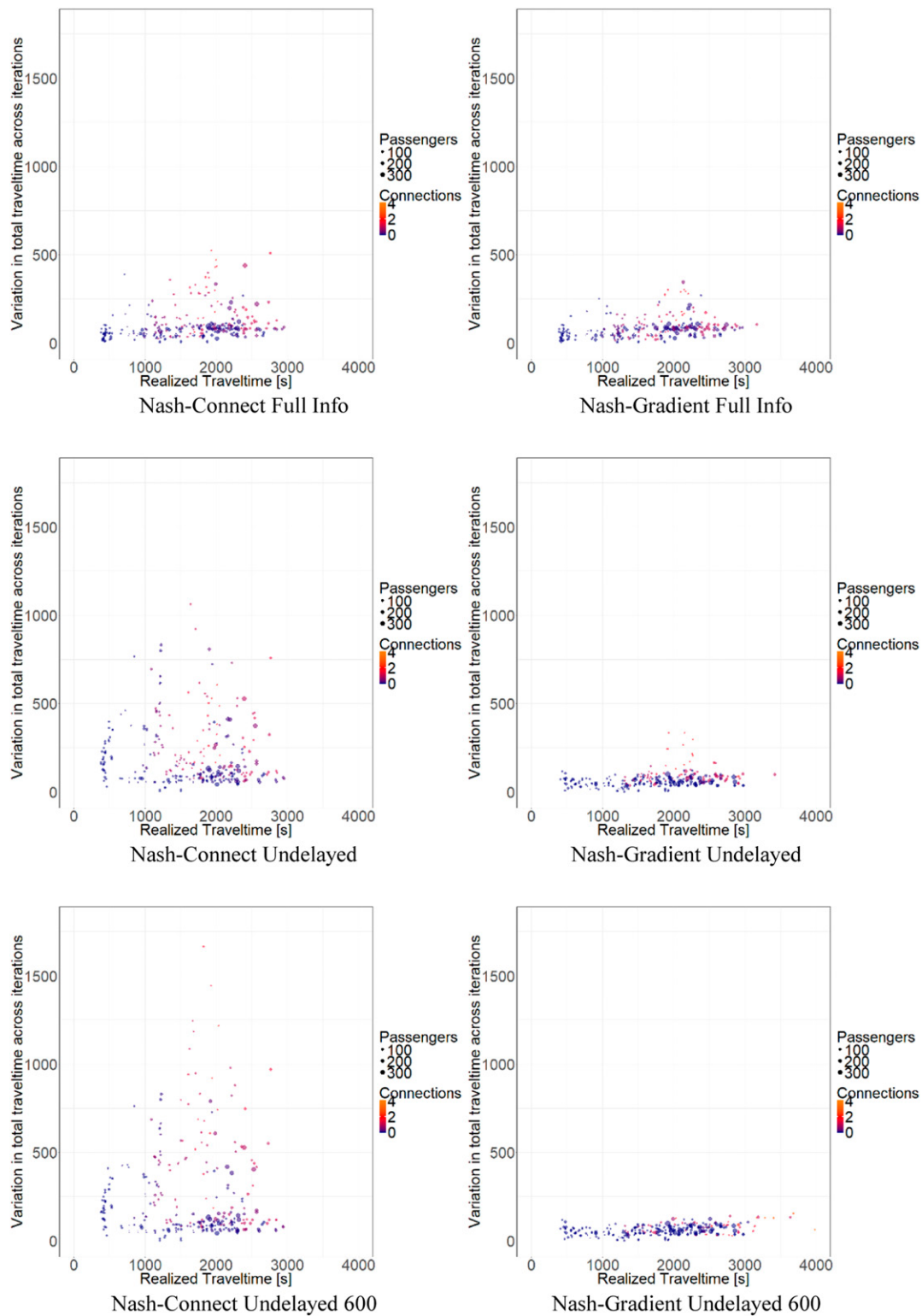


Figure 16. (Color online) Variation in Travel Time Across Iteration vs. Realized Travel Times for Different Algorithms and Routing Advice

less information is better for stability but not for absolute travel time.

6. Conclusion

This paper integrates train rescheduling and delay management into a series of mathematical models to control railway traffic in real time (train rescheduling) with the objective of minimizing passenger travel time (directly tackled by delay management). The state of practice of traffic control is still mainly focused on reducing train delays, which pertains to a small set of decision makers, well identified and with high compliance. This is partly due to policy aspects about the nondiscriminatory role of the IM and on unavailability of information or solution approaches that are able to describe and optimize passenger flows in railway networks. We consider that if traffic controllers assume that travelers have free route choice, they will react to reordering decisions taken in the interest of minimizing their delay by finding the path of minimum travel time. We show that, in general, the two decisions (reordering and route choice) and objectives (train delay and passenger travel time) do not match and study the conditions by which the interaction of the two decision makers results in equilibrium points and its performance and convergence. Specifically, we determine the trade-off and a formal description of the interaction between the objectives of the above-mentioned stakeholders related to their decisions based on some information in different game theoretical settings.

Among the games considered, we look at choices of passengers on a schedule, that is, a transit assignment problem, and the choice of train dispatchers (from the IM) about a schedule for the trains with minimal total delay of the trains, weighted by the number of passengers. Based on game theoretical models to specify the interactions between the players and different levels of information about each other, we are able to compute Nash equilibria of the resulting games that correspond to solutions to the studied microscopic delay management problem. Under some general conditions, we are also able to determine that equilibrium points might not exist.

Computational experiments on a real-life network with a large set of OD pairs show that the two objectives are indeed competitive; we also show how the different setups of updates and reactions to a move of the other players influence variation along the game, and performance of the final equilibrium point is found. There is also a trade-off between complete full information, leading to better solutions, and incomplete information, which increases travel times but simplifies the convergence of the problem and the dissemination to passengers.

Several possible directions are open for future research using these results. One could study other ways to reach the Nash equilibria and possibly determine all of them. Each of the algorithms studied assumes full information about the future railway states; an extension to an online problem with information available as time passes by would be a more realistic setting and allow for competitive analysis of decision-making strategies. Moreover, uncertainty is currently modeled in the decision of the players but not in the future railway states (based on stochastic prediction; see Corman and Kecman 2018) and/or stochastic optimization (see Schön and König 2019). This latter possibility would enable introducing risk-averse or risk-taking behavior in some of the players. Furthermore, the route choice behavior of users can be enriched by introducing heuristics, random utility models, greedy route advice, and hysteresis behaviors similar to Nielsen, Landex Rasmus, and Frederiksen 2009). Similarly, the transit assignment problem with capacity on vehicles generates further interactions across the players interested in using the limited capacity. Overall, it would be nice to understand the extent to which theoretical models fit reality; that is, under delayed conditions, are travelers sticking to their (offline) decisions or are they going to follow the real-time suggestions for alternative modes of connectivity? Approaches based on discrete choice theory or revealed preferences might be helpful to model and address these questions.

Acknowledgments

The author thanks Andrea D'Ariano and Marcella Samà for constructive feedback on this problem.

Appendix.

Tables A.1 and A.2 report the detailed points of average passenger travel time and average train consecutive delay (both in seconds, columns (4) and (6)) for the configuration studied (identified in column (1)), for each type of advice (column (2)) and group of instance (column (3)). For each block with the same advice and group of instance, the ratio between the individual row and the average of the block (i.e., advice + group) is reported in columns (5) and (7) (respectively, regarding passenger travel time and average train consecutive delay). These last two columns are color coded to show more immediately the patterns. Nash integral has not been presented as largely similar to Nash gradient. Tables A.1 focuses on the 30-minute time horizon instances and Table A.2 on the 60-minute time horizon instances.

In both cases, it is evident how the pattern that Nash connect always achieves one of the lowest train delays and the lowest passenger travel times. The lowest train delay is typically achieved by Nash gradient. The timetable consistently achieves the worst travel time and delay. For certain specific groups of instances, the gap between worst and best is more limited than for others.

Table A.1. Disaggregated Results Related to Relevant Points of Figure 11 at 30-Minute Instances

Algorithm	Routing advice	Instance group	Average passenger travel time (s)	Percent compared with advice + group average	Average train consecutive delay (s)	Percent compared with advice + group average
Nash-Connect	FULL INFO	30-min normal	1,798.4	95.2	23.2	38.1
Nash-Gradient	FULL INFO	30-min normal	1,809.7	95.8	19.6	32.1
Timetable	FULL INFO	30-min normal	2,109.0	111.7	169.3	277.1
Rescheduling	FULL INFO	30-min normal	1,836.5	97.3	32.2	52.8
Nash-Connect	FSFS	30-min normal	1,799.6	88.5	28.2	41.3
Nash-Gradient	FSFS	30-min normal	2,099.2	103.2	22.1	32.2
Timetable	FSFS	30-min normal	2,128.5	104.6	175.7	256.8
Rescheduling	FSFS	30-min normal	2,110.8	103.3	47.7	70.6
Nash-Connect	FSFS 600	30-min normal	1,797.3	87.8	24.8	36.7
Nash-Gradient	FSFS 600	30-min normal	2,112.1	103.2	22.0	32.5
Timetable	FSFS 600	30-min normal	2,153.2	105.2	175.7	260.2
Rescheduling	FSFS 600	30-min normal	2,127.2	103.9	47.7	70.6
Nash-Connect	UNDELAYED	30-min normal	1,802.3	93.1	22.6	34.0
Nash-Gradient	UNDELAYED	30-min normal	1,854.0	95.7	20.1	30.2
Timetable	UNDELAYED	30-min normal	2,183.3	112.7	175.7	264.1
Rescheduling	UNDELAYED	30-min normal	1,906.6	98.5	47.7	71.6
Nash-Connect	UNDELAYED 600	30-min normal	1,799.3	92.0	26.0	38.2
Nash-Gradient	UNDELAYED 600	30-min normal	1,883.9	96.3	22.4	33.0
Timetable	UNDELAYED 600	30-min normal	2,204.2	112.7	175.7	258.7
Rescheduling	UNDELAYED 600	30-min normal	1,934.3	98.9	47.7	70.1
Nash connect	FULL INFO	30-min extended	1,901.9	91.5	39.9	32.8
Nash gradient	FULL INFO	30-min extended	1,910.4	91.9	36.8	30.3
Timetable	FULL INFO	30-min extended	2,551.5	122.8	360.8	297.0
Rescheduling	FULL INFO	30-min extended	1,947.0	93.7	48.4	39.9
Nash connect	FSFS	30-min extended	1,904.6	81.7	46.0	36.8
Nash gradient	FSFS	30-min extended	2,410.8	103.5	43.4	34.7
Timetable	FSFS	30-min extended	2,576.4	110.6	361.4	288.9
Rescheduling	FSFS	30-min extended	2,429.5	104.3	49.6	39.7
Nash connect	FSFS 600	30-min extended	1,901.9	80.1	40.5	32.9
Nash gradient	FSFS 600	30-min extended	2,474.1	104.2	40.9	33.2
Timetable	FSFS 600	30-min extended	2,634.4	110.9	361.4	293.6
Rescheduling	FSFS 600	30-min extended	2,491.7	104.9	49.6	40.3
Nash connect	UNDELAYED	30-min extended	1,902.8	86.2	36.9	30.6
Nash gradient	UNDELAYED	30-min extended	2,058.2	93.3	34.7	28.8
Timetable	UNDELAYED	30-min extended	2,754.5	124.9	361.4	299.5
Rescheduling	UNDELAYED	30-min extended	2,109.5	95.6	49.6	41.1
Nash connect	UNDELAYED 600	30-min extended	1,902.3	85.8	38.9	32.1
Nash gradient	UNDELAYED 600	30-min extended	2,078.4	93.8	35.4	29.2
Timetable	UNDELAYED 600	30-min extended	2,759.3	124.5	361.4	297.8
Rescheduling	UNDELAYED 600	30-min extended	2,127.4	96.0	49.6	40.9

Table A.2. Disaggregated Results Related to Relevant Points of Figure 11 in 60-Minute Instances

Algorithm	Routing advice	Instance group	Average passenger travel time (s)	Percent compared with advice + group average	Average train consecutive delay (s)	Percent compared with advice + group average
Nash connect	FULL INFO	60-min normal	1,814.5	93.1	27.1	30.8
Nash gradient	FULL INFO	60-min normal	1,841.0	94.4	23.0	26.1
Timetable	FULL INFO	60-min normal	2,285.7	117.3	272.2	309.7
Rescheduling	FULL INFO	60-min normal	1,855.7	95.2	29.4	33.4
Nash connect	FSFS	60-min normal	1,809.6	84.0	26.9	28.1
Nash gradient	FSFS	60-min normal	2,246.0	104.3	24.2	25.3
Timetable	FSFS	60-min normal	2,313.4	107.4	282.5	295.1
Rescheduling	FSFS	60-min normal	2,243.4	104.2	49.3	51.5
Nash connect	FSFS 600	60-min normal	1,811.8	83.1	28.4	29.6
Nash gradient	FSFS 600	60-min normal	2,270.0	104.1	24.0	25.0
Timetable	FSFS 600	60-min normal	2,368.7	108.6	282.5	294.1
Rescheduling	FSFS 600	60-min normal	2,274.4	104.3	49.3	51.3
Nash connect	UNDELAYED	60-min normal	1,811.2	89.7	28.7	29.9
Nash gradient	UNDELAYED	60-min normal	1,891.1	93.7	22.8	23.8
Timetable	UNDELAYED	60-min normal	2,448.4	121.3	282.5	294.8
Rescheduling	UNDELAYED	60-min normal	1,924.2	95.3	49.3	51.4
Nash connect	UNDELAYED 600	60-min normal	1,811.3	88.6	28.7	29.7
Nash gradient	UNDELAYED 600	60-min normal	1,930.2	94.4	25.1	26.1
Timetable	UNDELAYED 600	60-min normal	2,475.9	121.0	282.5	293.1
Rescheduling	UNDELAYED 600	60-min normal	1,964.5	96.0	49.3	51.1
Nash connect	FULL INFO	60-min extended	1,905.9	85.6	44.0	19.1
Nash gradient	FULL INFO	60-min extended	1,929.9	86.7	39.9	17.3
Timetable	FULL INFO	60-min extended	3,099.5	139.3	780.2	339.1
Rescheduling	FULL INFO	60-min extended	1,967.2	88.4	56.1	24.4
Nash connect	FSFS	60-min extended	1,905.7	73.3	48.0	20.3
Nash gradient	FSFS	60-min extended	2,684.9	103.3	44.0	18.6
Timetable	FSFS	60-min extended	3,100.8	119.3	783.4	331.0
Rescheduling	FSFS	60-min extended	2,706.6	104.1	71.4	30.2
Nash connect	FSFS 600	60-min extended	1,907.6	71.7	47.8	20.2
Nash gradient	FSFS 600	60-min extended	2,771.3	104.1	46.1	19.4
Timetable	FSFS 600	60-min extended	3,172.4	119.2	783.4	330.3
Rescheduling	FSFS 600	60-min extended	2,794.5	105.0	71.4	30.1
Nash connect	UNDELAYED	60-min extended	1,904.1	79.7	45.8	19.4
Nash gradient	UNDELAYED	60-min extended	2,103.7	88.1	43.4	18.4
Timetable	UNDELAYED	60-min extended	3,373.6	141.2	783.4	332.0
Rescheduling	UNDELAYED	60-min extended	2,173.3	91.0	71.4	30.3
Nash connect	UNDELAYED 600	60-min extended	1,903.2	78.8	44.6	18.9
Nash gradient	UNDELAYED 600	60-min extended	2,139.1	88.6	43.0	18.3
Timetable	UNDELAYED 600	60-min extended	3,406.8	141.1	783.4	332.5
Rescheduling	UNDELAYED 600	60-min extended	2,208.4	91.5	71.4	30.3

For train delays, the gap between best and worst is for those 60-minute instances (with a minimum of 17% and a maximum of 340% regarding the group) larger than for the 30-minute instances (the minimum/maximum gap was then ranging from 28.8% to 300%); that is, the added variability and the longer time horizon identify cases where optimization can deliver added benefit.

For passenger travel times, the gap between best and worst is again for those 60-minute instances (with a minimum of 71% and a maximum of 141% regarding the group) larger than for the 30-minute instances (the minimum/maximum gap was then ranging from 80% to 124%). In other words, the added variability and the longer time horizon identify cases where optimization can deliver added benefit. Moreover, the variability within the group is much larger concerning the train delays than the passenger travel times because the train delays can theoretically be reduced to zero, whereas passenger travel time has a clear lower bound that is on the order of 1,000 seconds.

References

- Bablinski K (2015) A game-based analysis of freight paths allocation with a case study on Great Britain Brighton Main Line. *Transportation Res. Procedia* 13:196–208.
- Bauer R, Schöbel A (2014) Rules of thumb: Practical online strategies for delay management. *Public Transportation* 6:85–105.
- Bell MGH (2000) A game theory approach to measuring the performance reliability of transport networks. *Transportation Res. Part B: Methodology* 34:533–545.
- Bender M, Büttner S, Krumke SO (2013) Online delay management on a single line: beyond competitive analysis. *Public Transportation* 5:243–266.
- Binder S, Chen J, Bierlaire M (2014) Generation and evaluation of passenger-oriented railway disposition timetables. *Proc. 14th Swiss Transport Research Conf.* (Swiss Transport Research Conference, Ascona, Switzerland), 1–12.
- Binder S, Maknoon Y, Bierlaire M (2017) Exogeneous priority rules for the capacitated passenger assignment problem. *Transportation Res. Part B: Methodology* 105:19–42.
- Böhmová K, Mihalák M, Neubert P, Pröger T, Widmayer P (2015) Robust routing in urban public transportation: Evaluating strategies that learn from the past. *OpenAccess Series Informatics* 48:68–81.
- Bouman PC, Kroon LG, Vervest PHM, Maróti G (2017) Capacity, information and minority games in public transport. *Transportation Res. Part C: Emerging Tech.* 70:157–170.
- Brewer PJ, Plott CR (1996) A binary conflict ascending price (BICAP) mechanism for the decentralized allocation of the right to use railroad tracks. *Internat. J. Indust. Organ.* 14:857–886.
- Cacchiani V, Huisman D, Kidd M, Kroon L, Toth P, Veelenturf L, Wagenaar J (2014) An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Res. Part B: Methodology* 63:15–37.
- Caprara A, Kroon LG, Monaci M, Peeters M, Toth P (2006) Passenger railway optimization. Barnhart C, Laporte G, eds. *Handbooks in Operations Research and Management Science*, vol. 14 (Elsevier, Amsterdam), 129–187.
- Carey M, Crawford I (2007) Scheduling trains on a network of busy complex stations. *Transportation Res. Part B: Methodology* 41(2):159–178.
- Cats O, Koutsopoulos H, Burghout W, Toledo T (2011) Effect of real-time transit information on dynamic path choice of passengers. *Transportation Res. Rec.* 2217:46–54.
- Chen O, Ben-Akiva M (1998) Game-theoretic formulations of interaction between dynamic traffic control and dynamic traffic assignment. *Transportation Res. Rec.* 1617:179–188.
- Corman F, Kocman P (2018) Stochastic prediction of train delays in real-time using Bayesian networks. *Transportation Res.* 95: 599–615.
- Corman F, D'Ariano A, Hansen IA (2014) Evaluating disturbance robustness of railway schedules. *J. Intelligent Transportation Systems* 18(1):106120.
- Corman F, D'Ariano A, Hansen IA, Pacciarelli D (2011) Optimal multi-class rescheduling of railway traffic. *J. Rail Transport Planning Management* 1(1):14–24.
- Corman F, D'Ariano A, Pacciarelli D, Pranzo M (2012) Bi-objective conflict detection and resolution in railway traffic management. *Transportation Res. Part C: Emerg. Tech.* 20(1):79–94.
- Corman F, D'Ariano A, Pacciarelli D, Pranzo M (2014) Dispatching and coordination in multi-area railway traffic management. *Comput. Oper. Res.* 44(1):146–160.
- Corman F, D'Ariano A, Marra AD, Pacciarelli D, Samà M (2016) Integrating train scheduling and delay management in realtime railway traffic control. *Transportation Res. Part E: Logist. Transportation Rev.* 105:213–239.
- D'Ariano A, Pacciarelli D, Pranzo M (2007) A branch and bound algorithm for scheduling trains in a railway network. *Eur. J. Oper. Res.* 183(2):643–657.
- Delling D, Italiano GF, Pajor T, Santaroni F (2014) Better transit routing by exploiting vehicle GPS data. Chin X, ed. *Proc. 7th ACM SIGSPATIAL Internat. Conf.* (ACM, New York), 31–40.
- Dell'Olio L, Moura JL, Ibeas A (2006) Bi-level mathematical programming model for locating bus stops and optimizing frequencies transportation research record. *Transportation Res. Rec.* 1971(1):23–31.
- Dollevet T, Huisman D (2014) Fast heuristics for delay management with passenger rerouting. *Public Transportation* 6(1-2):67–84.
- Dollevet T, Corman F, D'Ariano A, Huisman D (2014) An iterative optimization framework for delay management and train scheduling. *J. Flexible Services Manufacturing* 26(4):490–515.
- Dollevet T, Huisman D, Schmidt M, Schöbel A (2012) Delay management with rerouting of passengers. *Transportation Sci.* 46(1): 74–89.
- Dollevet T, Huisman D, Kroon L, Schmidt M, Schöbel A (2015) Delay management including capacities of stations. *Transportation Sci.* 49(2):185–203.
- Duc Quynh T, Thuan NG (2018) On optimization problems in urban transport. Pardalos PM, Migdalas A, eds. *Open Problems in Optimization and Data Analysis*, Springer Optimization and Its Applications, vol. 141 (Springer, Heidelberg), 151–170.
- Easley D, Kleinberg J (2010) Modeling network traffic using game theory. Easley D, Kleinberg J, eds. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World* (Cambridge University Press, Cambridge, UK), 229–247.
- Espinosa-Aranda J, García-Rodenas R (2013) A demand-based weighted train delay approach for rescheduling railway networks in real-time. *J. Rail Transport Planning Management* 3(1-2):1–13.
- European Rail Research Advisory Council (2012) *Rail Route 2050: The Sustainable Backbone of the Single European Transport Area* (European Rail Research Advisory Council, Brussels).
- Fagnelli V, Sanguineti S (2014) A game theoretic model for re-optimizing a railway timetable. *Eur. Transportation Res. Rev.* 6: 113–125.
- Gatto M (2007) On the impact of uncertainty on some optimization problems: combinatorial aspects of delay management and robust online scheduling. PhD thesis, ETH Zurich.
- Ghaemi N, Cats O, Goverde RMP (2017) Railway disruption management challenges and possible solution directions. *Public Transportation* 9(1-2):343–364.

- Goerigk M, Knott M, Müller-Hannemann M, Schmidt M, Schöbel A (2013) The price of strict and light robustness in timetable information. *Transportation Sci.* 48:225–242.
- Hansen IA, Pacht J, eds. (2014) *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation*. (Eurailpress, Hamburg, Germany).
- Hamdouch Y, Lawphongpanich S (2008) Schedule-based transit assignment with loading priorities. *Math. Programming* 101(1).
- Hamdouch Y, Lawphongpanich S (2010) Congestion pricing for schedule-based transit networks. *Transportation Sci.* 44(3): 350–366.
- Hamdouch Y, Ho H, Sumalee A, Wang G (2011) Schedule-based transit assignment model with vehicle capacity and seat availability. *Transportation Res. Part B: Methodology* 42(7–8):663–684.
- Harrod S (2013) Auction pricing of network access for North American railways. *Transportation Res. Part E: Logist. Trans. Rev.* 49:176–189.
- Hollander Y, Prashker JN (2006) The applicability of non-cooperative game theory in transport analysis. *Transportation* 33(5):481–496.
- Josyula SP, Törnquist Krasemann J (2017) Passenger-oriented railway traffic re-scheduling: A review of alternative strategies utilizing passenger flow data. *7th Internat. Conf. Railway Oper. Res. Anal. Lille, France*.
- Klabes S (2010) Algorithmic railway capacity allocation in a competitive european railway market. Dissertation, RWTH Aachen, Germany.
- Kuo A, Miller-Hooks E (2012) Developing responsive rail services through collaboration. *Transportation Res. Part B: Methodology* 46: 424–439.
- Lamorgese L, Mannino C (2015) An exact decomposition approach for the real-time train dispatching problem. *Oper. Res.* 63(1):48–64.
- Lamorgese L, Mannino C, Pacciarelli P, Törnquist Krasemann J (2018) Train dispatching. Borndörfer R, et al., eds. *Handbook of Optimization in the Railway Industry*, International Series in Operations Research & Management Science, vol. 268, (Springer, Heidelberg, Germany), 265–284.
- Laliv R, Schmutzler A (2008) Exploring the effects of competition for railway markets. *Internat. J. Indust. Organ.* 26:443–458.
- Luan X, Meng L, Corman F (2017) Non-discriminatory train dispatching in a rail transport market with multiple competing and collaborative train operating companies. *Transportation Res. Part C: Emerging Tech.* 80:148–174.
- Mascis A, Pacciarelli D (2002) Job shop scheduling with blocking and no-wait constraints. *Eur. J. Oper. Res.* 143(3):498–517.
- Meschini L, Gentile G, Papola N (2007) A frequency based transit model for dynamic traffic assignment to multimodal networks. Allsop RE, Bell MGH, Heydecker BG, eds. *Proc. Internat. Symp. Transportation Traffic Theory* 17 (Elsevier, Amsterdam), 407–436.
- 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 Res. Part B: Methodology* 67(1):208–234.
- Mesbah M, Sarvi M, Ouyeyi I, Currie G (2011) Optimization of transit priority in the transportation network using a decomposition methodology. *Transportation Res. Part C: Emerging Tech.* 19(2):363–373.
- Nassir N, Hickman M, Ma ZL (2019) A strategy-based recursive path choice model for public transit smart card data. *Transportation Res. Part B: Methodology* 126(August):528–548.
- Nielsen OA, Frederiksen RD (2006) Optimisation of timetable-based, stochastic transit assignment models based on MSA. *Ann. Oper. Res.* 144(1):263–285.
- Nielsen OA, Landex Rasmus OA, Frederiksen D (2009) Passenger delay models for rail networks. Nuzzolo A, Wilson NHM, eds. *Schedule-Based Modeling of Transportation Networks*, Operations Research/Computer Science Interfaces Series, vol. 46 (Springer, New York), 1–23.
- Parbo J, Nielsen OA, Prato CG (2016) Passenger perspectives in railway timetabling: A literature review. *Transport Rev.* 36:500–526.
- Patriksson M (2008) Robust bi-level optimization models in transportation science. *Philos. Trans. Roy. Soc. A: Math. Phys. Engrg. Sci.* 366(1872):1989–2004.
- Pellegrini P, Rodriguez J (2013) Single European sky and single European railway area: A system level analysis of air and rail transportation. *Transportation Res. Part A: Policy Practice* 57(1):64–86.
- Pellegrini P, Marlière G, Rodriguez J (2014) Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Res. Part B: Methodology* 59(1):58–80.
- Poon M, Wong S, Tong C (2004) A dynamic schedule-based model for congested transit networks. *Transportation Res. Part B: Methodology* 38(4):343–368.
- Rosenthal RW (1973) A class of games possessing pure-strategy Nash equilibria. *Internat. J. Game Theory* 2:65–67.
- Rückert R, Lemnian M, Blendinger C, Rechner S, Müller-Hannemann M (2017) PANDA: A software tool for improved train dispatching with focus on passenger flows. *Public Transportation* 9(1):307–324.
- Samà M, D’Ariano A, Corman F, Pacciarelli D (2016) A variable neighborhood search for fast train scheduling and routing during disturbed railway traffic situations. *Comput. Oper. Res.* 78:480–499.
- Samà M, Meloni C, D’Ariano A, Corman F (2015) A multi-criteria decision support methodology for real-time train scheduling. *J. Rail Transport Planning Management* 5(3):146–162.
- Sato K, Tamura K, Tomii N (2013) A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. *J. Rail Transport Planning Management* 3(3):38–53.
- Schachtebeck M, Schöbel A (2010) To wait or not to wait—And who goes first? Delay management with priority decisions. *Transportation Sci.* 44(3):307–321.
- Schöbel A (2001) A model for the delay management problem based on mixed-integer programming. Proceedings of the 1st Workshop on Algorithmic Methods and Models for Optimization of Railways. *Electron. Notes Theoret. Comput. Sci.* 50(1):1–10.
- Schöbel A (2007) Integer programming approaches for solving the delay management problem. robust and online large-scale optimization. *Algorithmic Methods for Railway Optimization*, Lecture Notes Comput. Sci., vol. 4359, (Springer, Berlin), 145–170.
- Schöbel A (2009) Capacity constraints in delay management. *Public Transport Planning Oper.* 1(2):135–154.
- Schön C, König E (2019) A stochastic dynamic programming approach for delay management of a single train line. *Eur. J. Oper. Res.* 271:501–518.
- Schmidt M, Kroon L, Schöbel A, Bouman P (2017) The travelers route choice problem under uncertainty: Dominance relations between strategies. *Oper. Res.* 65(1):184–199.
- Spiess H, Florian M (1989) Optimal strategies: A new assignment model for transit networks. *Transportation Res. Part B: Methodology* 23(2):83–102.
- Staňková K (2009) On Stackelberg and inverse Stackelberg games and their applications in the optimal toll design problem, the energy markets liberalization problem, and in the theory of incentives. PhD thesis, Delft University of Technology, Netherlands.
- Suhl L, Biederick C, Klierer N (2001). Design of customer-oriented dispatching support for railways. Voss S, Daduna JR, eds. *Computer-Aided Scheduling of Public Transport*, vol. 505 (Springer, New York).
- Taale H (2008) Integrated anticipatory control of road networks: A game-theoretical approach. PhD thesis, Delft University of Technology, Netherlands.
- Tomii N, Tashiro Y, Tanabe N, Hirai C, Muraki K (2005) Train rescheduling algorithm which minimizes passengers’ dissatisfaction. Ali M, Esposito F, eds. *Innovations in Applied Artificial Intelligence*, Lecture Notes in Artificial Intelligence, vol. 3533 (Springer, Berlin), 829–838.

- Törnquist J, Persson JA (2007) N-tracked railway traffic re-scheduling during disturbances. *Transportation Res. Part B: Methodology* 41(3):342–362.
- Van Der Hurk E, Kroon L, Maróti G, Vervest P (2015) Deduction of passengers' route choices from smart card data. *IEEE Trans. Intelligent Transportation Systems* 16(1):430–440.
- Van Thienen S, Corman F, Vansteenwegen P (2018) Considering a dynamic impact zone for real-time railway traffic management. *Transportation Res. Part B: Methodology* 111:39–59.
- Vansteenwegen P, Van Oudheusden D (2007) Decreasing the passenger waiting time for an intercity rail network. *Transportation Res. Part B: Methodology* 41(4):478–492.
- Wang YB, De Schutter TJJ, van den Boom B, Tang N (2014) Efficient bilevel approach for urban rail transit operation with stop-skipping. *IEEE Trans. Intelligent Transportation Systems* 15(6): 2658–2670.
- Wilson NHM, Nuzzolo A, eds. (2004) *Schedule-Based Dynamic Transit Modeling: Theory and Applications*, Operations Research/Computer Science Interfaces Series, vol. 28. (Springer, New York).
- Zhang H, Su Y, Peng L, Yao D (2010) A review of game theory applications in transportation analysis. Wang Z, ed. *Proc. IEEE Internat. Conf. Computer and Information Application ICCIA 2010* (IEEE, Piscataway, NJ), 152–157.