Transportation Research Part C 95 (2018) 599–615

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Stochastic prediction of train delays in real-time using Bayesian networks☆

Francesco Corman[a,*], Pavle Kecman[b]

[a] Institute for Transport Planning and Systems, ETH Zurich, Switzerland
[b] Department of Science and Technology, Linköping University, Sweden

ARTICLE INFO

Keywords:
Bayesian networks
Prediction
Railway traffic
Stochastic processes
Train delays

ABSTRACT

In this paper we present a stochastic model for predicting the propagation of train delays based on Bayesian networks. This method can efficiently represent and compute the complex stochastic inference between random variables. Moreover, it allows updating the probability distributions and reducing the uncertainty of future train delays in real time under the assumption that more information continuously becomes available from the monitoring system. The dynamics of a train delay over time and space is presented as a stochastic process that describes the evolution of the time-dependent random variable. This approach is further extended by modelling the interdependence between trains that share the same infrastructure or have a scheduled passenger transfer. The model is applied on a set of historical traffic realisation data from the part of a busy corridor in Sweden. We present the results and analyse the accuracy of predictions as well as the evolution of probability distributions of event delays over time. The presented method is important for making better predictions for train traffic, that are not only based on static, offline collected data, but are able to positively include the dynamic characteristics of the continuously changing delays.

1. Introduction

Accurate prediction of train delays (deviations from timetable) is an important requirement for proactive and anticipative real-time control of railway traffic. Traffic controllers need to predict the arrival times of the trains within (or heading towards) their area in order to control the feasibility of timetable realisation. Similarly, the transport controllers on behalf of train operating companies may use the predictions to estimate the feasibility of planned passenger transfers, as well as rolling-stock and crew circulation plans. Valid estimates of arrival and departure times are therefore important for preventing or reducing delay propagation, managing connections, and providing reliable passenger information. The difficulty for predicting the train event times comes from the uncertainty and unpredictability of process times in railway traffic. The models for real-time traffic control have so far mostly focused on overcoming the great combinatorial complexity of train rescheduling (Corman et al., 2014b; Meng and Zhou, 2014; Törnquist and Persson, 2007), delay management (Dollevoet et al., 2014) and rolling-stock and crew rescheduling (Nielsen et al., 2012; Potthoff et al., 2010). The developed approaches are able to solve complex instances in real-time, however they typically assume perfect deterministic knowledge of the input traffic state and subsequent traffic evolution.

In recent years, the uncertainty of train event times has been recognised as one of the major obstacles for computing feasible and

☆ This article belongs to the Virtual Special Issue on “Big Data Railway”.

* Corresponding author.
E-mail addresses: francesco.corman@ivt.baug.ethz.ch (F. Corman), pavle.kecman@liu.se (P. Kecman).

https://doi.org/10.1016/j.trc.2018.08.003
Received 23 October 2017; Received in revised form 11 June 2018; Accepted 6 August 2018

implementable solutions for rescheduling problems in railway traffic (Corman and Meng, 2014; Quaglietta et al., 2013). The uncertainty of an event is usually represented by the probability distribution of its realisation. However, most of the existing approaches assume fixed probability distributions for train delays and do not consider the effect that real-time information on train positions and delays may have on (the parameters of) the corresponding distributions. In order to create realistic online tools for real-time traffic management, the dynamics of uncertainty of delays needs to be considered. When new information about train positions and delays becomes available, the uncertainty for predicting subsequent events is typically reduced. The main objective of this paper is to examine the effect that the prediction horizon and incoming information about a running train may have on the predictability of subsequent arrival and departure times of all trains. In other words, we try to give an answer to the question: how does the probability distribution of delay of an event change over time?

In this paper we first describe a method for modelling uncertainty of train delays based on Bayesian networks. Railway traffic is modelled by means of a probabilistic graphical model which exploits conditional independences between events to allow the efficient computation of their joint distribution (Koller and Friedman, 2009). An important advantage of this method in the context of real-time prediction of train traffic is that it allows the information or evidence about a certain event to be propagated. In other words, evidence about realisation of one event affects (reduces) the uncertainty of other events. Therefore, probability distribution of e.g. an arrival delay in a station changes over time in discrete steps as more information becomes available. This can be used by traffic controllers to estimate the probability of a route conflict in their area, probability of arrival delay of a feeder train for a passenger transfer, etc. Moreover, having a better estimate of train delays could be greatly beneficial for validation and evaluation purposes of the state-of-the-art online traffic models. In particular, this approach enables the estimation of delay dynamics for the closed-loop (Corman and Quaglietta, 2015; Caimi et al., 2012), online rescheduling (Gatto et al., 2007; Bauer and Schöbel, 2014) and simulation (Nash and Huerlimann, 2004; Quaglietta, 2014) tools. Finally, even though we focus on the real-time prediction of railway traffic, the modelling framework and methodology presented in this paper could be extended to handle predictions in other scheduled and constrained systems, such as public transport, logistic networks, and supply chains.

The next section gives the description of the problem and a comprehensive literature review. The methodological framework is presented in Section 3, followed by the description of the case study (Section 4) and analysis of results (Section 5). Section 6 summarises the main findings and gives the recommendations for future research.

2. Problem description and literature review

The real-time prediction of railway traffic is one of the main tasks on the operational traffic control level. Trains are operated according to a timetable and a daily process plan. Due to inevitable disturbances and deviations from the planned schedule, train runs need to be continuously monitored. By monitoring we assume keeping track of all performance indicators such as the actual train positions, delays, realised running and dwell times of all trains, etc. Monitoring therefore provides the actual traffic state that can be used to predict the future evolution of traffic on the network. A predictive traffic model needs to continuously provide the controller level with the information about the expected traffic conditions. Moreover, it should enable the controller to evaluate the impact of potential dispatching actions.

Railway prediction models can be classified to static (offline) and dynamic (online), and deterministic and stochastic, depending on the time span between the time they are run, and the operations they aim to predict; and on how they tackle uncertainty, respectively. Deterministic prediction models assume full knowledge of the future traffic evolution (Dolder et al., 2009). Some approaches (see for instance (Burdett and Kozan, 2014; Wei et al., 2015) focus on simulating the traffic based on current state, and determine most likely conflicts, with limited usage of data on past operations. Even though the more advanced data-driven deterministic models are able to explain a large percentage of process time variability using the values of explanatory variables, a certain degree of uncertainty, especially for dwell times, still remains unresolved (Kecman and Goverde, 2015).

Stochastic models attribute each event with a probability distribution in order to model the uncertainty of its realisation. They can be classified based on how they use the real-time information to update their predictions, into static and dynamic. Whereas static prediction models are based on the offline computed probability distributions and their parameters, dynamic models are updated in real-time as new information becomes available. Most of the stochastic delay propagation models (Büker and Seybold, 2012; Medeossi et al., 2011; Meester and Muns, 2007) were used for offline analyses of timetables. A-Posteriori analysis (Yaghini et al., 2013; Lee et al., 2016) focuses instead on understanding factors and root causes of realised delays, and incorporate those into strategic planning and changes to the timetable. A recent contribution that focuses on predicting delays in the process of appraisal of infrastructure investments was presented by Marković et al. (2015). For all those offline approaches, their goal is to determine factors describing the influence of design parameters to a railway system, in a planning phase, i.e. many months or years before operations. The mentioned approaches are inherently static and do not consider the effect that the real-time information obtained from the monitoring system may have on reducing the uncertainty of the future events.

The concept of delay dynamics refers to the degree of information one has available about future events, when online real time information is considered, within the prediction horizon. The idea of delay dynamics is illustrated in Fig. 1. With every update of train delay (arrivals to station A and B) probability distributions of arrival times to subsequent stations (C and D) are updated.

We summarize in Table 1 the most related works of the literature, which are discussed in what follows. In column 2, the works are categorized according to their *approach*, i.e. basic algorithm used to determine the conditional probability of future events based on past data and current events. Column 3 describes the consideration of *dynamic* nature of the problem, i.e. whether the different online predictions over time are linked somehow by parametric relation inherent in the model. For the works mentioned, often dynamics is not mentioned (Böhmová et al., 2015) or is defined a priori based on rules (Bauer and Schöbel, 2014); in general, dynamics within the

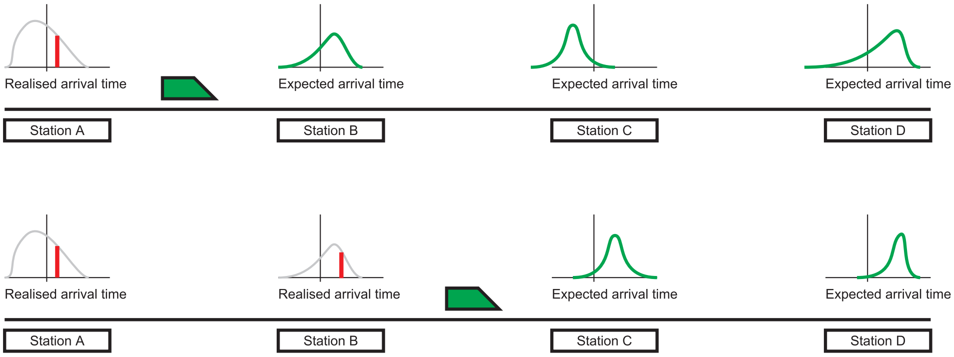

Fig. 1. Dynamic evolution of probability density of delay of a single train in time.

Table 1
Summary of the literature on dynamic data driven delay prediction approaches for railway networks.

Work	Approach	Dynamics	Relations
(Berger et al., 2011)	Probability update	Repeated	Single train route
(Keyhani et al., 2012)	Probability update	Repeated	Single train route
(Böhmová et al., 2015)	Nearest Neighbor	None	Data driven
(Bauer and Schöbel, 2014)	Rule based	Delay increases	Single event
(Lemnian et al., 2014)	Fuzzy a.o.	Repeated	Train types
(Kecman et al., 2015)	Markov Chain	Non-stationarity	Single train route
(Sahin, 2017)	Markov Chain	Steady state	Single train route
(Oneto et al., 2017)	Neural networks	Repeated, learning	Data driven
(Oneto et al., 2018)	Neural networks	Repeated, learning	Data driven
(Lessan et al., in press)	Bayesian network	Repeated	Data driven, hybrid
This work	Bayesian network	Update over time	Operating plan

prediction horizon is not mentioned, but over a long period of time, the distributions can be repeatedly recomputed, if an enlarged data set would be available, or timetables and exogenous conditions might change (Berger et al., 2011; Keyhani et al., 2012; Lemnian et al., 2014; Oneto et al., 2017, 2018; Lessan et al., in press); a learning process can update incrementally the model with some new values of parameters, as suggested in (Oneto et al., 2017, 2018); the dynamics can be inherent in a stochastic process model (Kecman et al., 2015), where only the process evolution towards a steady state is investigated (Sahin, 2017). The last column classifies whether the probabilistic *relations* between events are inferred from data only, or from some domain knowledge, either at the single event, single route of a train, aggregating together trains of the same type, or considering the entire network operating plan and interactions between trains over the network.

The prediction of operations in public transport networks has been studied by (Böhmová et al., 2015) resorting on probability distributions from past operations, and using proxies to map a current state to the past operations, based e.g. on factors such as day of the week, time of the day, etc. In this stream of research, the approach is online (dynamic), i.e. it processes data as they arrive, but tackles the problem to determine the future (*What do I know of a future event now?*) once and for all, and are not studying the problem of studying the evolution of my understanding of the future, as time goes by (*What do I know of a future event, at any time before it happens? How certain I am?*)

An approach that considers the dynamics of uncertainty of train delays was presented by Bauer and Schöbel (2014). The authors developed a 'delay generator' for the purpose of integrating uncertainty in online traffic management. A uniformly distributed delay value is assigned in real-time to a set of randomly chosen events. However, their approach represents a rather theoretical concept that mimics the evolution of train delays in time in order to create realistic instances for validation of an online delay management tool. This idea has been implemented recently with the purpose of developing a proactive passenger information system by estimating the probability of future delay of a single train based on the currently known delay (Lemnian et al., 2014). Those works are the first to report or describe that "delays fluctuate over time", and systems which are robust to such a deviation results in better transport performance. However, that work, as well as its predecessors (Berger et al., 2011; Keyhani et al., 2012), assumes that the delays of trains which do not have a scheduled passenger transfer are independent. In other words, delay propagation due to capacity constraints is not considered.

Regarding the determination of uncertainty and dynamics of delays over time, the usage of past data has been shown in (Sahin, 2017) enough to determine a Markov model of the operations as stochastic processes. Ultimately, running time supplements that would absorb the evolution of the delays in a steady state, as measured in the recorded data, can be computed. In a preliminary work on this topic, the dynamics of uncertainty was included by modelling train delay evolution over time as a stochastic process (Kecman et al., 2015). A train run is represented as a Markov chain with state transitions in discrete moments that represent arrival and departure events from a scheduled stop. After every registered departure or arrival event, the conditional probability distributions of the downstream events are updated with respect to the essential assumption for Markov processes that, given the present, future events do not depend on the past. A train delay evolution is modelled as a non-stationary Markov chain, meaning that the probability of a state change depends on the moment of transition.

Online approaches predict the short-term future, based on data about the present status of the network. Recently a number of very sophisticated approaches have been published (see for a very recent overview, (Ghofrani et al., 2018)). A recent interesting contribution is (Oneto et al., 2017), using Machine Learning approaches to train Extreme Learning Machines, i.e. special cases of neural networks, tackling a large scale network, using a fully data driven approach, and including exogenous dynamics such as weather information. Compared to this parallel research direction, we tackle more directly the dynamics of uncertainty, and confidence bounds on the forecasted events, rather than just the performance of prediction accuracy. We also refer to the time distance between the time now and the future, instead of the amount of stations ahead; and we tackle intense traffic on capacity bottlenecks, where train-train interactions and delay propagation are the norm. To this end, we include domain dependent knowledge, to help statistical learning methods. In a follow up work (Oneto et al., 2018), the authors evaluate the system on real life data, where good emphasis is put onto the computational performances and the dynamics across time horizons, i.e. the ease by which the entire system can be repeatedly trained again, once new data would be available, or regularly every day or so. The computational benefits of a distributed implementation towards running time are also proven. To predict the delay distribution of an event, they consider all events along the path of the single train under investigation, as well as information on the rest of the network, and exogenous conditions such as type of the day, as additional variables to be included in a neural network.

Another data-driven approach uses Support Vector Regression to predict arrival times of freight trains (Barbour et al., 2018). The estimated time of arrival, and associated delay of individual freight trains, is estimated, based on the properties of the train, network, and some knowledge about the traffic. The testcase of freight traffic in US has a lot of irregular operations, very variable traffic, and large delays, which limit the applicability of strong domain-based modelling constraints influencing the delay.

In this paper we develop the complementary idea to explicitly include the causal and temporal dependencies of events of other trains in the computation by explicitly modelling the (domain-specific) interdependence between trains that share the same infrastructure, or have a scheduled passenger transfer, by means of Bayesian networks. Therefore, an observed delay of a train will not only be used to update the probabilities of further events along the route of that train. That is, probability distributions of delay for all events of other trains that may be affected are updated. An illustrative example of the system setup is given in Fig. 2. The departure of the first train from Station A and its arrival to Station B initiate the procedure to update the probability distributions of all other estimated event times (EET) that may be affected by the observed delays. A Bayesian network with a structure that corresponds to a macroscopic traffic model can therefore be used to compute *stochastic* delay propagation with respect to the capacity constraints as well as the constraints due to passenger, rolling-stock or crew connections (Goverde, 2010). While the structure of the Bayseian network is fixed by the operating plan (e.g. a timetable), we use historical traffic data to calibrate the resulting Bayesian network with conditional probability distributions and regression coefficients for every two dependent events. Therefore, the incoming information from the monitoring system is used to reduce the uncertainty of the future events.

Similar approaches based on Bayesian networks have been also used in a recently published paper (Lessan et al., in press). There, the focus was in studying algorithmic performances of different network training methods (hill climbing, primitive linear), as well as

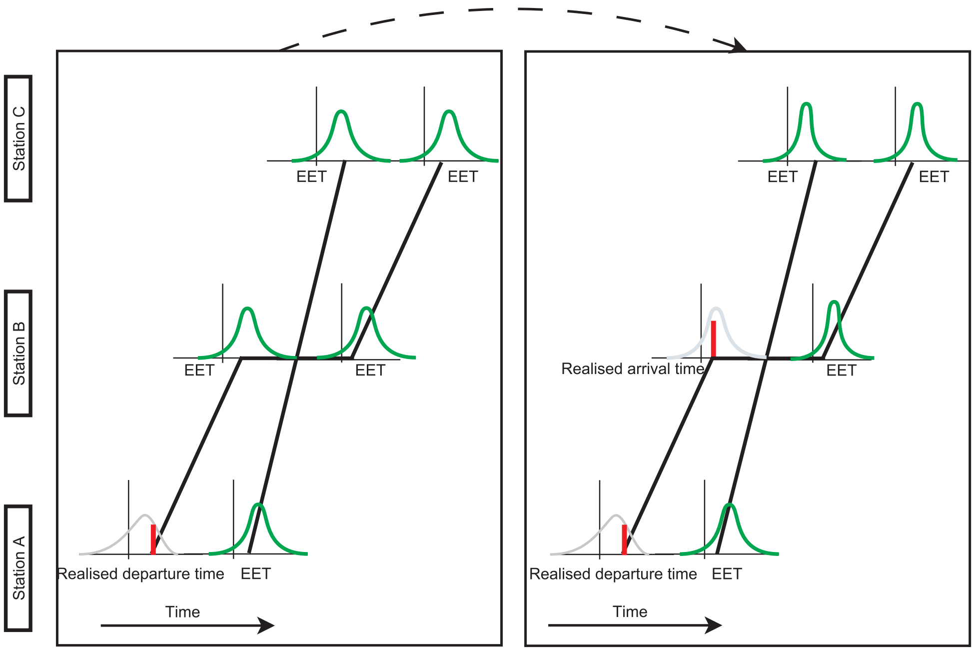

Fig. 2. Dynamic evolution of probability density of all dependent events in time.

the possibility of using hybrid learning methods to combine pure data-driven approaches with some domain knowledge. The authors found that hybrid structures can satisfactorily model complex relations arising in operations. Their main interest was in proving low prediction error of the models for future events. In comparison, we also base our study on a mix of domain knowledge, exploiting in particular the acyclity arising in graph theoretical models for railway operations, combined with large volumes of data. Our biggest contribution and theoretical focus is not per se in prediction accuracy, but in the interesting study of delay dynamics, i.e. how the prediction of a single delay event changes over time as the same event approaches the time *now*.

3. Methodological framework

3.1. General description and network structure

The ultimate goal of this paper is to compute the estimate of (i.e., predict) the probability distribution P of the random variables describing an event e in the future. To do so, we have to combine the realised value of some random variables, which are somehow connected to the event, with past observations of the random variables which allow us to describe this relation. In general $P(e_i) = P(e_0, e_1, ..., e_N)$, i.e. event i might depend on all N events, and maybe also on other exogenous ones. For prediction purposes, only those events $0, 1, ..., i-1$ in the past are actually usable. In this general framework of stochastic prediction of railway operations, the most important choices are which *structure* to give to the connections, the form of *correlation structures*; and the procedure of *update* of the conditional probabilities as new information is available.

As for the structure, in principle all events might be dependent on all other events, but in practice, models would assume that only some of the connections are more relevant than others. In such case, only some connections are explicitly modelled, the others can be kept only implicitly, or completely neglected in the mathematical relations. Then, $P(e_i)$ can be reported as function of $P(e_A, e_B, ..., e_Z)$, i.e., for a subset $A, B, Z \in N$ of the events only.

As for the correlation structures, this refers to the functional for $P()$. A simple case is a linear relation, for instance $P(e_i) = (a_A \cdot e_A + a_B \cdot e_B + ... + a_Z \cdot e_Z)$, with coefficients a_i, for a (sub) set $A, B, Z \in N$ of events. Other approaches have more complex relations, and some have even not always an explicit relation (for instance if they are based on black box models, such as neural networks).

The final item, i.e. how to determine effectively $P(e_i)\ \forall\ i$, once $P(e_x), \forall\ x$ are known with certainty, refers to the efficient computation of the probability, and especially in the case that probabilities depend on each other. In such case, a procedure should be determined that can effectively explore all relations and solve the set of equations $P(e_i) = P(e_A, e_B, ..., e_Z)$ given only some events are known with certainty, and all other interdependent $P(e_i)$ are actually unknown.

Bayesian networks are graphical models for reasoning under uncertainty, where the variables and conditional dependencies between them are represented with a directed acyclic graph $G = (N, A)$ (Korb and Nicholson, 2010). Nodes $i, j \in N$ represent random variables. In our model, each random i variable is associated to a node, and models directly an event e_i, which can be a particular train arriving at a particular station. Each event e_i and associated node i have associated some attributes: train number, station name and event type (arrival, departure or through). A directed arc $(i, j) \in A$ connects two nodes i and j, and models the dependencies between the events, with the direction of an arc indicating the causality relationship between the variables. Bayesian networks rely on the fact that a random variable typically interacts directly with but a few other random variables to construct a concise representation of reality where only the direct dependencies are encoded in the network (Koller and Friedman, 2009). The structure of the network, i.e., the directed arcs between the nodes that represent the considered events, can either be learned from the data or determined by expert knowledge. The recent trend of implementing sensor technologies and advanced data management systems in many railway networks in Europe allows using the massive databases of historical traffic data for the structure and parameter learning of Bayesian networks.

In our approach, we resort to a Bayesian network as we can exploit a large body of knowledge on graph theoretical models of railway operations (Hansen and Pachl, 2014). Namely, we know that feasible railway operations plans can be represented as a Direct Acyclic Graph, and use this to efficiently determine a network structure, which has relatively few explicit connections (i.e. the degree of connectivity of the network is relatively low), while reaching satisfactory prediction performance. We consider all connections between events as linear. An important property of Bayesian networks is that they explicitly model the quantitative strength of the connections between variables, thus allowing the probabilistic beliefs about them to be updated automatically as new information becomes available. This property enables modelling the dynamic inference between random variables in discrete moments in time (Murphy, 2002).

Learning the dependencies from a pure data driven approach (see (Oneto et al., 2017, 2018) for instance) or an hybrid approach (see (Lessan et al., in press)) would in principle be able to represent all observed correlations and stochastic dependencies between events. In this case, $P(e_i) = P(e_A, e_B, ..., e_Z)$ where only those events A, B, Z are considered, which have a link (causal, empirical or temporal correlation) with e_i which is considered strong enough. The limitations of such an approach are in the difficulty to determine causality or a temporal relationship within correlated events, in this way. Moreover, in theory there can be sensible correlations between all possible N Nodes (i.e. order of N^2 probabilistic relations), and moreover with external exogenous variables (for instance, weather). Thus a lot of effort or even ad hoc filtering (Lemnian et al., 2014) might help keeping only those relations which are physically and causally meaningful, and remain quick enough when building topology and updating the conditional probabilities.

We instead resort to domain-dependent knowledge, namely we use as structure of our probabilistic connections the plan of operation, which would the implicitly model all possible relations between events. In fact, a delay of a train may be a direct predictor of delay of the next event of that train which in turn can be used to estimate the delay its immediate successor and so on. The same

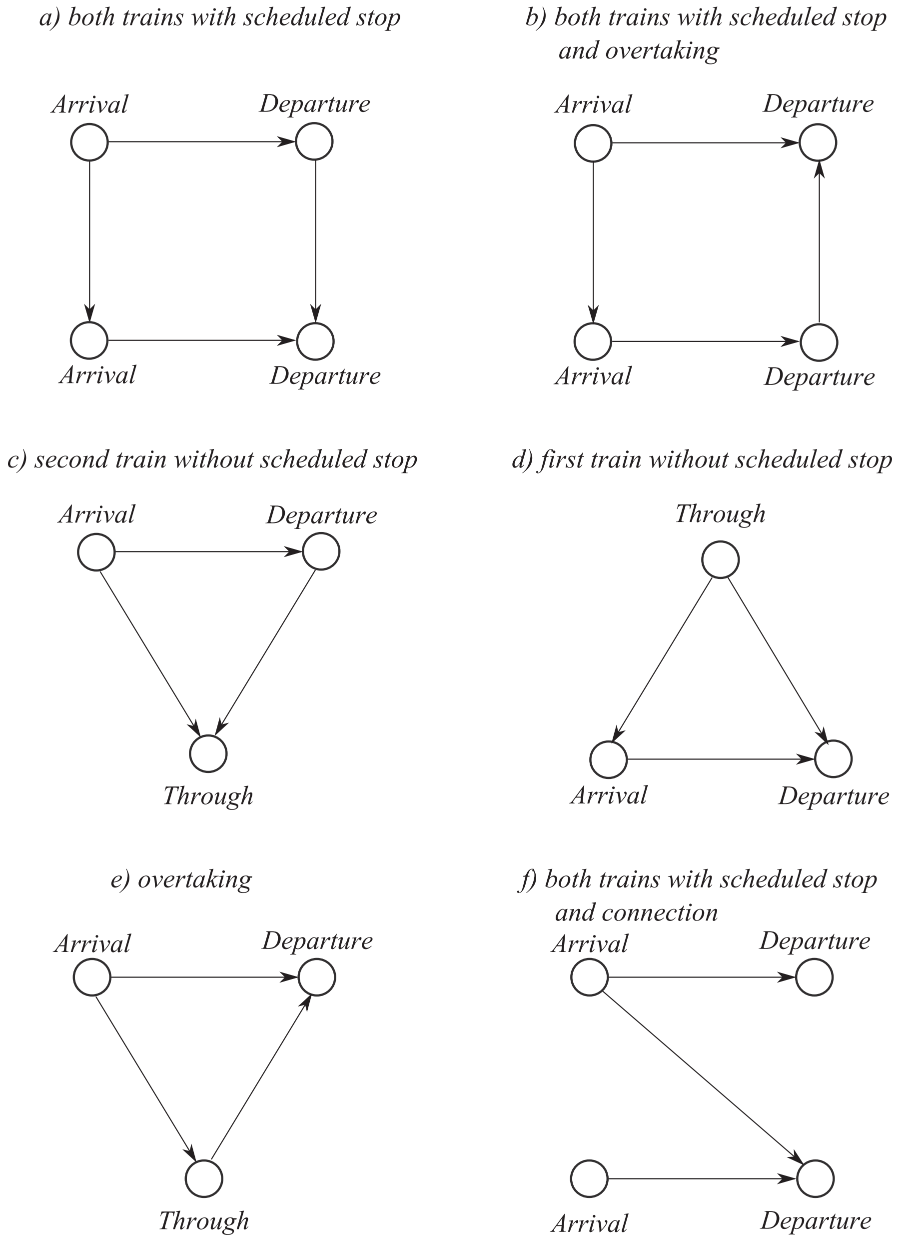

Fig. 3. Building blocks for creating the structure of a Bayesian network.

principle can be applied to trains that use the same infrastructure and therefore need to be separated by minimum headway times. If two trains use the same part of the infrastructure (block section or a station track) within short time, which is often the case on busy corridors, a delay of the first event can be used to predict the delay of the second. Finally, for trains with a scheduled passenger connection in a station, an arrival delay of a feeder train could cause a departure delay of the connecting train. We remark here that those domain-dependent constraints are not considered in the approaches mentioned in Table 1 and in fact are needed to ensure acyclicity of the graph. Experimentally, we find that the number of arcs considered in those approaches is relatively small, in the same order of magnitude as the amount of nodes.

All those structured systematic relations between events can be schematized into some template structures, reflecting the constraint from railway operations, to the relationship between events. The basic structures that are used to model causal dependencies in most typical situations for one-directional traffic are given in Fig. 3.

The described principles of causal relationship between delays of different events correspond to deterministic macroscopic traffic models where only train events at stations (departures, arrivals, through runs) are represented as nodes in the graph. This is similar to event activity networks where events e_i are of two types, arrival events, or departure events, and activities are describing a particular relation between two events. This model can be easily adapted to a graph, where events are associated to nodes i, j, and activities to arcs (i, j). In practice each of those constraints can be represented as $T(e_j) = max\{e_1 + d(1, j), ...e_N + d(N, j)\}$, where $T(e_i)$ would represent the time at which event e_i occurs, $d(i, j)$ represent the duration between events i and j, i.e. represented by the arc (i, j). The *max* operators describes the fact that events can only occur when a set of events e_i and durations $d(i, j)$ have respectively occurred and elapsed. In the different cases of running, dwell, headways, connections, planned departures, a suitable choice of the $d(i, j)$ is possible.

This way to model railway traffic on single- and double-track corridors and networks is rather general and has been used earlier to model delay management (Schöbel, 2007), delay propagation (Goverde, 2007) and more recently for analysis of timetable robustness (Jovanović et al., 2017). We are interested in only the structure of this graph, which we translate into the Bayesian network. Running and dwell arcs are used to model precedence relations between events, which in a Bayesian network correspond to the causality relationship between events of a single train run. The headway arcs between events of different trains model the train separation principles. The macroscopic character of the model implies that the detailed infrastructure constraints are neglected. However, they are implicitly represented in precedence constraints between station events. For example, two conflicting inbound train routes in a station are represented, in a deterministic macroscopic model, by an arc that indicates that the second train cannot arrive before the first train has arrived. In a probabilistic graph used in this paper, the corresponding arc models the probability of delay of arrival event of the second train given the arrival delay of the first train (Fig. 3 situations (a) and (b)).

In this paper we assume that the train orders are known. In general, the default plan of operations is the timetable, but every new plan of operations which can be implemented as a reaction to delays (changing order of trains, for instance) should have no two operations depending mutually on each other, to be feasible.

This is in accordance with the intended purpose of the model which is: (i) to provide traffic controllers with the estimated traffic evolution given the current traffic state and (ii) to allow them to evaluate the effects of their decisions. The envisaged real-time deployment of the model requires a procedure to modify the network structure depending on the dispatching actions such as re-ordering and rerouting of trains. Every change of orders refers to a dispatching action, which is a purpose-made update of the plan. Every such change of order changes the structure of the network, and its parameters, and need to be "propagated" to all downstream events between the two trains. The computational complexity of this case is discussed later.

In any case, all those possible operating plans would result into an acyclic graph. An acyclic graph guarantees that a delay of an event cannot be propagated to itself. Efficient procedures to tackle this problem, that could be implemented to update the network structure, have been implemented in graph-based rescheduling algorithms (Mascis and Pacciarelli, 2002; D'Ariano et al., 2007; Corman et al., 2014b).

3.2. Parameters learning and inference

Once the structure of the network has been determined, probability distributions of delays of considered events need to be computed. This would mean, computing the joint probabilities $P(e_i, e_j, ..., e_k)$ of all events $e_i, e_j, ..., e_k$ that are considered in the structure of the delay relations; i.e. that are associated to an arc in an equivalent DAG representing the operating plan of railway movements. The cumbersome computation of a joint distribution for a large number of random variables is avoided by relying on the network structure that encodes only the direct dependencies between events, according to the Bayesian networks. In this sense, the structures reported in Fig. 3, are instrumental in identifying templates of linear relations between events, which are described as joint probability distributions. Joint distribution can thus be represented by a number of local distribution which have only a small number parameters to estimate (Nagarajan et al., 2013), given a network structure, encoding the sensible relations between events.

Koller and Friedman (2009) show that a local distribution can in fact be represented as a linear model in which the parent nodes of the incoming arcs are the explanatory variables. This is in accordance with the earlier research results that demonstrate the accuracy of linear regression for modelling train delay dependencies and justify the selection of normal distribution to model train delays in the Bayesian network model (Bayissa, 2013; Kecman and Goverde, 2015). Distribution parameters for delay of node j associated to event e_j are computed with respect to the network structure. In particular, referring to the probability $P(e_j = x)$, representing the probability that an event e_j would have value (i.e., occur at delay) x minutes, we compute the expectation μ_j of the event e_j as $\mu_j = a_0 + \sum_{i=1}^{|in(j)|} a_i \cdot d(i)$ where $in(j)$ represents the set of direct predecessors (parents) of the node j associated to event e_j. The coefficients a_0, a_i, $\forall\ i|i \in in(j)$ are obtained by a process of fitting the linear model, while $d(i)$ are observed or estimated delays of parent events. The standard deviation σ is computed as a standard deviation of the residuals from the linear model.

A local distribution (of an event) is considered to be a univariate normal random variable. This assumption of the Gaussian property of random variables may be seen as a limitation for application in modelling train delays, having in mind the earlier results of statistical analyses of train delays, which typically use Weibull or Gamma distribution to model the delays (Yuan and Hansen, 2007). However, a local distribution in our model describes the delay of an event *conditioned on the known delay of its direct predecessors*. The additional information about the realisation time of the previous event can be used to significantly reduce uncertainty and modify the probability distribution of the event time. For example an arrival delay can be predicted more easily if the departure delay from the previous station (and/or arrival delay of the previous train to the same station) is known. The entrance delays (delays that do not have any predecessors) can in turn be estimated using arbitrary fitted distribution (not necessarily normal Gaussian) and given as crisp values as an input to the model.

Rather than simply describing the dependencies between random variables, Bayesian networks represent a powerful tool for inferential statistics and probabilistic reasoning. When new information about a random variable (evidence) becomes available, it is propagated through the network by updating the posterior probabilities (beliefs) of the relevant nodes. A number of exact (recursive application of Bayes' theorem) and approximate (Monte Carlo sampling) algorithms exists that can perform this computationally demanding task (Korb and Nicholson, 2010).

When an evidence about the observed event delay, becomes available, a conditional probability query determines the posterior distribution for each reachable event.

For every two nodes $i, j \in N$ and respective events e_i and e_j, where j is reachable from i, i.e. there is a sequence of directed arcs allowing a path from i to j, a conditional probability query may be used to answer the questions such as: what is the probability that a delay of event e_j (i.e. the realization of node j), is larger than x minutes, given (assuming) that the delay of event e_i (i.e. the realization of i) is y minutes? Moreover, given that the network is calibrated with continuous data and parameterised with linear coefficients, repeated application of linear models gives the most probable outcome of delays for all reachable subsequent events (maximum a posteriori query).

The size and complexity of the network have a significant impact on the computation times of predictions and posterior probabilities. Having in mind the online character of the proposed model, a quick processing of the information received from the monitoring system is essential. Exploiting once more the acyclicity of the graph representing the plan of operations network and the associated Bayesian network, we can efficiently perform the determination of the posterior probabilities of any subsequent event by a visit to all descendants of the event which has just become realised. The problem here is, that if $P(e_i) = P(e_A, e_B, ..., e_Z)$ if is possible that $P(e_A)$ is actually dependent on $P(e_B)$, and multiple interrelations between those vehicles are possible. Due to the acyclicity of the network, we can efficiently associate a topological order to the Bayesian network, and then just perform the visit of all nodes after the event, i.e. compute $P(e_A)$, $P(e_B)$, only when the probabilities of the predecessors of A, B are known or have been already computed. Formally, a topological sorting is a linear ordering of N nodes of a graph, such that for any directed arc between nodes u and v, node u appears before node v in the ordering. The computational complexity of a topological sorting of an acyclic graph is linear in the amount of nodes and arcs. The greatest advantage of the topological sorting is that computation of the conditional probabilities for any descendants of a node can be done by an efficient linear visit to the ordering (linear in the amount of nodes). This would require at most N linear sums, each of which having up to N terms, where N is the amount of nodes.

In our implementation, the time required to sort topologically the graph (in the order of thousands of nodes and arcs) is about 0.1 s, and the time required to visit all descendants (events) happening within the next hour is about 0.7 s, with about 200 nodes to be visited. As a comparison, visiting only the next 15 min would require about 80 nodes and 0.6 s; visiting up to 2 h would result in visiting about 300 nodes and 0.8 s; visiting the entire network, corresponding to a day of events, would take up to 3.5 s. Due to the further stochastic combination and fading phenomena, there is not much point in computing conditional probabilities for events very far away in time, as delays typically fades away due to buffer times which are inherently considered in the plan (Andersson et al., 2014), and the variability of the outcomes, and its low accuracy would make the prediction of very little use. Therefore, the received information typically does not affect the probability distribution of all events within the prediction horizon (Kecman, 2014).

Fig. 4 gives a network structure of the illustrative example shown in Fig. 2. The delay of an event is a random variable where each event is represented by a 3-tuple: train number (t_1, t_2), station name (A, B, C), event type (arr, dep, $pass$ for arrival, departure and through run, respectively). The parameters of distribution for node $d(t_1, B, dep)$ are computed by fitting the linear model in which the node is the response variable and nodes $d(t_1, A, arr)$ and $d(t_2, B, pass)$ are explanatory variables.

We discuss here briefly the computation complexity required when the topology of the network changes, for instance as a consequence of a dispatching action, and the network must be updated. From a practical perspective, change of orders require some explicit action from dispatchers, and for this reason, the updates to the plan are typically few and very well traceable. Updating the Bayesian network is then matter of updating only the marginal amount of nodes which have changed the incoming/outgoing arcs, i.e. the vast majority of the topology remains the same, while one or two orders of trains would be changed at any time of implementing a dispatching action. The update has thus a local step, i.e. updating the models at the level of the nodes; and a global step, i.e. checking the properties of the entire network.

For each node, the update of topology can practically be inverting the direction of an arc; or removing one arc; or inserting an additional arc between two nodes. In the worst case, the amount of nodes affected equals all the nodes of the two trains involved (the order changes in a single point, so at least 2 nodes are directly affected; and in the worst case this order has to be propagated to all

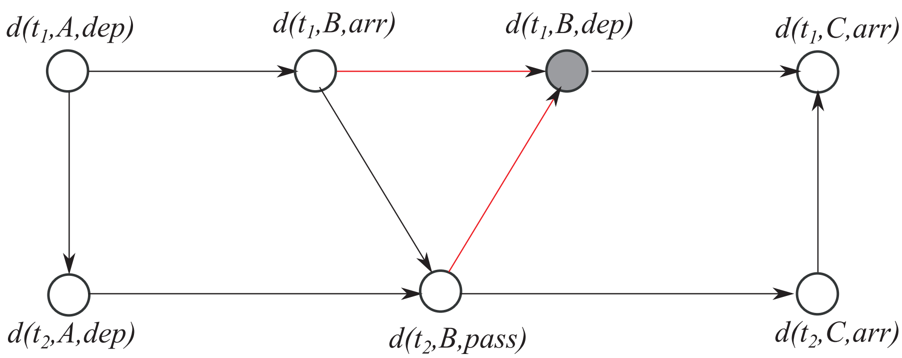

Fig. 4. Illustrative example of a Bayesian network.

events of the two trains, for instance if they follow each other along a corridor). For each of those nodes, a linear model has to be estimated, which would be very fast, due to the small amount of nodes involved. Moreover, the most likely delays (for instance inspired by the recorded data) and associated dispatching actions can be precomputed beforehand, and kept in a library of models to be plugged in, anytime the order of trains would changed, similarly to (Van Thielen et al., 2018).

The most computationally challenging part of such a procedure is the check that the entire network remains acyclic at global level, otherwise the entire Bayesian network loses meaning. In this case, we remark that the very nature of dispatching actions is to avoid any occurrence of deadlocks. Those deadlocks are in fact cyclic dependences of train orders, which are equal to a cyclic dependence of events in graph models, which have been used extensively in railway optimization (Borndörfer et al., 2018). Such a graph theoretical model can be further deterministic (such as the alternative graph model used widely for dispatching in (Mascis and Pacciarelli, 2002; D'Ariano et al., 2007; Corman et al., 2014b) among others) or stochastic (such as the Bayesian network presented here). In this sense, we are certain that, based on a feasible and conflict free dispatching action, the updated topology and resulting Bayesian network of events would never be able to determine cycles; and we can safely reduce the computational effort required to estimate the new linear networks for the updated nodes, if precomputation would not be possible (in which case, it would be even faster), avoiding a full check of the acyclity of the network. Under those assumptions, the need to adjust the structure of the network as response to an action of the dispatcher also does not seem a critical issue from a computational time point, as the time required to fit a linear model to a node is in the order of the milliseconds, in our case.

4. Case study

4.1. Description of the data set

The methodology described in the previous section was applied on a realistic case study from a busy corridor between Stockholm and Norrköping in Sweden. The corridor comprises the 180 km long northern part of the Swedish southern mainline between Stockholm and Malmö. It is a double-track line with mixed traffic. Passenger traffic is dominant with 90% share that comprises both local and intercity trains. Table 2 gives the names of the stations on the line, their distance from the terminal and the average scheduled running times between them. The considered corridor has in total 27 stations and junctions, 10 of which accommodate scheduled stops of passenger and freight trains. Approximately 300 trains per day traverse the corridor (fully or partially).

For the purpose of this study, a database containing two months (1 January to 28 February 2015) of historical traffic realisation data from system Lupp has been made available by the Swedish infrastructure manager Trafikverket. This dataset is representative of the average operations, in the sense that punctuality performance of the training + testing dataset (91.6 % at the threshold of 5 min) is very close to the average performance of the entire 2015 (91.2 % at the threshold of 5 min) and also the multi-year average 2001–2016 (91.1 % at the threshold of 5 min). Moreover, no strong seasonality effect is present in the performance of the Swedish network studied. The database contains the scheduled and realised times for departures, arrivals and through runs for all trains and

Table 2
Checkpoints where train events are recorded by the Lupp system.

	Name	Distance from terminus [km]	Av. sched. run. time from previous [min]
1	Stockholm Central	0	0
2	Stockholm South	2	3
3	Årstaberg	5	1
4	Älvsjö	8	2
5	Stuvsta	10	2
6	Huddinge	12	1
7	Flemingsberg	14	2
8	Björnkulla	16	2
9	Malmsjö	28	4
10	Södertälje	35	2
11	Järna	45	3
12	Mölnbo	56	4
13	Gnesta	63	2
14	Kolke	68	2
15	Björnlunda	72	1
16	Stjärnhov	81	3
17	Nyckelsjön	90	3
18	Sparreholm	93	1
19	Skebokvarn	99	2
20	Flen	107	3
21	Sköldinge	115	3
22	Stolpstugan	122	2
23	Katrineholms C	130	3
24	Strångsjö	141	4
25	Simonstorp	155	5
26	Åby	170	5
27	Norrköpings C	178	5

stations. All event times are rounded to full minutes. Table 2 shows that, on average, an information about a deviation of a train from its scheduled route is given with a frequency of 2 min (average running time between two stations). No information about the train movements is available between the checkpoints. Such low precision granularity of measurements prevents developing a detailed deterministic model such as the one presented by Kecman and Goverde (2014) and justifies the approach that takes into account uncertainty of train delays.

We exclude one randomly selected weekday from the analysis and use it as a test day to evaluate the performance of the Bayesian network model in a simulated real time environment. The remaining data is used as a training set for model calibration. The structure of the network is determined assuming that the train routes and orders are known (Section 3.1). The scheduled event times and orders, determined by the timetable, are used to build the basic structure of the Bayesian network. However, the structure of the network can be modified dynamically by reversing, removing or adding arcs. This is used to model the reordering actions, train cancellations or adding ad hoc train paths by the traffic control. Note that any dynamic change of the network structure requires recalibration of the affected nodes. In this work we assume the train orders to be fixed and available for one hour.

The calibration of the network assumes the computation of parameters for probability distribution of delay for each considered event represented by a node. The training set for computing the parameters of local distributions depends to a great extent on the network structure. For example, an arc representing a delay propagation during a train run between two stations is calibrated based on historical data for the *same train*, as identified by the train number, which repeats daily. Similarly, the relevant training set for an arc representing a headway arc from an event of another train, comprises only the days when the two trains ran in the same relative order. This means that the recurring phenomena for a particular event that may happen due to: (i) variations in travel demand (peak/off-peak hours), (ii) distribution of time reserves in the timetable, are implicitly represented in the model.

4.2. Experimental setup

The training and test datasets described in the previous section are used to create and validate the Bayesian network model. In order to test the performance of the described model, an experimental environment was set up that includes a static and a dynamic component. The static component consists of the database of historical traffic data used for dynamic assignment of distribution parameters. The dynamic component of the experimental environment comprises the actual process plans for all trains within the prediction horizon and actual train event times. The actual route for each train is given on the level of station events, i.e., ordered list of stations and event types. This is crucial for building the probabilistic graph model. As the prediction horizon moves, new trains are added to the model. The graph size does not increase linearly in time as only the nodes representing the last events of all trains are kept in the model due to the inherent Markov property of Bayesian networks (Koller and Friedman, 2009).

As explained in Section 4.1, the Lupp system log files contain the chronologically ordered train event messages. A real-time environment for model validation was created by sweeping through the test set Lupp log file for one day of traffic. Every train event message represents the new information that is propagated through the graph using statistical inference algorithms that predict the events in the topological order of the graph as in Kecman and Goverde (2014). The actually realized train event times are used to test the accuracy of predictions. No events have been excluded from the test data set. This means that even events with a large delay which are typically associated to serious disruptions and are symptoms of very different operations (see for instance Corman et al. (2014a)), are kept in the data used to evaluate the performance of the presented model. Fig. 5 shows the delay distribution of events included in the test set. Most events occur with a small delay, however delays may extend up to 50 min off schedule.

4.3. Modelling of passenger and freight train delays

The passenger trains operate strictly according to a timetable. Their deviation from the scheduled paths is relatively small and therefore they frequently follow the scheduled orders, thus providing sufficient amount of data in the training set to build the robust estimates of the distribution parameters. On the other hand, freight trains often significantly deviate from their schedule and they are often running on the ad hoc created paths that do not recur frequently. Fig. 6 illustrates the distribution of delays of passenger and freight trains from the training set. The standard deviation of freight train delays is significantly higher compared to the delays of passenger trains (64.19 to 11.20 min, respectively). Consequently, the delay dependencies between freight trains, operating along ad hoc paths, and other trains are difficult to capture from historical data due to their low frequency of occurrence. More details about the practice of ad hoc scheduling of freight trains in Sweden could be found in Törnquist Krasemann (2015).

The interaction between ad hoc freight train paths and successive and preceding trains could be modelled using the dependence

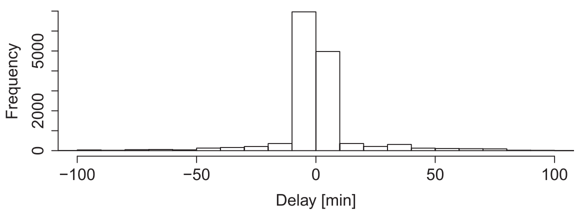

Fig. 5. Delay distibution for events included in the test set.

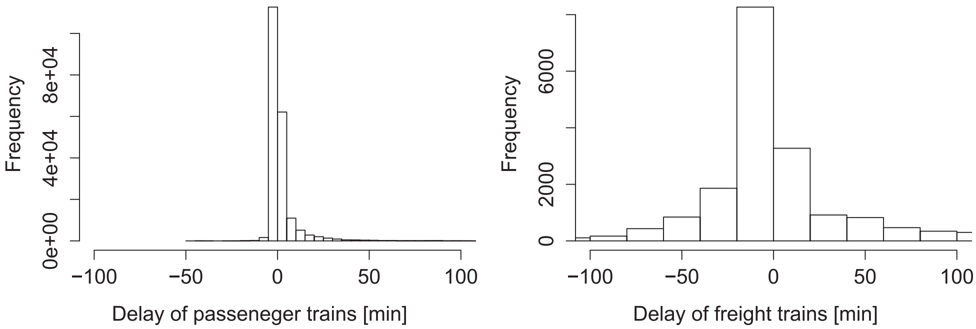

Fig. 6. Delay distributions for passenger (left) and freight (right) trains.

between the planned headway time between two events and the resulting delay increase for the second train. A delay increase is computed as a difference in delay between two successive events of a single train. Delay increase for all trains which may be delayed due to a hindrance from a freight train were extracted from the data and compared with the observed headway time between the two trains. The results are presented in Fig. 7. As anticipated there is a general trend that delay increase is higher and more likely to happen for short headway time after a freight train. However, regression analysis showed no significant impact of headway time on delay increase with the value of R^2 indicating that only 8% of variation of a delay increase can be explained by the short headway time after the preceding train. In fact the analysis showed that mean and median of headway times after an inserted freight train path are 15.85 and 12.43 min, respectively. This could be explained by the fact that traffic controllers carefully choose the path for ad hoc freight trains so that the scheduled traffic is affected as little as possible. For that reason we exclude the headway arcs between the events of out-of-schedule freight trains and the preceding and succeeding passenger trains from the model. Similar analysis is required for model applications on mixed-traffic corridors with a higher share of ad hoc freight trains. Such trains should be explicitly included in the network with the corresponding headway arcs constructed between them and other trains.

5. Results

We now evaluate the approach proposed on the test case identified. We specifically analyse the performance along 4 dimensions, as follows. Section 5.1 describes the computational characteristics of the models itself, in terms of nodes and arcs, and fitting accuracy. Section 5.2 reports on the absolute performance in prediction power of events in the future, for varying time horizons Section 5.3 compares different prediction algorithms in the state of the art, to predict the same event under the same conditions. Section 5.4 concludes by showcasing one of the key feature of Bayesian networks, namely the result of a dynamic update of the prediction: how the predicted distribution of probability changes over time as the event gets closer to the moment "now".

5.1. Model structure, size and parameters

The Bayesian network in the experimental setup contains the nodes that represent the train events planned to occur within one hour. An update about a realised event time is therefore propagated one hour in the future which is the longest prediction horizon we consider. The average network size is 137.12 nodes and 262.43 arcs. The average number of children per node is therefore 1.92 which indicates that approximately two explanatory variables are needed to compute the distribution parameters for each node.

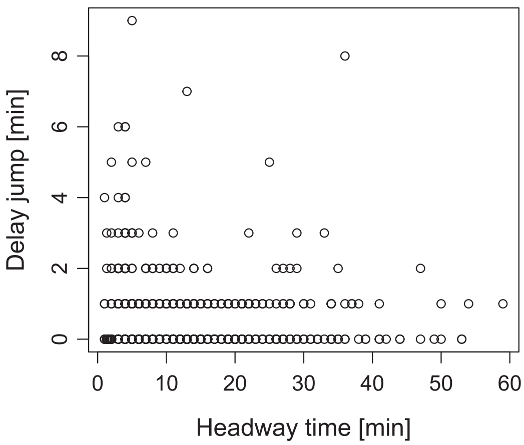

Fig. 7. Delay increase depending on headway time after the conflicting event.

Table 3
Average values of predictive power of local models.

Arc type	Coefficient	RSE	p value	R^2
Headway	0.42	0.33	***	0.52
Running	0.95	0.06	***	0.90
Dwell	0.81	0.09	***	0.79

Table 3 shows the predictive power of local linear models averaged over the 2194 events considered in the test set. The importance of each explanatory variable is considered separately in order to analyse the correlation of delay increase during different processes. The table shows correlation coefficients, residual standard error (RSE), p value as an indicator of variable importance ($***$ means $p < 0, 01$ which indicates very strong importance) and R^2 which presents the percentage of variance explained by the predictor. The strongest correlation is captured between the delays of events associated with a running process of a train. Arrival delay of a train is also a good predictor of its departure delay from the same station (dwell process). However, lower delay correlation between an arrival and the subsequent departure event reflects the fact that dwell times are more difficult to predict and may act as a source of delay as well as a delay buffer (Kecman and Goverde, 2015). A delay of the preceding conflicting event (headway process) is also an important predictor although with a weaker predictive power (lower value of R^2). The average multiple R^2 with complete local models explains 92% of delay variation.

5.2. Prediction model performance

This section reports the prediction accuracy of the model when applied on the peak hours (6:30–9:00 and 16:30–19:00) of the test day. After the observation of each train event in the specified period, the algorithm predicts the future traffic evolution in the next hour. In total the prediction algorithm is executed 563 times, each time performing on average 137.12 predictions. The predicted values are compared against the realised event times and the distribution of prediction error for all predictions is given in Fig. 8. The box-plot indicates the median (line in the middle of the box), the 1st and the 3rd quartiles (upper and lower bound of the box) and data maximum and minimum (ends of the upper and lower whisker). Despite the outliers in prediction errors, which are not excluded from the analysis, the plot shows a high prediction accuracy of the Bayesian network model.

The impact of the prediction horizon on the prediction accuracy can be observed by separately analysing the prediction error for each prediction horizon. The prediction horizon of 60 min is divided into 1 min wide intervals. The absolute prediction error is computed as the absolute value of the difference between the actually realised event time and the predicted event time. Mean absolute error (MAE) is obtained in each interval by computing the mean value of all corresponding absolute prediction errors. Figs. 9 and 10 respectively show the MAE (as average, median, and first and third quartile), and standard deviation, for each considered prediction horizon. As expected, both MAE and standard deviation decrease as the smaller prediction horizon is considered. The accuracy of predictions that are within a 30 min prediction horizon is significantly increased since more accurate information is available on events that have a direct impact on the realization time of an event. While average and mean increase for longer time horizons, also the gap (i.e. the area of the band in Fig. 9) between the first and the third quartile increases, (and similarly the standard deviation ins Fig. 10) reporting the lower degree of confidence on the values predicted, and not only a higher error, but also a higher variability of the error. Interestingly, the median error increases very slowly, compared to the average, showing that the bulk of the predictions keep remaining correct, while the large deviations have more difficulty to be predicted. Anyway, for longer prediction horizons, both MAE and standard deviation of error increase, indicating that the prediction accuracy is lower and that a significant amount of uncertainty remains about the event times of events more than 40 min ahead.

5.3. Comparison of different prediction approaches

We also compare different algorithms from the state of the art of dynamic prediction, in Fig. 11. Namely, we compare the approach proposed using the Bayesian network to encode the structure of the dependencies (named *Bayesian* in Figure), with a variety of other approaches. One is to consider only dependencies between the events of the same train, but deterministically. In other terms, this approach (named *Propagation* in Figure) assumes a train in the future would keep the current delay. We also report two

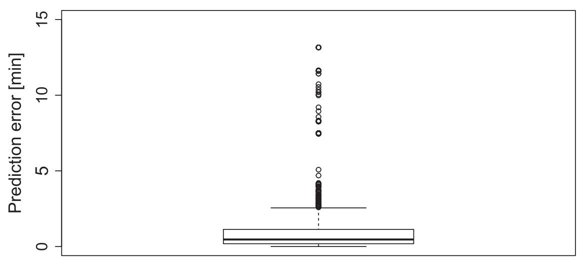

Fig. 8. Distribution of prediction error for all horizons.

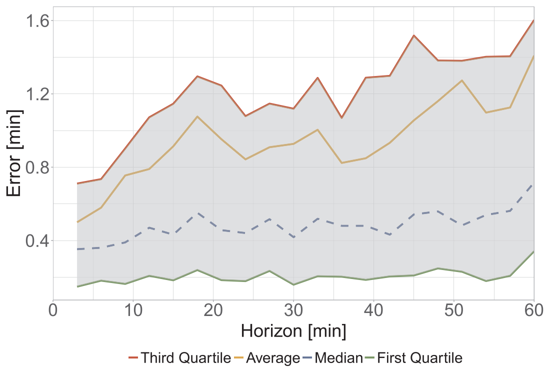

Fig. 9. MAE for all considered prediction horizons. The band represent 1st and 3rd quartile, median (dotted) and average (solid) are reported as lines.

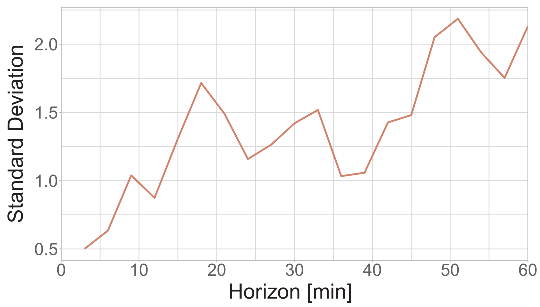

Fig. 10. Standard deviation of error for all considered prediction horizons.

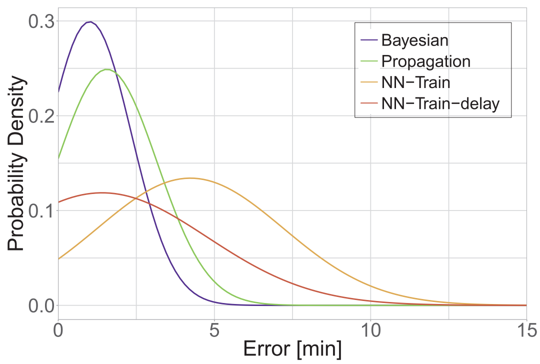

Fig. 11. Comparison of different prediction approaches.

different approaches, which perform a nearest neighbor search in the space of the recorded data, to deliver the prediction (based on the same ideas as (Böhmová et al., 2015), to find events happened in the past, which showed similar characteristics of the current moment). This works as follows. We assume a future event e_i described by a train number t, at a station, and event type, is to be predicted, given a current event e_j. The predicted delay of e_i is computed as the average of the delay of all events e_k recorded in the past, which have either (1) the same train number at the same station (named *NN-Train* in Figure); or (2) the same train number at the same station, while the delay associated to e_k is the same (or the most similar) as the delay associated to event e_j, this latter corresponding to the most recent observation of the run of the train of both events e_i and e_j. This latter Nearest Neighbor structure is named *NN-Train-delay* in Figure. We also tried other Nearest Neighbor structures based on time intervals, which did not perform better than those presented.

Fig. 11 reports the kernel estimate of the probability distributions of the error, for all events, such that the time horizon of prediction between events e_i and e_j is between 35 and 40 min. A curve more on the left side of the Figure represents smaller errors, in probabilistic terms. From this point of view, the advantage of the Bayesian network against the other approaches is evident, delivering a higher probability of smaller errors. It is interesting to see how the actual information on the delay (in deterministic terms, the propagation approach); in statistic terms, the difference between NN-train and NN-train-delay is actually very important. This underlines the strength and potential of dynamic approaches which are able to combine past information on delays, with current

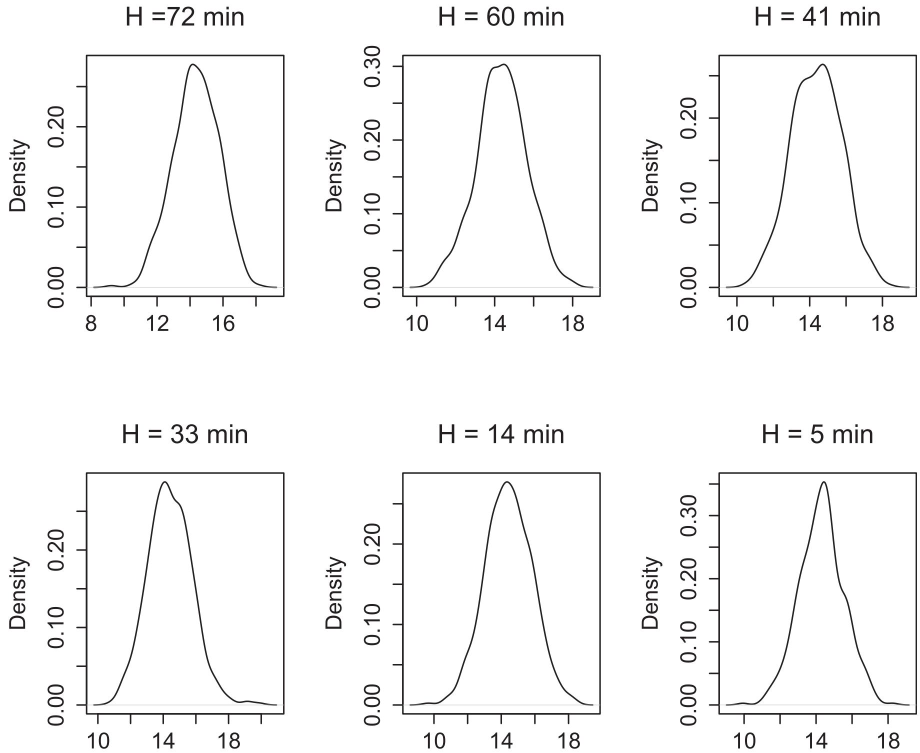

Fig. 12. Distribution of an arrival time in discrete steps between 72 and 5 min ahead of the event.

(online) information on delay, such as the Bayesian approach, which outperforms all other approaches presented here.

5.4. Dynamic updates of probability distributions

Finally, we analyse the dynamics of probability distribution of an event over time as more information about the event becomes available. Bayesian inference is used to compute the posterior distributions (Korb and Nicholson, 2010). Fig. 12 shows an example of how the distribution of arrival time of a train to the final station evolves over time in six discrete steps. As the event becomes closer (horizon H decreases), the tendency is that the standard deviation becomes smaller thus achieving sharp distributions that converge toward a 1-point distribution at the moment when the event is realised. On the other hand, the expected value of the distribution fluctuates a bit over its final value.

Taking a slice for this same event, we plot the evolution of the probability that the train will arrive with more than 16 min delay is depicted in Fig. 13. In this particular example the probability is decreasing in time as the event becomes closer. The actually observed delay of the event is 15 min.

In Figs. 14 and 15 we report a larger computational study, which we performed on a random sample of 1000 events in the

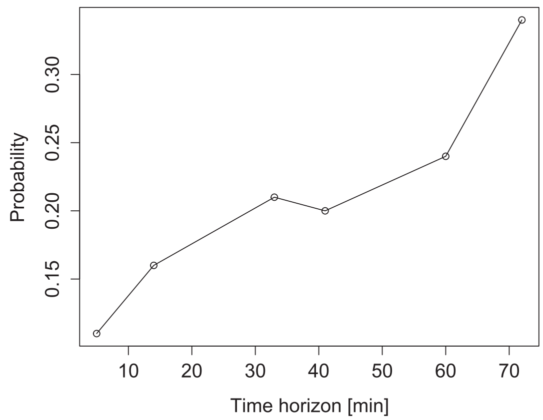

Fig. 13. Dynamic updates of probability that the train will arrive with more than 16 min delay.

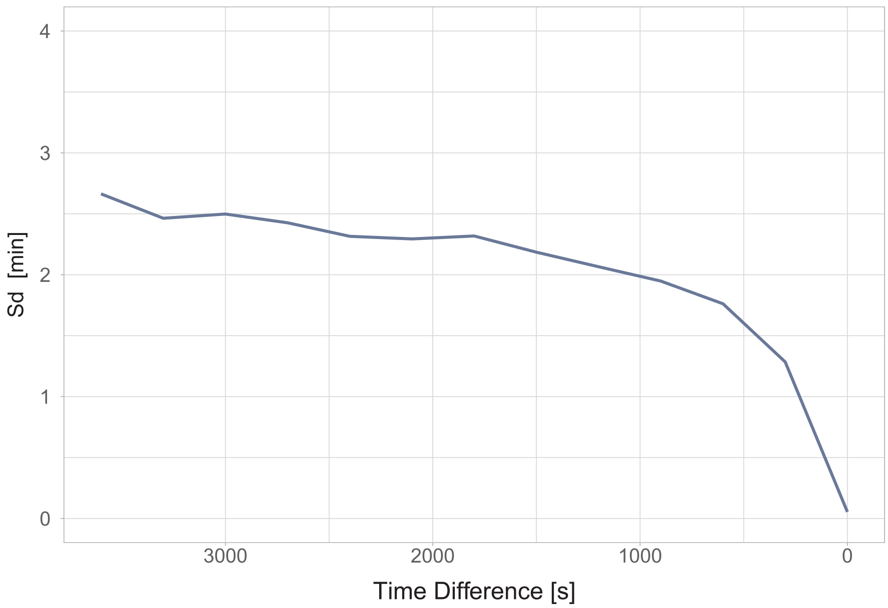

Fig. 14. Standard deviation along time, of the probability that 1000 random events will be happening with more than 5 min delay.

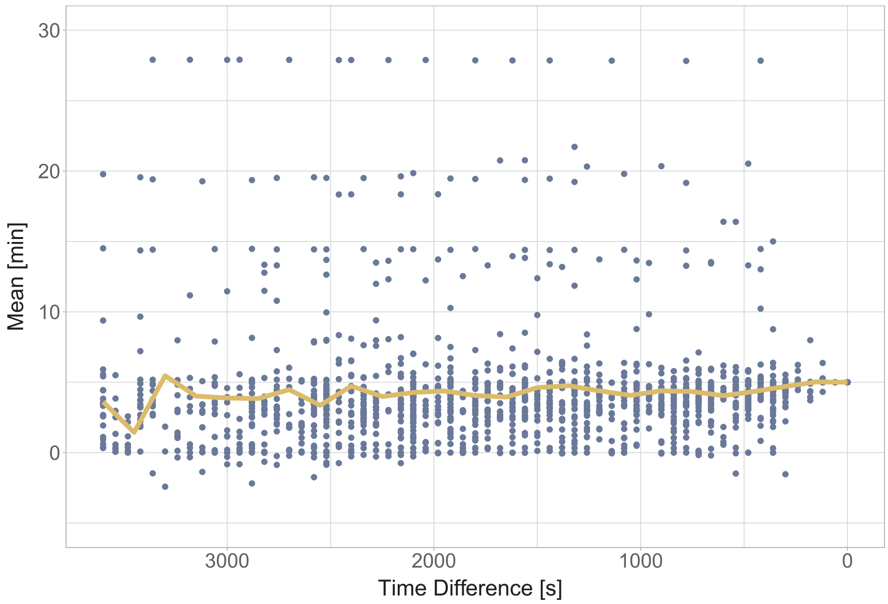

Fig. 15. Averages along time, of the probability that 1000 random events will be happening with more than 5 min delay (dots), plus smoothed average (solid line).

network. For each event, we computed the posterior distribution, conditional to the assumption that the final delay would be 5 min. We report the averages over all various probability densities. Both Figures report on the x-axis the amount of time difference before the event takes place, in seconds. In other terms, at the left extreme of the x -axis, the event is one hour ahead in the future, while at the right end of the x-axis the event has just happened, and the probability has no variance and a fixed value equal to 5 min. In Fig. 14 we report the standard deviation of each probability density, averaged over all events, by their time difference. One can see how quantitatively the variance decreases over time, with a particular shape. In our case, the standard deviation of the probability of occurrence of an event, as predicted half an hour ahead of time, is for instance 2 min, which is within the range of values useful for decision making, comparing to a similar discussion in (Oneto et al., 2018). It would be an interesting question for future research to characterize stochastic processes by their variability of variance over time and possibly derive some theoretical results on quasi stationarity.

In Fig. 15 we report instead the average of each probability density, as blue dots, and their smoothed average as a continuous line.

One can see how the average can sometimes be relatively worse off, but converges to the realised delay of 5 min. One can also see how, apart from minor variations, the estimator is unbiased, i.e. the average (solid line) of all averages of predictions (each one, a blue point) is actually very close to the 5 min which is the assumed event delay.

6. Conclusions

This paper presented an analysis of train delays and their evolution in real time with respect to the dynamic stochastic phenomena. The basic idea was to characterize the effect that the prediction horizon and incoming information about running trains may have on the probability of the future train delays. Bayesian networks are proved to be an appropriate method to concisely represent the complex interdependencies between train events. The major contribution of this paper is that train delays due to interactions with other trains are adequately represented in the stochastic model.

The presented method provides a way to incorporate the value of information from a live data stream into prediction of future events. A key feature for such an online learning approach is the possibility to perform good predictions under non-recurrent disruptions. That is an improvement compared to conventional prediction approaches based solely on the fixed values obtained offline from the historical data. The disruption of operation of one train causes an update of predictions for all possibly affected trains. The model was evaluated in a simulated real time environment and the computational results indicate that the predictions are reliable for horizons of up to 30 min.

The practical application of this method could increase the amount of information delivered to passengers, in the form of up-to-date probability for on-time arrival. It is shown by many policy studies that an informed passenger is more likely to accept this delay, and giving the probability margins could be an additional feature of projected travel time planners. Being able to characterize, analyse and predict the unavoidable dynamic uncertainty of process times can also result in better railway traffic planning and control and the corresponding tools.

A strong assumption of the model is that there is a perfect knowledge of train orders and routes within the prediction horizon. Event though the graph structure is convenient for representing the changes of routes and orders, the applicability of the model still to a great extent depends on the incoming information from traffic controllers. This makes it somewhat difficult to develop an independent application based on the presented model that would function outside of the traffic control loop. The future work in this direction will be dedicated to overcoming this drawback and to develop a tool that would effectively predict the traffic control actions and therefore be able to independently estimate the future traffic evolution.

Finally, the presented methodology is validated on a busy corridor. Extending the model to handle networks, that may contain multiple intersecting corridors, seems straightforward having in mind the earlier applications of macroscopic modelling that are able to represent complete national networks. However, such expansion of the model may lead to an increase in computation times of the inference algorithms. A challenge for the future work thus remains to examine the upper bound on model size that can still be used in a real-time environment. In fact, this opens up various directions of future research: studying alternative prediction or statistical inference algorithms, which could also be domain-unaware, and learn the dynamics of the system based on recorded or simulated operations, similarly to (Oneto et al., 2017), considering simpler, more approximated and scalable models of operations; studying how performance might be affected by different timetable structures, different levels of heterogeneity, speed and traffic density; study parametric online setups which adapt the Bayesian networks along time with a minimum computational effort; and/or considering strategies for model decomposition and coordination.

Acknowledgments

This study is supported by FP7 European project Capacity4Rail and State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University (RCS2015K002, RCS2015K003). The authors are grateful to Swedish transport administration Trafikverket for providing the data sets necessary to perform this research, and to Alessio Daniele Marra for his valuable help.

References

Andersson, E.V., Peterson, A., Törnquist-Krasemann, J., 2014. Quantifying railway timetable robustness in critical points. J. Rail Transport Plann. Manage. 3 (3), 95–110.
Barbour, W., Mori, J.M., Kuppa, S., Workac, D., 2018. Prediction of arrival times of freight traffic on us railroads using support vector regression. Transp. Res. Part C 93, 211–227.
Bauer, R., Schöbel, A., 2014. Rules of thumb: practical online-strategies for delay management. Public Transport 6 (1–2), 85–105.
Bayissa, T.F., 2013. How do Delays Along the Way Influence Delays to Final Destination? Master thesis, Department of Civil and Transport Engineering, Norwegian University of Science and Technology, Trondheim.
Berger, A., Gebhardt, A., Müller-Hannemann, M., Ostrowski, M., 2011. Stochastic delay prediction in large train networks. In: Caprara, A., Kontogiannis, S., (Eds.), 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, Dagstuhl, pp. 100–111.
Böhmová, K., Mihalák, M., Neubert, P., Pröger, T., Widmayer, P., 2015. Robust routing in urban public transportation: Evaluating strategies that learn from the past. In: Proc. 15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS), pp. 68–81.
Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M., Schlechte, T. (Eds.), 2018. Handbook of Optimization in the Railway Industry. Springer, Berlin.
Büker, T., Seybold, B., 2012. Stochastic modelling of delay propagation in large networks. J. Rail Transport Plann. Manage. 2 (1-2), 34–50.
Burdett, R., Kozan, E., 2014. Determining operations affected by delay in predictive train timetables. Comput. Operat. Res. 41, 150–166.
Caimi, G., Fuchsberger, M., Laumanns, M., Lüthi, M., 2012. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. Comput. Operat. Res. 39 (11), 2578–2593.
Corman, F., Meng, L., 2014. A review of online dynamic models and algorithms for railway traffic rescheduling. IEEE Trans. Intell. Transp. Syst. 16 (3), 1274–1284.
Corman, F., Quaglietta, E., 2015. Closing the loop in real-time railway control: Framework design and impacts on operations. Transport. Res. Part C: Emerging

Technol. 54 (0), 15–39.
Corman, F., D'Ariano, A., Hansen, I.A., 2014a. Evaluating disturbance robustness of railway schedules. J. Intell. Transport. Syst. 18 (1), 106–120.
Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2014b. Dispatching and coordination in multi-area railway traffic management. Comput. Operat. Res. 44, 146–160.
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.
Dolder, U., Krista, M., Voelcker, M., 2009. RCS – Rail Control System – Realtime train run simulation and conflict detection on a net wide scale based on updated train positions. In: Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis (RailZurich2009), Zurich, pp. 1–15.
Dollevoet, T., Huisman, D., Kroon, L.G., Schmidt, M., Schöbel, A., 2014. Delay management including capacities of stations. Transp. Sci. 49 (2), 185–203.
Gatto, M., Jacob, R., Peeters, L., Widmayer, P., 2007. Online delay management on a single train line. In: Geraets, F., Kroon, L., Schöbel, A., Wagner, D., Zaroliagis, C. (Eds.), Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science, vol. 4359. Springer, Berlin Heidelberg, pp. 306–320.
Ghofrani, F., He, Q., Goverde, R., Liu, X., 2018. Recent applications of big data analytics in railway transportation systems: A survey. Transp. Res. Part C 90, 226–246.
Goverde, R.M.P., 2007. Railway timetable stability analysis using max-plus system theory. Transport. Res. Part B: Methodol. 41 (2), 179–201.
Goverde, R.M.P., 2010. A delay propagation algorithm for large-scale railway traffic networks. Transport. Res. Part C: Emerging Technol. 18 (3), 269–287.
Hansen, I.A., Pachl, J. (Eds.), 2014. Railway Timetabling & Operations – Analysis, Modelling, Optimisation, Simulation, Performance Evaluation, second ed. Eurailpress, Hamburg.
Jovanović, P., Kecman, P., Bojović, N., Mandić, D., 2017. Optimal allocation of buffer times to increase train schedule robustness. Eur. J. Oper. Res. 256 (1), 44–54.
Kecman, P., 2014. Models for predictive railway traffic management. Ph.D. thesis, Delft University of Technology, Delft.
Kecman, P., Goverde, R.M.P., 2014. Online data-driven adaptive prediction of train event times. IEEE Trans. Intell. Transp. Syst. 16 (1), 465–474.
Kecman, P., Goverde, R.M.P., 2015. Predictive modelling of running and dwell times in railway traffic. Public Transport 7 (3), 295–319.
Kecman, P., Corman, F., Meng, L., 2015. Train delay evolution as a stochastic process. In: Proceedings of the 6th International Seminar on Railway Operations Modelling and Analysis (RailTokyo2015), Tokyo, pp. 1–19.
Keyhani, M.H., Schnee, M., Weihe, K., Zorn, H.P., 2012. Reliability and delay distributions of train connections. In: Delling, D., Liberti, L. (Eds.), 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASIcs), vol. 25, pp. 35–46.
Koller, D., Friedman, N., 2009. Probabilistic Graphical Models. Principles and Techniques. MIT Press, Cambridge.
Korb, K.B., Nicholson, A.E., 2010. Bayesian Artificial Intelligence, second ed. Chapman & Hall/CRC Computer Science & Data Analysis, CRC Press, Inc., London.
Lee, W.H., Yen, L.H., Chou, C.M., 2016. A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services. Transport. Res. Part C: Emerging Technol. 73, 49–64.
Lemnian, M., Rückert, R., Rechner, S., Blendinger, C., Müller-Hannemann, M., 2014. Timing of train disposition: Towards early passenger rerouting in case of delays. In: Funke, S., Mihalák, M. (Eds.), 14th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, OpenAccess Series in Informatics (OASIcs), pp. 122–137.
Lessan, J., Fu, L., Wen, C., 2018;al., in press. A hybrid bayesian network model for predicting delays in train operations. Comput. Ind. Eng. https://doi.org/10.1016/j.cie.2018.03.017. (in press).
Marković, N., Milinković, S., Tikhonov, K.S., Schonfeld, P., 2015. Analyzing passenger train arrival delays with support vector regression. Transport. Res. Part C: Emerging Technol. 56 (0), 251–262.
Mascis, A., Pacciarelli, D., 2002. Job-shop scheduling with blocking and no-wait constraints. Eur. J. Operat. Res. 143, 498–517.
Medeossi, G., Longo, G., de Fabris, S., 2011. A method for using stochastic blocking times to improve timetable planning. J. Rail Transport Plann. Manage. 1 (1), 1–13.
Meester, L.E., Muns, S., 2007. Stochastic delay propagation in railway networks and phase-type distributions. Transport. Res. Part B: Methodol. 41 (2), 218–230.
Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. Transport. Res. Part B: Methodol. 67 (0), 208–234.
Murphy, K.P., 2002. Dynamic bayesian networks: representation, inference and learning. Ph.D. thesis, University of California, Berkeley.
Nagarajan, R., Scutari, M., Lèbre, S., 2013. Bayesian Networks in R. Springer, New York.
Nash, A., Huerlimann, D., 2004. Railroad simulation using OpenTrack. In: Allan, J., Brebbia, C.A., Hill, R.J., Sciutto, G., Sone, S. (Eds.), Computers in Railways IX. WIT Press, Southampton, pp. 45–54.
Nielsen, L.K., Kroon, L., Maróti, G., 2012. A rolling horizon approach for disruption management of railway rolling stock. Eur. J. Oper. Res. 220, 496–509.
Oneto, L., Fumeo, E., Clerico, G., Canepa, R., papa, F., Dambra, C., Mazzino, N., Anguita, D., 2017. Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout. IEEE Trans. Syst. Man Cybernet. 47 (10), 2754–2767.
Oneto, L., Fumeo, E., Clerico, G., Canepa, R., papa, F., Dambra, C., Mazzino, N., Anguita, D., 2018. Train delay prediction systems: A big data analytics perspective. Big Data Res. 47, 54–64.
Potthoff, D., Huisman, D., Desaulniers, G., 2010. Column generation with dynamic duty selection for railway crew rescheduling. Transport. Sci. 44 (4), 493–505.
Quaglietta, E., 2014. A simulation-based approach for the optimal design of signalling block layout in railway networks. Simul. Model. Pract. Theory 46, 4–24 simulation-Optimization of Complex Systems: Methods and Applications.
Quaglietta, E., Corman, F., Goverde, R.M.P., 2013. Stability analysis of railway dispatching plans in a stochastic and dynamic environment. J. Rail Transport Plann. Manage. 3 (4), 137–149.
Sahin, I., 2017. Markov chain model for delay distribution in train schedules: Assessing the effectiveness of time allowances. J. Rail Transport Plann. Manage. 7 (3), 101–113 best Papers of the RailLille 2017 conference.
Schöbel, A., 2007. Integer programming approaches for solving the delay management problem. In: Geraets, F., Kroon, L., Schöbel, A., Wagner, D., Zaroliagis, C. (Eds.), Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science, vol. 4359. Springer, Berlin/Heilderberg, pp. 145–170.
Törnquist, J., Persson, J., 2007. N-tracked railway traffic re-scheduling during disturbances. Transport. Res. Part B: Methodol. 41 (3), 342–362.
Törnquist Krasemann, J., 2015. Configuration of an optimization-based decision support for railway traffic management in different contexts. In: Proceedings of the 6th International Seminar on Railway Operations Modelling and Analysis (RailTokyo2015), Tokyo, pp. 1–20.
Van Thielen, S., Corman, F., Vansteenwegen, P., 2018. Considering a dynamic impact zone for real-time railway traffic management. Transp. Res. Part B 111, 39–59.
Wei, D., Liu, H., Qin, Y., 2015. Modeling cascade dynamics of railway networks under inclement weather. Transport. Res. Part E: Logist. Transport. Rev. 80, 95–122.
Yaghini, M., Khoshraftar, M.M., Seyedabadi, M., 2013. Railway passenger train delay prediction via neural network model. J. Adv. Transport. 47 (3), 355–368.
Yuan, J., Hansen, I.A., 2007. Optimizing capacity utilization of stations by estimating knock-on train delays. Transport. Res. Part B: Methodol. 41 (2), 202–217.