Data-Enabled Stochastic Modelling for Evaluating Schedule Robustness of Railway Networks

Harshad Khadilkar
TATA Consultancy Services (TCS)
harshad.khadilkar@tcs.com

This paper evaluates the robustness of a railway network with respect to operational delays. It assumes that trains in the network operate on fixed routes and with reference to a timetable. A stochastic delay propagation model is proposed for identifying primary (externally imposed) delays and for computing the resultant secondary (knock-on) delays. Delay probability distributions are computed for each train at each station on its journey, using timetable and infrastructure data for identifying potential station resource conflicts with other trains. The delay predictions are used to evaluate schedule robustness using two newly proposed metrics. *Individual robustness* measures the ability of trains to limit the adverse effects of their own primary delays. On the other hand, *collective robustness* measures the ability of the network as a whole, to limit the knock-on effects of primary delays imposed on a small fraction of trains. The two metrics provide stochastic guarantees on the punctuality of trains when the published schedule is put in operation. The applicability of the proposed methodology is validated using empirical data from a portion of the Indian Railway network, containing more than 38,000 train arrival/departure records. While a railway network is used as a case study, the same ideas can be applied to any scheduled transportation network.

Key words: Rail transportation network, schedule robustness, stochastic delay propagation
History: Submitted - January 2016, Revision 1 - April 2016, Revision 2 - May 2016

1. Introduction

Mass modes of transportation (road, rail, air) provide reliable options for people and goods to move from a large set of origins to a just as large set of destinations. As a result, they are established in the form of highly interconnected networks. Bell and Iida (1997) describe transportation networks as composed of nodal points (bus stops, railway stations, airports) connected by links (roads, rail tracks, air corridors). The infrastructure can be utilised by a heterogeneous mix of preplanned and on-demand traffic. Scheduled service transportation networks (such as bus services, railways or commercial aviation) are composed of largely preplanned traffic and operate with reference to a timetable. The schedules may take one of two broad forms, as pointed out by Corman et al. (2014): *regular* services in which vehicles may travel long distances on complex multi-hop itineraries, or *shuttle* services in which vehicles continually operate to and fro between a pair of nodes. This paper analyses the robustness of both types of scheduled networks, using railways as a case study.

1.1. Motivation

Farahani et al. (2013) analyse transportation networks at three time scales. *Strategic planning* consists of long term decisions (years to decades) such as infrastructure development, fleet planning, and origin-destination service frequency. *Tactical planning* is carried out on an intermediate time scale (weeks to months), and includes fleet assignment and timetable design. Real time *conflict resolution* and planning is used to handle unexpected events during operation (within minutes or hours). A finalised plan, taking into account all the objectives and constraints of the three planning levels, is called a timetable. By definition, a timetable is conflict-free and ensures smooth operations if it is followed strictly. Stochastic events in actual operation undermine this expectation and lead to resource contention. The effects of such situations are described in Section 2. In railway networks, real-time conflicts are resolved predominantly by human dispatchers using rule-based heuristics, with some examples given in Sinha et al. (2015). Pasquale et al. (2015) describe cases where dispatchers are helped by optimisation algorithms embedded in decision support systems.

The (physical and mathematical) size of railway networks, combined with their human-intensive nature, makes operations highly stochastic. The global effect of local decision-making is thus a topic of great interest. A frequent result of heuristic or locally optimal conflict resolution is the spread of *delays* in the network. In scheduled systems, delays are defined to be the difference between scheduled and actual times of each timetable event. Yuan and Hansen (2007), Sheu and Lin (2012) define *primary* delays as those caused by external events such as emergency maintenance or inclement weather. *Secondary* delays are the knock-on effects of primary delays, and are a function of the network infrastructure, the degree of slack in the schedules, and the operational strategy.

The first goal of this paper is to perform a fast prediction of delay propagation in a scheduled railway network, using data about infrastructure constraints, a timetable, and the applicable conflict resolution strategy. The procedure for achieving this goal is described in Section 3. The second goal is to use the delay predictions to evaluate schedule robustness. The proposed metrics quantify the ability of the combined (i) infrastructure, (ii) timetable, and (iii) conflict resolution method, to limit the propagation of delays. More description is provided in Section 4.

1.2. Related work

Prior literature gives several perspectives on the issue of delay and robustness in transportation networks in general, and railways in particular. These studies can be functionally classified into two parts. One set of studies focusses on the generation of robust timetables/schedules using optimisation or heuristic techniques. A second set of studies analyses the robustness of the schedules, and attempts to model the propagation of delays. However, these approaches either impose special conditions on the timetables to be analysed (for example, periodicity), or require a large number of computationally intensive simulations to provide stochastic results. The time required to input the topology of the underlying networks is an added disadvantage.

1.2.1. Robust scheduling: Robust scheduling in a timetable optimisation framework is a common theme in prior studies. Corman et al. (2014) make the case for using optimisation techniques for proactive railway traffic management in the case of disruptions. The robustness of timetables to delays is evaluated using microsimulation and alternative graphs. While the inputs are stochastic, delay propagation within the model is deterministic. Additionally, a large amount of detail about the infrastructure is required to be provided, in order to carry out the simulation. Kroon et al. (2008) provide a stochastic optimisation model for developing robust railway timetables. They focus on the provision of slack time within the schedule, in order to limit the propagation of knock-on delays. The approach uses the Periodic Event Scheduling model (PESP), making it better suited to cyclic timetables.

On the other hand, Fischetti et al. (2009) consider aperiodic timetables in European railways, where this property is typical of heavy-traffic, long-distance corridors. The focus is on fast computation of robust, near-optimal timetables. Using a mix of linear programming and stochastic optimisation, robustness is 'trained' into the timetable by ad-hoc learning methods. However, it is not clear how this approach would be used to model the propagation of knock-on delays. Schedule recovery after disruptions is related to robust scheduling. Meng and Zhou (2011) focus on recovery of train schedules while minimising additional delays. They consider a single-track rail line and a single type of train. By contrast, the current paper addresses the impact of different operating rules, with a heterogeneous mix of train classes and infrastructure.

1.2.2. Delay modelling in railways: Activity graphs are commonly used for delay modelling in railway networks, as described in Buker and Seybold (2012) and Lorek et al. (2011). These graphs work with a logical succession of prerequisite events in the network. A large amount of microscopic detail about the network infrastructure and operating logic is required before such analyses can be carried out. Additionally, several simulation runs need to be completed in order to produce stochastic delay estimates with such methods. For example, several studies reviewed in Corman and Meng (2015) model the effect of stochastic disturbances on railway timetables using deterministic simulators. They evaluate several scenarios with multiple (order of 10^3) runs for computing stochastic results.

While analytical approaches are faster in terms of computation, they are more difficult to implement. Other restrictions imposed by such methods include difficulties in changing the predefined order of trains, and a limitation on the probability distributions applicable for modelling input delays. Kecman et al. (2015) use Bayesian networks to model the propagation of delays. Their focus is on inference using real-time events. Accordingly, the computational procedure is based on specific orders of events, and the prediction horizon is relatively short (60 minutes).

The use of max plus algebra does allow analytical modelling of delays, as explained in Goverde (2007) and Goverde (2010). However, max plus algebra is better suited to cyclic (periodic) timetables than to more general problems. It too requires detailed information about the network infrastructure and scheduling rules. Yuan (2006) considers delay propagation at single stations and uses multiple theoretical distributions to model delays. The station is modelled in microscopic detail, and the delay probability calculations are dependent on prerequisite events similar to event graphs. Studies such as Meester and Muns (2007) and Yuan (2006) use specified distributions for modelling delays, limiting their general application. In contrast to analytical methods, simulation studies such as Murali et al. (2010) and Siefer and Radtke (2006) generate approximate relationships between travel time delay, train mix, and network topology. These approaches are difficult to generalise to multiple railway networks, and indeed to generic transportation networks.

1.2.3. Robustness evaluation of railway timetables: Few studies have directly addressed the robustness of a railway schedule as a quantity to be evaluated. A major reason for the sparsity is the difficulty of quantifying robustness in this context. Salido et al. (2008) define a robustness degree for railway timetables, and focus on various ways in which temporal slack can be provided between trains. The issue of spatial slack is not addressed: trains that concurrently occupy proximal resources may interact even if there is no direct overlap in the timetable. Dewilde (2014) focus on the 'real weighted transfer time' of passengers as a measure of robustness. Landex and Jensen (2013) propose measures for infrastructure and timetable complexity at specific stations, but these are evaluated without reference to the actual operations of trains and the consequent delays.

Andersson (2014) proposes the use of Robustness in Critical Points (RCP), computed by the sum of time margins between successive trains at critical points. A robust timetable is defined as a timetable *in which trains should be able to keep their originally planned train slot despite small delays and without causing unrecoverable delays to other trains.* Similarly to Salido et al. (2008), the focus is on a static evaluation of the temporal overlap between trains. Neither study attempts to predict operational punctuality when the timetable is put into practice. The object of analysis is only the timetable quality, which forms a part of tactical planning. The close connection with infrastructure (strategic planning) and dispatch rules (conflict resolution) is not taken into account.

1.2.4. Other transportation domains: The methods of transportation network analysis defined in generic contexts, as in Cascetta (2009), can only partially be applied to railways. This is because trains are more tightly coupled to the infrastructure (railway tracks) than in other contexts such as air transportation, shipping, or roadways. The constraints on scheduling, and their implications for delay propagation, are more pronounced in railways. For example, slack is provided in airline flight schedules for delay management, as described in AhmadBeygi et al. (2008)

and Xu et al. (2005). This slack is estimated based on the characteristics of the origin/destination airports, as opposed to the interaction between concurrently scheduled flights. By contrast, this paper uses a stochastic modelling approach that is similar in principle to Khadilkar (2013), which focusses on the airport surface as a network in which interactions may occur. The taxiways on an airport surface impose tighter constraints on aircraft movement than flight corridors, making this system closely analogous to a railway network.

1.3. Contributions of this work

Prior literature primarily focusses on (i) robust scheduling using optimisation techniques, and (ii) robustness evaluation using stochastic simulation or timetable slack computation. By contrast, this paper explicitly recognises that *robustness is a function of all three levels of planning*: strategic (infrastructure), tactical (timetable), and operational (conflict resolution). The basis for measuring robustness is a method for *fast stochastic prediction* of delay propagation without simulation. Published timetable data are used by the algorithm to define train routes and network connectivity. Basic station capacity information is used to estimate the probability of delay propagation. The algorithm explicitly recognises the possibility of conflict not only because of temporal overlap (the traditional definition of slack), but also spatial overlap between trains (concurrently scheduled at the same station on parallel tracks). The network topology does not have to be programmed explicitly; it is built automatically using the timetable. Delay estimates are produced in the form of probability distributions for each train at each station on its journey.

The delay propagation algorithm is run with different initial conditions to evaluate two proposed *schedule robustness metrics*. These mutually complementary metrics address both the fates of

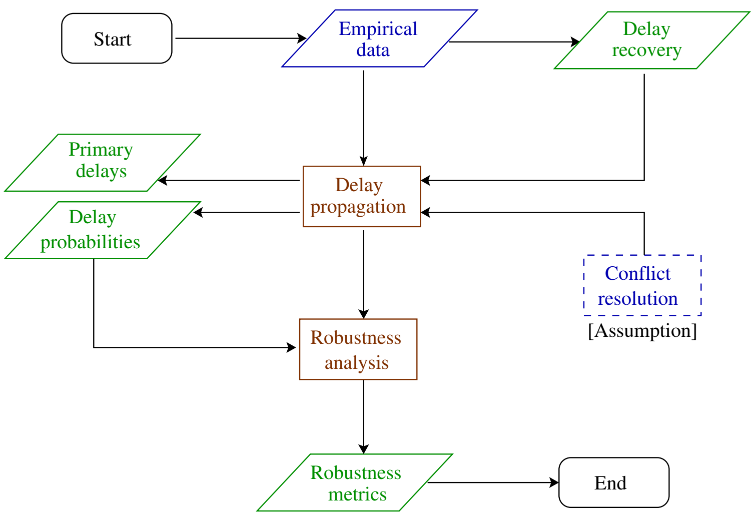

Figure 1 **Logical flow of the paper. The methodological components described in Section 3 and Section 4 are shown in the center. The conflict resolution strategy is to be provided as an input.**

individual delayed trains, and the spread of knock-on delays in the network. They represent a tradeoff between the objective of recovering delay for a given train, and the objective of limiting the impact on the rest of the network. The flow of the paper is depicted in Figure 1, with the two methodological components (delay propagation and robustness measurement) shown in the center.

The present work includes a framework for representing the dispatch strategy and infrastructure, in order to compute delays and robustness. The limited availability of data has necessitated some assumptions in the empirical evaluation. No specific operational details for Indian Railways were available, such as exact dispatch strategies or realistic slack times. These values were estimated from empirical data. A crowd-sourced comprehensive description of Indian Railway operations, provided by IRFCA (2016), was also used as a guide.

2. Preliminaries

This section describes an initial analysis of railway operations from a portion of the Indian Railway network. This analysis is used as an input to the delay propagation model.

2.1. Description of empirical data set

Empirical data for 15 days of operations (Nov 21 to Dec 05, 2014) were downloaded for the Mumbai-New Delhi railway line from a public source, India Rail Info (2014). The railway line extends over 1,400 km and 176 stations, including junction stations connecting to other portions of the nationwide system. The entire length is a conventional double-track, with interactions between opposing trains limited to stations. Figure 2 shows the time table for one of the 15 days in the empirical data. Mumbai and New Delhi stations form the termini for this particular line. Vadodara, Ratlam and Kota are junctions where trains branch towards other parts of the Indian network.

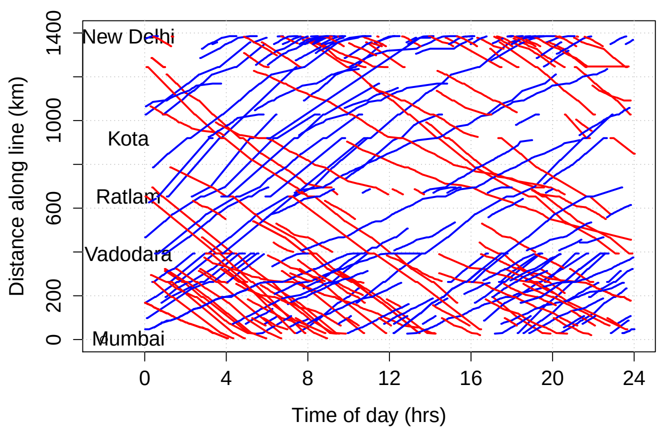

Figure 2 Sample schedule of trains for one day. The x-axis denotes time of day, while the y-axis denotes distance. Blue lines (heading upwards) denote trains towards New Delhi, while red lines (heading downwards) denote trains towards Mumbai.

The complete data set contains 3,988 train instances, with a total of 38,112 station arrival/departure records. It only covers scheduled passenger trains, and does not include freight train movements. Each record lists scheduled and actual arrival/departure times at the station as well as metadata about the train, at a time resolution of 1 minute. Supplementary information about the infrastructure, such as the number of tracks at each station as well the number of tracks between each pair of stations, was obtained separately. The metadata include information such as the train class, which is mapped to one of two static priorities in this paper. The mapping is based on well known information about the types of trains that receive resources with high priority. Typically, such trains run long distances at high speed, and stop at few stations. The number of priority levels in actual operation is unclear, but is likely to be higher than two. However, the highest priority trains run at low frequency, and form only a small portion of the data set. It is difficult to analyse their operations reliably. Therefore, this paper only classifies trains into two priority levels. High priority trains form 37% of the data set.

2.2. Delay recovery in empirical data

The runtime between two stations for most trains is larger than the minimum possible time, and is frequently utilised for reducing the impact of previous delays. In the absence of specific information regarding slack in the schedule, this study estimates the delay recovery potential from empirical data. In this study, delay recovery instances were identified based on two criteria: (i) there should be a reduction in delay between successive stations for a given train, and (ii) the residual delay should still be non-zero positive. The second criterion ensures that only those instances are selected, where the slack was fully utilised for recovery.

A scatter plot of the identified instances is shown in Figure 3. The reduction in delay (in minutes) is plotted against the distance covered by the train between successive stations. It is desirable to use a simple representative quantity for use in the delay propagation model. The slope of the linear fit passing through the origin is chosen to represent the recovery potential in the network being considered. The empirical recovery value (0.13 minutes/km) is later incorporated into the delay probability computation procedure. The correlation coefficient between the variables plotted in Figure 3 is a modest 0.55, which suggests a useful future study of better non-linear fits to the data.

2.3. Basic definition of modelling framework

The input to the delay propagation model is a published timetable that includes all the trains that will operate within the network, their routes (stations visited in order), and the scheduled times of arrival and departure at each station. It is important to note that reliable delay prediction requires the timetable to be *complete*; that is, it should not omit any concurrent traffic that may compete for the same infrastructure resources. The empirical data set used in this paper includes

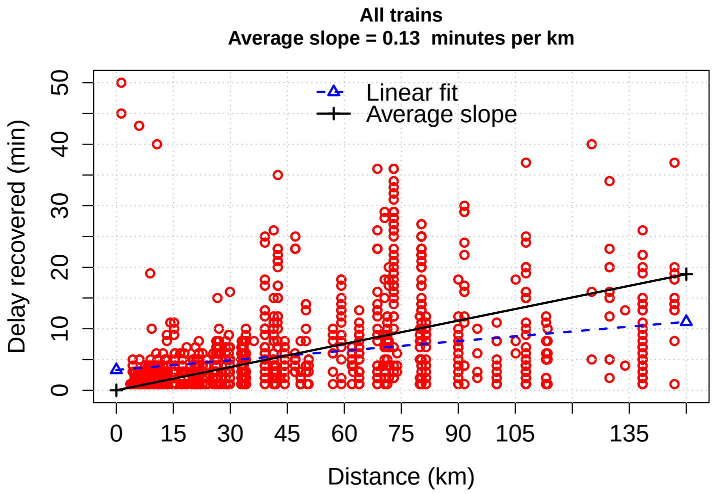

Figure 3 **Scatter plot of all delay recovery instances (all train priorities) in the empirical data set.**

only passenger trains (no freight trains). However, the delay and robustness predictions should not be strongly affected by the lack of freight data, because: (i) regular freight services in India are provided gaps in the passenger timetables, so that their operations can be temporally isolated, and (ii) freight trains are assigned the lowest priority in the Indian context, and wait for passenger trains to pass. They do not occupy main lines when they are held up due to passing traffic - there are designated holding areas provided at major junction stations.

Supplementary inputs to the model are the infrastructure constraints and the control algorithm used for resolving any resource conflicts. Static priorities, if applicable, should also be given. The output is a delay probability distribution for each train as it progresses along its route. In this paper, tracks at stations are called *loops*, to differentiate them from tracks between stations. Delays are propagated by interactions at stations, while some delay is recovered between two stations.

Let us denote[1] the set of all stations by $\mathcal{S}$, the set of all loops at station $s \in \mathcal{S}$ by $\mathcal{L}_s$, and the set of all trains by $\mathcal{TR}$. The model is easiest to visualise in a discrete time setting, and this will be used throughout the paper. Let us denote the set of discrete times by the ordered set $\mathcal{N}$. Delays are also discrete, and the delay distributions are probability mass functions. The delay distribution for train tr at station s is represented by a stochastic vector $\pi_{tr}(s)$, with each element equal to the probability of the train carrying the corresponding amount of delay at s. Successive elements in $\pi_{tr}(s)$ need not have the same resolution as the elements of $\mathcal{N}$. Instead, each element in $\pi_{tr}(s)$ represents a user-defined $\Delta \geq 1$ time units of delay. Further, $\pi_{tr}(s)$ is truncated at some horizon $k_{\max}\Delta$ for computational purposes. The horizon is set larger than any reasonably expected delays

[1] In this paper, capitalised calligraphic symbols (such as $\mathcal{S}$ or $\mathcal{TR}$) denote collective variables (sets or lists). Probability vectors are denoted by variations of π, trains by tr, stations by s. Train indices follow tr and generic ones are denoted by i, while station indices follow s and generic ones are denoted by j. Delay-related indices are denoted by k.

in the network. The delay vector is tagged to the scheduled time of arrival of train tr at station s. If two trains simultaneously wish to use the same loop at a station, one of them is assumed delayed by Δ time units. The probability of such an event is known *a priori*, as explained below.

2.4. Rules for conflict resolution

In several rail networks, including Indian Railways, trains are assigned to a fixed priority class. High priority trains have precedence over lower priority trains if there is a resource conflict. Let us assume that $\rho_{s\mathrm{j}}(tr1, tr2)$ is the probability that $tr1$ will be delayed because of a resource conflict with $tr2$, *conditional* on them arriving concurrently at $s\mathrm{j}$. It thus denotes the probability of a spatial interaction between two trains, given that there is already a temporal overlap. It is natural to assume that if $tr1$ has precedence over $tr2$, then $tr1$ cannot be delayed by $tr2$ and $\rho_{s\mathrm{j}}(tr1, tr2)$ is equal to 0. If the two trains have equal priority, or if $tr2$ has precedence over $tr1$, then $\rho_{s\mathrm{j}}(tr1, tr2)$ is based on the conditional probability of a resource conflict, given that the trains arrive concurrently. The conflict probability is based on several operating factors and infrastructure constraints. The assignment of values to $\rho_{s\mathrm{j}}(tr1, tr2)$ should be carried out on a case to case basis. In the absence of specific information, a simple assumption is made in this paper. It is assumed that trains are assigned to loops at a station with a uniform probability distribution. Therefore, the probability of two trains being assigned to the same loop is Bernoulli with parameter $\frac{1}{|\mathcal{L}_{s\mathrm{j}}|}$. The specific relations are as follows.

$$\rho_{s\mathrm{j}}(tr1, tr2) = \begin{cases} \frac{1}{|\mathcal{L}_{s\mathrm{j}}|}, & \text{if } tr2 \text{ has priority over } tr1 \\ \frac{1}{2}\frac{1}{|\mathcal{L}_{s\mathrm{j}}|}, & \text{if } tr2, tr1 \text{ have equal priority} \\ 0, & \text{if } tr1 \text{ has priority over } tr2 \end{cases} \tag{1}$$

Note that alternative priority schemes also exist in practice. Andersson (2014) outlines one that prioritises on-time trains. Moreover, no single dispatch rule may be used across an entire network and in all contexts. The simplified rules presented in this paper are useful for illustration.

3. Delay propagation methodology

This section describes an algorithm that analyses a reference timetable, in combination with a user-defined set of parameters, and produces stochastic estimates of delay for each train. The mechanics of the algorithm are demonstrated by predicting delays that occur during the normal course of operation (without considering primary delays). Section 4 shows how this algorithm can use primary delays for evaluating schedule robustness.

3.1. Initialisation and parameter setting

The principal input to the delay propagation algorithm is the schedule of each train as it visits a succession of stations on its route. Corresponding to each train-station pair $\{tr\mathrm{i}, s\mathrm{j}\}$ is a delay probability vector $\pi_{tr\mathrm{i}}(s\mathrm{j})$. Let us denote by $P_{tr\mathrm{i}}(s\mathrm{j}, k\Delta)$ the probability of train $tr\mathrm{i}$ arriving with a delay of $k\Delta$ units ($k \in [0, k_{\max}]$) at station $s\mathrm{j} \in \mathcal{S}$. Then $\pi_{tr\mathrm{i}}(s\mathrm{j})$ is the $(k_{\max}+1)$-sized vector of probabilities $P_{tr\mathrm{i}}(s\mathrm{j}, k\Delta)$.

$$\pi_{tr\mathrm{i}}(s\mathrm{j}) = [P_{tr\mathrm{i}}(s\mathrm{j},0), P_{tr\mathrm{i}}(s\mathrm{j},\Delta), \ldots, P_{tr\mathrm{i}}(s\mathrm{j},k_{\max}\Delta)]^T,$$
$$\text{and} \sum_{k=0}^{k_{\max}} P_{tr\mathrm{i}}(s\mathrm{j},k\Delta) = 1.$$

In this section, we assume that each train $tr\mathrm{i}$ starts on time at its first station $s[\mathrm{i}, ini]$, that is, $\pi_{tr\mathrm{i}}(s[\mathrm{i}, ini]) = [1,0,0,\ldots,0]^T$. Furthermore, let us denote the scheduled arrival time of $tr\mathrm{i}$ at $s\mathrm{j}$ by $n_{sch,\mathrm{i},\mathrm{j}} \in \mathcal{N}$. Using the definition given above, it follows that $P_{tr\mathrm{i}}(s\mathrm{j}, k\Delta)$ is the predicted probability of $tr\mathrm{i}$ actually arriving at $s\mathrm{j}$ at time $(n_{sch,\mathrm{i},\mathrm{j}} + k\Delta)$. Two trains $tr1$ and $tr2$ are said to be in potential conflict, if they both have non-zero probabilities of being at $s\mathrm{j}$ for at least one common time period. In such an eventuality, the new values of $\pi_{tr1}(s\mathrm{j})$, $\pi_{tr2}(s\mathrm{j})$ are computed from the respective values at previous stations, as follows.

3.2. Delay prediction algorithm

The delay probability computation is carried out in discrete steps for each station visited by a given train. Two computational components are described before addressing the full algorithm. Assuming that a pair of trains $tr1$ and $tr2$ are scheduled at overlapping times at a station $s\mathrm{j}$, Section 3.2.1 describes the knock-on delay computation for this scenario. Each train may recover some delay during its journey between two stations. The effect of recovery on the computational procedure is described in Section 3.2.2. For notational clarity, note the following points.

- There are two delay probability vectors attached to each station $s\mathrm{j}$ for a given train $tr\mathrm{i}$. The first (incumbent) vector, denoted by $\pi'_{tr\mathrm{i}}(s\mathrm{j})$, is computed from the vector at the previous station, after accounting for recovery between stations. The second (updated) vector, denoted by $\pi_{tr\mathrm{i}}(s\mathrm{j})$, is computed from $\pi'_{tr\mathrm{i}}(s\mathrm{j})$ after accounting for all interactions between $tr\mathrm{i}$ and other trains at $s\mathrm{j}$.
- Entries of $\pi_{tr\mathrm{i}}(s\mathrm{j})$ are denoted by $P_{tr\mathrm{i}}(s\mathrm{j}, k\Delta)$, while those of $\pi'_{tr\mathrm{i}}(s\mathrm{j})$ are denoted by $P'_{tr\mathrm{i}}(s\mathrm{j}, k\Delta)$.
- The generic station that follows $s\mathrm{j}$ on the journey of a given train is denoted by $s\mathrm{j}^*$.

3.2.1. Pairwise interaction at stations: For the sake of illustration, assume that trains $tr1$ and $tr2$ are scheduled to arrive at $s\mathrm{j}$ at the same time (no offset). They are tagged with incumbent values of $\pi'_{tr1}(s\mathrm{j})$ and $\pi'_{tr2}(s\mathrm{j})$, propagated from their respective prior stations. Train $tr1$ will be delayed by $tr2$ if all of the following conditions are met: (i) $tr1$ arrives at $s\mathrm{j}$ with delay $k\Delta$: $P'_{tr1}(s\mathrm{j}, k\Delta)$, (ii) $tr2$ arrives at $s\mathrm{j}$ with the same delay $k\Delta$, hence at the same time as $tr1$ (*temporal* overlap): $P'_{tr2}(s\mathrm{j}, k\Delta)$, and (iii) $tr2$ conflicts with $tr1$ for the same loop at $s\mathrm{j}$, given that they arrive at the same time (*spatial* overlap): $\rho_{s\mathrm{j}}(tr1, tr2)$. The probability of $tr1$ retaining $k\Delta$ delay after this interaction is reduced by $P'_{tr1}(s\mathrm{j}, k\Delta) \cdot P'_{tr2}(s\mathrm{j}, k\Delta) \cdot \rho_{s\mathrm{j}}(tr1, tr2)$. Figure 4 graphically illustrates

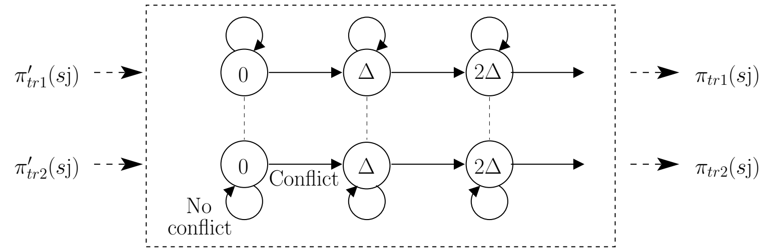

Figure 4 Graphical illustration of delay probability vector update at a station.

this interaction, while Equation (2) states it mathematically. Each delay state in π'_{tr1} has a self transition (no conflict), an incoming transition (conflict in lower delay state) and an outgoing transition (conflict in this delay state). The only exceptions are the maximum delay state $k_{\max}\Delta$ (no outbound transition), and the no-delay state (no incoming transition). Defining $\mathbb{1}_{a>b}$ to be 1 if $a>b$ and 0 otherwise, the update relation for state $k\Delta$ is,

$$\begin{aligned} P_{tr1}(s\text{j},k\Delta) = & \overbrace{P'_{tr1}(s\text{j},k\Delta)}^{\text{Incumbent value}} + \mathbb{1}_{k>0}\overbrace{P'_{tr1}(s\text{j},(k-1)\Delta)\,P'_{tr2}(s\text{j},(k-1)\Delta)\,\rho_{s\text{j}}(tr1,tr2)}^{\text{Conflict: incoming from previous state}} \\ & -\mathbb{1}_{k<k_{\max}}\underbrace{P'_{tr1}(s\text{j},k\Delta)\,P'_{tr2}(s\text{j},k\Delta)\rho_{s\text{j}}(tr1,tr2)}_{\text{Conflict: outgoing to next higher state}}. \end{aligned} \tag{2}$$

The vector for $tr2$ will have exactly similar computations using $\rho_{s\text{j}}(tr2,tr1)$ instead of $\rho_{s\text{j}}(tr1,tr2)$. Note that the value of Δ is assumed to be large enough that trains arriving in different time periods do not cause delays to each other.

3.2.2. Recovery between stations: The empirical characteristics of delay recovery in Section 2.2 showed a mean recovery rate of 0.13 minutes/km. Let us represent the generic recovery rate by λ. If the journey between two successive stations sj and sj* for a given train tri is of length l_{jj^*} km, the delay reduces in all states except $k=0$, and the expected reduction is thus $\lambda l_{\text{jj}^*}(1-P_{tr\text{i}}(s\text{j},0))$. The delay vector $\pi'_{tr\text{i}}(s\text{j}^*)$ is computed using Equation (3). Note that the use of $\min(\lambda l_{\text{jj}^*},\Delta)$ ensures that the updated delay vector remains stochastic, and that $0\le P'_{tr\text{i}}(s\text{j}^*,k\Delta)\le 1$. The right hand side of Equation (3) is non-negative since all four terms are non-negative. At the same time, it is upper bounded by $[P_{tr\text{i}}(s\text{j},k\Delta)+P_{tr\text{i}}(s\text{j},(k+1)\Delta)]$, which is at most equal to 1.

$$P'_{tr\text{i}}(s\text{j}^*,k\Delta) = \overbrace{P_{tr\text{i}}(s\text{j},k\Delta)\left[1-\mathbb{1}_{k>0}\frac{\min(\lambda l_{\text{jj}^*},\Delta)}{\Delta}\right]}^{\text{Fraction left in current delay state}} + \mathbb{1}_{k<k_{\max}}\overbrace{P_{tr\text{i}}(s\text{j},(k+1)\Delta)\frac{\min(\lambda l_{\text{jj}^*},\Delta)}{\Delta}}^{\text{Moved from next higher delay state}}. \tag{3}$$

The following manipulation shows that Equation (3) achieves the required reduction in delay.

$$\begin{aligned} \mathbb{E}[\text{Recovery}] &= \sum_{k=0}^{k_{\max}} k\Delta\left[P_{tr\text{i}}(s\text{j},k\Delta)-P'_{tr\text{i}}(s\text{j}^*,k\Delta)\right] \\ &= \min(\lambda l_{\text{jj}^*},\Delta)\sum_{k=0}^{k_{\max}} k\left[\mathbb{1}_{k>0}P_{tr\text{i}}(s\text{j},k\Delta)-\mathbb{1}_{k<k_{\max}}P_{tr\text{i}}(s\text{j},(k+1)\Delta)\right] \end{aligned}$$

$$= \min(\lambda l_{\mathrm{jj}^*}, \Delta)\,[P_{tr\mathrm{i}}(s\mathrm{j},\Delta) - P_{tr\mathrm{i}}(s\mathrm{j},2\Delta) + 2\,P_{tr\mathrm{i}}(s\mathrm{j},2\Delta) - 2\,P_{tr\mathrm{i}}(s\mathrm{j},3\Delta)\ldots]$$
$$= \min(\lambda l_{\mathrm{jj}^*}, \Delta)\sum_{k=1}^{k_{\max}} P_{tr\mathrm{i}}(s\mathrm{j},k\Delta) = \min(\lambda l_{\mathrm{jj}^*}, \Delta)(1 - P_{tr\mathrm{i}}(s\mathrm{j},0)).$$

3.2.3. Combined implementation: The update procedure given in Section 3.2.1 works equally well for trains that are not scheduled at sj at the exact same time, but with an offset δ. The probability of the trains arriving simultaneously is equal to the latter train arriving with a delay $k\Delta$, and the earlier train with a delay $(k\Delta + \delta)$. Should there be more than two trains scheduled concurrently at a station, the combined procedure given in Algorithm 1 naturally picks them up in chronological followed by priority order. Note that the algorithm depends only on interactions at common resources - in this case, only at stations. No specific definition of network topology is required, as long as a list of stations visited by train tri is available.

Algorithm 1 Predicting delay distributions $\pi_{tr\mathrm{i}}(s\mathrm{j})$

```
Input: List $\mathcal{E}$ of events $e_m = \{m, n_{sch,\mathrm{i,j}}, tr\mathrm{i}, s\mathrm{j}\}$, $m \in [1, \mathrm{size}(\mathcal{E})]$, $n_{sch,\mathrm{i,j}}$ = scheduled time of $tr$i
  at $s$j. The list $\mathcal{E}$ is sorted first by $n_{sch,\mathrm{i,j}}$ and then by priority of $tr$i.
Initialise $\pi_{tr\mathrm{i}}(s\mathrm{j})$ for each event to $[1, 0, \ldots, 0]^T$, or some other appropriate value
Define sub-list $\mathcal{E}_m$ as all events $e^\# = \{m^\#, n^\#, tr^\#, s\mathrm{j}\}$ s.t. $m^\# > m$ and $n^\# \le n_{sch,\mathrm{i,j}} + k_{\max}\Delta$
for $m \in [1, \mathrm{size}(\mathcal{E})]$,
   for each pair of events $(e_m, e^\#)$ where $e^\# \in \mathcal{E}_m$,
      Update $\pi_{tr\mathrm{i}}(s\mathrm{j})$, $\pi_{tr\#}(s\mathrm{j})$ according to Equation (2)
      if $\pi_{tr\mathrm{i}}(s\mathrm{j}) \ne \pi'_{tr\mathrm{i}}(s\mathrm{j})$,
         Propagate $\pi_{tr\mathrm{i}}(s\mathrm{j})$ to all future stations for $tr$i, including recovery from Equation (3)
      end if
      if $\pi_{tr\#}(s\mathrm{j}) \ne \pi'_{tr\#}(s\mathrm{j})$,
         Propagate $\pi_{tr\#}(s\mathrm{j})$ to all future stations for $tr^\#$, including recovery from Equation (3)
      end if
   end for
end for
Output: Predicted delay distributions $\pi_{tr\mathrm{i}}(s\mathrm{j})\ \forall e_m \in \mathcal{E}$
```

A simple hypothetical illustration of the algorithm is shown in Figure 5 for one pair of trains. The x-axis in the figure is time (marked by values from n_1 to n_5), and the y-axis is the geographical location of three successive stations ($s1$ to $s3$). The scheduled path of two trains $tr1$ and $tr2$ is marked using a dashed line and a solid line respectively. Train $tr1$ moves from $s1$ through $s2$ to $s3$ as time progresses, while $tr2$ moves in the opposite direction. Both trains are scheduled to arrive at

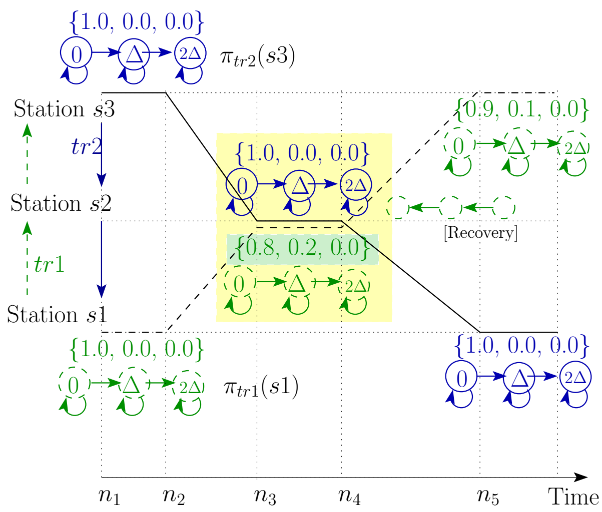

Figure 5 **Illustration of interaction between two trains, with hypothetical delay probability values. Train $tr1$ (dotted line) is travelling from Station $s1$ to $s3$ via $s2$, while train $tr2$ (solid line) is travelling in the opposite direction. Both trains have a scheduled time of arrival n_3 at $s2$, and scheduled time of departure n_4. It is assumed that $tr2$ has priority over $tr1$, and π_{tr2} remains unchanged. Vector π_{tr1} changes to a new hypothetical value, and this change is propagated to $s3$.**

$s2$ at the same time. Assuming that $tr2$ has precedence over $tr1$, the vector π_{tr2} remains unchanged at $[1,0,0]^T$ in the journey from $s3$ to $s1$. On the other hand, π_{tr1} is modified at $s2$. This updated value is propagated to all future stations on its journey (in this case, $s3$), after going through the delay recovery procedure. The procedure to handle an entire timetable is given in Algorithm 1. Note that if $k_{\max}\Delta \geq (n_5 - n_1)$ in Figure 5, the vectors π_{tr1} and π_{tr2} will also overlap at stations $s1$ and $s3$. Algorithm 1 will make the necessary update to $\pi_{tr1}(s1)$ and propagate an updated value to $\pi_{tr1}(s2)$ and $\pi_{tr1}(s3)$. This new value will be used for the subsequent computation at $s2$.

Since $\mathcal{E}$ is a finite list, Step 3 ensures finite termination of Algorithm 1. The number of updates within each iteration of Step 4 is upper bounded by $2 \cdot \text{size}(\mathcal{E}_m)$, since $\text{size}(\mathcal{E}_m)$ pairs of trains are considered for each m. Interactions between multiple trains at the same station are handled by Algorithm 1 in the order of arrival time and priority, as per Step 3.

3.3. Validation with empirical data

The quality of comparison between the delay predictions and the empirical data depends on the values of parameters Δ and $k_{\max}$. These values should be tuned based on context. The case study presented in this paper relates to long distance passenger trains in Indian Railways. The value of Δ is approximately equal to the time for which a train 'blocks' a station loop. The approximate time taken to move from the approach signal and come to a halt at the station is taken to be 2 minutes. The average halt for a long distance train is 5 minutes. This is followed by another 2 minute interval for train to accelerate from its halt and pass the departure signal. The whole process thus takes approximately 9 minutes. Therefore, a value of $\Delta = 10$ minutes is chosen for most test cases in this study. Smaller values of Δ, which may be more relevant in other contexts, are also demonstrated briefly.

The number of non-zero delay states used for validation are given by $k_{\max} = 8$. The various delay states are referred to as $0-10$ minutes for $k=0$, and so on up to $80-90$ minutes for $k = k_{\max}$. The algorithm is executed for the entire data set introduced in Section 2.1. Each train is assigned to one of two priority classes. In order to ensure that the full set of interactions is considered for each train, an intermediate period of 10 out of the available 15 days is chosen for comparison.

A comparison between the empirical data and the prediction from the algorithm is shown in Figure 6. Each predicted data point in the figure (probability of $k\Delta$ delay) is computed by averaging $P_{tri}(s\mathrm{j}, k\Delta)$ over all $tr\mathrm{i}$ and $s\mathrm{j}$. The equivalent empirical data point is the relative fraction of trains with delays in the interval $[k\Delta, (k+1)\Delta)$ in the empirical data set. We see that the predicted delays broadly match empirical delays, with a maximum error of 3%. Figure 7 shows the same comparison split by train priority. We see that the prediction for $k=0$ is better for priority 1 (high priority) trains than for priority 2 (low priority) trains. The difference is partly explained by the presence of primary delays in the empirical data, which are not related to operational policies. Once primary delays are introduced, low priority trains are disproportionately affected by knock-on delays.

The delay propagation procedure was implemented in R, on an Intel i5 4×2.4 GHz CPU running Ubuntu 14.10. The computation time for the algorithm with $\Delta = 10$ minutes is 5.88 minutes. For comparison, the algorithm was also run with $\Delta = 5$ minutes and $\Delta = 3$ minutes. The computation times in these cases were 6.27 minutes and 6.43 minutes respectively. The comparisons with empirical data in these two cases are shown in Figure 8. The match between the predicted and

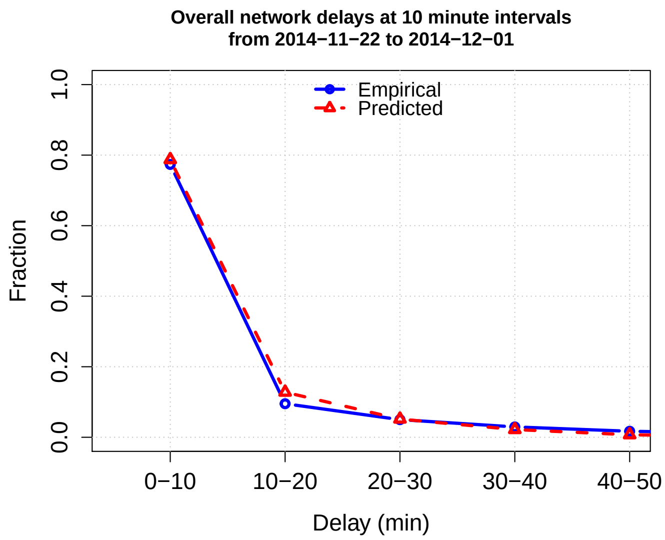

Figure 6 **Comparison of predicted and empirical delay distributions, aggregated over all trains.**

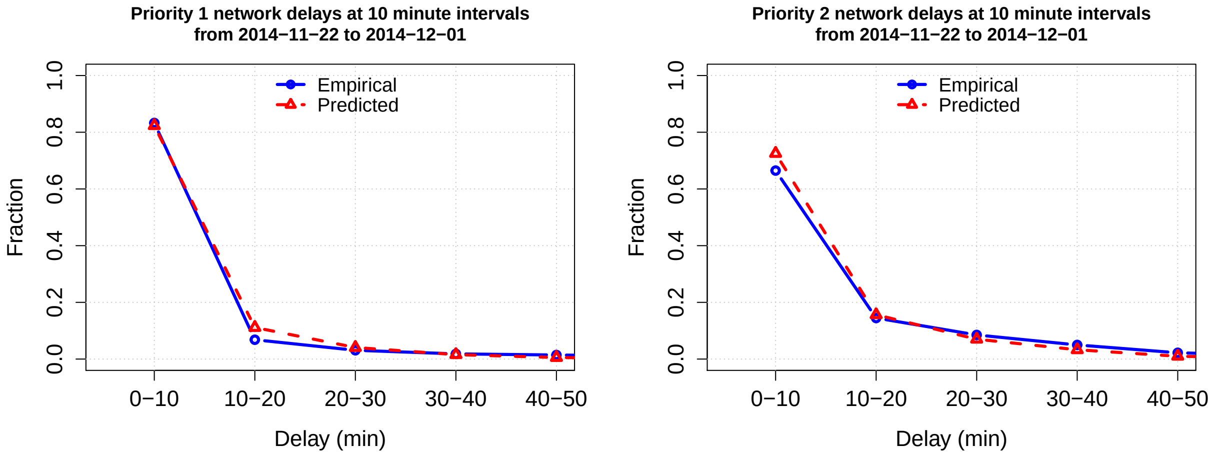

Figure 7 **Comparison of predicted and empirical delay distributions, split by train priority.**

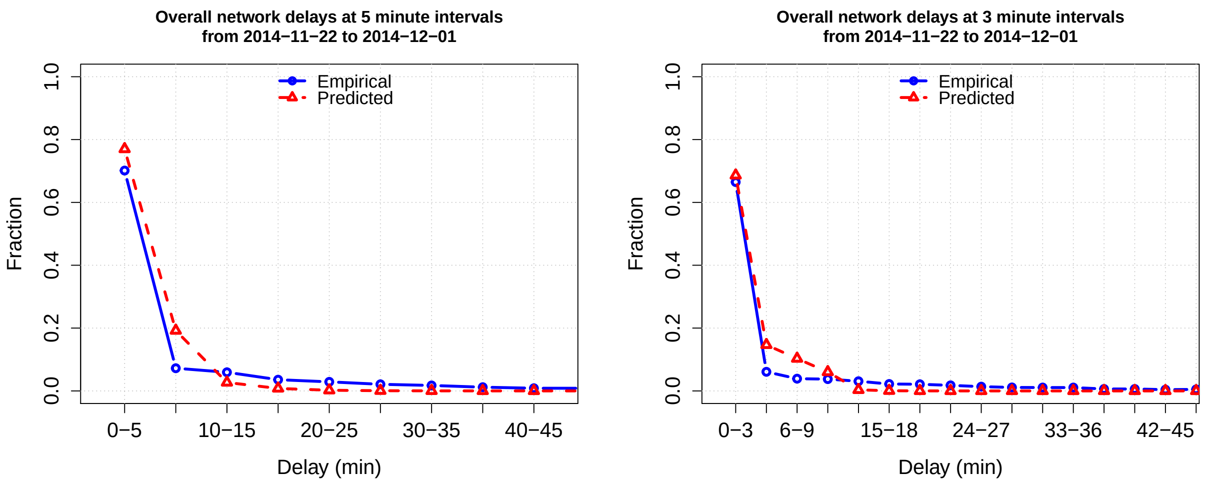

Figure 8 **Comparison of predicted and empirical delay distributions for $\Delta = 5$ minutes and $\Delta = 3$ minutes.**

empirical values is broadly similar to the $\Delta = 10$ case. Note that the computation times increase by relatively small amounts as Δ decreases. This is because the number of interactions between trains are driven by the horizon $k_{\max}\Delta$ (which has been kept constant at 90 minutes), as opposed to Δ. Since this number remains constant, the number of computational steps is also roughly constant.

3.4. Comparison of alternate conflict resolution strategies

The results presented in the previous discussion relate to the fixed priority conflict resolution scheme. Alternate schemes can be explored by a simple redefinition of $\rho_{sj}(tr1, tr2)$. Andersson (2014) notes that in Sweden, trains that are running on time preferentially receive resources. This implies that in case of conflict, the train that has lower delay gets higher priority. Complementary to this scheme is one where the train that is delayed gets higher priority. This latter scheme is analogous to one followed in air traffic management, as described in Khadilkar (2013). A comparison

Table 1 Likelihood of trains being delayed, under different conflict resolution strategies. The first and third strategy have similar results, with the bulk of all trains falling within the first and third likelihood buckets. On the other hand, the OnTimePreferred strategy exhibits significantly different behaviour. Note that P_0 is used as an abbreviated version of $P_{tri}(s\text{j}, 0)$.

Strategy	Delay likelihoods of various trains, according to $P_0 = P_{tri}(s\text{j}, 0)$			
	Very unlikely $P_0 = 1$	Unlikely $P_0 = 0.95 \pm 0.05$	Likely $P_0 = 0.7 \pm 0.2$	Very likely $P_0 = 0.25 \pm 0.25$
FixedPriority	**32%**	12%	**46%**	10%
OnTimePreferred	**61%**	02%	17%	**20%**
DelayedPreferred	**34%**	13%	**43%**	10%

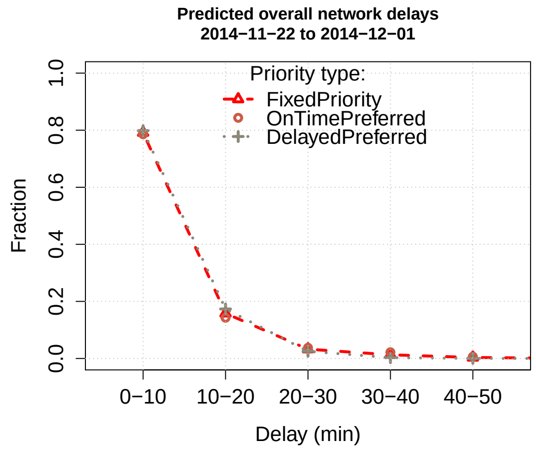

Figure 9 Comparison of predicted delays for three alternate conflict resolution strategies.

of the predicted delays as a consequence of all three conflict resolution strategies is shown in Figure 9. It is seen that the differences in resulting delays are minor. This is because the aggregate delay probabilities over all trains are similar. However, the distribution of delays across different trains is significantly different, as explained below.

Table 1 explores the first data point in Figure 9 (probability of 0−10 minute delay) in greater detail. All three strategies put the predicted fraction of trains with 0−10 minute delay at approximately 0.8. However, the trains that contribute to this value are different for the three strategies. Only 32% of trains satisfy $P_{tri}(s\text{j}, 0) = 1$ for the FixedPriority strategy, while 10% of trains have a very low probability of being on time ($P_{tri}(s\text{j}, 0) \approx 0.25$). The aggregate value of 0.8 is roughly composed of 32%×1 + 12%×0.95 + 46%×0.7 + 10%×0.25. On the other hand, for the OnTimePreferred strategy, a large number (61%) of trains satisfy $P_{tri}(s\text{j}, 0) = 1$. A relatively large portion of the aggregate value of 0.8 is composed of trains that individually have very small probabilities of being on time (20%×0.25). Once a train is delayed under this strategy, it quickly loses priority and accumulates additional delays.

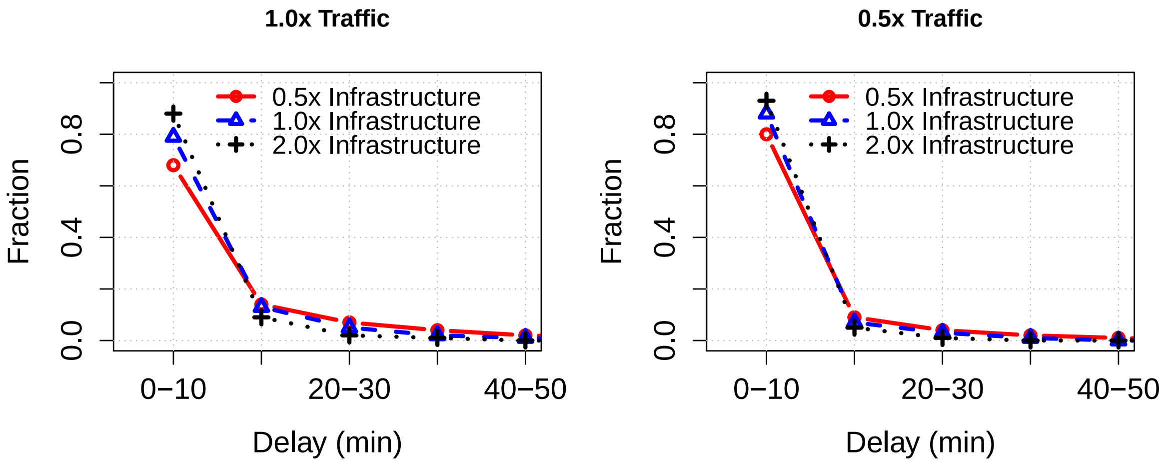

Figure 10 **Comparison of predicted delays for different infrastructure and traffic scenarios.**

3.5. Comparison of alternate infrastructure and timetable characteristics

Potential changes in the infrastructure and timetable characteristics also affect predicted delays. An increase in the number of loops available at stations would reduce the conflict probabilities according to Equation (1). Similarly, reducing the number of trains in the schedule would also reduce the number of interactions at stations. Both scenarios would reduce the coupling between different trains in the network. Reducing the station infrastructure or increasing the number of trains would increase the probability of conflict, and hence the magnitude of knock-on delays.

Figure 10 considers the effect of two different traffic levels and three different infrastructure levels on the predicted delays. The baseline case (equivalent to Figure 6) is 1.0x Traffic and 1.0x Infrastructure. Figure 10 [left] shows that if the infrastructure (number of loops at each station) is doubled, the fraction of trains running on time increases to 0.9. On the other hand, if the infrastructure is halved in comparison with current levels, this fraction reduces to 0.65. The trend is similar but less pronounced if there is only half as much traffic, as seen in Figure 10 [right].

4. Evaluation of schedule robustness

In this section, the delay propagation algorithm is used for evaluating two proposed metrics of schedule robustness. Prior literature is not in complete agreement as to a definition of robustness. However, it is clear that the definition should relate to the ability of the combined infrastructure, timetable, and conflict resolution strategy, to limit the propagation of primary delays in the form of knock-on delays. The primary delays observed in the empirical data set are described first. A useful characteristic of the way these delays propagate, is that $\pi_{tri}(s[\mathrm{i}, ini])$ undergoes a series of linear operations according to Algorithm 1. This property is used to define and evaluate the robustness of the network, using an *individual* metric and a *collective* metric.

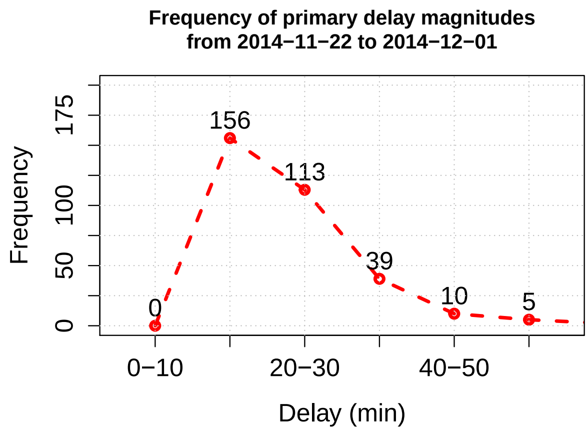

Figure 11 Magnitude of primary delays, shown according to frequency of observation.

4.1. Identification and properties of primary delays

The output of Algorithm 1 is a delay probability vector for each train at each station on its journey. This vector is based on the assumption that there are no primary delays in the network. The *likelihood* of an observed empirical delay of d minutes for $tr\mathrm{i}$ at $s\mathrm{j}$ is defined to be the corresponding probability value from the model, $P_{tr\mathrm{i}}(s\mathrm{j}, \lfloor \frac{d}{\Delta} \rfloor \Delta)$. As far as the predicted probability of an observed delay value remains high, the train can be said to be proceeding as expected. A sharp decrease in the likelihood of an observed delay, in successive stations on a train's journey, indicates a primary delay. This implies that a train which was travelling with delay values of high likelihood, is now experiencing delay values which were assigned low likelihood. The resulting frequency of various magnitudes of primary delay, as observed in 10 days of empirical data, is plotted in Figure 11. The number of identified instances represents approximately 1.5% of all train-station pairs, and 14% of all trains. The distribution of primary delays is used for generating initial conditions for the robustness computation.

4.2. Linearity of delay vector updates

Schedule robustness implicitly depends on the progression of delay in train journeys, as they move from their origin stations to destination stations. Algorithm 1 modifies the initial delay probability vector $\pi_{tr1}(s[1, ini])$ through a series of linear updates into the final (destination) vector $\pi_{tr1}(s[1, fin])$, as explained below. First, consider the interaction between a pair of trains at a station, as explained in Section 3.2.1. Assuming that the trains are scheduled to arrive at the same time ($\delta = 0$), the vector update according to Equation (2) can be written as follows.

$$\pi_{tr1}(s\mathrm{j}) = \begin{bmatrix} 1-\rho P'_{tr2}(0) & 0 & 0 & \dots \\ \rho P'_{tr2}(0) & 1-\rho P'_{tr2}(\Delta) & 0 & \dots \\ 0 & \rho P'_{tr2}(\Delta) & 1-\rho P'_{tr2}(2\Delta) & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \pi'_{tr1}(s\mathrm{j}) = [A_{tr2\to tr1,\mathrm{j}}]\, \pi'_{tr1}(s\mathrm{j}).$$

Since all entries of $\pi'_{tr2}(s\mathrm{j})$ are computed independently of $tr1$, $A_{tr2 \to tr1,\mathrm{j}}$ is a left stochastic matrix. Each train $tr\mathrm{i}$ will interact with $tr1$ through a stochastic matrix similar to this. Similarly, the delay recovery update described in Section 3.2.2 can be written as a linear update equation.

$$\pi'_{tr1}(s\mathrm{j}^*) = \begin{bmatrix} 1 & \frac{\min(\lambda l_{\mathrm{jj}^*}, \Delta)}{\Delta} & 0 & 0 & \cdots \\ 0 & 1 - \frac{\min(\lambda l_{\mathrm{jj}^*}, \Delta)}{\Delta} & \frac{\min(\lambda l_{\mathrm{jj}^*}, \Delta)}{\Delta} & 0 & \cdots \\ 0 & 0 & 1 - \frac{\min(\lambda l_{\mathrm{jj}^*}, \Delta)}{\Delta} & \frac{\min(\lambda l_{\mathrm{jj}^*}, \Delta)}{\Delta} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \pi_{tr1}(s\mathrm{j}) = [B_{tr1,\mathrm{j}\to\mathrm{j}^*}]\, \pi_{tr1}(s\mathrm{j}).$$

The delay vector at the final (destination) station can be written as a linear transform of the initial delay vector, as follows. We denote by $\mathcal{TR}[tr1, s\mathrm{j}]$ the set of trains that $tr1$ interacts with, at $s\mathrm{j}$.

$$\begin{aligned} \pi_{tr1}(s[1, fin]) &= \prod_{j \in \mathrm{route}(tr1)} \left[\overbrace{B_{tr1,\mathrm{j}\to\mathrm{j}^*}}^{\text{To next station } j^*} \underbrace{\left(\prod_{tr\mathrm{i} \in \mathcal{TR}[tr1, s\mathrm{j}]} A_{tr\mathrm{i}\to tr1,\mathrm{j}} \right)}_{\text{all other trains at station } s\mathrm{j}} \right] \pi_{tr1}(s[1, ini]) \\ &= [\mathcal{M}_{tr1}]\ \pi_{tr1}(s[1, ini]). \end{aligned}$$

As a product of stochastic matrices, $\mathcal{M}_{tr1}$ is also a stochastic matrix. It can be computed by taking the product as shown above. Alternatively, one may execute Algorithm 1 for $(k_{\max} + 1)$ linearly independent values of $\pi_{tr1}(s[1, ini])$, and solve the resulting linear system of equations with the corresponding values of $\pi_{tr1}(s[1, fin])$. Of the two types of matrices that form $\mathcal{M}_{tr1}$, the recovery matrices $B_{tr1,\mathrm{j}\to\mathrm{j}^*}$ depend (in this study) only on the distance travelled by $tr1$, and not on the train priority. On the other hand, $A_{tr\mathrm{i}\to tr1,\mathrm{j}}$ is driven by the delays that $tr1$ can experience as a result of conflicts with other trains. The higher the priority of $tr1$, the closer that $A_{tr\mathrm{i}\to tr1,\mathrm{j}}$ is to the identity matrix (no modification in predicted delays).

4.3. Individual robustness

The first of two proposed robustness metrics focusses on secondary delays for individual trains, when they themselves are subjected to primary delay. The delays are affected by the type of schedule, as explained below.

4.3.1. Periodic/shuttle services: Railway services are typically classified into one of two types, as described by Corman et al. (2014). They are referred to as *shuttle* services and *regular* services. Shuttle services are composed of trains that operate between a fixed pair of stations, with short turnaround times. As a result, delay accrued in one journey is carried into the next one. Due to a lack of empirical data for shuttle services, this study does not develop this aspect in detail. However, it is worth noting a few characteristics that could motivate this effort in the future.

Over successive trips, the delay probability vector goes through multiple transformations by $\mathcal{M}_{tr1}$. This is analogous to computing the state probabilities of a Markov chain, with a state transition matrix $\mathcal{M}_{tr1}$. The properties of $\mathcal{M}_{tr1}$ control the steady state behaviour of delays for $tr1$. The matrix is stochastic, there is at least one eigenvalue equal to 1, and this is the largest possible magnitude of any eigenvalue. If this largest eigenvalue is unique, the corresponding eigenvector is the steady state delay distribution of the given train. Should there be multiple eigenvalues of magnitude 1, the steady state distribution will be a linear combination of the dominant eigenvectors.

4.3.2. Regular services: Regular train services typically run long distance, and the turnaround times are quite large. Furthermore, railway carriages are frequently shared between multiple trains running regular long distance services. We may assume that regular services cannot be delayed at the starting station because of delayed incoming locomotives and carriages. The initial delay vector $\pi_{tr1}(s[1,ini])$ is thus constant, and undergoes only a single transformation by $\mathcal{M}_{tr1}$ to give $\pi_{tr1}(s[1,fin])$. The delay for a regular service train can be said to be *strictly bounded* if $\mathcal{M}_{tr1}$ is of a block diagonal form as shown in Figure 12. If, (i) there exists a diagonal block of size $k < k_{\max}+1$ within $\mathcal{M}_{tr1}$, and (ii) the initial delay vector $\pi_{tr1}(s[1,ini])$ has non-zero entries only in the first k locations, then the final delay vector $\pi_{tr1}(s[1,fin])$ can also have non-zero entries only in the first k locations. The implication is that small perturbations in initial delay result in a maximum final delay of $k\Delta$.

A strict block diagonal requirement on $\mathcal{M}_{tr1}$ ensures that small perturbations in $\pi_{tr1}(s[1,ini])$ will be bounded with probability 1. This may result in inflated predictions of delay propagation in a given network, as very low probability events are also accounted for. If desired, the strictness of the condition may be relaxed by requiring the rows of the block diagonal sub-matrix to sum to at least $\theta < 1$. This provides a θ-confidence bound on final (destination) delay for train $tr1$. Let us define $F_\theta(x\Delta)$ to be the cumulative fraction of trains in the schedule, that have delays bounded by $x\Delta$ or less, with a probability of at least θ. The θ confidence bound implies that the fraction of trains in the actual data set with delays of at most $x\Delta$, exceeds the predicted value $F_\theta(x\Delta)$ with probability θ.

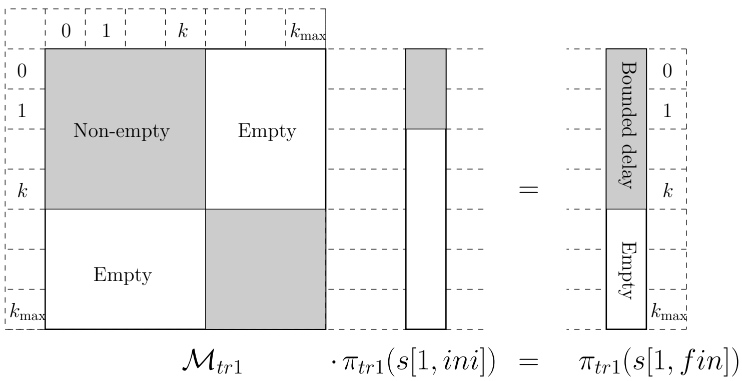

Figure 12 **Schematic of a delay vector transformation matrix $\mathcal{M}_{tr1}$ with bounded delays.**

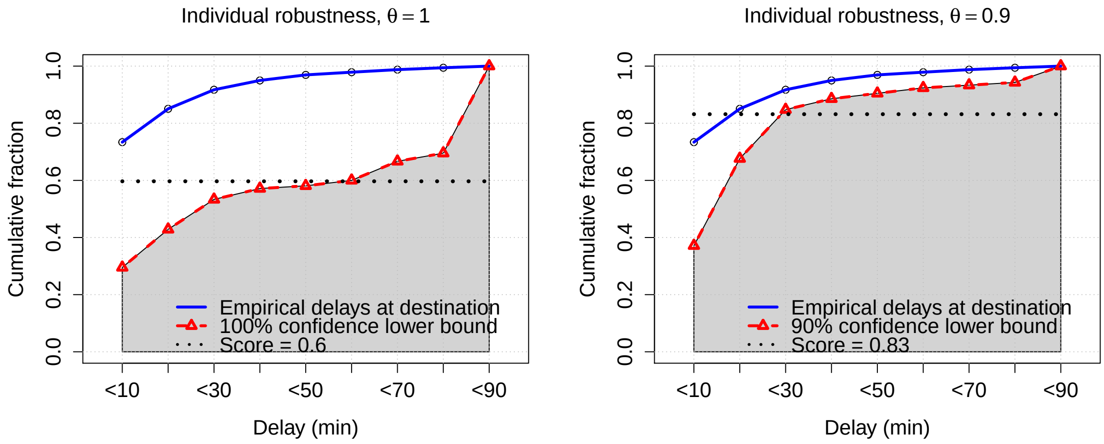

Figure 13 **Cumulative distribution of delays at destination, aggregated over all trains in the available schedule.**

Definition of individual robustness: The individual robustness is defined to be the normalised area under the curve of $F_\theta(x\Delta)$ as a function of $x\Delta$, and is given by $\frac{1}{k_{\max}+1}\sum F_\theta(x\Delta)$.

Figure 13 illustrates this measure for $\theta = 1.0$ and $\theta = 0.9$. The individual robustness score is the fraction of total area that is occupied by the shaded region in the figure. A score of 1.0 implies a maximally-robust schedule, where no train incurs knock-on delays with probability θ. This could indicate an under-utilisation of network resources, and is not always desirable. At the other extreme, a score of 0.0 would indicate that even small primary delays cause unbounded secondary delays. For any confidence level θ, the cumulative curve for empirical delays will be above the $F_\theta(x\Delta)$ bound (shaded area) with probability θ, as confirmed by Figure 13. Note that the individual robustness metric is a function of the parameter $k_{\max}$, which should be held constant when comparing results for different test scenarios.

4.4. Collective robustness

The second measure of robustness addresses the propagation of delays in the network as a whole, when some fraction of trains experience primary delays. Two opposing mechanisms drive secondary delay propagation. Delay recovery between stations reduces the total delay in the network, while interactions at stations tend to increase total delay. Train delays at their destination stations depend on the net result of these two effects, in addition to the initial level of primary delays.

Definition of collective robustness: The collective robustness is defined to be the *fraction of all trains* in the network that are expected to reach their destination stations on time, *conditional* on a given proportion of randomly selected trains with primary delays, and with a *probability* of at least θ. A train tri is defined to be on time with a confidence θ iff $P_{tri}(s[\mathrm{i}, fin], 0) \geq \theta$.

Figure 14 [right] shows that this fraction is approximately equal to 0.4 for the fixed priority scheme, with a confidence of $\theta = 0.9$. This implies that at least 40% of trains will reach their

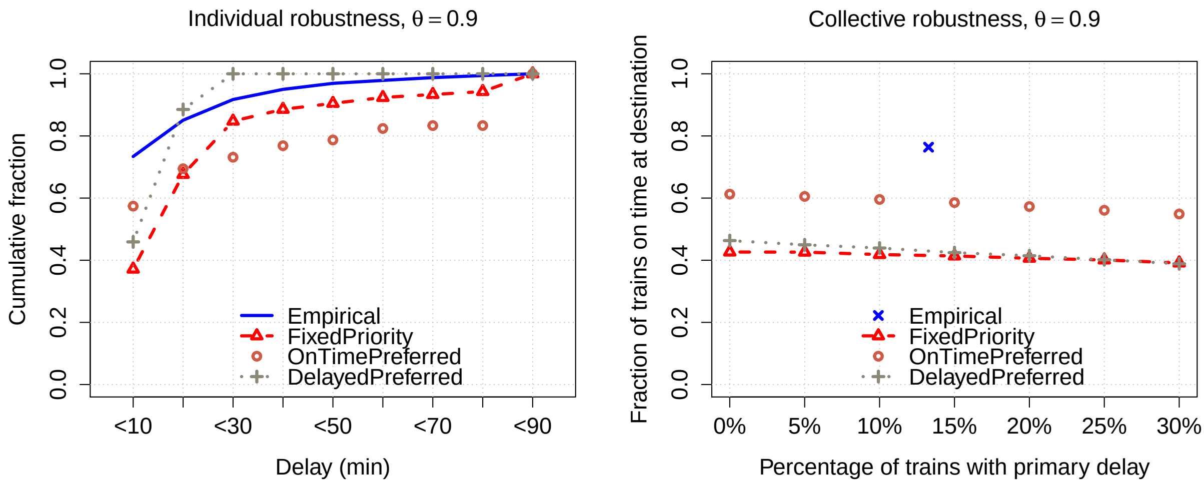

Figure 14 **Individual and collective robustness, compared for different conflict resolution strategies.**

destinations with no delay, with probability 0.9. In this figure, between 0% and 30% of trains were randomly selected for primary delay introduction. The amount of primary delay given to any one train was randomly drawn from the distribution shown in Figure 11. We saw in Section 4.1 that 14% of all trains in the empirical data set were subjected to primary delays. The relevant collective robustness metric from empirical data is 0.78, and is shown as a single point in Figure 14 [right]. Similarly to Figure 13, the empirical value falls above the θ-confidence lower bound.

4.5. Comparison of conflict resolution strategies

Section 3.4 showed that the average aggregate delays in the network were not affected by the choice of control strategy. However, Figure 14 [left] shows that individual robustness is strongly affected by this choice. The fixed priority strategy, as employed by Indian Railways in the empirical data set, shows moderate robustness with a score of 0.83 for $\theta = 0.9$. On the other hand, the strategy that gives priority to trains running on time has a lower score of 0.77. This difference is intuitively explained by the expectation that once a train is delayed, it loses priority and accrues significant secondary delay. Table 1 showed that this strategy has the largest proportion of trains running on time (delay under 10 minutes), but also the largest proportion of significant delays. On the other hand, the strategy that gives higher priority to delayed trains ensures that the delays for any single train are limited. This strategy has a score of 0.93, the tightest confidence bound of the three.

Figure 14 [right] compares the collective robustness scores for the same three strategies, at a confidence level of 0.9. In agreement with intuition, the second strategy (OnTimePreferred) enables the largest fraction of trains (60%) to finish without delays (tightest confidence bound). The remaining 40% of trains may incur delays. As noted in Table 1, these excluded trains may incur larger delays than for the other two strategies. Figure 14 [right] shows that the confidence bounds become more conservative as the proportion of primary delays increases. This behaviour is a result of the larger knock-on effects in the network, when the disruption to nominal operations is larger.

4.6. Comparison of different infrastructure levels

The differences in the robustness metrics for different infrastructure levels (shown in Figure 15) are more pronounced than the differences in predicted delays (shown in Figure 10). This result is intuitively similar to the differences shown by the alternate conflict resolution strategies. Figure 15 [left] shows that a hypothetical halving of available infrastructure reduces the individual robustness from 0.83 to 0.72. On the other hand, a hypothetical doubling of available infrastructure (number of loops at each station) increases the metric to 0.93. The collective robustness is not as significantly affected by a reduction in infrastructure, as seen in Figure 15 [right]. An increase in the infrastructure increases the collective robustness metric to 0.55.

4.7. Applications and limitations of the robustness metrics

The individual and collective robustness scores can be used to explore the broad characteristics of a railway timetable when put into operation. Proposed changes at any one of the three planning levels (strategic, tactical, operational) will have an effect on these scores, giving planners a way to quantify their desirability. An exemplary strategic decision would be the addition of capacity (tracks) to a railway station. By computing the effect on the schedule robustness, it should be possible to identify the best station at which to add capacity. A tactical decision support case could be the addition of a new train service on a particular route. It should be possible to identify the optimal time/frequency of such a service, by minimising the impact on robustness. Operational tactics can also be fine tuned in a similar way. The effect of a change in conflict resolution strategy has broadly opposite effects on individual and collective robustness, as seen in Figure 14.

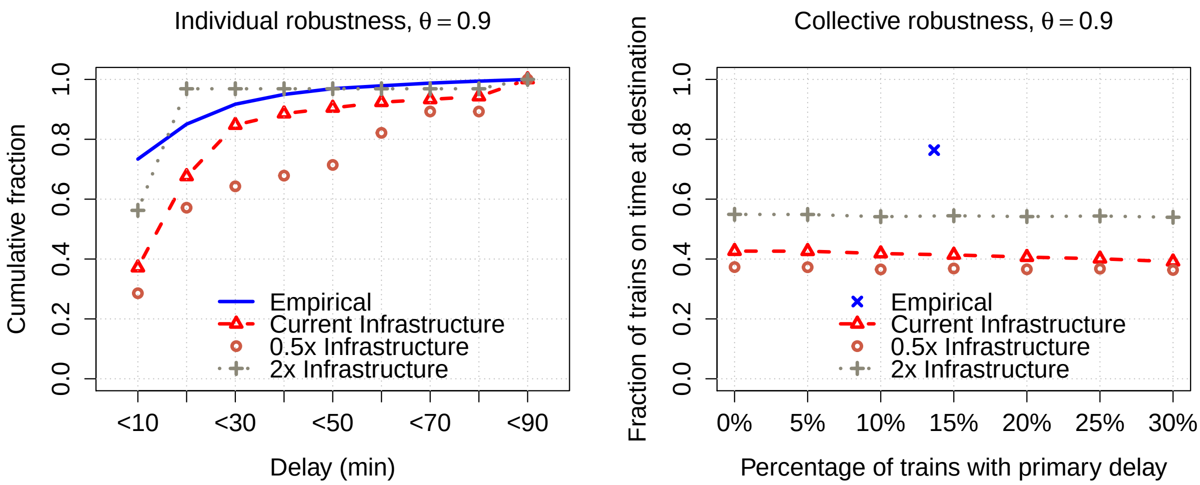

Figure 15 **Individual and collective robustness, compared for different infrastructure levels.**

The best choice of strategy would depend on the performance objectives. For ensuring passenger connections, it may be desirable to limit delays for individual trains, placing an emphasis on individual robustness. For maximising on-time performance, it may be desirable to run as many trains as possible on time, placing an emphasis on collective robustness. System designers should also be able to predict changes in performance indicators, when more than one variable is changed simultaneously. For example, one may explore changes in infrastructure levels, such that a future increase in train services does not affect punctuality. Alternatively, one could find the best priority level to assign to a newly introduced train, such that the effect on the network is minimised.

At the same time, it is important to recognise the extent to which this stochastic methodology is applicable. Limitations of this study include the lack of consideration given to train interactions while running between two stations. This can be addressed in the future by defining an interaction step similar to the one implemented at a station, and appropriately defining the conflict probability. A second limitation is the simplistic view of the conflict resolution algorithms, which may not be completely realistic. Actual rules used by dispatchers frequently depend on the context, such as time of day, geographical location, or personal experience. The approximations made in this paper are thus likely to have introduced errors in the predictions. Nevertheless, the delay propagation algorithm and robustness metrics can be used to quickly establish the operating characteristics of a transportation system, not necessarily restricted to railways.

5. Conclusion

This paper developed a delay propagation algorithm that is able to analyse a railway schedule, and (i) implicitly identify the topology of the underlying network, (ii) compute the probability of each train arriving at each station with a given amount of delay, and (iii) identify delays in empirical data which cannot be explained by network effects (primary delays). This model was shown to be in close agreement with empirical data from a portion of the Indian Railway network. Further, this model was used to illustrate two proposed metrics for schedule robustness. These metrics provide a quantitative basis for analysing the quality of the combination of strategic planning (network infrastructure), tactical planning (timetable) and operating logic (conflict resolution rules). The characteristics of the given empirical data set were explained, from the points of view of specific trains (individual robustness), and the network as a whole (collective robustness). The modular nature of the methodology facilitates changes to the infrastructure and operating logic, and updated predictions can be computed quickly. As a result, the methodology presented in this paper should be useful for exploring the operational characteristics of transportation networks, before the development of detailed simulations or analyses.

Acknowledgments

I would like to acknowledge the contribution of Shripad Salsingikar, Sunil Joshi, Sumit Raut, Siddhartha SenGupta and Sudhir Sinha from TCS, who provided the empirical data used in this paper as well as contributed through constructive criticism in the development of this work.

References

AhmadBeygi, S, A Cohn, Y Guan, P Belobaba. 2008. Analysis of the potential for delay propagation in passenger airline networks. *Journal of Air Transport Management* **14**(5) 221–236.

Andersson, E. 2014. Assessment of robustness in railway traffic timetables. Licentiate thesis, Linkoping University, Norrkoping, Sweden.

Bell, M, Y Iida. 1997. *Transportation network analysis*. John Wiley & Sons, Inc., Chichester, England.

Buker, T, B Seybold. 2012. Stochastic modelling of delay propagation in large networks. *Rail Transport Planning & Management* **2** 34–50.

Cascetta, E. 2009. *Transportation systems analysis: models and applications*, vol. 29. Springer Science & Business Media.

Corman, F, A D'Ariano, I Hansen. 2014. Evaluating disturbance robustness of railway schedules. *Journal of Intelligent Transportation Systems* **18**(1) 106–120.

Corman, F, L Meng. 2015. A review of online dynamic models and algorithms for railway traffic management. *Journal of Intelligent Transportation Systems* **16**(3) 1274–1284.

Dewilde, T. 2014. Improving the robustness of a railway system in large and complex station areas. Doctor in engineering thesis, Arenberg Doctoral School.

Farahani, R, E Miandoabchi, W Szeto, H Rashidi. 2013. A review of urban transportation network design problems. *European Journal of Operational Research* **229**(2) 281–302.

Fischetti, M, D Salvagnin, A Zanette. 2009. Fast approaches to improve the robustness of a railway timetable. *Transportation Science* **43**(3) 321–335.

Goverde, R. 2007. Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B* **41**(2) 179–201.

Goverde, R. 2010. A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C* **18** 269–287.

India Rail Info. 2014. Train timetable and operational data archives. http://indiarailinfo.com/ Retrieved January 2015.

IRFCA. 2016. The Indian Railways FAQ. http://www.irfca.org/faq/ Retrieved April 2016.

Kecman, P, F Corman, A Peterson, M Joborn. 2015. Stochastic prediction of train delays in real-time using Bayesian networks. *Conference on Advanced Systems in Public Transport*. Rotterdam, Netherlands.

Khadilkar, H. 2013. Networked control of aircraft operations at airports and in terminal areas. Ph.D. thesis, Massachusetts Institute of Technology.

Kroon, L, G Maroti, M Helmrich, M Vromans, R Dekker. 2008. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B* **42**(6) 553–570.

Landex, A, L Jensen. 2013. Measures for track complexity and robustness of operation at stations. *Journal of Rail Transport Planning & Management* **3** 22–35.

Lorek, M, S Fekete, A Kroller, M Pfetsch. 2011. Disruption management with rescheduling of trips and vehicle circulations. *Joint Rail Conference*. Pueblo, USA.

Meester, L, S Muns. 2007. Stochastic delay propagation in railway networks and phase-type distributions. *Transportation Research Part B* **41** 218–230.

Meng, L, X Zhou. 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. *Transportation Research Part B* **45**(7) 1080–1102.

Murali, P, M Dessouky, F Ordonez, K Palmer. 2010. A delay estimation technique for single and double-track railroads. *Transportation Research Part E* **46** 483–495.

Pasquale, C, I Papamichail, C Roncoli, S Sacone, S Siri, M Papageorgiou. 2015. A nonlinear optimal control approach for two-class freeway traffic regulation to reduce congestion and emissions. *European Control Conference*. Linz, Austria.

Salido, M, F Barber, L Ingolotti. 2008. Robustness in railway transportation scheduling. *World Congress on Intelligent Control and Automation*. Chongqing, China.

Sheu, J-W, W-S Lin. 2012. Adaptive optimal control for designing automatic train regulation for metro line. *IEEE Transactions on Control Systems Technology* **20**(5) 1319–1327.

Siefer, T, A Radtke. 2006. Evaluation of delay propagation. *World Congress on Railway Research*. Montreal, Canada.

Sinha, S, S Salsingikar, S SenGupta. 2015. Train scheduling using an iterative bi-level hierarchical approach. *International Conference on Railway Operations Modelling and Analysis*. Tokyo, Japan.

Xu, N, G Donohue, K Laskey, C-H Chen. 2005. Estimation of delay propagation in the national aviation system using Bayesian networks. *Air Traffic Management Research and Development Seminar*. Baltimore, USA.

Yuan, J. 2006. Stochastic modelling of train delays and delay propagation in stations. Phd thesis, Delft Univeristy of Technology, The Netherlands.

Yuan, J, I Hansen. 2007. Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B* **41** 202–217.