Forecasting Passenger Loads in Transportation Networks

Stefan Haar, Simon Theissing

Forecasting Passenger Loads in Transportation Networks

Stefan Haar, Simon Theissing

MExICo team, INRIA and LSV, CNRS & ENS de Cachan
Cachan, France

Abstract

This work is part of an ongoing effort to understand the dynamics of passenger loads in modern, multimodal transportation networks (TNs) and to mitigate the impact of perturbations. The challenge is that the percentage of passengers at any given point of the TN that have a certain destination, i.e. their distribution over different trip profiles, is unknown. We introduce a stochastic hybrid automaton model for multimodal TNs that allows to compute how such probabilistic load vectors are propagated through the TN, and develop a computation strategy for forecasting the network's load a certain time into the future.

Keywords: Stochastic hybrid automata, Transportation networks, Fokker-Planck Equation

1 Introduction

We continue here the work begun in [6] for capturing both the discrete vehicle movements and continuous passenger transfers in a multimodal public transportation network (TN). In [6], a deterministic hybrid automaton (DHA) model was used, so as to overcome via fluidification the state space explosion that makes fully discrete models intractable. For the specification, we used both discrete and continuous Petri nets (PNs) as basic modelling blocks [4], where the marking of the continuous places and the flows between them were vectorial instead of scalar. In fact, we integrated the numbers of passengers belonging to different trip profiles, i.e. having different destinations, as components of vector markings and -flows, with routing matrices relating them.

Now - and this is the starting point of the present work - a *real* TN is everything but deterministic. On the one hand, there are highly unpredictable asynchronous events for which statistical data is hard to obtain. It is thus difficult to include them in the daily network operation [11], e.g. by means of minute-by-minute or hourly forecasts. Typical examples are passenger incidents. Now, note that - apart from few exceptions - such incidents originate locally, in one mode or line, and then propagate to other modes or lines by passenger transfers. These transfers are predictable, not necesseraly deterministic, if one knows the destination or trip

profile of the passengers; but in general, this can only be known through probabilistic estimates. Finally, there are the "continuous" passenger arrival processes for which statistical data is easier to obtain: How many passenger will arrive at a station at what time? According to which route, including which vehicle missions, will they travel?

Here, we will extend the DHA model from [6] in that we will replace all deterministic passenger arrival processes by their stochastic counterparts, and, in doing so, introduce a stochastic hybrid automaton (SHA) model with jumps between its discrete modes, at a priori equidistantly-spaced discrete points in time, defined a priori. The literature reveals many predecessors of our SHA model, notably in the past two decades, with every approach introducing the uncertainty at a different point in the model dynamics. For instance, the authors of [13] extended the dynamics underlying a PN-DHA model in that the jumps between the discrete modes are either exponentially distributed or immediate; with a weighting function as a means to resolve conflicts among simultaneously enabled immediate transitions [1]. However, in this modelling approach the discrete jumps are decoupled from the continuous states, and the latter evolve according (acc.) to deterministic differential balance equations; the authors of [7] bridge the gap between the continuous states and the mode jumps by means of a guard function. Finally, the deterministic balance equations were replaced by normally distributed balance equations in [14]. Outside the framework of PNs, the authors of [8] introduced an SHA model that exhibits state-driven (forced) jumps between the discrete modes subjected to sets of stochastic differential equations (SDEs), one such set per mode. This approach was extended in [3] in that the mode transitions are no longer limited to forced jumps, but can be initiated by spontaneous jumps with state-dependent transition rates as well. The author of [2], then showed how the SHA model from [3] can be formulated in an equivalent system of integro-differential equations together with boundary conditions. Also notice that the authors of [10] presented a grid-based asymptotic approximation method for a backward reachability problem subjected to the dynamics of an SHA model that encounters spontaneous jumps between its discrete modes; with a system of SDEs assigned to every mode. That system is approximated by a Markov chain following a space and subsequent time discretization; whereas in our approach the discretization of the time precedes the discretization of the space, and the latter comes along with a numerical integration of the continuous states in a discrete mode.

In the rest of this paper, we will discuss the specification of TN's infrastructure in our vectorial SHA model, and the vehicle operation as well as the routing of all passenger flows thereon (Sec. 2 on p. 3). We will then elaborate the SHA model's time-continuous dynamics, before pinpointing all mode transition times to an equidistantly-spaced mesh, which can be regarded as a first major step to render forecasts of the model's hybrid state feasible (Sec. 3 on p. 8). Next, we will integrate the SHA model's discrete-time approximation into a computation strategy which can be used to forecast TN's passenger loads (Sec. 4 on p. 16). Finally, we will provide a short summary and an outlook on future work (Sec. 5 on p. 19).

2 Capturing the Transportation Network at Hand

2.1 TN's Infrastructure

We capture TN's infrastructure in our automaton model by a finite set of *transportation grids* that accommodate TN's different lines and transportation modes (Sec. 2.1.1), another finite set $\mathcal{S}$ of *stations* (Sec. 2.1.2 on p. 4), and an *interface* that defines all passenger transfer possibilities between the vehicles operated in the transportation grids and the stations (Sec. 2.1.3 on p. 4).

2.1.1 A Transportation Grid

Every transportation grid accommodates the lines of a particular transportation mode such as the rail grid of a metro system: It is made up of a finite set of discrete *waypoints* that are connected by *tracks*. In particular, a track that connects the waypoint w_1 to another waypoint w_2 indicates the possibility of a vehicle movement from w_1 to w_2; no more (such as a length, a curvature, or a slope) and no less. Thus, every pair of a waypoint is connected by at maximum two tracks, namely one track in each direction. Since by convention every waypoint moreover is either empty or occupied by at maximum one vehicle at a time, then also note that conflicting vehicle movements can occur iff two or more vehicles try to access the same waypoint at the same time.

We specify a particular transportation grid, say g, in form of a bipartite graph, in which all waypoints in g are represented as simple circles and all tracks as simple boxes. One edge then connects w_1 to the track t and another edge t to w_2 iff t shall implement the possibility of a vehicle movement from w_1 to w_2. Moreover, we enumerate every track in order to specify the resolution of all possible conflicting vehicle movements in a deterministic manner.

As an example, consider Fig. 1.

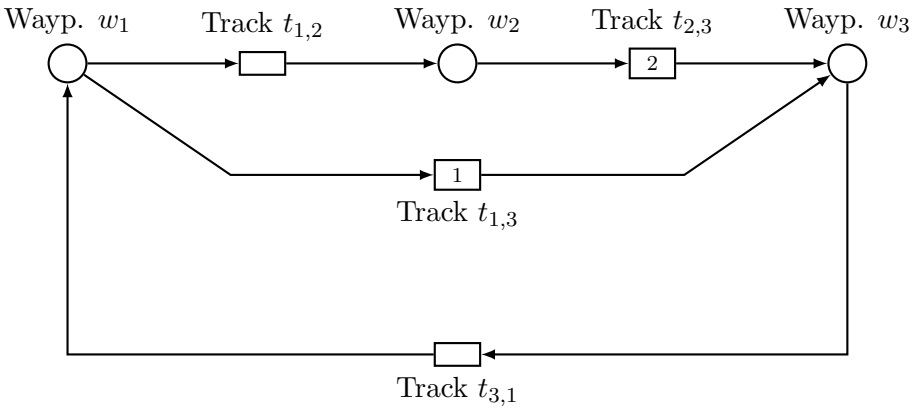


Fig. 1. Orbital transportation grid composed of three waypoints (simple circles) and four tracks (simple boxes): conflicting vehicle movements that might run together at w_1 are resolved in a deterministic manner by means of the unique integer numbers inscribed to $t_{2,3}$ and $t_{1,3}$ if necessary

It schematically depicts the transportation grid for the orbital line of a people mover: A vehicle at w_1 can go to w_2 via $t_{1,2}$, from w_2 to w_3 via $t_{2,3}$, and from w_3 back to w_1 via $t_{3,1}$, respectively. Alternatively, a vehicle at w_1 can take $t_{1,3}$, and thus directly move from w_1 to w_3 without the necessity to pass by w_2. Then note that

a conflict between two vehicle movements might occur iff one vehicle, say a_1, wants to access w_3 from w_2 via $t_{2,3}$, and at the same time another vehicle, say a_2, wants to access w_3 from w_1 via $t_{1,3}$. Assuming this to be true, we privilege a_2 over a_1 due to the fact that a vehicle movement taking $t_{1,3}$ has a higher priority (lower integer number inscribed to $t_{1,3}$) than another vehicle movement taking $t_{2,3}$.

2.1.2 A Station

Every station $s \in \mathcal{S}$ is made up of a finite set of capacity-limited gathering points (GPs) that accommodate the transferring and waiting passengers, and another finite set of elements, called corridors [1], with each corridor in s implementing the possibility of a passenger movement away from/to a GP in s w.r.t. another GP or the outdoor area of s. Once again, we use a bipartite digraph to capture the connection between all GPs and all corridors in s. In particular, every vertex either captures a particular GP in s (double circle) or a particular corridor (double box); and not a waypoint or a track.

As an example, look at the simple station from Fig. 2.

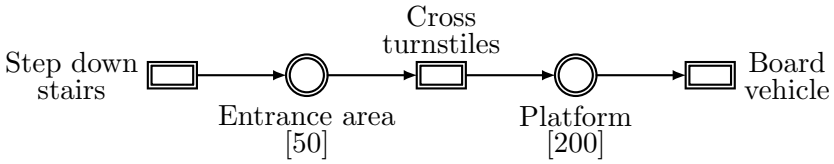


Fig. 2. Sample station with two capacity-limited gathering points (double circles) and three corridors (double boxes): the entrance area can hold up to 50 passengers and the platform 200 passengers

The station has two GPs and three corridors: Among the two GPs there is one entrance area that can accommodate up to 50 passengers (written in brackets next to it), and one platform that can accommodate up to 200 passengers. Moreover we can say that a passenger at the entrance area can transfer to the platform since a corridor connects both in the right direction. The unambiguous interpretation of the remaining two corridors depends on the context of the station, though; i.e., on the interface specification between this station and all transportation grids of TN; see below. Nevertheless, from the labels written next to all corridors we can anticipate that the entrance area accommodates all passengers who enter TN through this station, and that all passengers who leave the platform do so in order to board a vehicle.

2.1.3 Connecting the Transportation Grids and the Stations

The interface between the transportation grids and the stations defines which corridor in a station is connected to which waypoint in a transportation grid. In doing so, it defines all passenger transfer possibilities between the GPs of the stations and the vehicles that are stopped at particular waypoints in the transportation grids in our SHA model.

As an example, look at Fig. 3.

[1] We can associate with a corridor a limited throughput for a directed flow of passengers that is conserved between two discrete points; and this metaphor perfectly fits into our gas dynamic approach for all passenger flows

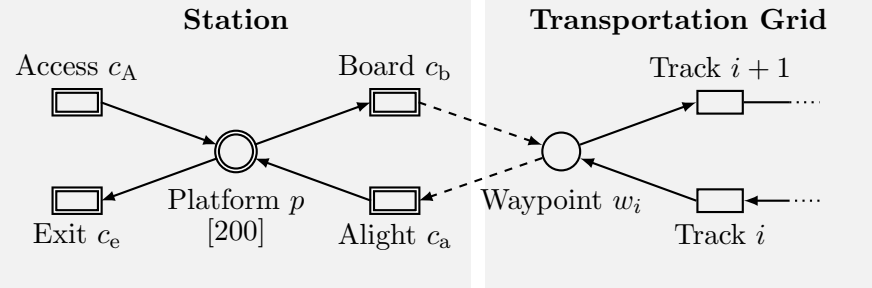


Fig. 3. Every dashed edge specifies the possibility of a passenger transfer from a GP in a station to a vehicle stopped at a waypoint in a transportation grid (through a corridor), or vice versa

It depicts the interface between a simple station, say s, and an extract of a transportation grid: If a vehicle a is stopped at the waypoint w_i, then all passengers on-board a can alight from it to the single platform p in s through c_a. Similarly, the passengers can board a from p through c_b.

In general, the interface between all stations and transportation grids is specified in form of a set of edges that connects the corresponding graphs, in which every edge either connects a corridor c in a station to a waypoint w in a transportation grid indicating the possibility of a passenger transfer from the single parent vertex of c (= GP) to a vehicle that is stopped at w, or vice versa.

We refer to this combined graph (of all stations, transportation grids, and the interface between both) as the infrastructure of our model, and demand that the passengers can access a vehicle a that is stopped at a particular waypoint only from one GP of a particular station, say s, if at all. Accordingly, all passengers on-board can alight from a to one (not necessarily the same) GP of s if at all.

2.2 TN's Vehicle Operation

At any time every vehicle (out of a finite set of vehicles) is either in the driving condition parked, stopped, or driving. Now every vehicle in operation, i.e., stopped or driving, executes a mission (Sec. 2.2.3 on p. 6) in a run (Sec. 2.2.2 on p. 5) acc. to a dispatch plan (Sec. 2.2.1).

2.2.1 The Dispatch Plan

The central element of the vehicle operation in our SHA model is a *dispatch plan* that defines at what time which vehicle is supposed to process which *run* in form of an ordered list. It can cover part of a day, a complete day, or even more. Moreover, note that in practice different dispatch plans might have been specified for different operational modes (nominal operation, perturbed mode of operation, etc.). However, in our model there is only one dispatch plan, which requires us to merge two or more real-world dispatch plans if necessary; or to compute for every different dispatch plan a separate forecast.

2.2.2 A Vehicle Run

Every vehicle run specifies a sequence of *vehicle missions* that have to be executed in the given order by a vehicle that processes the run; which should result in a

coherent driving cycle of fixed-route dead headings and transportation services.

2.2.3 A Vehicle Mission

The fixed route that underlies every vehicle mission is specified as a path in a transportation grid. An indication of stops at the waypoints along that path, minimum and maximum dwell times for these stops, and deterministic driving times in between the waypoints then complement the specification of a particular vehicle mission.

As an example, look at Fig. 4.

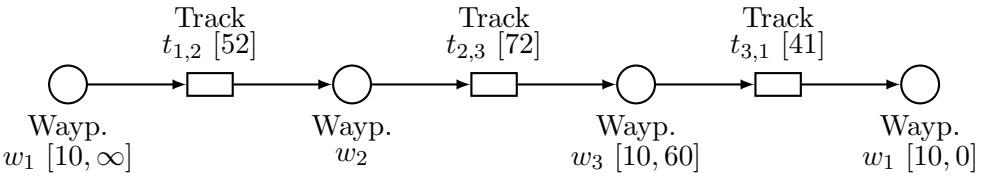


Fig. 4. Vehicle mission for the orbital transportation grid from Fig. 1 on p. 3: the single number written in brackets next to every track defines a constant driving time, and the pair of two comma-separated numbers and written in brackets next to every waypoint define minimum and maximum dwell times at that waypoint in the respective order

It depicts the graphical specification of a sample vehicle mission, say x', for the orbital transportation grid from Fig. 1 on p. 3: A vehicle a that executes x' is supposed to pass trhough the orbital grid starting from and ending at w_1. Moreover, note that a is supposed to stop at w_1 and w_3 since a separate pair of a minimum and a maximum dwell time is assigned to both (comma-separated values written in brackets next to w_1 and w_3). In particular, a cannot depart from w_3 before its dwell time at w_3 has exceeded 10 seconds. Assuming that passengers can board a/alight from a at w_3, then a departs from w_3 before its dwell time at w_3 has exceeded the maximum dwell time of 60 seconds iff no more passengers want to board/alight from it. On the contrary, a departs from w_3 after the maximum dwell time of 60 seconds has elapsed iff some passengers still want to alight from it or its next waypoint is blocked by another vehicle. In this way, the minimum dwell times impose hard constraints on the vehicle operation, whereas the maximum dwell times have to be respected iff possible.

2.3 The Routing of All Passenger Flows in TN

Look at Fig. 5 which depicts an extract of a sample infrastructure (graph) together with the specification of three different trip profiles 1, 2, and 3: All passengers who travel acc. to the first trip profile enter TN through the station S_1, and arrive at its platform next. From this platform, they then would like to board a vehicle that executes the mission x_1, which passes through and stops at all three waypoints w_1, w_2, and w_3. On-board such a vehicle stopped at w_2, these passengers would like to alight from it to a GP in the station S_2 (not depicted). In contrast to the first trip profile, all passengers who travel acc. to the second or third trip profile would like to alight from a vehicle that is stopped at w_3 to a GP in the station S_3 (not depicted). Then note that, in principle, both vehicle missions x_1 and x_2 provide a transportation service from S_1 to S_3. However in contrast to x_1, a vehicle that

executes x_2 is supposed to skip w_2 and thus provides an express service from S_1 to S_3. Now the difference between the passengers who travel acc. to the second trip profile from the passengers who travel acc. to the third trip profile is that the latter show a preference for x_2; whereas all passengers of the second trip profile are indifferent to both vehicle missions and thus board whatever vehicle mission is available.

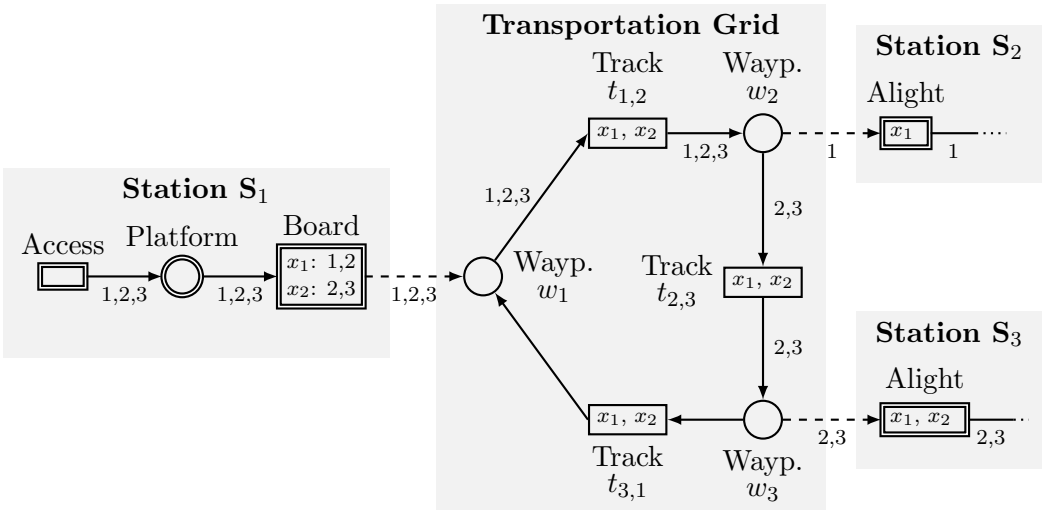


Fig. 5. Sample infrastructure (graph) of the SHA model together with the specification of three different trip profiles (1, 2, and 3), in which all passengers of the trip profile 3 prefer to board a vehicle that executes the mission x_2 in order to travel from S_1 to S_3 over a vehicle that executes x_1

In general, every passenger travels acc. to a particular trip profile out of a finite set of trip profiles $\mathcal{Y}$. Every trip profile $y \in \mathcal{Y}$ in turn specifies a path in the SHA model's infrastructure together with labels assigned to every corridor that is dedicated to the boarding of a vehicle so as to account for the passengers' different mission preferences. Then with goal to ease the graphical specification of all trip profiles, we can inscribe the infrastructure graph with the paths and the stops specified for the different vehicle missions. So in the example from above, the fact that x_1 is inscribed to the corridor "Alight" tells us that a vehicle that executes x_1 stops at w_2; which in turn implies that all passengers on-board that vehicle can alight from it in the given direction (or the GP this corridor points to; not depicted).

From the graphical specification of all trip profiles, we deduce routing matrices; for every corridor in every station one. Refer to Tab. 1 for an overview of some shorthands pertaining to matrix/vector operations used here and throughout the rest of the paper: Initially, every routing matrix $\mathrm{R}(\cdot)$ is a diagonal matrix with zeros and ones only. More precisely, the element $\mathrm{R}(c)[i,j]$, with $i,j \in \mathcal{Y}$, of the routing matrix $\mathrm{R}(c)$ assigned to the corridor c is one iff $i = j$ and the trip profile $i \in \mathcal{Y}$ covers c; in which we assume that all trip profiles are enumerated from 1 to $|\mathcal{Y}|$. We next might rewrite the columns of some routing matrices to account for the re-routing of some passenger flows. As a result, these (constant) routing matrices are not diagonal any more, in which their interpretation/specification goes as follows: $\mathrm{R}(c)[i,j]$ is the relative amount of passengers who enter c acc. to the trip profile i and who leave it acc. to the trip profile j. We then demand that every

column $\mathrm{R}(c)[\cdot, j]$, with $j \in \{1, 2, \ldots, |\mathcal{Y}|\}$, of $\mathrm{R}(c)$ either sums up to one or to zero, which will ensure the conservation of all passenger flows in the set up of the balance equations at a later point.

Table 1
An overview of some shorthands pertaining to matrix/vector operations used throughout the paper

Shorthand	Meaning
$a[i]$	Element in the i-th row of a column vector a
A^T	Transpose of a matrix A
$A[i, j]$	Element in the i-th row and the j-th column of a matrix A
$A[\cdot, j]$	j-th column of a matrix A

3 Capturing the Probability Flow

Our automaton model belongs to the class of continuous-time hybrid-state automata with deterministic-time and state-driven probabilistic jumps among a finite set of discrete modes. In general, every mode in such a model refers to a discrete state to which is assigned to a continuous domain with a continuous differential dynamics [2]. An edge-labelled graph - here and in the following referred to as a mode graph - captures all jump conditions between the different modes that might have been defined in terms of the continuous states that have to enter particular target sets or time-thresholds that have to elapse.

Now in our use case, we limit every mode to a discrete state in TN's vehicle operation that tells us

- the driving condition of every vehicle (parked, stopped, or driving),
- the position of every vehicle, in which, by convention, the position of a driving vehicle is identical to the waypoint it moves towards, and
- the dispatch, the run, and the mission of every stopped and every driving vehicle.

Then note that for every mode we can define a set of balance equations that capture the SHA model's continuous passenger flow dynamics in this particular mode in that they relate the passenger loads of the GPs among themselves and some stopped vehicles (Sec. 3.1 on p. 9). We can next use these balance equations to numerically integrate estimations for all passenger loads so as to evaluate the probability of occurrence of the SHA model's transition to another mode (Sec. 3.2 on p. 10). However at the latest, we then notice that we have to confine all mode transition times to finite sets (Sec. 3.3 on p. 12) for any feasible computation (Sec. 3.4 on p. 14).

[2] System of differential equations

3.1 The Balance Equations for a Particular Mode

Remember that we impose a very crucial constraint on the specification of the interface for the passenger transfers between the GPs in the stations and the vehicles stopped at the waypoints of the transportation grids (cf. Sec. 2.1.3 on p. 4): If a vehicle a is stopped at a waypoint, then a passenger transfer to/from a is possible from/to one and the same station only; which does not imply that the GP that accommodates all passengers who alight from a must be identical to the GP that accommodates all passengers who board a, though. In this context, we say that a vehicle a is *docked* to a station $s \in \mathcal{S}$ iff (i) a is stopped at a waypoint that can be accessed from a GP in s acc. to the specification of the infrastructure's interface, and (ii) some passengers want to alight from or board a acc. to the specification of the passenger (re-)routing. For a particular mode of our automaton, we can thus decompose the set of all balance equations for all passenger loads - which define the automaton's continuous passenger flow dynamics as long as no jump to another mode occurs - w.r.t. the different stations: For every station $s \in \mathcal{S}$ we set up a (decoupled) system of $n := n_1 + n_2$ (Itô-) stochastic differential balance equations for the $|\mathcal{Y}|$-dimensional passenger load vectors[3] of all $n_1 \in \mathbb{N}_{>0}$ GPs in s and all $n_2 \in \mathbb{N}_{\geq 0}$ vehicles docked to s. For time $\tau > 0$, this system can be written down in the form of

$$\mathrm{d}X_s(\tau) = A_s\left(X_s(\tau)\right)\,\mathrm{d}\tau + B_s\,\mathrm{d}\mathcal{W}_s(\tau), \tag{1}$$

with the $(n \times |\mathcal{Y}|)$-dimensional state vector X_s that groups all n_1+n_2 passenger load vectors; the n-dimensional drift vector A_s, and the $(n \times m)$-dimensional diffusion matrix B_s iff we assume that $m \in \mathbb{N}_{>0}$ trip profiles enter TN through s; in which case $\mathcal{W}_s$ is an m-dimensional Wiener process[4] [12]. In other words, we assume that the uncertainty that is inherent to the continuous passenger flow dynamics in form of Eqn. 1 comes along with the passengers who enter TN through its stations (= probabilistic passenger arrival processes). However, all passenger flows that originate within TN such as a flow between two GPs, a flow between a GP and a stopped vehicle, or a passenger flow that leaves TN from one of its stations, are supposed to be specified in a deterministic manner given the respective passenger loads.

With that said, we now look at the exemplary specification of the balance equations for the station s from Fig. 3 on p. 5 in a particular mode, in which we assume that one vehicle, say a, is stopped at the waypoint w_i, and some passengers want to alight from/board a to/from the platform p: First of all, note that every corridor $c \in C_s := \{c_\mathrm{a}, c_\mathrm{b}, c_\mathrm{e}, c_\mathrm{A}\}$ in s must have a maximum throughput $\phi_\mathrm{max}(c)$, with $\phi_\mathrm{max} : C_s \to \mathbb{R}_{\geq 0}$ that limits the number of passengers per second who can pass through it. This maximum throughput thus limits the (scalar) magnitude $\phi_\mathrm{m}(c,\tau)$, with $\phi_\mathrm{m} : C_s \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, of the passenger flow (vector) $\phi(c,\tau)$, with $\phi : C_s \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{|\mathcal{Y}|}_{\geq 0}$, through c at time $\tau \geq 0$ s.t. $\phi_\mathrm{m}(c,\tau) = \sum_{i\in\mathcal{Y}} \phi(c,\tau)[i]$ and $\phi_\mathrm{m}(c,\tau) \leq \phi_\mathrm{max}(c)$ for every $c \in C_s$, and every $\tau \geq 0$; in which $\phi(c,\tau)[i]$ gives the flow of all passengers who enter c at τ acc. to the trip profile $i \in \mathcal{Y}$[5]. It then

[3] $\mathcal{Y}$ is the finite set of different trip profiles (cf. Sec. 2.3 on p. 6), and $|\mathcal{Y}|$ is the cardinality thereof

[4] A continuous-time stochastic process with independent and stationary increments $\mathcal{W}_t - \mathcal{W}_s$ whose law is Gaussian with parameter $t - s$

[5] Refer to Tab. 1 on p. 8 for an overview of some matrix/vector operations employed here

follows that[6] the product $\mathrm{R}(c)[j,\cdot]\,\phi(c,\tau)$ of the j-th row of the routing matrix $\mathrm{R}(c)$ assigned to c with $\phi(c,\tau)$, gives the flow of all passengers who leave c at τ acc. to the trip profile $j \in \mathcal{Y}$. As balance equation for the passenger load $\mathrm{M}(a,\tau)$, with $\mathrm{M} : \{a,p\} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}^{|\mathcal{Y}|}$, of a at $\tau \geq 0$ we thus write down[7]

$$\mathrm{dM}(a,\tau) := \mathrm{R}(c_\mathrm{b})\ \phi(c_\mathrm{b},\tau)\ \mathrm{d}\tau - \phi(c_\mathrm{a},\tau)\ \mathrm{d}\tau. \tag{2}$$

Note that the passenger load of a must neither be negative nor exceed a's capacity limit $\kappa_a > 0$. For any $\tau \geq 0$ and any $i \in \mathcal{Y}$, we thus have to require that $\mathrm{M}(a,\tau)[i] \to 0$ imply $\phi(c_\mathrm{a},\tau)[i] \to 0$, and $\sum_{j\in\mathcal{Y}} \mathrm{M}(a,\tau)[j] \to \kappa_a$ implies $\phi_\mathrm{m}(c_\mathrm{b},\tau) \to 0$; in which case we say that $\phi(c_\mathrm{a},\tau)$ is *demand-sensitive*, and $\phi(c_\mathrm{b},\tau)$ is *capacity-sensitive*. Moreover, we need the passenger flow assigned to every corridor to be *routing-proper*, i.e., for any $c \in C_s$, $j \in \mathcal{Y}$, and $\tau \geq 0$, it holds that $\sum_{i\in\mathcal{Y}} \mathrm{R}(c)[i,j] = 0$ implies $\phi(c,\tau)[i] = 0$. Now the specification of the balance equation for the passenger load of the platform p is a little bit more complex, though, due to the fact that we have to integrate into it the uncertainty that comes along with the passenger flow $\phi(c_\mathrm{A},\tau)$ through c_A (= arrival process). This is captured by adding, to the impact that $\phi(c_\mathrm{A},\tau)$ has on $\mathrm{M}(p,\tau)$, as expressed by the product $\mathrm{R}(c_\mathrm{A})\ \phi(c_\mathrm{A},\tau)$, another product term $\mathrm{R}(c_\mathrm{A})\ D(c_\mathrm{A})$. This latter term connects the multidimensional Wiener process $\mathcal{W}_s$ with $\mathrm{M}(p,\tau)$ through the product of $\mathrm{R}(c_\mathrm{A})$ with the diffusion matrix $D(c_A)$[8], with $D : c_\mathrm{A} \to \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{W}_s|}$:

$$\begin{aligned}\mathrm{dM}(p,\tau) := &\overbrace{\mathrm{R}(c_\mathrm{A})\ \phi(c_\mathrm{A},\tau)\ \mathrm{d}\tau + D(c_\mathrm{A})\ \mathrm{d}\mathcal{W}_s}^{\text{Drift- and diffusion term for the spec. of the arrival process through } c_\mathrm{A}} + \mathrm{R}(c_\mathrm{a})\ \phi(c_\mathrm{a},\tau)\ \mathrm{d}\tau - \\ &\phi(c_\mathrm{e},\tau)\ \mathrm{d}\tau - \phi(c_\mathrm{b},\tau)\ \mathrm{d}\tau\end{aligned} \tag{3}$$

In Eqn. 3, we assume that all passenger flows are routing-proper as well as - iff applicable - capacity- and demand-sensitive, in a way that is similar to the specification of the passenger flows in the balance equation Eqn. 2 for the passenger load of a above. However, note that this requirement taking alone cannot ensure the non-negativity and the capacity limit of p any more. Instead, we have to explicitly enforce both during the numerical integration of Eqn. 1, by defining appropriate boundary conditions. Similarly, we have to explicitly enforce the maximum throughput of c_A.

3.2 Numerical Integration of Balance Equations

In practice, there are two dominant approaches to computing the temporal change of an initial distribution[9] subjected to Eqn. (1). On the one hand, there is the Monte Carlo method [9]: From the pool of all possible solution paths, some paths are selected "randomly" and possibly several times. In this approach, one thus would have to ensure that the correlation between the selected solution paths matches the

[6] From the specification of the passenger routing, which is not specified for this example here

[7] In integral form since this balance equation is supposed to be integrated into Eqn. 1

[8] Tuning parameter from estimation; in general not time-invariant

[9] Estimation for the passenger loads of all GPs in a particular station and of all vehicles which are docked to this station in our use case

correlation found in the actual stochastic process. Of course, this is a difficult task especially if the statistical data is not at hand, as is often the case in degraded modes of operation. The second approach is the one we will use here: Integrate the balance equations from Eqn. 1 into systems of linear parabolic partial differential equations (one for every station $s \in \mathcal{S}$):

$$\begin{aligned}\frac{\partial}{\partial \tau}\mathrm{pdf}\left(X_s\left(\tau\right)\right) = &-\sum_{i=1}^{N}\frac{\partial}{\partial X_{s,i}}\left(A_s\left(X_s\left(\tau\right),\tau\right)[i]\,\mathrm{pdf}\left(X_s\left(\tau\right)\right)\right)\\ &+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\Psi_s[i,j]\,\frac{\partial^2}{\partial X_{s,i}X_{s,j}}\mathrm{pdf}\left(X_s\left(\tau\right)\right).\end{aligned} \tag{4}$$

with A_s, B_s from Eqn. 1 and the abbreviation $\Psi_s := B_s\,B_s^T$. This system is also known as the (multidimensional) *Fokker-Planck equation (FPE)* or the *Kolmogorov forward equation*; it describes the time evolution of the probability density function (pdf) for X_s.

Eqn. 4 defines a continuity equation in form of a system of deterministic parabolic partial differential equations: By introducing the probability flux

$$f\left(X_s(\tau)\right) := A_s\left(X_s\left(\tau\right)\right)\,\mathrm{pdf}\left(X_s\left(\tau\right)\right) - \frac{1}{2}\,\Psi_s\begin{bmatrix}\frac{\partial}{\partial X_{s,1}}\mathrm{pdf}\left(X_s\left(\tau\right)\right)\\ \frac{\partial}{\partial X_{s,2}}\mathrm{pdf}\left(X_s\left(\tau\right)\right)\\ \vdots\\ \frac{\partial}{\partial X_{s,N}}\mathrm{pdf}\left(X_s\left(\tau\right)\right)\end{bmatrix}, \tag{5}$$

we can rewrite Eqn. 4 acc. to

$$\frac{\partial}{\partial \tau}\mathrm{pdf}\left(X_s\left(\tau\right)\right) + \mathrm{div}\left(f\left(X_s(\tau)\right)\right) = 0, \tag{6}$$

in which $\mathrm{div}\left(\cdot\right)$ denotes the divergence operator.

Now for this continuity equation we can derive reflecting boundary conditions for the numerical integration of Eqn. 4 that confine the probability flux to a closed convex polytope K_s, which captures the complete passenger load space associated with s in the mode at hand [10]. Here, we sketch only the major steps in their derivation in an informal manner, and refer to the literature [5, chapter 5] for more details. The starting point is the insight that the cumulative probability of X_s adopting a value in K_s at any time instant $\tau \geq 0$ must be one. Thus, the time derivative of this cumulative probability must be zero. Then, after some transformations employing the divergence theorem, we obtain the reflecting boundary condition

$$n\left(X_s(\tau)\right) \cdot f\left(X_s(\tau)\right) = 0, \forall\, X_s(\tau) \in \partial K_s \text{ and } \tau \geq 0, \tag{7}$$

in which $n\left(X_s\left(\tau\right)\right)$ denotes a unit vector in the outward orthogonal direction of K_s

[10] Cartesian product of $|\mathcal{Y}|$-dimensional real vector spaces: for the passenger load vector of every GP in s and of every vehicle docked to s one; in which $\mathcal{Y}$ is the set of different trip profiles and $|\mathcal{Y}|$ the cardinality thereof; lower- and upper bounded by all non-negativity- and capacity constraints

w.r.t. the boundary ∂K_s of K_s at $X_s(\tau) \in \partial K_s$. For Cartesian coordinates the scalar product of $f(X_s(\tau))$ with $n(X_s(\tau))$ simplifies to $n(X_s(\tau))^T f(X_s(\tau))$.

From this, it is possible to include Eqn. 4 and Eqn. 7 subjected to Eqn. 1 in form of a boundary value problem in a numerical integration scheme that preserves the conservation of the probability flux. We do not go into further details here, but mention that we go for the finite volume method in combination with the Crank-Nicolson method in an on-going implementation. Finally, note that we can enforce the maximum throughput of every corridor that implements a passenger arrival process, in adjusting the barriers for some reflecting boundary conditions in our time-discrete computation strategy; more on this later.

3.3 Discretization of Time

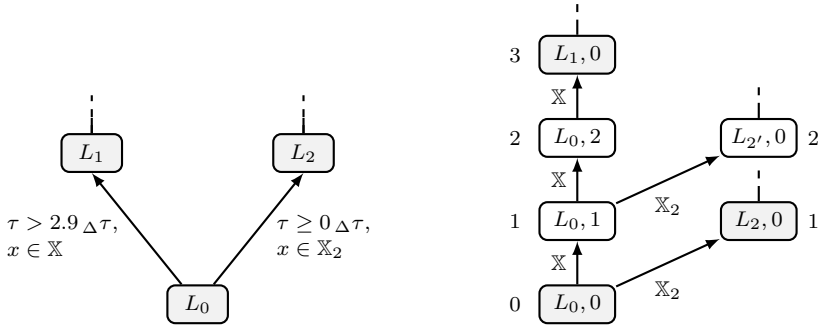
Remember that we resolve conflicting vehicle movements in a deterministic manner if necessary. Thus, two different types of mode transitions can occur in our automaton model in principle: deterministic-timed- and probabilistic mode transitions.

Deterministic-timed Mode Transitions. They involve vehicle dispatches, vehicle arrivals, and vehicle departures assuming that the latter are not tied to any passenger loads. They are thus completely conditioned on (time) constraints that are imposed on the vehicles' dispatch times, as well as their elapsed driving- and dwell times (the driving time of vehicle a_1 must equal τ_1 seconds, the dwell of vehicle a_2 must exceed τ_2 seconds, and so forth).

Probabilistic Mode Transitions. They involve departures of vehicles that are docked to a station because less than one passenger wants to alight from/board a vehicle, or less than one passenger can board a vehicle (because of the vehicle's limited capacity), and so on. They are thus conditioned on constraints that are imposed on some passenger loads (= passenger load constraints) in form of closed convex polytopes; although they might involve time constraints as well. Moreover, note that the departure events of two vehicles, which are docked to two different stations, are mutually exclusive due to the set up of our balance equations (for every station one separate and decoupled system).

Vehicle Load Tree. The possible departure times of vehicles docked to a station form an uncountable set, which renders the computation of our continuous-time automaton model intractable. In order to overcome this burden, we thus discretize the time in the computation of the model's hybrid state. More specifically, we compute the time evolution of the model's *vehicle load* starting recursively from some initial one from one equidistantly-spaced discrete point in time to the next; in which a particular vehicle load refines a mode of our automaton model in that it also defines elapsed dwell times for all stopped vehicles, and elapsed driving times for all driving vehicles. Thus, several vehicle loads can belong to one and the same mode. We then a) assume that the SHA model can change its vehicle load only at equidistantly-spaced discrete points in time. In addition, we b) replace every equality constraint for a vehicle arrival to be true by an inequality constraint s.t. the elapsed driving time of a vehicle has to exceed a certain time threshold $\tau > 0$; and does not have to equal τ. Finally, we c) ceil every dispatch time (from the dispatch plan) to the next closest discrete point in time. As a result, we obtain a

vehicle load tree G; an edge- and vertex-labelled directed rooted tree (cf. Fig. 6): Every vertex captures a particular vehicle load L of TN. This L is stored as a tuple (L', n), in which L' is a vehicle load itself, and $n \in \mathbb{N}_{>0}$ is a non-negative integer. We then obtain L from (L', n) in that we increment every elapsed driving- and dwell time in L' by n. The vertex label assigned to L gives the discrete time step that elapses before our automaton model - if at all - is in L starting from the vehicle load in the root of G at $\tau = 0$; it equals the height of (L', n) in G. On the other hand, every edge label, say from the vehicle load L_1 to L_2, gives a convex polytope for TN's passenger load that equals the complete passenger load space [11] iff the transition of the automaton's vehicle load from L_1 to L_2 is not conditioned on a probabilistic mode transition, and a subset of the passenger load space otherwise; in which (latter) case this subset is adopted from the specification of the passenger load constraint for the respective mode transition. It then follows that branches in the vehicle load tree arise from probabilistic mode transitions only.



(a) Mode graph (b) Vehicle load tree

Fig. 6. Comparison of a mode graph and a vehicle load tree, in which $\mathbb{X}$ denotes the complete passenger load space, $\mathbb{X}_2 \subset \mathbb{X}$, and ${}_{\Delta}\tau > 0$ is the time step for the equidistant discretization

Computation of the Vehicle Load Tree. The iterative computation of a vehicle load tree from one time layer to the next, with a fixed time step of ${}_{\Delta}\tau > 0$ seconds, is straightforward; and we thus describe the major steps of every (iterative) run only in an informal manner here: Starting point is the tree $G_0 = (V_0, E_0, f_{\mathrm{e},0}, f_{\mathrm{v},0})$ with a single vertex v_0 s.t. $V_0 = \{v_0\}$ and $E_0 = \emptyset$. This vertex captures the automaton's vehicle load L_0 at time $\tau = 0$ s.t. $v_0 = (L_0, 0)$ and $f_{\mathrm{v}}(v_0) = 0$. For a summary of the shorthands V_i, E_i, $f_{\mathrm{v},i}$, and $f_{\mathrm{e},i}$ pertaining to the specification of the vehicle load tree G_i, with $i \in \mathbb{N}_{\geq 0}$, refer to Tab. 2 on p. 14. The run $j \in \mathbb{N}_{>0}$ then computes G_j from G_{j-1} in that it first adopts the specification of G_{j-1} s.t. $G_i := G_{j-1}$, and then processes for every vertex $v \in V_{j-1}$ s.t. $f_{\mathrm{v}}(L, n) = i$, $v := (L, n)$, and M is the mode that is contained in L the following 5 steps:

(i) Compute the set F of all enabled mode transitions from M to any other mode M', in which a mode transition from M to M' is enabled iff all time constraints (independently of the passenger load constraints) are met until time step i

(ii) Define $F_{\mathrm{d}} \subseteq F$ as the subset of all deterministic-timed mode transitions, and $F_{\mathrm{p}} \subseteq F$ as the subset of all probabilistic mode transitions s.t. $F = F_{\mathrm{d}} \cup F_{\mathrm{p}}$ and

[11] Cartesian product of a finite set of closed intervals on the real number axis; for every GP in every station and for every vehicle one interval that is lower bounded by zero and upper bounded by the capacity limit of the respective GP or vehicle

$F_\mathrm{d} \cap F_\mathrm{p} = \emptyset$

(iii) If F is empty:
  (a) Define $v' := (L, n+1)$, and $\mathbb{X}$ as the complete passenger load space
  (b) Append v' as a child of v to G_i
  (c) Define $f_{\mathrm{e},i}\,(v, v') := \mathbb{X}$, and $f_{\mathrm{v},i}\,(v) := i$

(iv) If $F_\mathrm{d} \neq \emptyset$:
  (a) Define $\mathbb{X}$ as the complete passenger load space
  (b) Identify the mode transition $f \in F_\mathrm{d}$ with the biggest impact on L in terms of the number of vehicles that change their operational states [12] (position, mission, etc.)
  (c) W.r.t. M, let M' be the mode following f, and $\mathcal{A}$ be the set of all vehicles that change their operational states due to f
  (d) Compute the vehicle load L'
    - Adopt M' for the mode that is contained in L'
    - Set the elapsed driving/dwell time of every driving/stopped vehicle $a \in \mathcal{A}$ in L' to zero
    - For the elapsed driving/dwell time of every other driving/stopped vehicle $a' \notin \mathcal{A}$ in L', take that from L and increment it by one
  (e) Define $v' := (L', 0)$
  (f) Continue with step iii.b

(v) If $F = F_\mathrm{p}$, then do for every $f \in F$:
  - Define $\mathbb{X}$ as the convex polytope that captures the passenger load constraint imposed on f
  - Continue with step iv.b

Table 2
Shorthands pertaining to the computation of a vehicle load tree G assuming that V is the vertex set of G, and E is its edge set

Shorthand	Meaning
$f_\mathrm{v}\,(v)$	Height of $v \in V$ in G
$f_\mathrm{e}\,(v, v')$	Subset of the complete passenger load space that has to be entered by the automaton's passenger load so as to induce a jump from v to v', with $(v, v') \in E$
${}^\bullet v$	Set ${}^\bullet v := \{v' \in V : (v', v) \in E\}$ of all parents of $v \in V$
$v^\bullet$	Set $v^\bullet := \{v' \in V : (v, v') \in E\}$ of all children of $v \in V$

3.4 Computing Transition Probabilities

So far we have described how estimations for the passenger loads of all GPs in a station, and of all vehicles docked to that station, can be propagated in time in a particular mode of our automaton model by the numerical integration of deterministic partial differential equations. We then introduced the vehicle load of TN, and

[12] This mode transition is unique

described how - starting from some initial vehicle load L_0 at time $\tau = 0$ - all possible solution paths of that vehicle load can be computed in a time-discrete manner, in which all mode transition times are confined to an equidistantly-spaced time-layered mesh. The next step is thus to connect both; the continuous-time propagations of all passenger load estimations in the different modes on the one hand, and all possible solution paths for the vehicle load on the other hand. In doing so, we describe how a discrete-time approximation for the hybrid state (= particular vehicle load of TN + all passenger load vectors) of our (continuous-time) automaton can be computed [13]. Note first that a missing edge from $v' \in V$ to $v \in V$ implies that the corresponding transition of the automaton's vehicle load cannot occur, given the discrete time step ${}_{\Delta}\tau > 0$ underlying G. On the other hand, the existence of this edge individually only implies that the corresponding transition may occur, but does not necessarily have to do so. In this context focussing on the SHA model's discrete state first, note that

$$\mathbb{P}(v) := \sum_{v' \in {}^{\bullet}v} \mathbb{P}(v|v') \, \mathbb{P}(v') \tag{8}$$

must hold for every $v \in V$, in which $\mathbb{P}(v)$ is the marginal probability that the vehicle load of the automaton at time step $\tau = f_{\mathrm{v}}(v)$ is v, and $\mathbb{P}(v|v')$ is the (marginal) conditional probability that the automaton's vehicle load at time step $\tau = f_{\mathrm{v}}(v)$ is v, given that it is v' at time step $\tau' = f_{\mathrm{v}}(v') = \tau - 1$. This conditional probability of course depends on the automaton's passenger load; see below. Next note that at any time step $i \in \mathcal{H} := \left\{0, 1, \ldots, \left\lfloor \frac{\tau_{\mathrm{f}}}{{}_{\Delta}\tau} \right\rfloor\right\}$, with $\tau_{\mathrm{f}} := \max_{v \in V} f_{\mathrm{v}}(v)$, our automaton model must be in some vehicle load, which is identical to saying that

$$\sum_{v \in {f_{\mathrm{v}}}^{-1}(i)} \mathbb{P}(v) = 1, \, \forall i \in \mathcal{H}. \tag{9}$$

Moreover, at every time step $i \in \mathcal{H}$ the automaton's passenger load must adopt a value from a subset of the complete passenger load space with a positive probability that we store in the density $\mathrm{pdf}(i)$, which brings us to the computation of the latter: As outlined above, we assume that (i) $\mathbb{P}(v_0) = 1$, in which $v_0 := (L_0, 0)$ and $f_{\mathrm{v}}(v_0) = 0$; and (ii) $\mathrm{pdf}(0)$ is known. We then obtain $\mathrm{pdf}(1)$ in that we numerically integrate $\mathrm{pdf}(0)$ from $\tau = 0$ to $\tau = {}_{\Delta}\tau$ subjected to all stochastic balance equations (for every station $s \in \mathcal{S}$ one) from Eqn. 1 on p. 9 that were specified for the mode that is contained in v_0. Now from $\mathrm{pdf}(1)$ we can compute all transition probabilities from v_0 to every other vehicle load $v \in {v_0}^{\bullet}$ acc. to Eqn. 8 on p. 15 since a) $\mathbb{P}(v|v_0)$ is the probability that the automaton's passenger load adopts a value from $f_{\mathrm{e}}(v_0, v)$, b) $f_{\mathrm{e}}(v_0, v)$ is a subset of the complete passenger load space that in turn is covered by $\mathrm{pdf}(1)$, and c) all stochastic balance equations integrated into Eqn. 1 have the Markov property. Thus at time $\tau = {}_{\Delta}\tau$ the automaton's hybrid state is completely defined: its passenger load by $\mathrm{pdf}(1)$ and its vehicle load by $\mathbb{P}(v) = \mathbb{P}(v|v_0) \, \mathbb{P}(v_0) = \mathbb{P}(v|v_0)$. Now at time $\tau' = {}_{\Delta}\tau$ the automaton can be in any vehicle load $v \in {v_0}^{\bullet}$ with its proper but not necessarily different set of

[13] Approximation that should asymptotically converge to the exact (analytical) solution by reducing the discretization time step for the discrete mode graph and by increasing the granularity of all numerical integrations

stochastic balance equations that governs the continuous passenger flow dynamics in the time interval $[1, 2)$. Thus, although input to the numerical integration of every set of balance equations pertaining to a vehicle load $v \in v_0^\bullet$ from $\tau' = {}_\Delta\tau$ to $\tau'' = 2\,{}_\Delta\tau$ is $\mathrm{pdf}\,(1)$, the density pertaining to v computed for $\tau'' = 2\,{}_\Delta\tau$, $\mathrm{pdf}_\mathrm{o}\,(v)$, differs from vehicle load to vehicle load in that $v_1 \neq v_2$, with $v_1, v_2 \in v_0^\bullet$, does not imply $\mathrm{pdf}_\mathrm{o}\,(v_1) = \mathrm{pdf}_\mathrm{o}\,(v_2)$. However once again due to the Markov property of all balance equations it follows that $\mathrm{pdf}\,(2)$ computes from the product $\mathrm{pdf}_\mathrm{o}\,(v)$ with $\mathbb{P}\,(v)$ summed up over all $v \in v_0^\bullet$; and in general it holds that for any $i \in \mathcal{H}$, $\mathrm{pdf}\,(i+1)$ computes acc. to

$$\mathrm{pdf}\,(i+1) = \sum_{v \in f_\mathrm{v}^{-1}(i)} \mathrm{pdf}_\mathrm{o}\,(v)\ \mathbb{P}\,(v)\,, \tag{10}$$

which together with Eqn. 8 tells us that the time-evolution of the automaton's hybrid state has to be computed (time) layer by layer.

4 Forecasting Passenger Loads

4.1 The Problem Statement

We start from an observation of TN's (network) state at time $\tau = 0$ and a question about its passenger load (= objective of the forecast). The former is captured by an exact knowledge of its vehicle load L_0 at $\tau = 0$ and an estimation of its passenger load (= passenger load of every GP in every station, and of every vehicle operated in the infrastructure at hand) for $\tau = 0$ in form of the density $\mathrm{pdf}\,(0)$. From the latter, we can deduce a prediction horizon τ^*, a target set for TN's passenger load $\mathbb{X}^*$, and a threshold α^* for the probability that TN's passenger load adopts a value from $\mathbb{X}^*$. For instance, a particular question might read as follows:

> "Will the passenger load of the platform p in the station $s \in \mathcal{S}$ exceed 200 passengers ($\to \mathbb{X}^*$) with a probability greater than 0.7 ($\to \alpha^*$) within the next 20 minutes ($\to \tau^*$)?"

4.2 The General Solution

We compute $\alpha(0) := \int_{\mathbb{X}^*} \mathrm{pdf}\,(0)\ \mathrm{d}V$, i.e., the probability of TN's passenger loads to take on a value from $\mathbb{X}^*$ at $\tau = 0$. Then, proceed as follows: If $\alpha(0) > \alpha^*$, we have already obtained an affirmative answer to our question and we can stop here. Otherwise, we make use of our time-discrete automaton model in that we provide $\alpha(0)$, the time index $i = 0$, and the vehicle load tree $G_0 := (V_0, E_0, f_{\mathrm{e},0}, f_{\mathrm{v},0})$ as input to the following iteration that terminates iff $\alpha(j) > \alpha^*$ or $(j+1)\,{}_\Delta\tau > \tau^*$, with $j \in \mathbb{N}_{\geq 0}$; here $V_0 := \{v_0\}$, with $v_0 := (L_0, 0)$, $E_0 := \emptyset$, and $f_{\mathrm{v},0}\,(v_0) := 0$:

(i) Compute the vehicle load tree G_{i+1} from G_i as described in Sec. 3.3

(ii) Compute $\mathrm{pdf}\,(i+1)$ acc. to Eqn. 10 on p. 16

(iii) Compute $\alpha(i+1) := \int_{\mathbb{X}^*} \mathrm{pdf}\,(i+1)\ \mathrm{d}V$

(iv) If $\alpha(i+1) \leq \alpha^*$, then compute $\mathbb{P}\,(v)$ acc. to Eqn. 8 on p. 15 for every $v \in V_{i+1}$ s.t. $f_{\mathrm{v},i+1} = i+1$

(v) Set $i := i + 1$

Thus by construction, we know the binary true/false answer to our initial question upon termination of this iterative computation. So the next question then is whether or not the iterative computation itself is feasible in a reasonable amount of time (number of trivial computation steps) with reasonable memory constraints, which is probably not the case unless some further simplifications are made. This will lead us to a more efficient implementation of the iterative computation of the forecast from above (Sec. 4.3) together with a re-definition of all balance equations for the numerical integration of the passenger flows (Sec. 4.4 on p. 17).

4.3 The Efficient Implementation of the Iterative Forecasting Algorithm

First of all, note that in order to compute $\mathrm{pdf}(i+1)$, with $i \in \mathcal{H}^*$ and $\mathcal{H}^* := \left\{1, 2, \ldots, \left\lceil \frac{\tau^*}{\Delta\tau} \right\rceil \right\}$, acc. to Eqn. 10 on p. 16, we only need to know the set of all leaf nodes of the vehicle load tree G_i, say $\bar{V}_i$, together with the density $\mathrm{pdf}(i-1)$ and $\mathbb{P}(v)$ for every $v \in \bar{V}_i$. So, instead of computing a vehicle load tree in the iterative computation of the forecast, as described above, anew every time, it makes sense to propagate only this reduced piece of information from one iteration to the next. However, taking this measure alone, we still end up with a potentially huge set $\bar{V}_i$ given even only very few passenger ride possibilities on-board concurrently moving vehicles; and this is a big problem since we do not only have to compute the set $\bar{V}_i$ itself, but for every $v \in \bar{V}_i$ we also have to numerically solve partial differential equations in the form of Eqn. 1 on p. 9 (so far, one for every station). What we have to do thus is to identify all vehicle movements that do not affect the forecast and to ignore them. In this context, note that - starting from the time step $(i-1)\,{}_\Delta\tau$ until the end of the prediction horizon at time τ^* - we can compute how far the operational state (including the position) of every vehicle can evolve from the constant driving times for the individual tracks and the minimum dwell times for the waypoints in the specification of the different vehicle missions. Combining this knowledge with the specification of the complete passenger re-routing and the definition of the target set $\mathbb{X}^*$ - from which we can deduce a target zone in form of a set of GPs in the stations and vehicles, - we can thus compute whether or not a passenger outside this target zone can penetrate it within the prediction horizon, given the passenger's current position. In this way, we can compute at every iterative step $i \in \mathcal{H}^*$ not only the set of all vehicle movements that we have to take into consideration in the computation of $\bar{V}_j$, but also for every $v \in \bar{V}_j$ the set of all stations that we have to include in the numerical integration of the passenger loads; that is, excluding those stations that are outside the target zone and from which passengers cannot penetrate the target zone within the remaining prediction horizon.

4.4 Redefining all Balance Equations for the Numerical Integration

In the worst case, the system of balance equations derived for a station $s \in \mathcal{S}$ in a particular mode of the automaton acc. to Eqn. 1 on p. 9 has the dimension $(n_1 + n_2)\,|\mathcal{Y}|$, in which n_1 is the number of GPs in s, n_2 is the number of vehicles docked to s, and $\mathcal{Y}$ is the finite set of different trip profiles introduced during the

specification of the passenger routing. Then note that in practice Eqn. 1 with even very few dimensions renders the numerical integration of the boundary value problem in form of Eqn. 4 on p. 11 subjected to Eqn. 7 on p. 11 *intractable* given reasonable computation constraints.

Now one approach to overcome this computational burden is to decouple all passenger flows during the numerical integration of all passenger loads associated with s, and to merge trip profiles with common last miles: First of all, we assume that all $n := n_1 + n_2$ passenger load vectors associated with s were estimated for the forecast at hand in isolation in form of separate and uncorrelated probability densities. Moreover, we assume that these estimations do not violate any additional constraints (non-existent in the original setting of the balance equations) w.r.t. the maximum number of passengers who can be located at a particular GP or on-board a vehicle as a function of their trip profile; see below. Assuming this to be true, we then integrate these estimations into a finite set of decoupled one- and two-dimensional systems of balance equations for the passenger loads associated with s, which we classify either as *arrival processes*, *passenger transfers*, or *outflows*:

Arrival Process. For every corridor in s that implements a passenger arrival process, say c, to one of its GPs, say p, set up a one-dimensional balance equation acc. to Eqn. 1 for the free capacity of p. This free capacity can be computed from p's (constant) total capacity minus its cumulative passenger load. Then note that we can ensure the maximum throughput $\phi_{\max}(c)$ assigned to c during the numerical integration of this one-dimensional balance equation by reducing the total capacity of p s.t. it equals the product of $\phi_{\max}(c)$ with ${}_{\Delta}\tau > 0$ if necessary; in which ${}_{\Delta}\tau$ is the discrete time step that underlies the computation of the vehicle load tree, and thus defines the horizon of the numerical integration (from one vehicle load to the next) [14].

Passenger Transfer. For every passenger transfer between two discrete points associated with s (a GP in s or a vehicle docked to s), say from p_1 to p_2, set up two one-dimensional balance equations acc. to Eqn. 1; one for the cumulative passenger load of p_1 and the other for the free capacity of p_2. Ensure that the corresponding passenger flow assigned to the corridor which connects p_1 to p_2 is capacity- and demand-sensitive (cf. Sec. 3.1 on p. 9). Recall that, by assumption, neither of the two balance equations can have any diffusion terms since every modelled uncertainty is captured by a passenger arrival process or the estimations for all initial passenger loads.

Outflow. For every corridor in s that implements a flow of passengers leaving s from one of its GPs, say p, set up a one-dimensional balance equation acc. to Eqn. 1 (no diffusion term) for the cumulative passenger load of p. Ensure that the specification of the passenger flow assigned to c is demand-sensitive.

Now that we have replaced the system of SDEs for the station s in a particular mode by a finite set of decoupled systems of one- and two dimensional SDEs, we have to specify their proper initialization including the passenger loads and the capacity constraints: Imagine that two corridors, say c_1 and c_2, discharge into the same GP in s, say p; further, that two corridors, say c_3 and c_4, originate in p; and that all

[14] The reflecting boundary condition from Eqn. 7 still ensures the non-negativity of p's (cumulative) passenger load and p's total capacity

four corridors implement a passenger transfer to/from p from/to two other GPs in s. If we set up four decoupled balance equations for all four passenger transfer possibilities as described above, we then have to i) distribute the initial estimation for p's passenger load among the four balance equations, ii) numerically integrate the four balance equations, and then iii) join the integrated estimations into one single density that captures the passenger load of p at the end of the integration horizon. We achieve this by breaking down p's total capacity into distinct parts; one part, say κ_1, that defines the total capacity of p in the balance equation set up for the passenger transfer associated with c_1 and the remaining part κ_2 that gives the total capacity of p in the balance equation set up for c_2, respectively. In particular, set $\kappa_i := \frac{\phi_{\max}(c_i)}{\phi_{\max}(c_1)+\phi_{\max}(c_2)}$, where $i \in \{1,2\}$ and $\phi_{\max}(c_i)$ is the maximum throughput of c_i. Thus, the part of p's total capacity that is attributed to the balance equation set up for c_i is proportional to the maximum throughput of c_i and inversely proportional to the sum of the throughputs of all corridors that discharge into p. This means that we have to extend the denominator in the definition of κ_i if further corridors (implementing passenger transfers or arrival processes) discharge into p. In this context assuming that one such additional corridor implements a passenger arrival process, we might have to further reduce p's total capacity in the set up of the respective balance equation as so as to ensure the maximum throughput as described above. We next use the specification of the passenger (re-)routing to distribute the estimation of p's initial passenger load among the two (decoupled) balance equations set up for the passenger transfers associated with c_3 and c_4. In fact, we know from the specification of the passenger (re-)routing which group of passengers (which trip profile) wants to take which of the two corridors c_3 or c_4 if at all. We thus extract from the initial estimation the number of all those passengers who want to take c_3 and assign it as the initial (cumulative) passenger load to p in the balance equation set up for c_3. We do likewise for c_4. Upon completion of the numerical integration of the balance equations for c_3 and c_4, we then recover the individual trip profile from the computed density of the cumulative passenger loads by extrapolation; more on this will be described in a paper in preparation.

5 Summary & Outlook

In this paper we have introduced a continuous-time stochastic hybrid automaton model and showed how its discrete time approximation can be used to forecast passenger loads in multimodal transportation networks: All passengers are grouped into a finite set of trip profiles that route the respective (continuous) passenger flows. All vehicles on the other hand are either parked, or executing missions with deterministic driving times between every pair of two consecutive discrete points (whereas docking times at stations vary according to the passenger load). Uncertainty explicitly enters the model in form of the passengers who start their trips in the network and the estimations for the initial passenger loads of all stations and vehicles.

Two bottlenecks have to be tackled in the iterative computation (from one time-layer to the next) of every forecast: First, there is the vehicle load tree that captures all possible solution paths of the automaton's discrete state that in turn captures

the operational state of all vehicles. Every concurrent service run (= vehicle movement with the possibility of passengers on-board) introduces some extra branches which tremendously increase the number of numerical integrations that have to be performed in every time-layer.

Now for both bottlenecks we have used two workarounds that we plan to improve on in the future. Moreover, we intend to develop algorithms to efficiently compute forecasts not only for particular passenger loads but also more complex circumstances, such as the *travel* times between origin- and destination pairs.

Acknowledgement

This research work has been carried out under the leadership of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program "Investissements d'Avenir".

References

[1] Balbo, G., *Introduction to generalized stochastic petri nets*, in: *Formal Methods for Performance Evaluation*, 2007 .

[2] Bect, J., *A unifying formulation of the fokker-planck-kolmogorov equation for general stochastic hybrid systems*, ArXiv e-prints (2008).

[3] Bujorianu, M. L., J. Lygeros and C. B. Bujorianu, *Toward a general theory of stochastic hybrid systems*, in: *Stochastic Hybrid Systems*, Springer Berlin Heidelberg, 2006 .

[4] David, R. and H. Alla, "Discrete, Continuous, and Hybrid Petri Nets," Springer Publishing Company, 2010.

[5] Gardiner, C. W., "Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences," Springer Publishing, 2004, 3 edition.

[6] Haar, S. and S. Theissing, *A hybrid-dynamical model for passenger-flow in transportation systems*, in: *5th IFAC Conference on Analysis and Design of Hybrid Systems*, 2015.

[7] Horton, G., V. G. Kulkarni, D. M. Nicol and K. S. Trivedi, *Fluid stochastic petri nets: Theory, applications, and solution techniques*, European Journal of Operational Research (1998).

[8] Hu, J., J. Lygeros and S. Sastry, *Towards a theory of stochastic hybrid systems*, in: *Hybrid Systems: Computation and Control*, Springer Berlin Heidelberg, 2000 .

[9] MacKay, D. J. C., *Introduction to monte carlo methods*, in: *Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models*, 1998.

[10] Prandini, M. and J. Hu, *A numerical approximation scheme for reachability analysis of stochastic hybrid systems with state-dependent switchings*, in: *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.

[11] Schmöecker, J. D. and W. Adeney, *Metro service delay recovery: comparison of strategies and constraints across systems*, Journal of the Transportation Research Board (2005).

[12] Szabados, T., *An elementary introduction to the wiener process and stochastic integrals*, ArXiv e-prints (2010).

[13] Trivedi, K. S. and V. G. Kulkarni, *Fspns: Fluid stochastic petri nets*, in: *Proceedings of the 14th International Conference on the Application and Theory of Petri nets*, 1993.

[14] Wolter, K., *Modelling hybrid systems with fluid stochastic petri nets*, in: *Proceedings of the 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems*, 2000.