



HAL
open science

Supervision in Multi-Modal Transportation System

Simon Theissing

► **To cite this version:**

Simon Theissing. Supervision in Multi-Modal Transportation System. Modeling and Simulation. Université Paris Saclay (COMUE), 2016. English. NNT : 2016SACLN076 . tel-01419126v3

HAL Id: tel-01419126

<https://inria.hal.science/tel-01419126v3>

Submitted on 8 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2016SACLN076

THÈSE DE DOCTORAT
DE
L'UNIVERSITÉ PARIS-SACLAY
PRÉPARÉE À
L'ECOLE NORMALE SUPÉRIEURE PARIS-SACLAY

ECOLE DOCTORALE N° 580
Sciences et technologies de l'information et de la communication

Spécialité de doctorat : Informatique

Par

M. Simon Theißing

Supervision en Transport Multimodal

Thèse présentée et soutenue à Cachan, le 5 décembre 2016 :

Composition du Jury :

M. Bruno Tuffin	Directeur de Recherche, Inria Rennes	Président
Mme Katinka Wolter	Professeure, Freie Universität Berlin	Rapporteur
M. William Knottenbelt	Professeur, Imperial College London	Rapporteur
M. Olivier Bournez	Professeur, Ecole Polytechnique	Examineur
M. Stefan Haar	Directeur de Recherche, Inria Saclay	Directeur de thèse

Acknowledgements

First and foremost I'd like to thank Stefan Haar, who was so kind to accept me as his PhD student. He taught me how to focus on the essential. I have always appreciated his well-founded criticism and his fruitful propositions, which led to *our* new forecast model for multimodal transportation systems. Apart from the actual supervision of my studies, I'd also like to thank him for the preparation of my research with grateful financial support from the French Program "Investissements d'Avenir" as well as administrative and legal support from the two French research institutes INRIA Saclay and IRT SystemX.

Many research engineers and trainees from Alstom Transport and IRT SystemX contributed to the development, implementation, and testing of the above-mentioned model. I hereby like to thank all of them.

I like to thank Prof. Katinka Wolter and Prof. William Knottenbelt. Both proof-read this report, made many good suggestions, and came to Paris on 5 December 2016 for the oral examination of my PhD thesis. That day I also appreciated the presence of Bruno Tuffin and Prof. Olivier Bournez who joined the examination division, and showed great interest in my research.

Last but not least, I'd like to thank a lot my family, my partner, and my friends for their support, sympathy, patience and interest.

I am very glad to have/have had all of you:

Thanks a lot, merci beaucoup, vielen Dank!

Simon Theiing
Delft, December 2016

Résumé Substantiel

Mots clés: *Automates stochastiques avec une dynamique hybride, Réseaux de Petri, Réseaux de transport multimodales, Systèmes des équation différentielles et stochastiques*

Les réseaux de transport multimodaux modernes sont sans doute essentiels pour la durabilité écologique et l'aisance économique des agglomérations urbaines, par conséquent aussi pour la qualité de vie de leurs habitants. D'ailleurs, le bon fonctionnement sur le plan de la compatibilité entre les différents services et lignes est essentiel pour leur acceptation, étant donné que (i) la plupart des trajets nécessitent des changements entre les lignes et que (ii) des investissements coûteux, dans le but de créer des liens plus directs avec la construction de nouvelles lignes ou l'extension de lignes existantes, ne sont pas à débattre. Voilà pourquoi une meilleure compréhension des interactions entre les modes et les lignes dans le contexte des transferts de passagers est d'une importance cruciale. Toutefois, comprendre ces transferts est singulièrement difficile dans le cas de situations inhabituelles comme des incidents de passagers et/ou si la demande dévie des plans statistiques à long terme. Ici le développement et l'intégration de modèles mathématiques sophistiqués peuvent remédier à ces inconvénients. A ce propos, la supervision via des modèles prévoyants représente un champ d'application très prometteur, analysé dans le présent travail.

La supervision selon des modèles prévoyants peut prendre différentes formes. Dans le présent travail, nous nous intéressons à l'analyse de l'impact basé sur des modèles de différentes actions, comme des départs en retard de certains véhicules après un arrêt, appliqué sur le fonctionnement du réseau de transport en question et sa gestion de situations de stress qui ne font pas partie des données statistiques. C'est pourquoi nous introduisons un nouveau modèle stochastique, un automate hybride avec une dynamique probabiliste, et nous montrons comment ce modèle profondément mathématique peut prédire le nombre de passagers dans et l'état de fonctionnement du véhicule en question du réseau de transport, d'abord par de simples estimations du nombre de tous les passagers et la connaissance exacte de l'état du véhicule au moment de l'incident.

Ce nouvel automate réunit sous un même regard les passagers demandeurs de services de transport à parcours fixes ainsi que les véhicules capables de les assurer. Il prend explicitement en compte la capacité maximale et le fait que les passagers n'empruntent pas nécessairement des chemins efficaces, dont la représentation sous la forme d'une fonction de coût facilement compréhensible devient nécessaire. Chaque passager possède plutôt son propre profil de voyage qui définit un chemin fixe dans l'infrastructure du réseau de transport, et une préférence pour les différents services de transport sur ce chemin. Les mouvements de véhicules sont inclus dans la dynamique du modèle, ce qui est essentiel pour l'analyse de l'impact de chaque action liée aux mouvements de véhicule. De surcroît, notre modèle prend en compte l'incertitude qui résulte du nombre inconnu de pas-

sagers au début et de passagers arrivant au fur et à mesure. Comparé aux modèles classiques d'automates hybrides, notre approche inspirée du style des réseaux de Pétri ne requiert pas le calcul de ces multiples équations différentielles à la main. Ces systèmes peuvent être dérivés de la représentation essentiellement graphique d'une manière totalement automatique pour le calcul en temps discret d'une prévision. Cette propriété de notre modèle réduit le risque de précisions faites par des humains et les erreurs qui en résulteraient.

Notre schéma mentionné plus haut en temps discret pour le calcul de certaines prévisions est très différent des approches de modèles d'automate hybride stochastique développés jusque-là, qui sont limités à l'analyse de propriétés qualitatives intrinsèques ou au calcul de certaines réalisations de l'état du système. D'abord, nous commençons nos calculs à partir d'un état incertain parce que nous ignorons le nombre exact de passagers à bord des véhicules et dans les stations. Nous ignorons en plus la quantité et la composition des arrivées successives de passagers. Deuxièmement, dans notre approche, l'utilisateur final définit l'instant du changement de positions des véhicules, les conditions de conduite et les états opérationnels, et en procédant ainsi, l'instant auquel notre modèle automate change d'état discret.

Nous démontrons l'effectivité et l'efficacité de nos éléments constitutifs, ainsi que la faisabilité des calculs de toute notre approche sur la supervision prévoyant le modèle dans des cas type simplistes et dans des cas plus réalistes d'utilisation. Cette dernière occurrence souligne la facilité d'utilisation et les bénéfices prospectifs de notre approche de supervision prospective modèle en vue de systèmes de transport multimodaux futurs.

CONTENTS

1	Introduction	1
1.1	Model-Predictive Supervision for Improved Multimodal Transportation	1
1.2	Objective of This Work	3
1.3	Context of This Work	3
1.4	Scientific Publications	4
1.5	Conventions & Notations	5
1.6	Outline of this Report	7
2	Simulating Transportation Networks	9
2.1	Structure, State, and Operation of a Multimodal Transportation Network	9
2.2	Intended Scope of Supervision Model	12
2.3	Existent Modelling & Simulation Approaches	13
2.4	Conclusion	23
3	A New SHA Model for Multimodal Transport Networks	25
3.1	Model Overview	26
3.2	Specification of the Infrastructure	43
3.3	Specification of the Vehicle Operation	50
3.4	Specification of the Passenger Routing	57
3.5	Specification of All Passenger Flows	65
3.6	Specification of All Balance Equations	75
3.7	Specification of All Mode Transitions	78
4	Obtaining Forecasts	101
4.1	Preliminary Considerations	102
4.2	Discrete Time Traversal of Complete Forecast Horizon	104
4.3	Bottlenecks	114
4.4	Bricks for the Efficient Computation of Forecasts	117
4.5	Forecast Algorithm	148
5	Implementation & Test	153
5.1	Our Forecast Engine for Transportation Networks	154
5.2	A Quantitative Analysis of the Computation of Propagation DAGs	158
5.3	Ensure Smooth Passenger Transfer Flows At Massy	174
6	Summary & Outlook	189
6.1	Contributions	189
6.2	Obtained Results From Numerical Experiments	190
6.3	Open Problems & Future Prospects	191
	Bibliography	195
	List of Figures	203
	List of Tables	205

INTRODUCTION

This chapter shall motivate our work, define its context, and ease the readability of the remainder of this report as it is indicated in the table of contents below. More specifically, we will try to motivate the development and the use of model-predictive supervision approaches for multimodal transportation systems in Sec. 1.1. In Sec. 1.2 we will then summarize the objectives of our work in more detail, which all focus on the development and test of a model which can be used to anticipate the impact of individual actions (e.g. the voluntary delay of some vehicle departure) in some real multimodal transportation system prior to their executions. Next in Sec. 1.3, we will briefly look at the context of this work. Finally, we will provide an overview of (i) all conventions & notations that we will frequently use in the remainder of this report in Sec. 1.5, and (ii) the structure of this remainder in Sec. 1.6.

Contents

1.1	Model-Predictive Supervision for Improved Multimodal Transportation	1
1.2	Objective of This Work	3
1.3	Context of This Work	3
1.4	Scientific Publications	4
1.5	Conventions & Notations	5
1.6	Outline of this Report	7

Model-Predictive Supervision for Improved Multimodal Transportation

Without any doubt transportation is at the very heart of our daily lives; be it for professional or leisure activities. Because it is doing so, it is of utmost importance for modern societies to maintain efficient transportation systems. However, practice shows that said maintenance is hard to accomplish especially if it requires the construction of new infrastructures, the purchase of new signalling systems, or the purchase of new rolling stock.

The difficulty in maintaining efficient transportation systems is mainly due to tight budgetary constraints and a lack of political consent. However, costly infrastructure projects and the like are not the only option to which we can put out our

hands. Instead we can also go for *soft approaches* which are characterized by small but nevertheless very effective modifications to the way we use and operate our transportation systems; especially approaches which try to facilitate the interoperability of different transportation networks, modes, and lines (see e.g. theme 6 of the 2009 Action Plan on Mobility of the European Commission for Mobility and Transport). Examples include, but are not limited to, multimodal ticketing and real-time passenger information services (see e.g. the D6 final report “Towards a European Multi-Modal Journey Planner” as product of the framework service contract TREN/G4/FV-2008/475/01 of the European Commission for Mobility and Transport).

Many soft approaches to a more efficient operation of our transportation systems simply process the massive amounts of data which we continuously collect about their states in some more or less straightforward manner. However, some other soft approaches require to process more than existing data, namely also complex mathematical models and algorithms for their efficient simulation & analysis. Why? Because access to *massive* data does not mean access to rich data, which implies that purely statistical models have a hard time in correctly predicting the consequences of singular events that they have never seen before. Of course, this weakness of purely statistical models is of concern to us since human transportation systems are prone to many singular events that may negatively effect their operation; be it events originating within these systems due to e.g. technical failures of signalling systems, or external events penetrating them from outside due to e.g. terrorist attacks that cause temporarily unusual high demands for said systems at a few entrance points.

Just because some event has never occurred before, this does not mean that we are not capable to correctly forecast its impact. For instance, if a panic attack in some stadium initiates a run for some nearby train station, then chances are high that we do not know the exact magnitude and composition (e.g. w.r.t. the passengers’ different trip destinations) of the flow of people who leave said stadium towards said station. Even if we knew everything about the flow, we probably would not manage to accurately forecast the its impact on the operation of the considered transportation system just by performing some mental acts. However, this cognitive limitation does restrain us from the execution of some model which does the job when being executed on some technical computing device; a model to which we have mapped all relevant, yet for us very easy to understand, action-reaction principles; a model which we initiate only with a snapshot of that part of our massive data stream that is relevant for the particular event at hand.

In this report we will elaborate a mathematically-rigorous model for a modern multimodal transportation system. To be more precise, our model will focus on a public transportation system, wherein the different modes and lines interact with each other solely by passenger transfers and operational decisions taken on a network level. Moreover, we will show how our model can assist the operators of such a transportation system to forecast the impacts of individual actions, such as the late departure of some vehicle from a stop or the closure of a station, when

being applied to its operation upon occurrence of some downgrading event which lacks statistical data; all this in some short time horizon such as the next twenty minutes upon said event occurrence.

With our model-predictive supervision approach we hope to stipulate the development & analysis of similar soft approaches for safer, more reliable, and more efficient transportation systems that consider the dynamically highly-complex nature of these networked systems as a mathematical and computer-scientific challenge instead of an impenetrable data wall.

Objective of This Work

The objective of this work is to develop a mathematical model for a multimodal public transportation system and a computation scheme for its simulation, which can be used by some human supervisor of said system in some degraded situation to evaluate different corrective actions prior to their actual execution. These corrective actions either modify individual vehicle movements in the considered transportation system, or re-route passengers at local points therein. In order to have any practical meaning, this model must account for capacity limits such as the limited number of passengers on-board the transportation means, and uncertainties; both in form of uncertain passenger arrival flows to the network, and imprecise estimations for all initial passenger numbers. Finally, the present work shall demonstrate the feasibility and prospective benefit of said model-predictive supervision approach in form of some simple use case.

Context of This Work

The work presented in this report contributed to the academic/industrial research project “Modelling, Interoperability, and Communication (MIC)”. It was prepared from September 2013 to August 2016; partly at the premises of the Technological Research Institute (IRT) SystemX, and partly at the premises of the computer science laboratory of ENS Cachan.

IRT SystemX. IRT SystemX was inaugurated in early 2013 as a consequence of some major investment plan initiated by the French government. It provides offices, a legal framework, and some more services to all industrials and academia who come to this place in the south of Paris with the common goal to work together on several R&D projects. These projects - with a 50/50 percent financing from public and private funding - are all related to what IRT SystemX refers to as digital engineering. Their goal is to develop software-implemented solutions (mainly simulation platforms) for their application in different target domains such as smart grids or intelligent transportation systems.

Project MIC. MIC is one of the above mentioned projects under the technological leadership of IRT SystemX with a focus on multimodal transportation systems in urban agglomerations. Amongst others, MIC investigates (i) opportunities for car sharing [Carlier 2015], (ii) efficient algorithms for multimodal route planning [Dib 2015], (iii) approaches for improved car flow modelling & simulation [Sossoe 2015], and (iv) the prospective benefit, cost, and feasibility of multimodal model-predictive supervision approaches (which we consider here).

Scientific Publications

Table 1.1 below lists all our publications that have been published until the compilation of this report.

Table 1.1: List of our publications indicating their underlying mathematical model (MAS: multi-agent system, DHA: deterministic hybrid automaton, SHA: stochastic hybrid automaton), and their contributions to the specification or computation of this model

Publication	Model	Major Contributions
[Theissing 2016b], HAL, Inria	MAS	<ul style="list-style-type: none"> • Stations and transportation grids which decompose network infrastructure • Trip profiles which define routing of passengers • Vehicle missions which define transportation services
[Theissing 2015], ADHS'15, Atlanta	DHA	<ul style="list-style-type: none"> • Fluidification of all passenger movements • Routing matrices which capture passengers' trip profiles
[Theissing 2016a], ACC'16, Boston	SHA	<ul style="list-style-type: none"> • Introduction of uncertain passenger arrival flows and uncertain initial passenger numbers into the model dynamics • Basic idea of discrete time computation of a forecast
[Theissing 2016d], PASM'16, Münster	SHA	<ul style="list-style-type: none"> • Complete set of equations defining discrete time computation of forecast along directed acyclic graph
[Theissing 2016c], QEST'16, Québec C.	SHA	<ul style="list-style-type: none"> • Canonical decoupling of all passenger flows, which targets high-dimensionality of all differential equations systems

Conventions & Notations

Below we list in alphabetical order some themes such as functional analysis and graph theory which decompose all conventions and notations that we will frequently use in the remainder of this report. Of course the reader of this report is not supposed to memorize them, but to use this section as a compact reference point instead.

Abbreviations. All commonly used abbreviations in this report, such as “iff” as abbreviation for “if and only if”, are listed at the end of this report.

Boolean Expressions. If $b_1, b_2, \dots, b_n, n \in \mathbb{N}_{>0}$, are boolean expressions, then

$$\bigwedge_{i=1}^n b_i := b_1 \wedge b_2 \wedge \dots \wedge b_n.$$

Formal Languages. If L is some formal language, then L^* denotes the Kleene closure thereof. We moreover use the shorthand

$$\bigoplus_{i=1}^k m_i := m_1 + m_2 + \dots + m_k$$

to denote the concatenation of $k \in \mathbb{N}_{>0}$ words m_1, m_2, \dots, m_k from \mathcal{L} to the new word $m_1 m_2 \dots m_k$.

Functional Analysis. We denote by $\mathcal{C}^0(X)$ the set of all continuous functions on some domain X . Moreover, we write $g = f(x_1, x_2, \dots, x_n)$ iff g is a function of some $n \in \mathbb{N}_{>0}$ input arguments x_1, x_2, \dots, x_n .

Gender-specific Text Formulations. Sometimes we may refer to passengers as individuals rather than lumped quantities. Assuming this to be true, we use the female sex for all personal and possessive pronouns; without having any prejudice in mind.

Graph Theory. Let $G := (N, E)$ be some directed graph with vertex set N and edge set E . Moreover, let n_1 and n_2 be two of its nodes, and G' be the unique undirected graph which is computed from G in that we (i) adopt the vertex set of G for the vertex set of G' , and (ii) replace every directed edge in G by an undirected edge in G' . We then say that n_1 and n_2 are weakly connected in G iff there exists a path in G' which connects n_1 to n_2 . Accordingly, we say that n_1 and n_2 are disconnected in G iff they are not weakly connected. Finally, say that some other digraph $G'' := (N'', E'')$ with vertex set N'' and edge set E'' is a subgraph of G iff $N'' \subseteq N$ and $E'' = E \cap (N'')^2$.

Linear Algebra. If A is some matrix, then $A[i, j]$ denotes the component in its i -th row and j -th column. Moreover, $A[i, \cdot]$ denotes the i -th row of A , and $A[\cdot, j]$ denotes the j -th column of A . Similarly, if a is a column (row) vector, then $a[i]$ denotes the component in its i -th row (column). Whether these vectors have to be interpreted as column or row vectors will be clear from the context. If S is some set, then $S^{m \times n}$ denotes the set of all matrices with $m \in \mathbb{N}_{>0}$ rows and $n \in \mathbb{N}_{>0}$ columns s.t. $A \in S^{m \times n}$ implies $A[i, j] \in S$ for $i \in \{1, 2, \dots, m\}$ and $j \in \{1, 2, \dots, n\}$. A^t denotes the transpose of some matrix A . If a_1, a_2, \dots, a_k are $k \in \mathbb{N}_{>0}$ column vectors, then we may write $[a_1; a_2; \dots; a_k]$ to denote the column vector

$$\left[(a_1)^t, (a_2)^t, \dots, (a_k)^t \right]^t.$$

Physical Units. If not explicitly mentioned otherwise, we use SI units where applicable. Amongst others, this convention then implies that the initial state of some variable v has to be specified in meters per second if v is a velocity.

Probability Theory. Let X be a continuous random variable. Then $\text{pdf}(X)$ denotes the probability density of X , and $\text{pdf}(X; x)$ denotes the evaluation of $\text{pdf}(X)$ at x ; assuming that x is some admissible realization of X . Let Y be another continuous random variable. Then $\text{pdf}(X, Y)$ denotes the cumulative probability density for X and Y . Let Z be some discrete random variable, and z a particular realization thereof. Then $\mathbb{P}(Z = z)$ denotes the probability that Z equals z , and $\text{pdf}(X|Z = z)$ denotes the conditional probability density for X given that Z equals z . Furthermore, let pdf' some density for Y . We then write $\text{pdf}(X|Y \sim \text{pdf}')$ for the density of X if it is conditioned on the fact that Y 's density is pdf' , and $\text{pdf}(X|Z = z, Y \sim \text{pdf}')$ for the density of X if it is conditioned on the fact that (i) $Z = z$ and (ii) Y 's density is pdf' . Similarly, we write $\mathbb{P}(Z = z|Y \sim \text{pdf}')$ for the probability that Z equals z given the fact that Y 's density is pdf' . Finally, we write - although a little bit awkward looking - $\text{pdf}(X|Z = z, Y \sim \text{pdf}'; x)$ to denote the evaluation of $\text{pdf}(X|Z = z, Y \sim \text{pdf}')$ at x .

Set Theory. \mathbb{R} denotes the set of all the real numbers, and \mathbb{N} the set of all integers. Subscripts written next to \mathbb{R} and \mathbb{N} define particular ranges therein. For instance, $\mathbb{R}_{\geq 0}$ denotes the set of all non-negative real numbers, and $\mathbb{N}_{>0}$ denotes the set of all positive integers. If S is a set, then 2^S denotes the power set thereof. For some set K , we may write

$$U := \bigcup_{i \in K} U_i$$

and say that it is a disjoint union of all sets U_i with $i \in K$. Assuming this to be true, we underline the fact that

$$U_i \cap U_j = \emptyset$$

for $i \neq j$ without writing it down. Moreover, we denote by

$$\bigtimes_{i=1}^n A_i := A_1 \times A_2 \times \dots \times A_n$$

the generalized Cartesian product of some $n \in \mathbb{N}_{>0}$ different sets A_1, A_2, \dots, A_n .

Outline of this Report

The rest of this report is structured into five more chapters. In chapter 2 we will take a closer look at the structure and the operation of a real multimodal (public) transportation system, as well as its particularities w.r.t. other common types of networked systems such as electronic data networks. This study will allow us to define some minimum requirements for the model we are looking for; and we will use them to discuss the suitability of popular modelling & simulation approaches which can be found in the literature. In the conclusion of chapter 2, we will then highlight the need for our new Petri net-styled stochastic hybrid automaton (SHA) model; which we will elaborate in chapter 3 step by step. It is also in this new chapter in Sec. 3.1.5 on p. 41ff., where we will provide an overview of the major milestones in the development of our model for the interested reader. In chapter 4 we will then elaborate a discrete time scheme for the approximate propagation of our SHA model's hybrid state forward in time, and show how this scheme can be used to compute forecasts (passenger numbers, vehicle positions, etc.) with finite time horizons. Next in chapter 5, we will analyse the applicability (= prove of concept) of our modelling & simulation approach, in that we will confront our implemented software module - which gathers together all our developed algorithmic bricks into one common executable - with some simplistic test case and one more realistic use case. It is also this latter use case which we will use to highlight the prospective potential of our model-based predictive supervision approach, namely to ensure smooth passenger transfer flows for the benefit of all including the passengers themselves, the transport operators, and the transport authorities. Last but not least, we will conclude this report in chapter 6. More specifically, we will recapitulate here very briefly our contributions made, results obtained results, and lessons learnt. We will moreover communicate in this chapter to the reader some open problems, and try to motivate her/him to further develop our model & simulation approach.

SIMULATING TRANSPORTATION NETWORKS

In the previous chapter, we have motivated the use of models in the supervision of multimodal transportation systems, where these models enable human supervisors to forecast the time-evolution of the passenger numbers in and the vehicle operational state of these systems when needed. Now in this chapter, we will discuss in more detail what type of models we are talking about, and whether such models already exist. For this purpose, we will briefly look at the structure, the state, and the operation of a real multimodal transportation system in Sec. 2.1; cf. the table of contents below. We will then define in Sec. 2.2 what part of this structure, state, and dynamics is likely to be of great importance to any forecast computation, and thus shall be captured by the model which we look for. Next in Sec. 2.3, we will look for existent models in the literature, and discuss their adequateness for our supervision purposes. Finally, in Sec. 2.4 we will sum up the results of our literature review, which highlights the need for a new model.

Contents

2.1	Structure, State, and Operation of a Multimodal Transportation Network	9
2.2	Intended Scope of Supervision Model	12
2.3	Existent Modelling & Simulation Approaches	13
2.3.1	Preface	13
2.3.2	Transit Assignment Models	13
2.3.3	Petri Nets	14
2.3.4	Max-plus Linear Models	22
2.4	Conclusion	23

Structure, State, and Operation of a Multimodal Transportation Network

Most likely, the reader of this report is familiar with the concept and basic functioning of (multimodal) transportation systems. That is why, neither shall the following paragraphs provide an exhaustive overview of these systems as e.g. in [Board 2013] tries to do, nor shall they replace any good text book or other good teaching material dedicated to some particular aspect of these systems such as

their infrastructure. Instead, the following paragraphs shall highlight their network structure, -state, and -operation. The following paragraphs shall moreover highlight some major particularities, multimodal transportation systems have w.r.t. other, but from a dynamical point of view very similar systems. Note that this latter knowledge is crucial for the literature review on existent modelling approaches, which we will conduct in Sec. 2.3.

Network Structure. Here, in this report we focus on mass transportation systems which are composed of different lines, where every line confines the movements of some vehicles belonging to the same transportation mode along some fixed route. This route connects different stations, and is confined within the boundaries of some well-defined geographical area, such as a city and the suburbs of this city. An example for a transportation line is a commuter (train) line which connects the south of some urban agglomeration with its north.

Note that the above decomposition of transportation systems into lines and stations which connect these lines, reveal their network structure which we can map to multigraphs¹, where nodes represent stations and edges represent sections of transportation lines which connect them. These multigraphs taken alone may then already tell us a lot about e.g. possible conflicts prior to their actual occurrences, although the multigraphs themselves do not capture the state and the operation of the respective transportation networks. We refer the interested reader to [Derrible 2012] which latter publication can be used as a starting point for further reading about the structural analysis of transportation systems, and conclusions which may be drawn from such structural analyses. In addition to the network structure character of transportation systems, note that their decomposition into lines and stations might render them *multimodal networks* if the vehicles operated in their different lines belong to different transportation modes. This possible multimodality property of transportation systems of course is independent of the fact whether or not passengers realize multimodal trips, as long as they could do so in practice between some origin & destination pairs.

Network State. The state of some possibly multimodal transportation network is essentially defined by all passengers in this network who demand the different transportation services, and by all vehicles which provide them. More specifically, we must know the number of all passengers in all stations and on-board all vehicles in some transportation network if we want to know its state. We moreover must know the intended routes of all passengers, so as to e.g. be able to anticipate whether all passenger at some particular platform in some station want to board the same vehicle which arrives next. In addition, we must know the positions, the driving conditions, and the operational states of all vehicles. Only this compound passenger- and vehicle-dependent information allows us to judge whether e.g. some capacity limits, be it the limited number of vehicles in some depot or the

¹graph in which two or more edges may connect the same pair of nodes

2.1. Structure, State, and Operation of a Multimodal Transportation Network¹

limited number of passengers on-board some vehicle, poses some serious threat to the operation of the considered network at some particular point in time which requires immediate action.

Network Operation. In practice, all fixed-route transportation services along the different lines in some transportation network are published in timetables. These timetables do either indicate the scheduled arrival & departure times of all vehicles, or the scheduled arrival & departure time of only one vehicle plus the scheduled arrival & departure times of all following vehicles w.r.t. to it in form of time headways. Whatever the situation is, note that all timetables are the result of some medium- to long term planning, which involves many agreements between some transportation authority on the one hand, and one or more transportation operators on the other hand. This many-party agreement in turn implies that all fixed-route transportation services along any line in the considered transportation network cannot be demand-sensitive; apart from stop requests. This negotiated agreement moreover implies that some transportation operator, who is in charge of the proper execution of all transportation services provided on some particular line, keenly seeks to stick to all published timetables if not explicitly told otherwise. In other words, his/her primary focus is on the punctuality of all transportation services, and not the synchronization of these transportation services with transportation services provided on other lines.

Particularities w.r.t. Similar Systems. By similar systems we mean highly- populated transportation networks, where moving entities can be locally grouped together according to common routes into lumped quantities. Apart from a (public) transportation network, another typical example for such type of a network is the widespread TCP/IP network. We will use this particular type of data transmission network here, in order to demonstrate the particularities of the passenger network which we consider in our report. In this context, note that (i) all passenger flows in a public transportation network are conserved, but this conservation of all moving entities does not necessarily hold in a TCP/IP network where retransmissions of single data packets might be an integral part of the network design. Moreover note that (ii) although both passengers and data packets originate at some point in their respective networks and have some well-defined destination where they want to leave them, the routing of passengers in public transportation networks is fundamentally different from the routing of data packets in TCP/IP networks. In fact, the passengers in a public transportation network try to stick to self-determined routes with the goal to reach their destinations on time. Unanticipated situations might *convince* them to adapt their routes if alternatives exist. The routing of a data packet in a TCP/IP network on the other, is not in the hand of the data packets themselves but defined by routers which in practice are configured in a such a way that the routing of all data packets ensures some specified service levels. Next note that (iii) in a public transportation systems several passengers

are transported in bulks from one physical location to another on-board the same capacity-limited “container”. These temporarily grouped passengers do not necessarily have the same trip destination. In addition, note that (iv) very similar to the original routes, these new routes might be everything else then cost-efficient; whatever the cost function might be. See also [Heye 2003] here, which discusses the perceived route complexity of passengers in public transportation systems, and the way they navigate in these systems. Last but not least, be aware of the fact that most public transportation systems have evolved over several decades if not centuries, and this not homogeneously w.r.t. all interconnected transportation lines and modes meaning that very different grades of automation collide.

Intended Scope of Supervision Model

Remember that the model we are looking for shall be a means to transportation operators, in that it enables them to forecast the state of some possibly multimodal transportation system upon some incident occurrence, so as to anticipate the impact of different operational actions on the network’s future state. Now because some of the considered actions might involve modifications to individual vehicle movements, this supervision model must be able to track individual vehicles, i.e. their positions, driving conditions, and operational states; cf. Tab. 2.1 below.

Table 2.1: Requirements for the model of a multimodal transportation system if it shall be employed in some model-predictive supervision of this system

ID	Description
1	Must respect all capacity limits
2	Must track the positions, the driving conditions, and the operational states of all vehicles
3	Must distinguish different routes of passengers which are not necessarily cost-efficient
4	Must account for the possibility that passengers change their routes at local points in the network during their trips
5	Shall account for variations in the magnitudes and compositions of all passenger arrival flows

On the other hand, the same model can but does not necessarily have to track individual passengers as long as it (i) distinguishes passengers with different routes, and (ii) accounts for the fact that passengers might change their possibly inefficient routes in the considered network during their trips therein. This ignorance of individual passengers is possible since we assume that all strategic considerations

are based on lumped passenger numbers if at all. For instance a transportation operation might consider the evacuation of some station if the number of passengers in this station exceeds some critical limit which is independent of the routes of all passengers. We next demand from any supervision model that it respects all capacity-limits, be it on-board some vehicle's passenger compartment or at some platform in a station. Note that this latter demand is everything else than cherry-picking, but takes into account that transportation systems were planned and are operated in such a way that queues during peak hours and as a result of incidents build. Finally, we demand from any prospective supervision model that it can capture uncertain passenger arrival flows; either explicitly integrated into the model dynamics-defining rules, or implicitly during the simulation thereof.

Existent Modelling & Simulation Approaches

Preface

In the previous section we have elaborated what the model we are looking for our model-predictive supervision purposes must be able to capture and to replicate during its simulation. However from a practical point of view we need to demand more. In particular, the model we are looking for must also have some rigorous mathematical framework as well as some easy to understand graphical specification. The rigorous mathematical framework is necessary since it allows us to trace back the influence of any model or simulation parameter from the computed simulation result to the model's initial state. Moreover it allows us to develop mathematically rigorous algorithms for its analysis. The model's easy to understand graphical specification on the other hand, is essential if we want to reduce the risk of any man made specification errors. Furthermore, this graphical specification is of utmost importance for the practitioner of our supervision approach bearing in mind that real transportation networks underlie regular modifications, although these modifications might be very small. In other words, a model where all equations have to be written by hand, and this after every minor modification to the structure, the rolling stock, and the operation of this rolling stock of the considered transportation network, is likely to be everything but acceptable. That being said, all models which we could find in the literature and which fulfil all above requirements, can be counted on one hand. They are either transit assignment models, Petri net models, or max-plus linear models. We discuss them one by one in the given order in the next three sections.

Transit Assignment Models

Transit assignment models belong to a particular class of network flow models. They allocate passenger flows between origin/destination pairs in public transportation networks to finite sets of admissible routes in between. For this purpose they translate the passengers' preferences for alternative transportation ser-

vices at local decision points in the considered networks to linear cost functions. These cost functions together with linear algebraic constraints which shall enforce maximum passenger flows between all decision points, form linear programming problems that can be efficiently solved on general purpose computers; hence their popularity.

As usual for network flow models, the specification of a transit assignment model is done in form of some edge-labelled digraph. All nodes therein either correspond to sources, sinks, or decision points in the considered network where passenger flows diverge or converge. Edges on the other hand correspond to the availability of transportation services between these nodes. Finally, edge labels define upper limits for the passenger flows along all edges.

According to the literature [Fu 2012], there are two different types of transit assignment models, namely static frequency-based (traffic assignment) models, and dynamic schedule-based (traffic assignment) models. We will have a brief look at both model variants next, which shall allow us later on to decide whether or not they are possible candidates for our model-predictive supervision approach.

Frequency-based Models. Frequency-based transit assignment models are referred to as static transit assignment models due to the fact that they completely ignore all timetables in the considered network. Instead they assume constant services frequencies along all possible trip legs that may be derived from the actual (scheduled) services frequencies as averaged values over longer time horizons. Nodes in these models refer to physical locations such as stations or platforms in stations.

Schedule-based Models. Like static or frequency-based transit assignment models, dynamic or schedule-based transit assignment models are also flow networks; however, with the subtle difference that they unfold all timetables of the target transportation networks. As a consequence of this unfolding, a node in their edge-labelled digraphs does not simply correspond to a particular location any more, but to a particular pair of a location and (simulation) time step; hence the name time expanded node [Hamdouch 2008, p. 665-669]. Similarly, every edge connecting two nodes does not refer to the availability of some frequency-based transportation service between these two nodes any more, but to some scheduled transportation service instead.

Petri Nets

Simply speaking a Petri net is a mathematical model for a discrete event dynamical system (DES), which can capture concurrency and the competition for shared resources in such a system besides further important phenomena. Note that this modelling capability of Petri nets is of utmost interest to us since they are omnipresent in the transportation networks we like to model. As an example for concurrency take for instance the positions and operational states of vehicles operated

on the same or on different transportation lines are likely to evolve independently from each other within some space and time constraints.

We will introduce the classical Petri net model in the next paragraph. This original model however has no notion of time, and we thus cannot use it for our model-predictive supervision purpose. That is why, we will look at timed extensions of this original Petri net model in a subsequent paragraph. It is also this paragraph, where we will briefly discuss different ways to include uncertainty in timed Petri net extensions. After this, we will look at two more extensions to the original Petri net modelling formalism, namely nets-within-nets models and hybrid-dynamical Petri nets.

Classical Model. Figure 2.3.1 below depicts a classical Petri net in some particular marking.

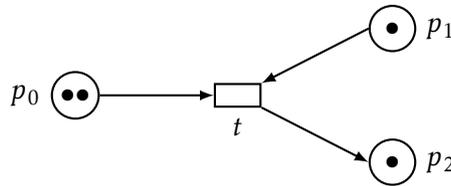


Figure 2.3.1: Reference Petri net with marking before transition firing

The circles p_0 , p_1 , and p_2 therein are called places. The dots inscribed to all places are called tokens. Adopting this Petri net-specific terminology, p_0 thus holds two tokens which may correspond to two passengers at a platform. The token in p_1 may correspond to the free capacity of a stopped vehicle, and the token in p_2 may correspond to a passenger on-board this stopped vehicle. Together all tokens define the marking of the depicted Petri net, and in doing so the purely discrete state of this model which e.g. captures a boarding procedure. Apart from places, the above Petri net has one transition t , and directed edges which connect t to p_0 , p_1 and p_2 ; and vice versa. This t might implement the possibility that passengers at p_0 enter p_2 , i.e. the cabin of the aforementioned stopped vehicle.

In general, the structure of any conventional Petri net is specified in form of a bipartite digraph, where all nodes can be grouped into two disjoint sets, namely one set of places and another disjoint set of transitions. This digraph is referred to as the place/transition net of the Petri net (model).

Definition 1 (Place/Transition Net) A place/transition net $N := (P, T, E)$ is a bipartite digraph. Its nodes are grouped into two disjoint sets; the set P of places, and the set T of transitions. Every edge $(a, b) \in E \subseteq (P \times T) \cup (T \times P)$ connects a place to a transition iff $a \in P$ and $b \in T$, or a transition to a place iff $a \in T$ and $b \in P$.

The number of tokens in every place of some place/transition net defines its marking, and in doing so the discrete state of the Petri net this place/transition net belongs to.

Definition 2 (Marking) *The marking of some place/transition net N is a function $\mu_N : P \rightarrow \mathbb{N}_{\geq 0}$ which maps every place $p \in P$ in N to some non-negative integers $\mu_N(p)$.*

So far we have formally introduced the structure and the state of a Petri net model in form of a place/transition net and the marking thereof. However, we have not introduced rules which define under which condition and how the marking of a Petri net model may change starting from some reference marking. We will do so in a short. However, let us introduce four more definitions first, which we will extensively use in the rest of this report. They are all related to the place/transition net of some Petri net model.

Definition 3 (Preset of a Place/Transition) *The preset of a place or a transition $a \in P \cup T$ from some place transition/net N is*

$$\bullet a := \{b \in P \cup T : (b, a) \in E\}.$$

Hence, the preset of some place (transition) $a \in P$ is the subset of all those transitions (places) $b \in \bullet a$ in N , which are connected to a by an edge pointing towards a . On the other hand, the postset of a in N is the subset of all those (transitions) places in N which are connected to a by an edge pointing away from a iff a is a place (transition).

Definition 4 (Postset of a Place/Transition) *The postset of a place or a transition $a \in P \cup T$ from some place transition/net N is*

$$a^\bullet := \{b \in P \cup T : (a, b) \in E\}.$$

Given these first two definitions, note that in Fig. 2.3.1 above $\bullet t = \{p_0, p_1\}$ and $t^\bullet = \{p_2\}$. The third new definition formally introduces a path in N .

Definition 5 (Path in a Place/Transition Net) *A path in a place/transition net N of length $n \in \mathbb{N}_{>0}$ is an alternating sequence $\pi := a_0 a_1 \dots a_n$ of $n+1$ places and transitions s.t. (i) $a_0 \in P \cup T$, (ii) $a_i \in \bullet(a_{i+1})$ for $i \in \{0, 1, \dots, n-1\}$, and (iii) $a_i \neq a_j$ for $i \neq j$ iff $(i, j) \neq \{(1, n+1), (n+1, 1)\}$.*

An example for a path in the place/transition net depicted in Fig. 2.3.1 is $p_0 t p_1$, and another example is $p_2 t p_0$. Finally, the fourth definition formally introduces condition for some place/transition net to be weakly connected; which means that there is a path in this net between every pair of a place and a transition when the orientation of all edges is disregarded.

Definition 6 (Weakly Connected Place/Transition Net) *A place/transition net N is weakly connected iff for any pair (a, b) , with $a, b \in P \cup T$, of two places, two transitions, or a place and a transition, there is a path from a to b , and/or vice versa.*

We now come back to the rules which define under which condition and how the marking of a Petri net can change assuming that this net has some reference marking which is known to us. In fact, every prospective altering of this reference

marking is uniquely defined by two rules. The first rule is called the enabling rule, and it defines under what condition any transition therein can *fire*; which does not mean that this transition firing has to occur.

Definition 7 (Enabled Transition) *A transition $t \in T$ is enabled in some marking μ_N of a place/transition net N iff $\mu_N(p) \geq 0, \forall p \in \bullet t$.*

In addition, the second rule defines the impact of every transition firing assuming that it occurs.

Definition 8 (Transition Firing) *A transition $t \in T$ can fire in the marking x of the place the place/transition net N iff it is enabled in this marking according to Def.*

Applying the first of the two above rules to our Petri net from Fig. 2.3.1, implies that t is enabled in the depicted marking. According to Def. 7 above, t thus can fire, which latter firing - if it occurs - produces a new marking which is depicted in Fig. 2.3.2 below.

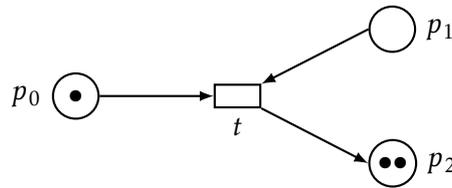


Figure 2.3.2: Reference Petri net from Fig. 2.3.1 with marking after transition t has fired

Thus, a firing of t *destroys* one token in both p_0 and p_1 each, and *creates* at the same time another token in p_2 . Note that in this new marking t is no longer enabled, which implies that t cannot fire one more time. Moreover, note that if we associate with the Petri net from Fig. 2.3.1 the boarding process of passengers at p_0 to a stopped vehicle, then the firing of t implies that the remaining passenger at p_0 (= remaining token therein) cannot board this stopped vehicle because it is full.

According to Def. 7 and Def. 8 above, the Petri net as shown in Fig. 2.3.2 above is stuck forever in the depicted marking. This fixation in some particular marking without the possibility of any transition firing is sometimes referred to as a deadlock and might e.g. correspond to the situation where the execution of some program code is frozen for whatever reason. In fact, most analysis techniques developed for Petri nets in the past focus on the verification of this kind of qualitative properties of DED systems; in that they prove their existence or absence. For more information on this topic, see e.g. the first two chapters of [Desel 1995]. However, note that we are not interested in those qualitative analysis techniques such as the absence of deadlocks in some signalling system here. The questions we try to answer in our model-predictive supervision approach instead, rather relate to the time evolution of quantitative measures such as the number of passengers in some transfer hub. Obviously a classical Petri net model cannot be used for this kind of performance analyses since it has no explicit notion of time.

Time and Uncertainty. As outlined in the previous paragraph, classical Petri nets have no explicit notion of time and thus cannot be used for our model- predictive supervision purpose. Luckily however, timed extensions of this model exist. They either confine all transition firings to some countable set of discrete points in time and are thus usually referred to as discrete time Petri nets, or they do not and are thus referred to as continuous time Petri nets. This distinction between discrete and continuous time Petri nets taken alone, does not tell us how the simulation time during the execution of such a model pushes the evolution of its marking. However in most model realizations, this link between simulation time on the one hand and firing occurrence time on the other hand is relatively straightforward. These timed models assign counters to all transitions, namely for every transition one separate counter. This counter either decreases or increases as a function of the model's simulation time as long as the transition it is assigned to is enabled, which latter property is a function of the net's marking. Once a counter hits some pre-specified threshold, the transition, to which this counter is assigned to, fires. This firing in turn resets the counter to some pre-specified value.

If the initial marking of some timed Petri model as described above shall completely define its future marking as a function of its simulation time, then rules must define how possible conflicts among simultaneously enabled transitions are resolved, assuming that these transitions have input places in common. In principle, two fundamentally different alternatives for this conflict resolution exist. Either unique priorities assigned to all transitions resolve possible conflicts among them in a deterministic manner, or these conflicts are resolved in a probabilistic manner. This latter possibility brings us to uncertainty, or more specifically to the question where and how uncertainty can be introduced into the formalism of timed Petri nets. In principle, there are three different possibilities to do so. Apart from the probabilistic resolution of conflicting transition firings, uncertainty can moreover be introduced in the initial markings of those nets in form of probability mass functions over a range of admissible markings, and in the time lapse between the enabling of some transition and its subsequent firing.

We cannot go into further details here, and in doing so discuss amongst others all thinkable timed Petri net constructions; be it deterministic or probabilistic. Instead, we refer the interested reader to two very generic models, namely the (continuous time) generalized stochastic Petri net model as elaborated in [Marsan 1994] where the firing times of all so-called timed transitions upon firing enabling are exponentially distributed as opposed to the immediate firing times of all remaining transitions, and the discrete time stochastic Petri nets as elaborated in [Zijal 1994] where all firing times upon firing enabling are geometrically distributed and constant firing times are a special case of this geometric distribution.

Nets-Within-Nets. The structure of any timed extension of a classical Petri net introduced so far is defined by some place/transition net, and its discrete state by the marking thereof. Unfortunately in practice, it can be very difficult to map the

state and the structure of a real system to this graphical specification of a Petri net. Even if the mapping is achieved, the obtained Petri net model might lack clarity since the meaning of its places and the tokens therein is everything but intuitive. A look into the literature suggests that also public transportation systems belong to those real systems which cannot easily be mapped to classical Petri nets or timed extensions thereof. Note that this difficulty is not due to the size of such a system. In fact, the network character of a transportation system permits a well-defined decomposition of its infrastructure, which can be exploited very effectively in a modular specification of a Petri net model for this system. See e.g. [Giua 2008], which depicts the modular specification of a Petri net for the performance analysis of a rail network. Instead, it looks like the real problem for a modeller to map the structure and the state of a transportation network to a classical low-level Petri net is the inconvenient fact that two very different parties in this network interact with each other in a very complicated manner. These two parties involve the passengers who demand transportation services on the one hand, and the vehicles which provide them on the other hand. Although their interaction is limited to a few boarding and alighting processes at local points in the network, it is these boarding and alighting processes, together with the fact that passengers travel on-board vehicles in batches but may go separate ways in the rest of the transportation infrastructure, which makes the Petri net modelling very cumbersome if not impossible. Thus, without any big surprise only very few publications including [Lopez 2011] integrate both - either implicitly or explicitly - the movements of all vehicles and all passengers, as well as their interactions in one and the same Petri net model. So-called Object Petri nets may provide remedy here. Compared to classical Petri nets, the markings of these high-level Petri nets are not limited to simple tokens but may comprise complex token objects with internal structures and states of their own. Applied to transportation networks, these token objects for instance could correspond to individual vehicle- and passenger agents. The internal states of these token agents could then correspond to their positions in the considered transportation infrastructure. This way of modelling a transportation network by means of Object Petri nets is demonstrated e.g. in [Abbas-Turki 2002].

Some major drawback of Object Petri nets w.r.t. to classical Petri nets is their enhanced complexity both in terms of specification and analysis. Depending of the particular use, nets-within-nets however might overcome this drawback. They build a subclass of Object Petri nets with the particularity that all token objects are Petri nets of their own which reside in the places of some host Petri net. For illustration purposes, we have depicted some simple realization of such a nets-within-nets model in Fig. 2.3.3 in two different states, where every state of this model is defined by the nested marking of its host net. This host net comprises the two places p_0 and p_1 , and the transition t which connects p_0 to p_1 . The state of our reference model in turn, is defined by the marking of its host net, i.e. the tokens in p_0 and p_1 . In Fig. 2.3.3a, p_1 is empty. However, one token resides in p_0 . This single token in p_0 is a *net token* whose structure is defined by a place/transition net with the two places π_0 and π_1 , and the transition τ which connects π_0 to π_1 . The net

token's state is defined by the marking of its net, i.e. the single token in π_0 .

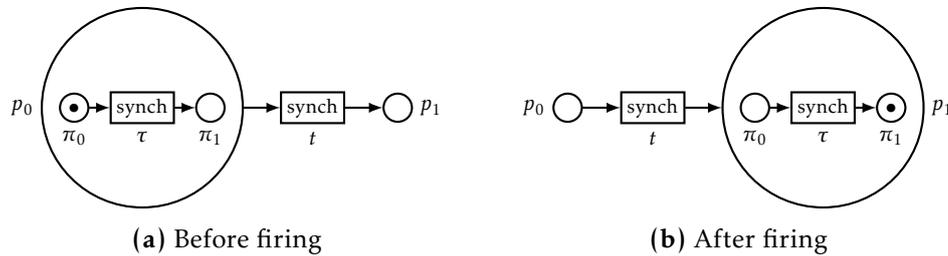


Figure 2.3.3: A simple nets-within-nets model

Both t in the host net of our reference model from Fig. 2.3.3a above, as well as τ in the net of the token which resides in p_0 have the same label “synch”. This label shall indicate a synchronization of both transitions when being fired. In other words both t and τ can only fire together but not alone. The evaluation of whether or not t and τ can fire together is relatively straightforward. We thus do not introduce any definitions here, but only explain the main ideas. In fact, all that we have to do, is to employ the enabling rule from Def. 7 which we have already introduced for a classical Petri net on p. 17. According to this rule, τ can fire in the marking which is depicted in Fig. 2.3.3a. Similarly, if t is allowed to fire if we neglect the structure and the internal state of the net token which resides in p_0 . Hence, both t and τ may fire in the marking from Fig. 2.3.3a. Upon occurrence this firing then produces the new marking which is depicted in Fig. 2.3.3b. Note that this new marking complies with the firing rule for a classical Petri net from Def. 8 on p. 17. According to this rule, a firing of τ destroys a simple token in π_0 and creates another token in π_1 . A subsequent firing of t then *moves* the updated net token from p_0 to p_1 , although strictly speaking there is not movement at all but the destruction of some existing net token at p_0 and the creation of some new net token at p_1 .

Nets-within-nets were first introduced in the literature under the name of Elementary Object Nets by the author of [Valk 1998] as some particular realization of Object Petri nets. Shortly after, some publications demonstrated their potential use for the analysis of environments including transportation networks where mobile agents represented by net tokens interact with each other in some common infrastructure [Bednarczyk 2005, Köhler 2003].

Hybrid-Dynamical Petri Nets. Although in principle possible, in practice we cannot employ any of the timed extensions of a classical Petri net model which we have introduced so far for our model-predictive supervision purpose if we want to explicitly include uncertainty such as the exact arrival time of a passenger at a stop therein. We cannot do so since the number of admissible markings of this purely discrete (state) mathematical model would exponentially grow as a function of its simulation time; implying that the exhaustive traversal of the model's complete state space and thus the computation of any stochastic measure for the considered

transportation network would become intractable. However note that a (at least partly) fluidification of the timed Petri net model extension at hand could provide remedy here.

The principle idea underlying the fluidification of some discrete Petri net model is to relax the firing of a set of transitions therein (called continuous transitions) from integer multiples to real values [Silva 2004]. As consequence of this relaxation, some places (called continuous places) are no longer filled with an integer number of tokens in some particular marking of the net, but liquids instead. Note that this fluidification idea has stipulated the development of hybrid-dynamical Petri nets (hereafter also referred to as fluid Petri nets), where some transitions continuously decrease (increase) the fluid levels of the places in their presets (post-sets) according to some marking-dependent rate. We do not go into details here, but refer the interested reader to [David 2010]. Here, only note that many fluid Petri net formalisms have been developed in the past, both for deterministic and stochastic model realizations. For instance, [Febbraro 2004] discloses a deterministic fluid Petri net model with a piece-wise linear dynamics for the performance analysis of an urban street network, or more specifically for the performance analysis of the traffic light system (discrete part of the net) controlling all vehicle flows (continuous part of the net) therein. [Castelain 2002] discloses a similar model for the same purpose but with a non-linear dynamics. [Mahi 2012] in turn discloses a deterministic fluid Petri net model for the analysis of passenger transfers in multimodal transportation systems for a particular configuration of all vehicle positions. However, the authors of [Mahi 2012] do not explain how all vehicle movements are mapped to their model; apart from the fact that they shall be captured by the flow of ordinary tokens in the discrete part of their net. On the part of stochastic model realizations, [Trivedi 1993] introduces a fluid stochastic Petri net model, where the firing times of some transitions are exponentially distributed; the occurrences of the transition firings themselves however are not conditioned on the fluid levels of its continuous places as it is e.g. the case in the fluid stochastic Petri net from [Horton 1998]. The authors of [Wolter 2000] slightly modify this model from [Horton 1998], in that they allow for uncertain firing rates, thus producing a fluid stochastic Petri net model where the evolution of the model's continuous state is governed by a set of stochastic differential equations.

Although, we look here for mathematical models such as FSPNs which come with some kind of high-level specification language out of the box, note that this limitation does not prevent us from employing algorithms for their simulation and analysis which have been developed for other models. In this context, note that many algorithms which have been developed for e.g. hybrid systems [Hu 2000] including - but not limited to - the grid-based asymptotic approximation method from [Prandini 2007] can also be adopted for FSPNs [Tuffin 2001]; in addition to the many algorithms which have been explicitly developed for Petri nets.

Max-plus Linear Models

A max-plus linear model for a discrete event system (DES) is a system of max-plus algebraic equations, where each equation relates the occurrence time of some particular event in the considered DES to the occurrence times of all other events [De Schutter 2008].

The only two (binary) operations in a max-plus linear system are the maximization of two real-valued operands and their addition. In practice these two binary operations together with the null element $\epsilon := -\infty$ are sufficient to model many synchronized systems. As an example, assume that the execution of some fictitious DES is completely characterized by the occurrence times of two different events therein. Let $x_1(k)$ with $k \in \mathbb{N}_{\geq 0}$ denote the k -th occurrence time of the first event, and $x_2(k)$ the k -th occurrence time of the second event. Moreover for the first event assume that its k -th occurrence happens after some $\tau_1 \in \mathbb{R}_\epsilon := \mathbb{R} \cup \{\epsilon\}$ seconds have been elapsed w.r.t. its previous occurrence. For the second event on the other hand assume that its k -th occurrence does not happen before (i) some $\tau_2 \in \mathbb{R}_\epsilon$ seconds have been elapsed w.r.t. its previous occurrence, and (ii) the first event has occurred $k - 1$ times. Accordingly, system of linear algebraic equations which defines the dynamics of our fictitious system is

$$\begin{aligned} x_1(k) &:= x_1(k-1) \otimes \tau_1 \\ x_2(k) &:= [x_1(k-1) \otimes \tau_2] \oplus x_2(k-1) \end{aligned} \quad (2.1)$$

if $a \otimes b := \max(a, b)$ for any $a, b \in \mathbb{R}_\epsilon$, and $a \oplus b := a + b$. The former operation \otimes is referred to as max-plus multiplication, and the latter operation \oplus is referred to as max-plus addition. Note that this choice of naming and symbols seems to be very confusing only at first glance of this model. It results from the fact that many properties and concepts from can be translated from the conventional linear algebra to the max-plus linear algebra which underlines this model [De Schutter 2008, p. 38].

Beside the fact that many efficient algorithms for the computation of max-plus linear models have been developed in the past, this model also has the important property that its equations may be fully derived from the graphical specification of some so-called Timed-event Graph (TEG). This TEG is a classical Petri net, i.e. a marked place/transition net with enabling and firing rules from Def. 7 and Def. 8 on p. 17, with the particularity that every place therein exactly has one upstream and one downstream transition [Heidergott 2005, p. 116].

Most max-plus linear models that can be found in the literature for the performance analysis of public transportation network, either fully disregard all passenger movements or they focus on the vehicle movements in one single line while disregarding the existence of all other lines. An very exception is the linear max-plus model from [Nait-Sidi-Moh 2002a] and [Nait-Sidi-Moh 2002b], which can be used to study the impact of passenger transfers between feeder- and target lines on their operational performance; where the basic modelling assumption is that a vehicle operated on a target line may not depart from a stop before the passenger

transfer from a feeder line is completed. However in practice, the target vehicle may depart before if e.g. some time threshold has elapsed. Moreover note that this model does not capture any other passenger movements apart from transfer flows between feeder- and target lines.

Conclusion

In Sec. 2.1 we have briefly looked at the line-structure, the line-operation, and the state of a modern and possibly multimodal transportation network. We used this recap for the formulation of some requirements in Sec. 2.2 which we demand from any model that shall be employed in our model-predictive supervision approach; apart from the very important requirement that this model shall have an intuitive graphical specification as well. Unfortunately, from our model review in Sec. 2.3, we must conclude that such a model does not yet exist. More specifically, we have looked at three different candidate models in more detail, namely transit assignment models, Petri net models, and max-plus linear models; and realized that none of them fulfills all requirements out of the box. For instance, transit assignment models seem to be good candidates for the performance analysis of transportation networks iff (i) vehicle movements and modifications thereof do not have to be explicitly taken into account during the model execution, and (ii) sufficient historical data is available for the definition of a cost function which defines the route allocation of all passengers in this network. However, for most degraded situations that are of interest to us, historical data does not exist. Moreover, remember that we like to analyse the impact of modifications applied but not limited to individual vehicle movements. Max-plus linear models on the other hand, seem to be a very powerful tool when the performance of individual vehicle synchronizations at local points in the considered network shall be analysed. However, passenger movements or congestion are not included in these models so far, and there seems to be no easy way to add them. Finally, Petri net model seem to be a good choice when either vehicle movements or passenger flows are to be analysed; but not both at the same time either due to the illegibility of the net (classical timed Petri nets, fluid Petri nets) or the explosion of its discrete state (nets-within-nets model).

A NEW SHA MODEL FOR MULTIMODAL TRANSPORT NETWORKS

In this chapter, we introduce a new stochastic hybrid automaton (SHA) model for mono- and multimodal transportation networks alike; which we denote by the four italicized letters *T-SHA*; cf. the table of contents below. In Sec. 3.1 we provide a definition for *T-SHA* including all basic rules and elementary equations which govern its hybrid dynamics. We also provide a short overview of all principle milestones in its development here; from a purely discrete event-dynamical Petri net-styled multi agent (MAS-)system model to a stochastic differential equations automaton model. In Sec. 3.2 to 3.4 we then elaborate *T-SHA*'s structure in more detail, which involves the infrastructure specification of the considered transportation network by means of place/transition nets, as well as the specification of the vehicle operation and the passenger routing therein by means of some other mathematical objects. In Sec. 3.5 and Sec. 3.6, we then focus on the elaboration of *T-SHA*'s continuous passenger flow-dynamics in that we elaborate the proper specification of five different types of passenger flows, and the formulation of all balance equations by means of so-called routing matrices. Last but not least, we discuss in Sec. 3.7 the specification of *T-SHA*'s mode transitions.

Contents

3.1	Model Overview	26
3.1.1	Modelling Framework	26
3.1.2	Infrastructure, Vehicle Operation, and Passenger Routing . .	28
3.1.3	Passenger Numbers, Modes, and Clocks	31
3.1.4	Passenger Flow- and Vehicle Movement Dynamics	34
3.1.5	For the Interesting Reader: How the T-SHA Model Evolved .	41
3.2	Specification of the Infrastructure	43
3.2.1	Stations	43
3.2.2	Transportation Grids	45
3.2.3	Interface Between All Stations and Transportation Grids . . .	46
3.3	Specification of the Vehicle Operation	50
3.3.1	Vehicle Missions	50
3.3.2	Vehicle Runs	52
3.3.3	Scheduled Vehicle Dispatches	53
3.3.4	Signalling System	54

3.3.5	Consistent Vehicle Operation	56
3.4	Specification of the Passenger Routing	57
3.4.1	Trip Profiles	57
3.4.2	Passenger Re-Routing	60
3.4.3	Derivation of Routing Matrices	61
3.4.4	Verification of the Proper Routing of All Passenger Flows	64
3.5	Specification of All Passenger Flows	65
3.5.1	Overview	65
3.5.2	Network Outflows	68
3.5.3	Transfer Flows	70
3.5.4	Boarding & Alighting Flows	73
3.5.5	Network Inflows	74
3.6	Specification of All Balance Equations	75
3.7	Specification of All Mode Transitions	78
3.7.1	Some Preliminary Considerations	78
3.7.2	Useful Definitions for the Specification of All Atomic Actions	81
3.7.3	Specification of Passenger Number-Dependent Actions	83
3.7.4	Specification of Passenger Number-Independent Actions	89

Model Overview

Modelling Framework

First of all, let us briefly review what we need. We need a mathematical model, which describes the time evolution of some - possibly multimodal - transportation network's uncertain state, so as to compute forecasts of this state some finite time into the future starting from some initial estimations/measurements; the passenger loads at all gathering points in the considered stations, the passenger loads on-board all considered vehicles which provide transportation services to the passengers, as well as the discrete positions, the driving conditions, and the operational states of all these vehicles.

Whereas the model must track individual vehicles both in space and time, all passengers with similar trip profiles can be grouped together to lumped quantities; where in the interpretation of our model a passenger's trip profile defines some particular route in the considered infrastructure together with a preference for alternative/competing transportation services along this route.

The literature review in chapter 2 showed that to the best of our knowledge such a mixed meso- and macroscopic *mathematical* model has not been formulated/introduced so far. The existing mathematical models/modelling approaches focus on some particular aspect such as the vehicle movements in a fixed-block

system or some boarding process of a stopped vehicle; or they disregard uncertainty inherent to transportation networks; or they abstract away from all vehicle movements/do not take them explicitly into account. That is why, we introduce our own Petri net-styled automaton model, where discrete moving vehicle agents provide transportation services to all passengers or execute dead-headings. All passengers with the same trip profile stay where they are or flow like a phase in a multiphase fluid in a network of pipelines connecting capacity-limited reservoirs; where these reservoirs correspond to discrete locations in the considered stations such as platforms and to all vehicles stopped in front of them for the purpose of boarding & alighting. Every admissible configuration (called mode) of the discrete positions, discrete driving conditions, and discrete operational states of all considered vehicles uniquely defines all possible passenger flows in the considered network, and in doing the passenger flow dynamics of our automaton.

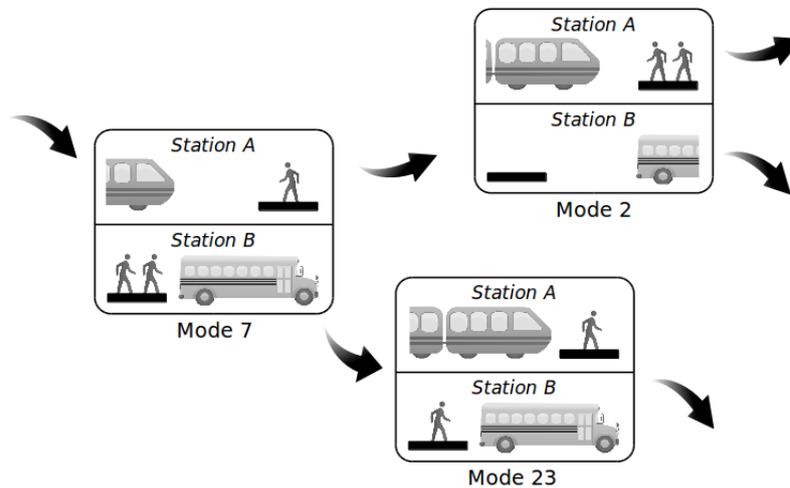


Figure 3.1.1: Every possible configuration (called mode; discrete state of our SHA model) of all vehicles' discrete positions, driving conditions, and operational states defines which passenger flow between the platform in a station and a vehicle stopped in front is possible. In doing so, this configuration/mode uniquely defines the continuous passenger flow dynamics in our model. Mode transitions might be coupled to some passenger numbers, to some time constraints, or both. Since in general all passenger numbers are (continuous) random variables, most mode transition times are (continuous) random variables themselves; which can produce alternative trajectories for the time evolution of our model's mode as it is indicated in this picture.

All mode transitions are defined by deterministic rules such as the departure of a vehicle from a stop, which might be conditioned e.g. on the number of passengers on-board this vehicle who want to alight from it with or without the necessity to respect some minimum or maximum dwell time. Moreover, we assume that all driving times are pre-defined constants, and as such are not subject to any uncertainty. In other words, we know at which time a vehicle will arrive at

its destination upon departure from its previous position. However, the fact that the passenger numbers are in general random variables (we do not know at which time the passengers arrive at the network; neither do we know their trip profiles) renders most mode transition times random variables themselves. Exceptions do exist; such as the scheduled dispatch of some parked vehicle, which we allow to occur at some predefined point in time, and this independently from the time evolution of the transportation network's state.

Infrastructure, Vehicle Operation, and Passenger Routing

We map the infrastructure (see Sec. 3.2 on p. 43ff.) of the considered transportation network, the vehicle operation (see Sec. 3.3 on p. 50ff.), and the passenger routing (see Sec. 3.4 on p. 57ff.) therein to a collection of mathematical objects which we denote by the two italicized capital letters TN .

Infrastructure. We decompose the infrastructure of the transportation network at hand into two disjoint sets; a finite set \mathcal{S} of stations (see Sec. 3.2.1 on p. 43ff.) and another finite set \mathcal{G} of transportation grids (see Sec. 3.2.2 on p. 45ff.). Every station $s \in \mathcal{S}$ accommodates waiting and/or transferring passengers; waiting for a particular vehicle ride, and transferring between discrete locations in s such as two platforms. Every transportation grid $g \in \mathcal{G}$ on the other hand, accommodates all vehicles of a particular mode or line, and defines all possible movements therein. A typical example for a transportation grid is the rail grid of a metro line. Finally, an interface \mathcal{I} defines the connection between all stations and all transportation grids for the purpose of boarding and alighting.

Vehicle Operation. By assumption, TN 's different transportation grids do not share infrastructure elements, which implies that both $\bigcup_{g \in \mathcal{G}} P_w^{(g)}$ and $\bigcup_{g \in \mathcal{G}} T_{tr}^{(g)}$ are disjoint unions. Vehicles thus cannot transfer in between, which in turn allows us to decompose TN 's finite set \mathcal{V} of vehicles into the disjoint union $\mathcal{V} := \bigcup_{g \in \mathcal{G}} \mathcal{V}^{(g)}$, where $\mathcal{V}^{(g)}$, for some transportation grid $g \in \mathcal{G}$, denotes the subset of all vehicles that are operated in g . Similarly, we can decompose the finite set \mathcal{W} of all vehicle missions (see Sec. 3.3.1 on p. 50ff.) into the disjoint union $\mathcal{W} := \bigcup_{g \in \mathcal{G}} \mathcal{W}^{(g)}$, the finite set \mathcal{Z} of all vehicle runs (see Sec. 3.3.2 on p. 52ff.) into the disjoint union $\mathcal{Z} := \bigcup_{g \in \mathcal{G}} \mathcal{Z}^{(g)}$, and the finite set \mathcal{D} of all vehicle dispatches (see Sec. 3.3.3 on p. 53ff.) into the disjoint union $\mathcal{D} := \bigcup_{g \in \mathcal{G}} \mathcal{D}^{(g)}$. Together with the signalling SI (see Sec. 3.3.4 on p. 54), they define the operation of $v \in \mathcal{V}$ in TN . At any time every vehicle in operation executes a mission $w \in \mathcal{W}^{(g)}$ in some transportation grid $g \in \mathcal{G}$. This mission either defines a dead-heading or a transportation service in g in form of a path and a sequence of stops along this path. All vehicle runs $z \in \mathcal{Z}^{(g)}$ concatenate some vehicle missions specified for the same transportation grid $g \in \mathcal{G}$ that have to be executed in the given order by a vehicle that processes z . Every scheduled vehicle dispatch $d \in \mathcal{D}^{(g)}$ defines at which time which vehicle in g is

supposed to start processing which vehicle run. Finally, the signalling system SI defines the vehicles' driving times in all transportation grids.

Passenger Routing. We group all passengers into a finite set \mathcal{Y} of trip profiles (see Sec. 3.4.1 on p. 57ff.). At any time every passenger travels according to one particular trip profile, which defines a route in the considered transportation network together with a preference for the different transportation services provided therein. However, this does not mean that the passengers cannot change their trip profiles (see Sec. 3.4.2 on p. 60).

Definition 9 (TN) *A transportation network is a collection*

$$TN := \left(\mathcal{S}, \left\{ P_{\text{gp}}^{(s)}, T_c^{(s)}, E_{\text{st}}^{(s)} : s \in \mathcal{S} \right\}, \mathcal{G}, \left\{ P_w^{(g)}, T_{\text{tr}}^{(g)}, E_{\text{gr}}^{(g)}, \lambda_p^{(g)} : g \in \mathcal{G} \right\}, \right. \\ \left. \mathcal{I}, \mathcal{V}, \mathcal{W}, \mathcal{Z}, \mathcal{D}, SI, \mathcal{Y}, \mathcal{R}, \left\{ \lambda_c^{(b)} \geq 0 : b \in \mathcal{V} \cup \{(s,p) : (s,p) \in \mathcal{S} \times P_{\text{gp}}^{(s)}\} \right\} \right)$$

of

- a finite non-empty set $\mathcal{S} := \{1, 2, \dots, n_1\}$ of $n_1 \in \mathbb{N}_{>0}$ different stations, where every $s \in \mathcal{S}$ can be decomposed into
 - a finite non-empty set $P_{\text{gp}}^{(s)} := \{1, 2, \dots, n_2^{(s)}\}$ of $n_2^{(s)} \in \mathbb{N}_{>0}$ different discrete gathering points for all passengers in s ,
 - a finite non-empty set $T_c^{(s)}$ of corridors that enable passengers to join s from TN 's exterior, transfer between gathering points in s , board stopped vehicles from s , and alight from stopped vehicles to s , and
 - a finite non-empty set

$$E_{\text{st}}^{(s)} \subseteq \left(P_{\text{gp}}^{(s)} \times T_c^{(s)} \right) \cup \left(T_c^{(s)} \times P_{\text{gp}}^{(s)} \right)$$

of links defining the connection between all gathering points and corridors in s ,

- a finite non-empty set \mathcal{G} of transportation grids, where every $g \in \mathcal{G}$ can be decomposed into
 - a finite non-empty set $P_w^{(g)}$ of waypoints, where
 - * all $p \in P_w^{(g)}$ regarded as a whole accommodate all vehicles operated in g , and
 - * every $p \in P_w^{(g)}$ is either empty or occupied by one vehicle at the same time,
 - a finite non-empty set $T_{\text{tr}}^{(g)}$ of tracks that enable discrete vehicle movements between the waypoints in g ,

– a finite non-empty set

$$E_{\text{gr}}^{(g)} \subseteq (P_w^{(g)} \times T_{\text{tr}}^{(g)}) \cup (T_{\text{tr}}^{(g)} \times P_w^{(g)})$$

of links defining the connection between all waypoints and tracks in g , and

– a unique priority $\lambda_p^{(g)}(t)$, with $\lambda_p^{(g)} : T_{\text{tr}}^{(g)} \rightarrow \{1, 2, \dots, |T_{\text{tr}}^{(g)}|\}$, assigned to every track $t \in T_{\text{tr}}^{(g)}$ for the deterministic resolution of all possible conflicting vehicle movements, where

$$* \lambda_p^{(g)}(t_1) = \lambda_p^{(g)}(t_2) \text{ for } t_1, t_2 \in T_{\text{tr}}^{(g)} \text{ implies } t_1 = t_2, \text{ and}$$

• an interface $\mathcal{I} := \mathcal{I}_a \cup \mathcal{I}_b$ between all $s \in \mathcal{S}$ and all $g \in \mathcal{G}$, with

$$\mathcal{I}_a \in \left\{ K \subseteq \left\{ ((g, p), (s, t)) \in (\mathcal{G} \times P_w^{(g)}) \times (\mathcal{S} \times T_{\text{ic}}^{(s)}) \right\} : \right. \\ \left. \begin{aligned} & ((g, p), (s, t)) \in K \wedge ((g, p'), (s, t)) \in K \text{ implies } p = p', \text{ and} \\ & ((g, p), (s, t)) \in K \wedge ((g, p), (s, t')) \in K \text{ implies } t = t' \end{aligned} \right\},$$

and

$$\mathcal{I}_b \in \left\{ K \subseteq \left\{ ((s, t), (g, p)) \in (\mathcal{S} \times T_{\text{oc}}^{(s)}) \times (\mathcal{G} \times P_w^{(g)}) \right\} : \right. \\ \left. \begin{aligned} & ((s, t), (g, p)) \in K \wedge ((s, t), (g, p')) \in K \text{ implies } p = p', \text{ and} \\ & ((s, t), (g, p)) \in K \wedge ((s, t'), (g, p)) \in K \text{ implies } t = t' \end{aligned} \right\},$$

which constitute a well-divisible infrastructure for all passenger flows and vehicle movements,

- a finite non-empty set \mathcal{W} of vehicle missions,
- a finite non-empty set \mathcal{Z} of vehicle runs,
- a finite set \mathcal{D} of scheduled vehicle dispatches which constitutes a consistent dispatch plan for TN , and
- a signalling system SI

which define the operation of a finite non-empty set $\mathcal{V} := \{1, 2, \dots, n_3\}$ of $n_3 \in \mathbb{N}_{>0}$ different vehicles in all $g \in \mathcal{G}$,

- a finite non-empty set $\mathcal{Y} := \{1, 2, \dots, n_4\}$ of $n_4 \in \mathbb{N}_{>0}$ feasible trip profiles, and
- a static re-routing \mathcal{R} among some trip profiles

which define the proper routing of all passengers in the above infrastructure, and

- a capacity limit for the maximum number $\lambda_c^{(s,p)}$ of passengers at every gathering point $p \in P_{gp}^{(s)}$ in every station s , and
- a capacity limit for the maximum number $\lambda_c^{(v)}$ of passengers on-board every vehicle $v \in \mathcal{V}$.

Passenger Numbers, Modes, and Clocks

In the previous section we have introduced a collection TN of mathematical objects, which shall capture the infrastructure, the vehicle operation, and the passenger routing therein of some considered transportation network when regarded as a whole. Now in this section we will look at the state of such a transportation network; or more specifically the way we can express it in terms of modes, (vectorial) passenger numbers, and clocks associated with TN .

Passenger Numbers. We do not want to track individual passengers but we are rather interested in lumped quantities. Thus, at any gathering point in any station and on-board any vehicle, we group all passengers who travel according to the same trip profile $y \in \mathcal{Y}$ to one non-negative real number together. This grouping of passengers implies that at any time a real-valued vector with as many components as we consider different trip profiles in TN completely defines the passenger numbers at any gathering point in any station and on-board any vehicle; cf. Fig. 3.1.2 below.

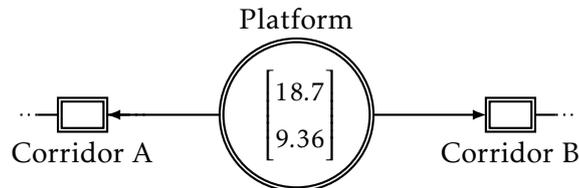


Figure 3.1.2: Fluid passenger numbers at some platform in some station: all passengers at the considered platform travel according to two different trip profiles; 18.7 passengers travel according to trip profile 1, and 9.36 passengers travel according to trip profile 2

Modes. We consider all vehicles $v \in \mathcal{V}$ operated in TN as indivisible agents with internal structures and states of their own that *reside* in the different waypoints of TN 's transportation grids; where every waypoint in every transportation grid is either empty or occupied by at maximum one vehicle at the same time. Then note that the set \mathcal{Q} , which shall comprise all possible realization of the vehicles' different discrete positions in the transportation grids, their discrete driving conditions (either parked, stopped, or driving), and their discrete operational states (in form of a pair of a run and a mission), is finite given the finite time horizon of

TN 's dispatch plan. As such every configuration $q \in \mathcal{Q}$ of TN 's combined vehicle operational state uniquely defines which passenger flow between some platform of a station and some vehicle stopped in front is possible; and because it is doing so, we refer to it as the mode of TN .

Definition 10 (Mode of TN) Every mode $q \in \mathcal{Q}$ of TN , with

$$q := (Q_p^{(q)}, Q_s^{(q)}, Q_d^{(q)}, \mathcal{V}_f^{(q)}),$$

defines a particular discrete state in the vehicle operation of the considered transportation network, which tells us for every transportation grid $g \in \mathcal{G}$,

- which vehicle is parked at which waypoint,
- which vehicle is stopped at which waypoint, together with the run this vehicle processes and the mission therein it executes,
- which vehicle moves towards which waypoint, together with the run this vehicle processes and the mission therein it executes, and
- the subset of all vehicles that are stopped at some waypoint and whose operation is frozen for regulation purposes.

A vehicle $v \in \mathcal{V}^{(g)}$

- is parked at some waypoint $p \in P_w^{(g)}$ iff $(g, p, v) \in Q_p^{(q)}$, with

$$Q_p^{(q)} \subseteq \{(g, p, v) : (g, p, v) \in \mathcal{G} \times P_w^{(g)} \times \mathcal{V}^{(g)}\},$$

- is stopped at some waypoint $p \in P_w^{(g)}$ and executes some mission $w \in \mathcal{W}^{(g)}$ according to some run $z \in \mathcal{Z}^{(g)}$ iff $(g, p, v, z, w) \in Q_s^{(q)}$, with

$$Q_s^{(q)} \subseteq \{(g, p, v, z, w) : (g, p, v, z, w) \in \mathcal{G} \times P_w^{(g)} \times \mathcal{V}^{(g)} \times \mathcal{Z}^{(g)} \times \mathcal{W}^{(g)}\},$$

or

- moves towards some waypoint $p \in P_w^{(g)}$ and executes some mission $w \in \mathcal{W}^{(g)}$ according to some run $z \in \mathcal{Z}^{(g)}$ iff $(g, p, v, z, w) \in Q_d^{(q)}$, with

$$Q_d^{(q)} \subseteq \{(g, p, v, z, w) : (g, p, v, z, w) \in \mathcal{G} \times P_w^{(g)} \times \mathcal{V}^{(g)} \times \mathcal{Z}^{(g)} \times \mathcal{W}^{(g)}\},$$

and where the operation of some vehicle $v \in \mathcal{V}^{(g)}$ is frozen at some waypoint $p \in P_w^{(g)}$ for regulation purposes iff $v \in \mathcal{V}_f^{(q)}$, with

$$\mathcal{V}_f^{(q)} \subseteq \{v \in \mathcal{V} : \exists (g, p, z, w) \in \mathcal{G} \times P_w^{(g)} \times \mathcal{Z}^{(g)} \times \mathcal{W}^{(g)} \text{ s.t. } (g, p, v, z, w) \in Q_s^{(q)}\}.$$

According to Def. 10 above, a vehicle could be parked and move towards some waypoint at the same time in some mode $q \in \mathcal{Q}$. However, this does not make any sense; just as little as the possibility that neither some vehicle $v \in \mathcal{V}$ is parked at some waypoint, nor stopped at some waypoint, nor driving towards some waypoint. That is why, we demand from any *admissible* mode $q \in \mathcal{Q}$, that $\mathcal{V} = \mathcal{V}_p^{(q)} \cup \mathcal{V}_s^{(q)} \cup \mathcal{V}_d^{(q)}$ is a disjoint union; where $\mathcal{V}_p^{(q)}$, $\mathcal{V}_s^{(q)}$, and $\mathcal{V}_d^{(q)}$ are defined in Tab. 3.1 below.

Table 3.1: Frequently used sets for the referencing of all vehicles in any mode $q \in \mathcal{Q}$

Set	Meaning
$\mathcal{V}_p^{(q)} := \left\{ v \in \mathcal{V} : \exists (g, p) \in \mathcal{G} \times P_{gp}^{(g)} \right.$ $\left. \text{s.t. } (g, p, v) \in Q_p^{(q)} \right\}$	All vehicles parked at some waypoint
$\mathcal{V}_s^{(q)} := \left\{ v \in \mathcal{V} : \exists (g, p, z, w) \in \mathcal{G} \times P_{gp}^{(g)} \times \mathcal{Z}^{(g)} \times \mathcal{W}^{(g)} \right.$ $\left. \text{s.t. } (g, p, v, z, w) \in Q_s^{(q)} \right\}$	All vehicles stopped at some waypoint
$\mathcal{V}_d^{(q)} := \left\{ v \in \mathcal{V} : \exists (g, p, z, w) \in \mathcal{G} \times P_{gp}^{(g)} \times \mathcal{Z}^{(g)} \times \mathcal{W}^{(g)} \right.$ $\left. \text{s.t. } (g, p, v, z, w) \in Q_d^{(q)} \right\}$	All vehicles driving towards some waypoint

This restriction of \mathcal{Q} to admissible modes allows us to introduce some more useful shorthands that are listed in Tab. 3.2 below.

Table 3.2: Frequently used shorthands for the positions, driving conditions, runs, and missions of vehicle $v \in \mathcal{V}$ in mode $q \in \mathcal{Q}$

Shorthand	Meaning
$\lambda_p^{(q,v)}$	Position of some vehicle $v \in \mathcal{V}_p^{(q)} \cup \mathcal{V}_s^{(q)}$; waypoint towards which some vehicle $v \in \mathcal{V}_d^{(q)}$ moves
$\lambda_r^{(q,v)}$	Run of some vehicle $v \in \mathcal{V}_s^{(q)} \cup \mathcal{V}_d^{(q)}$
$\lambda_m^{(q,v)}$	Mission of some vehicle $v \in \mathcal{V}_s^{(q)} \cup \mathcal{V}_d^{(q)}$

Clocks. Every mode $q \in \mathcal{Q}$ defines a particular configuration for all vehicles' discrete positions, discrete driving conditions, and discrete operational states in TN . It does not capture TN 's complete vehicle operational state, though. More specifically, q does not define the elapsed dwell times of all vehicles which are stopped at some waypoint, and the remaining driving times of all vehicles which are on their

way towards some waypoint. Neither does it define the time that has elapsed since the operation of some vehicle has been frozen at some waypoint, nor the elapsed parking time of some vehicle at some waypoint. That is why, we have to capture this complementary timing information by some other mathematical objects that we call clocks. We achieve so by assigning to every vehicle $v \in \mathcal{V}$ one clock, which is a real-valued variable whose value (called valuation) gives one of the above pieces of information depending on v 's driving condition and operational state in q .

Passenger Flow- and Vehicle Movement Dynamics

We map the time evolution of TN 's hybrid state, i.e. its (discrete) mode, the (continuous) vectorial passenger numbers at all gathering point in all stations and on-board all vehicles, and the (continuous) valuations of all clocks associated with the operation of all vehicles to T -SHA; our continuous-time stochastic hybrid (state) automaton model for multimodal transportation networks.

Definition 11 (T-SHA) *A continuous-time stochastic hybrid automaton (model) for some transportation network TN from Def. 9 on p. 29 is a collection*

$$T\text{-SHA} := \left(\mathcal{Q}, \{c^{(v)} : v \in \mathcal{V}\}, \{\mathcal{M}^{(s,p)} : s \in \mathcal{S}, p \in P_{\text{gp}}^{(s)}\}, \{\mathcal{U}^{(v)} : v \in \mathcal{V}\}, \right. \\ \left. \{\alpha^{(q)}, \beta^{(q)} : q \in \mathcal{Q}\}, \gamma_c, \gamma_{\text{gp}}, \gamma_v, \mathcal{A}, \Sigma, q_0, \varphi_0, \text{pdf}_0 \right)$$

of

- a finite non-empty set \mathcal{Q} of modes according to Def. 10 on p. 32 which are the discrete states of T -SHA, where
 - $q(\tau)$, with $q : \mathbb{R}_{\geq 0} \rightarrow \mathcal{Q}$ for $\tau \geq 0$, gives T -SHA's mode at time τ ,
- a clock $c^{(v)}$ associated with the operation of every vehicle $v \in \mathcal{V}$ in TN , where
 - $\text{val}(c^{(v)}, \tau)$, with $\text{val} : v \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ for $\tau \in \mathbb{R}_{\geq 0}$, denotes the valuation of $c^{(v)}$ at (simulation) time τ ,
- a local time invariant state space

$$\mathcal{M}^{(s,p)} := \left\{ k \in (\mathbb{R}_{\geq 0})^{|\mathcal{Y}|} : \sum_{y' \in \mathcal{Y}} k[y'] \leq \lambda_c^{(s,p)} \right\},$$

for the passenger number at every gathering point $p \in P_{\text{gp}}^{(s)}$ in every station $s \in \mathcal{S}$, where

- $M^{(s,p)}(\tau)[y]$, with $M^{(s,p)} : \mathbb{R}_{\geq 0} \rightarrow \mathcal{M}^{(s,p)}$ for $\tau \geq 0$ and $y \in \mathcal{Y}$, gives the number of passengers at p in s at time τ who travel according to trip profile y ,

- a local time invariant state space

$$\mathcal{U}^{(v)} := \left\{ k \in (\mathbb{R}_{\geq 0})^{|\mathcal{Y}|} : \sum_{y' \in \mathcal{Y}} k[y'] \leq \lambda_c^{(v)} \right\}.$$

for the passenger number on-board every vehicle $v \in \mathcal{V}$, where

- $U^{(v)}(\tau)[y]$, with $U^{(v)} : \mathbb{R}_{\geq 0} \rightarrow \mathcal{U}^{(v)}$ for $\tau \geq 0$ and $y \in \mathcal{Y}$, gives the number of passengers on-board v and time τ who travel according to trip profile y ,

- two vector fields $\alpha^{(q)}$ and $\beta^{(q)}$, with $\alpha^{(q)}, \beta^{(q)} : \mathcal{X} \rightarrow \mathcal{X}$ and

$$\begin{aligned} \mathcal{X} := \mathcal{M}^{(1,1)} \times \mathcal{M}^{(1,2)} \times \dots \times \mathcal{M}^{(1,|P_{\text{gp}}^{(1)}|)} \times \mathcal{M}^{(2,1)} \times \dots \times \mathcal{M}^{(|S|,|P_{\text{gp}}^{(|S|)}|)} \times \\ \mathcal{U}^{(1)} \times \mathcal{U}^{(2)} \times \dots \times \mathcal{U}^{(|\mathcal{V}|)}, \end{aligned}$$

for $q \in \mathcal{Q}$ which define the time-homogeneous drift and diffusion of all passenger numbers stacked in

$$\begin{aligned} X(\tau) := \left[M^{(1,1)}(\tau); \dots; M^{(1,|P_{\text{gp}}^{(1)}|)}(\tau); M^{(2,1)}(\tau); \dots; M^{(|S|,|P_{\text{gp}}^{(|S|)}|)}(\tau); \right. \\ \left. U^{(1)}(\tau); \dots; U^{(|\mathcal{V}|)}(\tau) \right]^t \end{aligned}$$

when T-SHA is in $q \in \mathcal{Q}$,

- a finite non-empty set \mathcal{A} of atomic actions associated with the operation of all vehicles $v \in \mathcal{V}$ in TN such as the departure of a vehicle from a waypoint which when performed by some v changes the position, driving condition, or the operational state of v ,
- a finite non-empty set

$$\Sigma \subseteq \mathcal{Q} \times \left\{ e \in 2^{\mathcal{V} \times \mathcal{A}} : (v, a_1) \in e \wedge (v, a_2) \in e \text{ implies } a_1 = a_2, \text{ for } v \in \mathcal{V} \right\} \times \mathcal{Q}$$

of mode transitions,

- some clock constraints for mode transition $\sigma \in \Sigma$ in form of the boolean expression $\gamma_c(\sigma)$, with

$$\gamma_c : \Sigma \rightarrow \left\{ \bigwedge_{v \in \mathcal{V}} b^{(v)} : b^{(v)} \in \left\{ \text{true}, \text{val}(c^{(v)}, \cdot) \bowtie k_1^{(v)} \right\} \right\},$$

where $k_1^{(v)} \in \mathbb{R}_{\geq 0}$ is some time threshold which is related to the valuation $\text{val}(c^{(v)}, \tau)$ of the clock associated with the operation of some vehicle $v \in \mathcal{V}$ via some binary operator $\bowtie \in \{=, \geq\}$ at any time $\tau \geq 0$,

- some constraints for the passenger numbers at all gathering points $p \in P_{\text{gp}}^{(s)}$ in all stations $s \in \mathcal{S}$ for mode transition $\sigma \in \Sigma$ in form of the boolean expression $\gamma_{\text{gp}}(\sigma)$, with

$$\gamma_{\text{gp}} : \Sigma \rightarrow \left\{ \bigwedge_{(s,p) \in \mathcal{S} \times P_{\text{gp}}^{(s)}} b^{(s,p)} : b^{(s,p)} \in \left\{ \text{true}, m^{(s,p)}(\cdot) \bowtie k_2^{(s,p)} \right\} \right\},$$

where $k_2^{(s,p)} \in \mathbb{R}_{\geq 0}$ is some threshold which is related to the cumulative passenger number

$$m^{(s,p)}(\tau) := \sum_{i \in \mathcal{Y}} M^{(s,p)}(\tau)[i],$$

at some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ via some binary operator $\bowtie \in \{<, \leq, =, \geq, >\}$ at any time $\tau \geq 0$,

- some constraints for the passenger numbers on-board all vehicles for mode transition $\sigma \in \Sigma$ in form of the boolean expression $\gamma_{\text{v}}(\sigma)$, with

$$\gamma_{\text{v}} : \Sigma \rightarrow \left\{ \bigwedge_{v \in \mathcal{V}} b^{(v)} : b^{(v)} \in \left\{ \text{true}, u^{(v)}(\cdot) \bowtie k_3^{(v)} \right\} \right\},$$

where $k_3^{(v)} \in \mathbb{R}_{\geq 0}$ is some threshold which is related to the cumulative passenger number

$$u^{(v)}(\tau) := \sum_{i \in \mathcal{Y}} U^{(v)}(\tau)[i],$$

on-board some vehicle $v \in \mathcal{V}$ via some binary operator $\bowtie \in \{<, \leq, =, \geq, >\}$ at any time $\tau \geq 0$,

- some mode $q_0 \in \mathcal{Q}$ with marginal probability one for the initial discrete state of T-SHA,
- some initial valuation $\varphi_0[v]$, with $\varphi_0 \in (\mathbb{R}_{\geq 0})^{|\mathcal{V}|}$, for the clock associated with the operation of any vehicle $v \in \mathcal{V}$, and
- some joint probability density

$$\text{pdf}_0 : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}, \quad \int_{x \in \mathcal{X}} \text{pdf}_0(x) dx = 1,$$

for all initial passenger numbers.

Passenger Flow-Dynamics in Some Mode. We assign to every mode $q \in \mathcal{Q}$ a system

$$dX(\tau) := \alpha^{(q)}(X(\tau)) d\tau + \beta^{(q)}(X(\tau)) dW(\tau), \quad (3.1)$$

of Itô-SDEs, where W is a multidimensional Wiener process, i.e. a continuous-time stochastic process with independent and stationary increments $W(\tau) - W(\tau')$ whose law is Gaussian with parameter $\tau - \tau' \geq 0$ (see [Szabados 2010] for a rigorous analysis of this stochastic process). This system relates the time evolution of every passenger number stacked in $X(\tau)$ to all passenger flows into and out of the corresponding gathering point or vehicle when T -SHA is in q . All this according to the informal law

$$\Delta(\text{Passenger load}) = \sum \text{Inflows} - \sum \text{Outflows}.$$

Given the fact that some vehicle might be stopped in q for the purpose of boarding & alighting (and thus enables some passenger flow to or from some platform in some station) but not in some other mode $q' \in \mathcal{Q}$, then implies that the set up of this balance equation in q' might be different from the corresponding set up in q . Moreover, given the constraint that both $\alpha^{(q)}$ and $\beta^{(q)}$ must not be explicit functions of the time in q , implies that the passenger flow dynamics associated with T -SHA does not change as long as T -SHA stays in q .

Neither does the above system of Itô-SDEs taken alone enforce non-negativity and capacity constraints for any passenger number, nor does it define the time evolution of $\text{pdf}(X(\tau))$, i.e. the joint probability density of X at time $\tau \geq 0$, starting from $\text{pdf}(X(0)) := \text{pdf}_0$ at time $\tau = 0$. This is where the Fokker-Planck equation (FPE) comes into play [Gardiner 1985]. The FPE, or also known as the Kolmogorov forward equation, is a system of linear parabolic partial differential equations, which describes the time evolution of a probability density, whose continuous random variable is subjected to drift and diffusion in form of a system of Itô-SDEs. With the drift term α and diffusion term β from Eqn. 3.27, it reads

$$\begin{aligned} \frac{\partial}{\partial \tau} \text{pdf}(X(\tau)) = & - \sum_{i=1}^{k_d} \frac{\partial}{\partial X_i} \left(\alpha^{(q)}(X(\tau))[i] \text{pdf}(X(\tau)) \right) + \\ & \frac{1}{2} \sum_{i=1}^{k_d} \sum_{j=1}^{k_d} \Psi^{(q)}(X(\tau))[i, j] \frac{\partial^2}{\partial X_i \partial X_j} \left(\text{pdf}(X(\tau)) \right), \end{aligned} \quad (3.2)$$

where

$$k_d := |\mathcal{Y}| \left(|\mathcal{V}| + \sum_{s \in \mathcal{S}} |P_{\text{gp}}^{(s)}| \right)$$

denotes the dimension of \mathcal{X} , $\frac{\partial}{\partial X_i}$ denotes the partial derivative w.r.t. some dimension $i \in \{1, 2, \dots, k_d\}$, and

$$\Psi^{(q)}(X(\tau)) := \beta^{(q)}(X(\tau)) \left(\beta^{(q)}(X(\tau)) \right)^t.$$

Note that this FPE defines a continuity equation. By introducing the probability

flux

$$f(X(\tau)) := \alpha^{(q)}(X(\tau)) \text{pdf}(X(\tau)) - \frac{1}{2} \Psi^{(q)} \begin{bmatrix} \frac{\partial}{\partial X_1} \text{pdf}(X(\tau)) \\ \frac{\partial}{\partial X_2} \text{pdf}(X(\tau)) \\ \vdots \\ \frac{\partial}{\partial X_{k_d}} \text{pdf}(X(\tau)) \end{bmatrix}, \quad (3.3)$$

we can rewrite Eqn. 3.2 according to

$$\frac{\partial}{\partial \tau} \text{pdf}(X(\tau)) + \text{div}(f(X(\tau))) = 0, \quad (3.4)$$

where $\text{div}(\cdot)$ denotes the divergence operator. Now since Eqn. 3.2 above can be rewritten according to Eqn. 3.4, we can derive reflecting boundary conditions for it which confine all passenger numbers to their admissible ranges; see [Gardiner 1985, chapter 5] for more information. For this purpose, note that the cumulative probability of $X(\tau)$ to adopt a value in \mathcal{X} at any time $\tau \geq 0$ must be one, which in mathematical terms is identical to saying that

$$\int_{\mathcal{X}} \text{pdf}(X(\tau)) dx = 1, \forall \tau \geq 0, \quad (3.5)$$

and

$$\frac{\partial}{\partial \tau} \int_{\mathcal{X}} \text{pdf}(X(\tau)) dx = 0, \forall \tau \geq 0. \quad (3.6)$$

Next note that we can change the order of integration and derivation in (3.6) above, which allows us to rewrite it in form of

$$\int_{\mathcal{X}} \text{div}(f(X(\tau))) dx = 0, \forall \tau \geq 0 \quad (3.7)$$

if we make use of the identity

$$\int_{\mathcal{X}} \frac{\partial}{\partial \tau} (\text{pdf}(X(\tau))) dx = - \int_{\mathcal{X}} \text{div}(f(X(\tau))) dx \quad (3.8)$$

obtained from Eqn. 3.4. The divergence theorem then states that

$$\int_{\mathcal{X}} \text{div}(f(X(\tau))) dx = \oint_{\partial \mathcal{X}} f(X(\tau)) \cdot n(X(\tau)) dx \quad (3.9)$$

which implies

$$\oint_{\partial \mathcal{X}} f(X(\tau)) \cdot n(X(\tau)) dx = 0, \forall \tau \geq 0. \quad (3.10)$$

Therein, $\partial\mathcal{X}$ denotes the surface of \mathcal{X} , “ \cdot ” the dot product operator, and $n(X(\tau))$ denotes a unit vector which is orthogonal to $\partial\mathcal{X}$ at $X(\tau) \in \partial\mathcal{X}$ and points outwards of \mathcal{X} . This latter equation defines a so-called reflecting boundary condition for the time evolution of X in any mode $q \in \mathcal{Q}$. It states that the probability density of X is conserved in \mathcal{X} at any time $\tau \geq 0$ in any q iff the probability net flux across $\partial\mathcal{X}$ is zero.

Eqn. 3.2 and 3.10 form an initial value problem which uniquely defines $\text{pdf}(X(\tau'))$ at any future time $\tau' > \tau$ given $\text{pdf}(X(\tau))$ at some time $\tau \geq 0$ as long long as $T\text{-SHA}$ stays in q .

Clock Valuations in Some Mode. In any mode $q \in \mathcal{Q}$, the valuations of all clocks increase linearly with the (simulation) time. Because they do so, the valuation of any clock $c^{(v)}$, which is associated with the operation of some vehicle $v \in \mathcal{V}(g)$ that in turn is operated in some transportation grid $g \in \mathcal{G}$, gives

- the elapsed parking time of v at some waypoint $p \in P_w^{(g)}$ in g iff $v \in \mathcal{V}_p^{(q)}$ and $\lambda_p^{(q,v)} = p$,
- the elapsed dwell time of v at some waypoint $p \in P_w^{(g)}$ in g iff $v \in \mathcal{V}_s^{(q)} \setminus \mathcal{V}_f^{(q)}$ and $\lambda_p^{(q,v)} = p$,
- the elapsed driving time of v towards some waypoint $p \in P_w^{(g)}$ in g iff $v \in \mathcal{V}_d^{(q)}$ and $\lambda_p^{(q,v)} = p$, or
- the time that has elapsed since the operation of v has been frozen at some waypoint $p \in P_w^{(g)}$ in g iff $v \in \mathcal{V}_f^{(q)}$ and $\lambda_p^{(q,v)} = p$.

At the initial simulation time $\tau = 0$, the valuation of $c^{(v)}$ is $\varphi_0[v]$ for $v \in \mathcal{V}$ with marginal probability one.

Mode Transitions. Mode transitions occur, whenever some vehicle changes its position, driving condition, or operational state; which latter performance of some atomic action is either conditioned on the valuation of its clock, some passenger numbers, the positions of all other vehicles, or all. For instance, the parking of a vehicle at some stop might be conditioned on the fact that there are no more passengers on-board its cabin. As another example, the departure of a vehicle from some waypoint might be conditioned on the fact that (i) the dwell time of this vehicle at this waypoint exceeds some minimum time threshold, (ii) no more passengers on-board its cabin want to alight from it, and (iii) the waypoint this vehicle is supposed to move towards next is not occupied by some other vehicle.

In this context note that in practice, the performance of every conceivable atomic action on the operational level of some individual vehicle consumes time. For instance, the parking of some stopped vehicle, say v , requires the shutdown of all of its engines and maybe some more routine procedures. Now in $T\text{-SHA}$ the performance of some action itself does not consume any amount of time. We

have to map this time consumption to the constraints imposed on the valuations of some clocks instead. In our above parking example, this means that we have to map the minimum amount of time before all engines are shut down to a constraint associated with the valuation of v 's clock for the transition from q to q' if q captures the situation (= mode) when v is stopped and going to park next, and q' captures the situation when v is finally parked.

In some situations two or more vehicles might perform some actions at the same time. That is why, more generally speaking the occurrence of some mode transition $\sigma \in \Sigma$ at some time $\tau \geq 0$ is the result of some vehicles which perform some - from vehicle to vehicle possibly different - atomic action $a \in \mathcal{A}$ at τ ; cf. Sec. 3.7 on p. 78ff. for more details. For $q, q' \in \mathcal{Q}$, and some

$$K \in \left\{ e \in 2^{\mathcal{V} \times \mathcal{A}} : (v, a_1) \in e \wedge (v, a_2) \in e \text{ implies } a_1 = a_2, \text{ for } v \in \mathcal{V} \right\},$$

$\sigma := (q, K, q')$ occurs at τ iff σ is enabled at τ .

Definition 12 (Enabled Mode Transition) *Some mode transition $\sigma \in \Sigma$, with $\sigma := (q, K, q')$ for $q, q' \in \mathcal{Q}$, and some*

$$K \in \left\{ e \in 2^{\mathcal{V} \times \mathcal{A}} : (v, a_1) \in e \wedge (v, a_2) \in e \text{ implies } a_1 = a_2, \text{ for } v \in \mathcal{V} \right\},$$

is enabled at some time $\tau \geq 0$ iff

- *T-SHA is in q at τ ,*
- *the boolean expression $\gamma_c(\sigma)$ for all clock constraints evaluates to “true” at τ ,*
- *the boolean expression $\gamma_{gp}(\sigma)$ for the passenger number constraints at all gathering points in all stations evaluates to “true” at τ , and*
- *the boolean expression $\gamma_v(\sigma)$ for the passenger number constraints on-board all vehicles evaluates to “true” at τ ,*

where

- *the placeholder “.” in every boolean expression above must be replaced by τ .*

If σ is enabled, then it provokes *T-SHA* to change its mode from q to q' at τ . Moreover, σ resets the clock $c^{(v)}$ associated with the operation of every vehicle $v \in \{v' \in \mathcal{V} : \exists a \in \mathcal{A} \text{ s.t. } (v, a) \in K\}$, i.e. every vehicle which performs some action according to σ . Also note that σ does not cause jumps in any passenger numbers, which is a natural assumption in our setting since neither passengers are created from nowhere, nor do they get lost just because some vehicle e.g. updates its operational state.

Although it will turn out not to be a real modelling constraint, we informally demand here that for any particular realization of *T-SHA*'s hybrid state at maximum one mode transition $\sigma \in \Sigma$ is enabled if at all. We do demand so in order to eliminate any possible source of ambiguity in the specification of our automaton.

For the Interesting Reader: How the T-SHA Model Evolved

Starting Point. *T-SHA* has its origin in a purely discrete-event Petri net-styled multi agent system (MAS). The modelling engine of this MAS is the nets-within-nets approach from [Valk 1998] that we have already introduced in Sec. 2.3.3 on p. 14ff. In this MAS both vehicles and passengers are represented as indivisible net tokens with internal structure and state of their own. These net tokens reside in a common place/transition net that captures the infrastructure of the considered transportation network. The internal state of every passenger token is defined by the discrete marking of a place/transition net that captures a particular trip profile in that it unfolds a path in the above infrastructure together with a preference for the different transportation services provided by the vehicle tokens. The vehicle tokens' internal states in turn are defined by (i) the discrete markings of place/transition nets that capture all transportation services in form of paths and sequences of stops along these paths in the above infrastructure, and (ii) the passenger numbers of their cabins in form of individual passenger tokens.

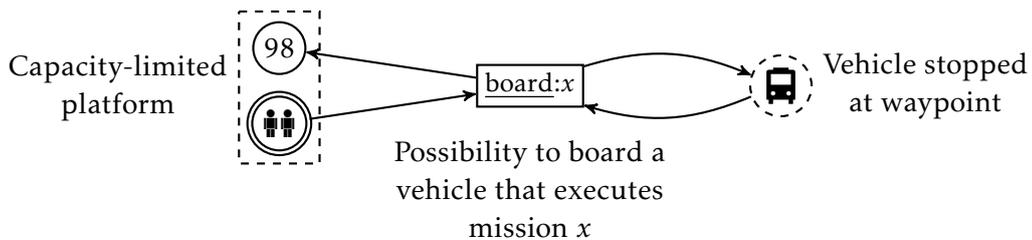


Figure 3.1.3: A purely discrete-event Petri nets-styled multi agent system is at the origin of our SHA model for multimodal transportation networks. Therein, individual vehicle agents provide transportation services to individual passenger agents which follow self-determined routes. All agents are indivisible tokens with internal structure and state of their own

Refer to [Theissing 2016b] for a detailed description of the MAS. Here only note that it incarnates some major modelling blocks and ideas, which we have adopted in the definition and elaboration of our SHA model:

- We divide the infrastructure of the considered transportation network into a finite set of stations, another finite set of transportation grids, and an interface connecting both for the purpose of boarding & alighting.
- Every transportation grid accommodates a particular mode or line in form of some discrete vehicle positions.
- Different transportation grids do not share infrastructure elements.
- Individual vehicle agents provide different transportation services or execute dead-headings which are defined for the different transportation grids in form of so-called vehicle missions.

- All passengers have trip profiles that produce self-determined but not necessarily cost-efficient routes connecting origin/destination pairs in the infrastructure of the considered transportation network. These routes might be sensitive to alternative competing transportation services.
- No difference is made between the modes and lines of a multimodal transportation network in terms of the way all vehicles provide the transportation services to the passengers, and the way all passengers use them. In this context, all vehicles are regarded - independently of their mode - as capacity-limited people movers that execute predefined missions.

Fluidification. The explosion of the above MAS's purely discrete state space makes its exhaustive formal analysis intractable even for very few vehicle- and passenger agents; which however does not prohibit its implementation and simulation with e.g. the free of charge tool Renew [Kummer 2004]. Now compared to the many many passenger agents, we probably have to model only some vehicle agents since we do not want to explicitly capture in our model other vehicles (e.g. vehicles that pertain to private car use) than those that provide the transportation services to the passengers. Moreover, the fact that passengers are individuals does not prevent us from grouping them together into a finite set of aggregated passenger numbers in our model. Thus, every passenger travels according to one particular trip profile out of a finite set of different trip profiles. At every point in the network, all passengers of the same trip profile are indistinguishable in that they do the same thing; e.g. all of them board a vehicle, all of them transfer to another platform, and so on. This consideration lead us to the development of the deterministic hybrid automaton (DHA) model from [Theissing 2015]. In this DHA model, a finite set of discrete moving vehicle agents execute missions, and in doing so provide transportation services to the passengers in the considered mulitmodal transportation network. Every mode of the DHA corresponds to a particular configuration of all vehicles' discrete positions, discrete driving conditions, and discrete operational states. The DHA's continuous state defines the vehicles' elapsed driving- and dwell times, as well as the vectorial passenger numbers at all discrete locations such as platforms considered in the different stations, and on-board all vehicles; where every real-valued component of some passenger vector corresponds to the number of passengers of some particular trip profile (cf. Fig. 3.1.2 on p. 31). A set of coupled ordinary differential equations (ODEs) assigned to every pair of a station and a mode (decoupled from the systems of ODEs set up for all other stations), relates the continuous time evolution of all fluid passenger numbers within this station and the vehicles that are docked to it for the purpose of boarding & alighting, to the passenger flows in between as long as the DHA does not change its mode. Some mode transitions such as the departure of a vehicle from a stop are coupled to the continuous passenger numbers, whereas others are triggered by deterministic time thresholds only. This model determinism produces sequences of well-defined consecutive mode transitions which de-

pend on the automaton's initial state; cf. Fig. 3.1.4 below. The major difficulty in the unfolding of this sequence is the precise computation of all isolated mode transition times; some numerical issue which is well-known in the literature on hybrid automata.

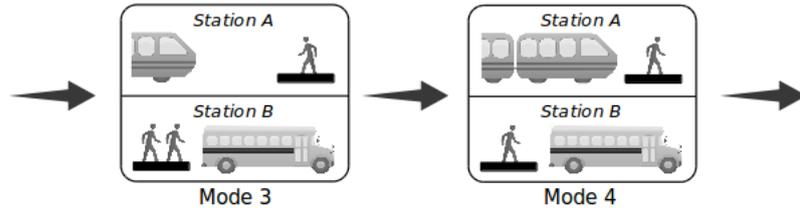


Figure 3.1.4: Compare with Fig. 3.1.1 on p. 27. In big contrast to *T-SHA*, an unfolding of the hybrid state of the DHA from [Theissing 2015] for some particular initial state, produces an unambiguous sequence of well-defined mode transitions that occur at isolated points in time

Uncertainties. In general, we cannot parametrize the above DHA model since we do not know (i) at which time passengers arrive at the station from the transportation network's exterior, (ii) their initial trip profiles, and (iii) all initial passenger vectors. This inconvenient fact finally lead us to the development of *T-SHA*, which we introduced in [Theissing 2016a]. Compared to our above DHA model, we replaced all systems of ODEs by systems of (Itô-) stochastic differential equations (SDEs), so as to be able to (i) start our analyses with uncertain initial passenger numbers, and (ii) include uncertain passenger arrival flows into the model's continuous time dynamics. The mechanism of triggering mode transitions via thresholds remains the same; however, the passenger number-trajectories' hitting times are not deterministic, isolated points in time any more, but rather random variables with a continuous range of values.

Specification of the Infrastructure

Stations

Every station $s \in \mathcal{S}$ of TN from Def. 9 on p. 29 defines the connection between all gathering points $p \in P_{\text{gp}}^{(s)}$ and all corridors $t \in T_c^{(s)}$ in form of a place/transition net, where the set $P_{\text{gp}}^{(s)}$ contains the set of all places, and the set $T_c^{(s)}$ contains the set of all transitions. Every gathering point $p \in P_{\text{gp}}^{(s)}$ corresponds to a well-defined location in some real-world station such as a platform or an entrance area. It can accommodate at maximum $\lambda_c^{(s,p)}$ passengers at the same time. On the other hand, every corridor $t \in T_c^{(s)}$ implements the possibility of a passenger flow into or out of some gathering point.

Definition 13 Every station $s \in \mathcal{S}$ is a weakly connected place/transition net

$$s := (P_{\text{gp}}^{(s)}, T_c^{(s)}, E_{\text{st}}^{(s)})$$

according to Def. 1 on p. 1 and Def. 6 on p. 16, where

- the finite set $P_{\text{gp}}^{(s)}$ of gathering points in s contains all places,
- the finite set $T_c^{(s)}$ of corridors in s contains all transitions,
- $|\bullet t| \leq 1$ and $|t \bullet| \leq 1$ for $t \in T_c^{(s)}$, and
- $\bullet(t_1) = \bullet(t_2)$ and $(t_1) \bullet = (t_2) \bullet$ implies $t_1 = t_2$ for any pair (t_1, t_2) of two corridors $t_1, t_2 \in T_c^{(s)}$.

It defines all possible passenger flows into and out of every gathering point $p \in P_{\text{gp}}^{(s)}$ in that passengers can access p via some corridor $t \in T_c^{(s)}$ iff $t \in \bullet p$. Correspondingly, passengers can leave p via t iff $t \in p \bullet$.

High-Level Decomposition of All Corridors. Although we have not looked at the transportation grids and the interface between the stations and the transportation grids yet, we can still distinguish three types of corridors in s on a first very high level of abstraction; namely inflow-, transfer-, and outflow corridors. More specifically, all inflow corridors in s implement the possibility of a passenger flow from the exterior of s to some gathering point within s .

Definition 14 (Inflow Corridor) A corridor $t \in T_c^{(s)}$ from some station $s \in \mathcal{S}$ is an inflow corridor iff $t \in T_{\text{ic}}^{(s)}$, with

$$T_{\text{ic}}^{(s)} := \{t \in T_c^{(s)} : \bullet t = \emptyset \wedge t \bullet \subseteq P_{\text{gp}}^{(s)}\}.$$

All transfer corridors in s implement the possibility of a (directed) passenger flow between two different gathering points within s .

Definition 15 (Transfer Corridor) A corridor $t \in T_c^{(s)}$ from some station $s \in \mathcal{S}$ is a transfer corridor iff $t \in T_{\text{tc}}^{(s)}$, with

$$T_{\text{tc}}^{(s)} := \{t \in T_c^{(s)} : \bullet t \subseteq P_{\text{gp}}^{(s)} \wedge t \bullet \subseteq P_{\text{gp}}^{(s)}, \wedge \bullet t \cap t \bullet = \emptyset\}.$$

Finally, all outflow corridors in s implement the possibility of a passenger flow leaving s from some gathering point in s to its exterior.

Definition 16 (Outflow Corridor) A corridor $t \in T_c^{(s)}$ from some station $s \in \mathcal{S}$ is an outflow corridor iff $t \in T_{\text{oc}}^{(s)}$, with

$$T_{\text{oc}}^{(s)} := \{t \in T_c^{(s)} : \bullet t \subseteq P_{\text{gp}}^{(s)} \wedge t \bullet = \emptyset\}.$$

Graphical Representation. In any graphical representation of some station $s \in \mathcal{S}$ of TN , we draw all gathering points as double circles, and all corridors as double boxes. We might also write the capacity limit of the gathering points in brackets next to them; cf. Fig. 3.2.1 below.

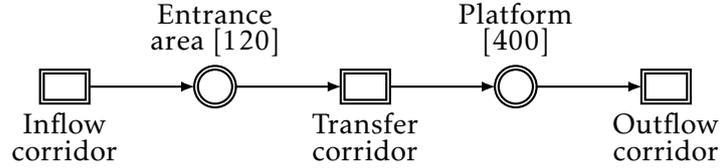


Figure 3.2.1: A sample station that connects three corridors with two gathering points; an entrance area which can accommodate up to 120 passengers at the same time, and a platform which can accommodate up to 400 passengers at the same time.

Transportation Grids

Every transportation grid $g \in \mathcal{G}$ of TN from Def. 9 on p. 29 defines the connection between all waypoints $P_w^{(g)}$ and all tracks $t \in T_{tr}^{(g)}$ in form of a place/transition net, where the set $P_w^{(g)}$ contains the set of all places, and the set $T_{tr}^{(g)}$ contains the set of all transitions. More specifically, every waypoint $p \in P_w^{(g)}$ refers to a particular discrete position along some transportation line/s such a stop in front of a station or a parking in a depot. It can accommodate at maximum one single vehicle at the same time. On the other hand, every track $t \in T_{tr}^{(g)}$ implements the possibility of a vehicle movement between some pair of two waypoints.

Definition 17 (Transportation Grid) Every transportation grid $g \in \mathcal{G}$ is a weakly-connected place/transition net

$$g := (P_w^{(g)}, T_{tr}^{(g)}, E_{gr}^{(g)})$$

according to Def. 1 on p. 15 and Def. 6 on p. 16, where

- the finite set $P_w^{(g)}$ of waypoints in g contains all places,
- the finite set $T_{tr}^{(g)}$ of tracks in g contains all transitions, and
- $|\bullet t| = 1$ and $|t\bullet| = 1$ for $t \in T_{tr}^{(g)}$.

It defines all possible vehicle movements along some transportation line/s. Some track $t \in T_{tr}^{(g)}$, with $t := (p_1, p_2)$, enables all vehicles $v \in \mathcal{V}^{(g)}$ operated in g to move from the waypoint p_1 to the waypoint p_2 iff $p_1 \in \bullet t$ and $p_2 \in t\bullet$.

Graphical Representation. In any graphical representation of some transportation grid, we draw all waypoints as simple circles and all tracks as simple boxes; so as to avoid any confusion with the graphical representation of some station; where we use double circles for all places and double boxes for all transitions.

Fig. 3.2.2 below depicts a sample transportation grid that might accommodate a circular metro line.

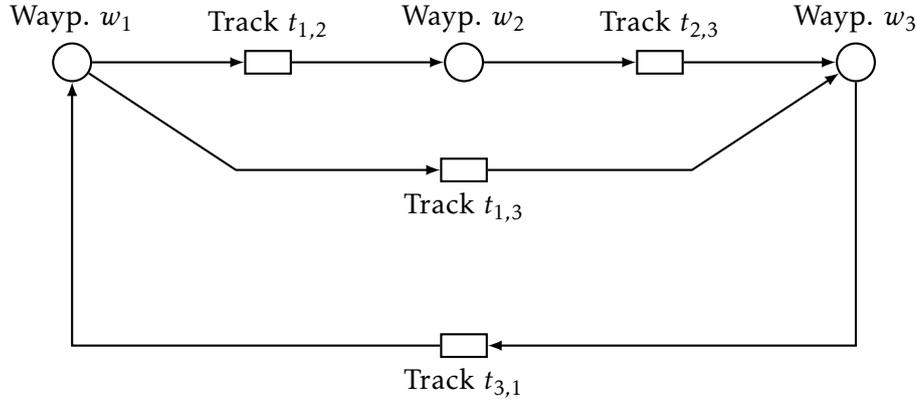


Figure 3.2.2: A sample transportation grid that connects three waypoints and four tracks, with the possibility of a vehicle conflict at w_3 . A vehicle can access w_3 from w_2 via $t_{2,3}$, or from w_1 via $t_{1,3}$. A conflict occurs iff one vehicle from w_2 and another vehicle from w_1 try to access w_3 at the same time.

Conflicting Vehicle Movements. Since by convention every waypoint can accommodate at maximum one vehicle at the same time, a conflict between two or more vehicles might occur during operation in some transportation grid iff these vehicles want to access the same waypoint at the time; e.g. the waypoint w_3 in the transportation grid from Fig. 3.2.2 above. This latter possibility of a vehicle conflict cannot occur if the considered grid is free of structural conflicts.

Definition 18 (Conflict-free Transportation Grid) A transportation grid $g \in \mathcal{G}$ is free of structural conflicts iff $\nexists p_1, p_2 \in P_w^{(g)}$ s.t. $p_1 \neq p_2$ and $(p_1)^\bullet = (p_2)^\bullet$.

Otherwise, we will use the time invariant priorities assigned to all tracks to resolve all possible vehicle conflicts in a deterministic manner.

Interface Between All Stations and Transportation Grids

Correct Interpretation of Interface. TN 's interface \mathcal{I} from Def. 9 on p. 29 defines the connection between all stations $s \in \mathcal{S}$ and all transportation grids $g \in \mathcal{G}$ for the purpose of boarding & alighting in form of the disjoint union of two sets, \mathcal{I}_a and \mathcal{I}_b . More specifically, a passenger can board some vehicle $v \in \mathcal{V}^{(g)}$ which is stopped at some waypoint $p_1 \in P_w^{(g)}$ in some transportation grid $g \in \mathcal{G}$ from a gathering point $p_2 \in P_{gp}^{(g)}$ in some station $s \in \mathcal{S}$ through some outflow corridor

$t_1 \in T_{oc}^{(s)}$ iff $p_2 \in \bullet(t_1)$ and $((s, t_1), (g, p_1)) \in \mathcal{I}_b$, with

$$\mathcal{I}_b \in \left\{ K \subseteq \left\{ ((s, t), (g, p)) \in (\mathcal{S} \times T_{oc}^{(s)}) \times (\mathcal{G} \times P_w^{(g)}) \right\} : \right. \\ \left. \begin{aligned} &((s, t), (g, p)) \in K \wedge ((s, t), (g, p')) \in K \text{ implies } p = p', \text{ and} \\ &((s, t), (g, p)) \in K \wedge (s, t'), (g, p) \in K \text{ implies } t = t' \end{aligned} \right\}.$$

Accordingly, a passenger on-board v can alight from it to p_2 through some outflow corridor $t_2 \in T_{oc}^{(s)}$ iff $p_2 \in (t_2)^\bullet$ and $((g, p_1), (s, t_2)) \in \mathcal{I}_a$, with

$$\mathcal{I}_a \in \left\{ K \subseteq \left\{ ((g, p), (s, t)) \in (\mathcal{G} \times P_w^{(g)}) \times (\mathcal{S} \times T_{ic}^{(s)}) \right\} : \right. \\ \left. \begin{aligned} &((g, p), (s, t)) \in K \wedge ((g, p'), (s, t)) \in K \text{ implies } p = p', \text{ and} \\ &((g, p), (s, t)) \in K \wedge ((g, p), (s, t')) \in K \text{ implies } t = t' \end{aligned} \right\}.$$

Well-Divisible Infrastructure. Def. 9 on p. 29 for TN states that its interface \mathcal{I} defines a well-divisible infrastructure, which demands that passengers must not access a vehicle stopped at a particular waypoint from two different stations.

Definition 19 (Well-divisible Infrastructure) *TN's infrastructure is well-divisible iff for waypoint $p \in P_w^{(g)}$ in any transportation grid $g \in \mathcal{G}$, the fact that*

$$((s_1, t_1), (g, p)) \in \mathcal{I}_b \text{ or } ((g, p), (s_1, t_1)) \in \mathcal{I}_a,$$

and

$$((s_2, t_2), (g, p)) \in \mathcal{I}_b \text{ or } ((g, p), (s_2, t_2)) \in \mathcal{I}_a,$$

implies $s_1 = s_2$ for corridor $t_1 \in T_{fc}^{(s_1)}$ in any station $s_1 \in \mathcal{S}$, and for corridor $t_2 \in T_{fc}^{(s_2)}$ in any station $s_2 \in \mathcal{S}$.

This important property of TN 's infrastructure ensures that a vehicle can no longer serve as a bridge for passenger flows between two or more stations in any mode $q \in \mathcal{Q}$ of $T\text{-SHA}$; which will allow us to isolate in any $q \in \mathcal{Q}$ the (continuous) passenger-flow dynamics set up for one station from the passenger-flow dynamics in all other stations.

Useful Shorthands. Note that every there is exactly one waypoint in the preset of every track in every transportation grid, and another waypoint in its postset. Moreover, note that there is at maximum one gathering point in the preset of every corridor in every station, and another gathering point in its postset. We can

make use of these two special properties of all stations and of all transportation grids in form of two shorthands if we also notice that TN 's interface \mathcal{I} connects every inflow/outflow corridor in a station to at maximum one waypoint in a transportation grid: The first shorthand, $\star t$, denotes the gathering point $p_1 \in P_{\text{gp}}^{(g)}$ in the station $s \in \mathcal{S}$ iff $t \in T_{\text{tc}}^{(s)} \cup T_{\text{oc}}^{(s)}$ and $\bullet t = \{p_1\}$. Similarly, the second shorthand, t^\star , denotes the gathering point $p_2 \in P_{\text{gp}}^{(s)}$ in the station $s \in \mathcal{S}$ iff $t \in T_{\text{ic}}^{(s)} \cup T_{\text{tc}}^{(s)}$ and $t^\bullet = \{p_2\}$.

Fine-Grain Decomposition of all Corridors. If a passenger can board a vehicle from some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ via some corridor $t \in p^\bullet \cap T_{\text{oc}}^{(s)}$, then we say that t is a boarding corridor.

Definition 20 (Boarding Corridor) A corridor $t \in T_c^{(s)}$ in some station $s \in \mathcal{S}$ is a boarding corridor iff $t \in T_{\text{bc}}^{(s)}$, with

$$T_{\text{bc}}^{(s)} := \left\{ t' \in T_{\text{oc}}^{(s)} : \exists (g, p) \in \mathcal{G} \times P_w^{(g)} \text{ s.t. } ((s, t'), (g, p)) \in \mathcal{I}_b \right\}.$$

Accordingly, we say that t is an alighting corridor iff passengers on-board some vehicle stopped in front of s can alight to p .

Definition 21 (Alighting Corridor) A corridor $t \in T_c^{(s)}$ from some station $s \in \mathcal{S}$ is an alighting corridor iff $t \in T_{\text{ac}}^{(s)}$, with

$$T_{\text{ac}}^{(s)} := \left\{ t' \in T_{\text{ic}}^{(s)} : \exists (g, p) \in \mathcal{G} \times P_w^{(g)} \text{ s.t. } ((g, p), (s, t')) \in \mathcal{I}_a \right\},$$

If some inflow corridor $t \in T_{\text{ic}}^{(s)}$ is not a boarding corridor, then we say that it is an entrance corridor since its only purpose might be to implement the possibility of a passenger flow from TN 's exterior into some gathering point in s .

Definition 22 (Entrance Corridor) A corridor $t \in T_c^{(s)}$ in some station $s \in \mathcal{S}$ is an entrance corridor iff $t \in T_{\text{ec}}^{(s)}$, with $T_{\text{ec}}^{(s)} := T_{\text{ic}}^{(s)} \setminus T_{\text{ac}}^{(s)}$.

Similarly, we say that some outflow corridor $t' \in T_{\text{oc}}^{(s)}$ is an exit corridor iff the purpose of its implementation is to provide a passenger flow out of some gathering point in s into TN 's exterior.

Definition 23 (Exit Corridor) A corridor $t \in T_c^{(s)}$ from some station $s \in \mathcal{S}$ is an exit corridor iff $t \in T_{\text{xc}}^{(s)}$, with $T_{\text{xc}}^{(s)} := T_{\text{oc}}^{(s)} \setminus T_{\text{bc}}^{(s)}$.

Moreover, we group together all boarding- and alighting corridors to interface corridors for their easy reference.

Definition 24 (Interface Corridor) A corridor $t \in T_c^{(s)}$ from some station $s \in \mathcal{S}$ is an interface corridor iff $t \in T_{\text{fc}}^{(s)}$, with $T_{\text{fc}}^{(s)} := T_{\text{bc}}^{(s)} \cup T_{\text{ac}}^{(s)}$.

This way, we group all corridors in any station $s \in \mathcal{S}$ to alighting-, boarding-, entrance-, exit-, inflow-, interface-, outflow-, and transfer flows from different points of view; cf. Tab. 3.3 below.

Table 3.3: Different corridor types in some station $s \in \mathcal{S}$

Type	Definition	Set	Purpose
Alighting C.	Def. 21	$T_{ac}^{(s)}$	Implements the possibility of a passenger flow from a vehicle stopped at *t to t^*
Boarding C.	Def. 20	$T_{bc}^{(s)}$	Implements the possibility of a passenger from *t onto a vehicle stopped at t^*
Entrance C.	Def. 22	$T_{ec}^{(s)}$	Implements the possibility of a passenger flow from TN's exterior to t^*
Exit C.	Def. 23	$T_{xc}^{(s)}$	Implements the possibility of a passenger flow from *t to TN's exterior
Inflow C.	Def. 14	$T_{ic}^{(s)}$	Implements the possibility of a passenger flow from the exterior of s to t^*
Interface C.	Def. 24	$T_{fc}^{(s)}$	Boarding or alighting corridor
Outflow C.	Def. 16	$T_{oc}^{(s)}$	Implements the possibility of a passenger flow from *t to the exterior of s
Transfer C.	Def. 15	$T_{tc}^{(s)}$	Implements the possibility of a passenger flow from *t to t^*

Platforms & Stops. If some vehicle $v \in \mathcal{V}^{(g)}$ is stopped at some waypoint $p_1 \in P_w^{(g)}$ in some transportation grid g , and passengers on-board v can alight from it to some gathering point $p_2 \in P_{gp}^{(s)}$ in some station $s \in \mathcal{S}$ via some alighting corridor $t_1 \in T_{ac}^{(s)}$ since

$$\left((g, p_1), (s, t_1) \right) \in \mathcal{I}_a,$$

then we say that p_2 is a platform and p_1 is a stop. Similarly, we say that p_2 is a platform and p_1 is a stop if passengers can board v from p_2 via some boarding corridor $t_2 \in T_{bc}^{(s)}$ since

$$\left((s, t_2), (g, p_1) \right) \in \mathcal{I}_b.$$

Definition 25 (Platform) A gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ is a platform iff $\exists t \in T_{\text{ac}}^{(s)} \cup T_{\text{bc}}^{(s)}$ s.t. $t \in \bullet p \cup p \bullet$.

Definition 26 (Stop) A waypoint $p \in P_w^{(g)}$ in some transportation grid $g \in \mathcal{G}$ is a stop iff $\exists t \in T_{\text{ac}}^{(s)} \cup T_{\text{bc}}^{(s)}$ in some station $s \in \mathcal{S}$ s.t.

$$\left((s, t), (g, p) \right) \in \mathcal{I}_b \text{ or } \left((g, p), (s, t) \right) \in \mathcal{I}_a.$$

Graphical Representation. In the graphical representation of TN 's complete infrastructure, we draw the connection between boarding- and alighting corridors on the one hand, and stops on the other hand by dashed arcs; cf. Fig. 3.2.3 below.

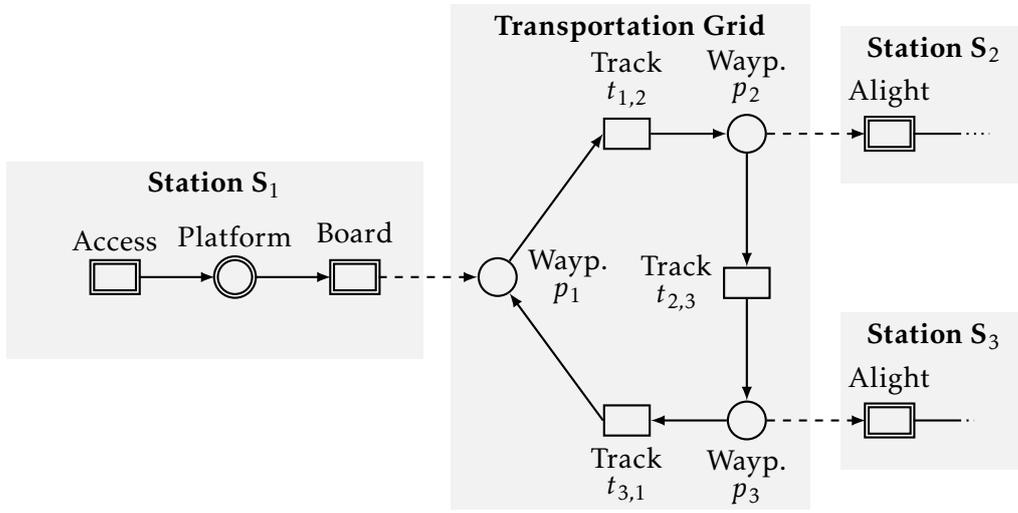


Figure 3.2.3: Graphical representation of a sample infrastructure for TN : Three stations S_1 , S_2 , and S_3 are connected to the same circular transportation grid. From the platform of S_1 , passengers can board a vehicle stopped at the waypoint p_1 . From a vehicle stopped at the waypoint p_2 , all on-board passengers can alight from it to S_2 . Similarly, from a vehicle stopped at the waypoint p_3 , all on-board passengers can alight from it to S_3 .

Specification of the Vehicle Operation

Vehicle Missions

Definition. Every vehicle mission $w \in \mathcal{W}^{(g)}$ from Def. 9 on p. 29 specified for transportation grid $g \in \mathcal{G}$ defines a path $\pi^{(w)} = a_0 a_1 \dots a_n$ of some length $n \in \mathbb{N}_{>0}$ in g , together with an indication of stops at all waypoints along that path.

Definition 27 (Vehicle Mission) Every vehicle mission $w \in \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is a collection

$$w := \left(\pi^{(w)}, \left\{ \lambda_-^{(w,p)}, \lambda_+^{(w,p)}, \lambda_{\text{nf}}^{(w,p)} \geq 0; \lambda_s^{(w,p)} \in [0, 1] : p \in \pi^{(w)} \cap P_w^{(g)} \right\} \right)$$

of

- a path $\pi^{(w)} := a_0 a_1 \dots a_n$ of some length $n \in \mathbb{N}_{>0}$ in g , with a waypoint at both of its ends which implies $a_0 \in P_w^{(g)}$ and $a_n \in P_w^{(g)}$,
- a minimum dwell time of $\lambda_-^{(w,p)}$ seconds at any waypoint $p \in \pi^{(w)} \cap P_w^{(g)}$,
- a maximum dwell time of $\lambda_+^{(w,p)}$ seconds at any waypoint $p \in \pi^{(w)} \cap P_w^{(g)}$, with $\lambda_+^{(w,p)} \geq \lambda_-^{(w,p)}$,
- a partition $\lambda_s^{(w,p)}$ of the complete dwell time at any waypoint $p \in \pi^{(w)} \cap P_w^{(g)}$, and
- a freezing time of $\lambda_{\text{nf}}^{(w,p)}$ seconds for the interruption of some vehicle's operation at any waypoint $p \in \pi^{(w)} \cap P_w^{(g)}$.

A stop at a particular waypoint $p \in P_w^{(g)} \cap \{a_0, a_1, \dots, a_n\}$ along $\pi^{(w)}$ must be indicated by a positive minimum dwell defined for it. The minimum dwell time $\lambda_-^{(w,p)}$ and the maximum dwell time $\lambda_+^{(w,p)}$ at p define the closed time interval $[\lambda_-^{(w,p)}, \lambda_+^{(w,p)}]$ within which some vehicle $v \in \mathcal{V}$ that executes w can try to depart from p : A departure in the time interval $[\lambda_-^{(w,p)}, \lambda_-^{(w,p)} + \lambda_s^{(w,p)} \Delta\tau]$, with $\Delta\tau := \lambda_+^{(w,p)} - \lambda_-^{(w,p)}$, is considered as an on-time departure, where this on-time departure of v from p must not occur as long as some passengers still want to board it or want to alight from it. On the other hand, v 's departure from p in the complementary time interval $[\lambda_-^{(w,p)} + \lambda_s^{(w,p)} \Delta\tau, \lambda_+^{(w,p)}]$ is considered to be a late departure. The occurrence of this departure ignores the number of passengers who want to board v , but only takes the number of passengers who want to alight from it into account. Moreover, if v cannot depart from p in $[\lambda_-^{(w,p)}, \lambda_+^{(w,p)}]$ because there are still some passengers on-board v who want to alight from it, then v 's operation is frozen for the next $\lambda_{\text{nf}}^{(w,p)}$ seconds. Once $\lambda_{\text{nf}}^{(w,p)}$ seconds have elapsed, v can try to depart from p in the time interval $[\lambda_-^{(w,p)} + \lambda_{\text{nf}}^{(w,p)}, \lambda_+^{(w,p)} + \lambda_{\text{nf}}^{(w,p)}]$ once again; and so forth.

Transportation Services & Dead-Headings. Some mission $w \in \mathcal{W}^{(g)}$ specified for some transportation service in $g \in \mathcal{G}$ defines a transportation service in g iff some vehicle $v \in \mathcal{V}^{(g)}$ which executes w (i) must stop at least at two different waypoints in g , say waypoints a_i and a_j along $\pi^{(w)}$, with $i < j$, (ii) passengers can board v at a_i from the platform of some station, and (iii) passengers can alight from a_j to the platform of some other station.

Definition 28 (Transportation Service) A vehicle mission $w \in \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ defines a transportation service in g iff $\exists i, j \in \{0, 1, \dots, n\}$, $i < j$, s.t

- $\lambda_-^{(w,a_i)} > 0$ and $a_i = (t_b)^\star$ for $t_b \in T_{\text{bc}}^{(s_1)}$ and station $s_1 \in \mathcal{S}$, so that passengers can board some vehicle $v \in \mathcal{V}$ that executes w and is stopped at a_i via t_b , and

- $\lambda_{-}^{(w,a_j)} > 0$ and $a_j = \star(t_a)$ for $t_a \in T_{ac}^{(s_2)}$ and $s_2 \in \mathcal{S}$, so that the same passengers can alight from v via t_a .

Otherwise, w defines a dead-heading in g .

Definition 29 (Dead-Heading) A vehicle mission $w \in \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ defines a dead-heading in g iff it does not define a transportation service in g .

Graphical Representation. The path that every vehicle mission $w \in \mathcal{W}^{(g)}$ unfolds in some transportation grid $g \in \mathcal{G}$ of TN , and all planned stops along this path, can be visualized in the graphical representation of TN 's infrastructure e.g. for the convenient specification of all vehicle missions; cf. Fig. 3.3.1 below: Label every track $t \in T^{(g)}$ that connects two waypoints along w 's path with the ID of w . Similarly, label all station corridors that are connected to a stop along w 's path with w 's ID iff w defines a positive minimum dwell time at these stops.

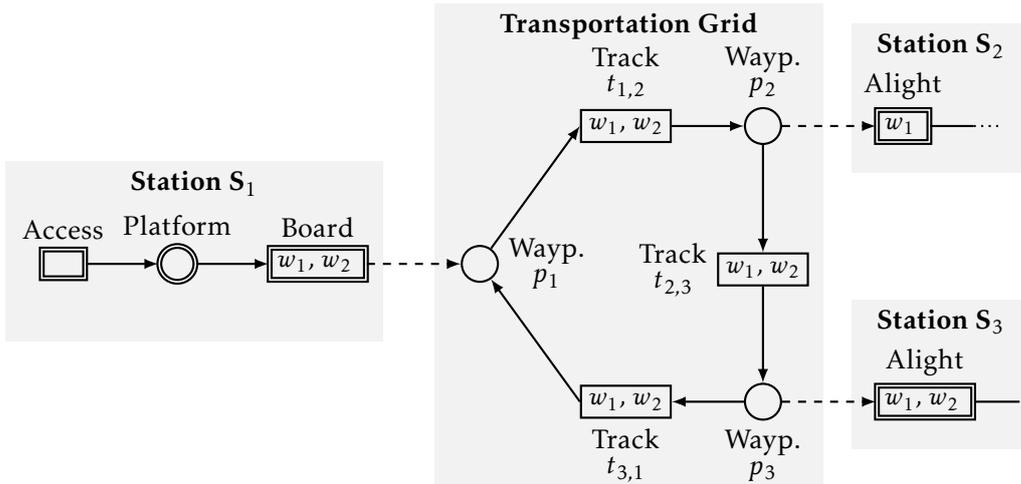


Figure 3.3.1: Visualization of some vehicle missions w_1 , w_2 , and w_3 in the graphical representation of the sample infrastructure for TN : A vehicle that executes w_1 is supposed to stop at the p_2 so that passengers can alight from it to the station S_2 . However, a vehicle that executes w_2 is supposed to skip p_2 .

Vehicle Runs

Every vehicle run $z \in \mathcal{Z}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ from Def. 9 on p. 29 defines a consistent sequence $z := w_1 w_2 \dots w_n$ of $n \in \mathbb{N}_{>0}$ different vehicle missions $w_1, w_2, \dots, w_n \in \mathcal{W}^{(g)}$ that have to be executed in the given order by a vehicle that processes z . Here, consistent means that the first waypoint in w_{i+1} is identical to the last waypoint in w_i for $i \in \{1, 2, \dots, n-1\}$.

Definition 30 (First Waypoint in a Vehicle Mission) *The first waypoint in a vehicle mission $w \in \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is a_0 if z 's path $\pi^{(w)} = a_0 a_1 \dots a_n$ in g has length $n \in \mathbb{N}_{>0}$.*

Definition 31 (Last Waypoint in a Vehicle Mission) *The last waypoint in a vehicle mission $w \in \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is a_n if z 's path $\pi^{(w)} = a_0 a_1 \dots a_n$ in g has length $n \in \mathbb{N}_{>0}$.*

Assuming this to be true, we say that w_i is compatible with w_{i+1} , which however does not imply the converse.

Definition 32 (Compatible Vehicle Missions) *A vehicle mission $w_1 \in \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is compatible with another vehicle mission $w_2 \in \mathcal{W}^{(g)}$ iff the first waypoint in w_2 is identical to the last waypoint in w_1 .*

Definition 33 (Vehicle Run) *Every vehicle run $z \in \mathcal{Z}$ is a sequence $z := w_1 w_2 \dots w_n$ of some $n \in \mathbb{N}_{>0}$ different vehicle missions $w_1, w_2, \dots, w_n \in \mathcal{W}^{(g)}$, with $w_i \neq w_j$ for $i \neq j$, specified for transportation grid $g \in \mathcal{G}$ s.t. w_i is compatible with w_{i+1} for $i \in \{1, 2, \dots, n-1\}$.*

Tab. 3.4 below illustrates some sample vehicle run.

Table 3.4: Illustration of some vehicle run

Order	Dead heading	Transportation service
1	Leave depot and move to TS A according the vehicle mission w_9	-
2	-	Realize express service from TS A to TS B according the vehicle mission w_4
3	-	Realize full service from TS B to TS A including stop at all intermediate stations according the mission w_{13}
4	Move to depot and enter it according the vehicle mission w_{21}	-

Scheduled Vehicle Dispatches

Every scheduled vehicle dispatch $d \in \mathcal{D}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ from Def. 9 on p. 29, defines a nominal time $\tau \geq 0$ together with some maximum time delay $\Delta\tau \geq 0$ w.r.t. τ , when some vehicle $v \in \mathcal{V}^{(g)}$ is supposed to start processing some vehicle run in g . By convention, d will be ignored iff v cannot be dispatched in the time interval $[\tau, \tau + \Delta\tau]$.

Definition 34 (Scheduled Vehicle Dispatch) A scheduled vehicle dispatch $d \in \mathcal{D}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is a tuple $d := (\tau, v, z, \Delta\tau)$ of

- a time $\tau \geq 0$ in seconds relative to the initial simulation time $\tau = 0$,
- a vehicle run $z \in \mathcal{Z}^{(g)}$,
- a vehicle $v \in \mathcal{V}^{(g)}$, and
- a time constant $\Delta\tau \geq 0$.

TN 's dispatch plan 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 such as a nominal mode of operation or a perturbed mode of operation. However, TN can process only one dispatch plan, which might require the modeller to merge two or more real-world dispatch plans if necessary; or to analyse them separately, i.e. to run two different computations/simulations with two different dispatch plans.

Table 3.5: Sample extract of TN 's dispatch plan that might cover a complete working day from 9.07am until 10.13pm for a nominal mode of operation

Dispatch Time	Vehicle	Run	Max. Delay
9.07am	17	VR 0815	1 min
9.11am	3	KW 4711	3 min
⋮	⋮	⋮	⋮
10.13pm	8	FR 0033	2 min

Signalling System

The signalling system SI of TN from Def. 9 on p. 29 defines the driving times of all vehicles between the waypoints in the different transportation grids $g \in \mathcal{G}$ in any mode $q \in \mathcal{Q}$: In order to better distinguish a vehicle's driving- from its dwell time, we could refine the definition of a mode in that we would not mix stopped vehicles whose doors are open and thus enable boarding & alighting passenger flows, with stopped vehicles whose doors are closed.

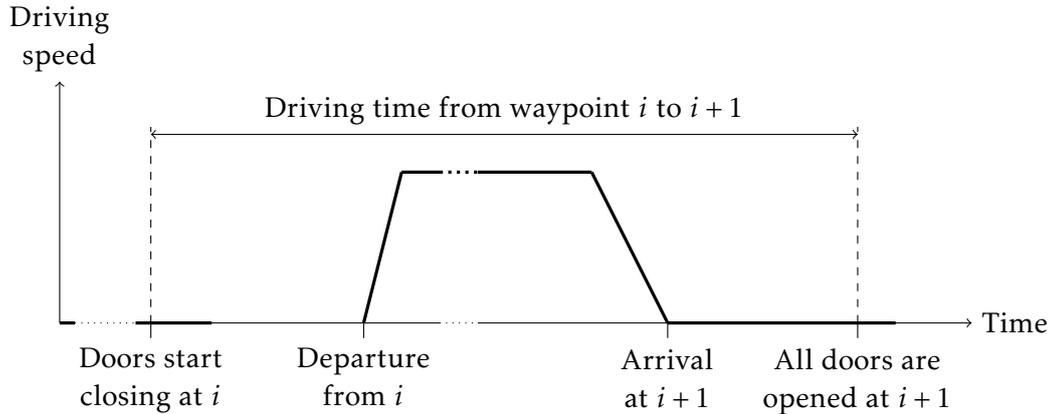


Figure 3.3.2: Sample decomposition of a vehicle's driving time between two consecutive stops at the waypoints i and $i+1$, which accounts for an acceleration and a breaking phase, and the time to close and open the doors

However, the focus in the development of *T-SHA* is on the passenger flows and their interaction with the vehicles, and not particularities in the vehicle dynamics: A vehicle that is stopped but whose doors are closed, does not enable a passenger flow. From a passenger-flow dynamical point of view, it is thus not different from a driving vehicle which has not yet arrived at the considered stop. In this context, we demand that the driving time of some stopped vehicle, say stopped at waypoint p_1 , to some other waypoint, say p_2 , where it is supposed to stop next for the purpose of boarding & alighting, must cover the complete time that elapses upon its doors are closed at p_1 until it has opened its doors at p_2 again.

The fact that all driving times do not have to be kept constant during the execution of *T-SHA* but can rather be coupled to some mode $q \in \mathcal{Q}$, enables us to replicate more precisely the driving times in the actual transportation network: By convention, (i) every waypoint can accommodate at maximum one vehicle at the same time, and (ii) the position in form of a waypoint which *T-SHA*'s mode assigns to some moving vehicle corresponds to the waypoint this vehicle moves towards in said mode. In *T-SHA* a vehicle thus cannot depart from a waypoint if its next waypoint is blocked by another vehicle, and we do not have to enforce this fixed block-spacing of all vehicles by appropriately specifying *SI*. However, in practice every vehicle most likely adjusts its driving speed to the environment it observes in that e.g. it reduces its driving speed if it detects some kind of a bottleneck along its planned trajectory. Moreover, transportation operators might want to selectively manipulate some speed limits so as to catch up deviations from some desired operation. That is why, we let *SI* the possibility to base its decision for the definition of all driving times in some particular transportation grid on the vehicles' positions, -driving conditions, and -operational states in all grids, i.e. on *T-SHA*'s complete mode rather than some restricted part of it.

Definition 35 (Signalling System) *The signalling system SI of TN is a function*

$$SI : (q, g, v, t) \in \mathcal{Q} \times \mathcal{G} \times \mathcal{V}_g \times T_{\text{tr}}^{(g)} \rightarrow \mathbb{R}_{>0}$$

that assigns to every tuple (q, g, v, t) of

- some mode $q \in \mathcal{Q}$ of TN ,
- some transportation grid $g \in \mathcal{G}$,
- some vehicle $v \in \mathcal{V}_g$ operated in g , and
- some track $t \in T_{\text{tr}g}$ connecting the waypoint *t to the waypoint t^* in g ,

a (time-invariant) driving time of $SI(q, g, v, t)$ seconds for v to go from *t to t^* .

Consistent Vehicle Operation

TN 's vehicle operation is consistent iff the specification of its dispatch plan is consistent. Now in order to check the consistency of TN 's dispatch plan in form of the set \mathcal{D} of all scheduled vehicle dispatches, we process for every vehicle $v \in \mathcal{V}^{(g)}$ operated in some transportation grid $g \in \mathcal{G}$ the following four steps: First of all, we extract from $\mathcal{D}^{(g)}$ all scheduled dispatches involving v and copy them to the set

$$\mathcal{D}^{(g,v)} := \left\{ \exists (\tau, \bar{v}, z, \Delta\tau) \in \mathcal{D}^{(g)} : \bar{v} = v \right\}, \quad (3.11)$$

We next verify that every $d \in \mathcal{D}^{(g,v)}$ specifies a different dispatch time for v since v cannot start processing two different vehicle runs at the same time. Assuming this to be true, we moreover have to verify that all pairs of two consecutive dispatches $d_1, d_2 \in \mathcal{D}^{(g,v)}$ of v are consistent: The last waypoint in the last mission in the vehicle run specified by d_1 , say vehicle run z_1 , must be identical to the first waypoint in the first mission in the vehicle run specified by d_2 , say z_2 , in which case we then say that d_1 is compatible with d_2 .

Definition 36 (First Mission in a Vehicle Run) *The first vehicle mission in a vehicle run $z \in \mathcal{Z}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is w_1 iff $z = w_1 w_2 \dots w_n$ for $n \in \mathbb{N}_{>0}$ and $w_1, w_2, \dots, w_n \in \mathcal{W}^{(g)}$.*

Definition 37 (Last Mission in a Vehicle Run) *The last vehicle mission in a vehicle run $z \in \mathcal{Z}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is w_n iff $z = w_1 w_2 \dots w_n$ for $n \in \mathbb{N}_{>0}$ and $w_1, w_2, \dots, w_n \in \mathcal{W}^{(g)}$.*

Definition 38 (Compatible Vehicle Runs) *A vehicle run $z_1 \in \mathcal{Z}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$ is compatible with a vehicle run $z_2 \in \mathcal{Z}^{(g)}$ iff the first waypoint in z_2 's first mission is identical to the last waypoint in z_1 's last mission.*

However, this is not enough. We must also verify that all $d \in \mathcal{D}^{(g,v)}$ respect the vehicle runs' minimum process times which can be computed from the missions' minimum execution times.

Definition 39 (Minimum Execution Time of a Vehicle Mission) *The minimum execution time of a vehicle mission $w \in \mathcal{W}^{(g)}$ specified for transportation grid g , with a path $\pi^{(w)} = a_0 a_1 \dots a_n$ of some length $n \in \mathbb{N}_{>0}$ in $N^{(g)}$, is*

$$\lambda_{\text{mt}}(g, w) := \sum_{p \in \pi^{(w)} \cap P_w^{(g)}} \lambda_{-}^{(w)}(p) + \min_{(v, q) \in \mathcal{V}^{(q, w)} \times \mathcal{Q}} \sum_{t \in \pi^{(w)} \cap T_{\text{tr}}^{(g)}} SI(q, g, v, t) \quad (3.12)$$

seconds, with $\lambda_{\text{mt}} : (g, w) \in \mathcal{G} \times \mathcal{W}^{(g)} \rightarrow \mathbb{R}_{>0}$, where

$$\mathcal{V}^{(q, w)} := \{\bar{v} \in \mathcal{V}^{(g)} : \lambda_{\text{m}}^{(q)}(\bar{v}) = w\},$$

for mode $q \in \mathcal{Q}$, denotes the subset of all vehicles that execute w iff TN is in q .

Definition 40 (Minimum Process Time of a Vehicle Run) *The minimum process time of any vehicle run $z \in \mathcal{Z}$, with $z = w_1 w_2 \dots w_n$ and $w_1, w_2, \dots, w_n \in \mathcal{W}^{(g)}$ for $n \in \mathbb{N}_{>0}$, specified for transportation grid $g \in \mathcal{G}$ is*

$$\lambda_{\text{rt}}(g, z) := \sum_{i \in \{1, 2, \dots, n\}} \lambda_{\text{mt}}(g, w_i) \quad (3.13)$$

seconds, with $\lambda_{\text{rt}} : (g, z) \in \mathcal{G} \times \mathcal{Z} \rightarrow \mathbb{R}_{>0}$, and the minimum execution time $\lambda_{\text{mt}}(g, w)$ of some vehicle mission $w \in \mathcal{W}^{(g)}$ computed according to Eqn. 3.12 from Def. 39 on p. 57.

More specifically, we demand that $\mathcal{D}^{(g, v)}$ respects the fact that v cannot start processing some new run before it has completed its current run.

Definition 41 (Consistent Dispatch Plan) *The dispatch plan of TN is consistent iff for transportation grid $g \in \mathcal{G}$, any pair of two dispatch times $\tau_1, \tau_2 \in \mathbb{R}_{\geq 0}$, any vehicle $v \in \mathcal{V}^{(g)}$, any pair of two vehicle runs $z_1, z_2 \in \mathcal{Z}^{(g)}$, and any pair of two time delays $\Delta\tau_1, \Delta\tau_2 \in \mathbb{R}_{\geq 0}$:*

- $(\tau_1, v, z_1, \Delta\tau_1) \in \mathcal{D}^{(g)}$ and $(\tau_2, v, z_2, \Delta\tau_2) \in \mathcal{D}^{(g)}$ implies $\tau_1 \neq \tau_2$ since v cannot start processing two different runs at the same time.
- $(\tau_1, v, z_1, \Delta\tau_1) \in \mathcal{D}^{(g, v)}$, $(\tau_2, v, z_2, \Delta\tau_2) \in \mathcal{D}^{(g, v)}$, and $\tau_2 > \tau_1$ implies z_1 is compatible with z_2 assuming that $\exists (\tau_3, v, z_3, \Delta\tau_3) \in \mathcal{D}^{(g, v)}$, with $\tau_3 \geq 0$, $z_3 \in \mathcal{Z}^{(g, v)}$, and $\Delta\tau_3 \geq 0$, s.t. $\tau_1 < \tau_3 < \tau_2$.
- $(\tau_1, v, z_1, \Delta\tau_1) \in \mathcal{D}^{(g, v)}$, $(\tau_2, v, z_2, \Delta\tau_2) \in \mathcal{D}^{(g, v)}$, and $\tau_2 > \tau_1$ implies $\tau_2 > \tau_1 + \lambda_{\text{rt}}(g, z_1)$, with the minimum process time $\lambda_{\text{rt}}(g, z_1)$ of z_1 computed according to Eqn. 40 from p. 57, since v cannot start processing z_2 when it still processes z_1 .

Specification of the Passenger Routing

Trip Profiles

At any time every passenger travels according to one particular trip profile $y \in \mathcal{Y}$, which defines a sequence of actions that (i) must provoke a passenger with y to

enter and finally to leave TN from/to its exterior, (ii) may provoke some transfers between gathering points in the stations, and (iii) must provoke at least one ride on-board a vehicle. Of course, every sequence of actions must respect some order. We enforce this order in that we define y as a word of a regular language generated by a regular expression [Jiang 2010, p. 1-6], where every letter of y corresponds to a particular action.

Definition 42 (Trip Profile) *Every trip profile $y \in \mathcal{Y}$ is a word of the regular language R_{tp} , which is generated by the regular expression*

$$r_{\text{tp}} := \bigoplus_{s_1 \in \mathcal{S}} A_e^{(s_1)} (A_t^{(s_1)})^* A_b^{(s_1)} \bigoplus_{s_2 \in \mathcal{S}} (A_a^{(s_2)} (A_t^{(s_2)})^* A_b^{(s_2)})^* \bigoplus_{s_3 \in \mathcal{S}} A_a^{(s_3)} (A_t^{(s_3)})^* A_x^{(s_3)},$$

with the five different types of actions

- $A_e^{(s)} := T_{\text{ec}}^{(s)}$ to enter station $s \in \mathcal{S}$ from TN's exterior,
- $A_t^{(s)} := T_{\text{tc}}^{(s)}$ to transfer between two gathering points in s ,
- $A_b^{(s)} := T_{\text{bc}}^{(s)} \times (2^{\mathcal{W}} \setminus \emptyset)$ to board some vehicle mission that defines a transportation service in some transportation grid according to Def. 28 on p. 51 from s ,
- $A_a^{(s)} := T_{\text{ac}}^{(s)}$ to alight from a vehicle to s , and
- $A_x^{(s)} := T_{\text{xc}}^{(s)}$ to leave s to TN's exterior.

The first letter of every trip profile $y \in \mathcal{Y}$ must be an entrance corridor $t_e \in T_{\text{ec}}^{(s)}$ in some station $s \in \mathcal{S}$, which indicates the point where all passengers who travel according to y enter TN from its exterior; namely through t_e in s which brings them to the gathering point $(t_e)^*$ next. From $(t_e)^*$ they then might want to transfer to some other gathering point in s , which must be defined in form of a sequence of transfer corridors that unfolds a path in s : For instance if a passenger who travels according to y wants to transfer from $^*(t_{t,1})$ to $(t_{t,1})^*$ and then from $^*(t_{t,2})$ to $(t_{t,2})^*$, for $t_{t,1}, t_{t,2} \in T_{\text{tc}}^{(s)}$ with $(t_{t,1})^* = ^*(t_{t,2})$, then $t_{t,1}$ and $t_{t,2}$ must be two letters of y , where $t_{t,2}$ immediately follows $t_{t,1}$.

Remark 1 *A sequence $t_1 t_2 \dots t_k$ of $k \in \mathbb{N}_{>0}$ corridors $t_1, t_2, \dots, t_k \in T_c^{(s)}$ in some station $s \in \mathcal{S}$ unfolds a path in s iff the sequence*

$$^*(t_1) t_1 (t_1)^* ^*(t_2) t_2 \dots (t_k)^*$$

is a path in $N^{(s)}$ according to Def. 5 on p. 16.

At some point all above passengers must arrive at a platform $^*(t_b)$ in s , for boarding corridor $t_b \in T_{\text{bc}}^{(s)}$, from which they want to board a vehicle $v \in \mathcal{V}$ that executes some mission $w \in \mathcal{W}$ through t_b next. Assuming this to be true, then (t_b, w) is a letter of y . Moreover, w must define (i) a transportation service, (ii) a stop at $(t_b)^*$, and (iii) another stop at $^*(t_a)$, for alighting corridor $t_a \in T_{\text{ac}}^{(s')}$ in a station $s' \in \mathcal{S}$,

where all passengers of y want to alight from v . Also note that (t_b, \mathcal{W}_b) must replace (t_b, w) if all passengers of y shall consider all vehicle missions in $\mathcal{W}_b \subseteq \mathcal{W}$ as equivalent alternatives; equivalent in a sense that all passengers of y do not have a preference for $w \in \mathcal{W}_b$ but try to board whatever vehicle, that executes a mission in \mathcal{W}_b and is stopped at $(t_b)^\star$, is available.

Remark 2 *By convention, every waypoint can accommodate at maximum one vehicle at a time. That is why, a passenger can never choose between two or more vehicle missions at the same time as long as she sticks to her trip profile.*

The last letter of y must be some exit corridor through which all passengers of y want to leave TN .

Definition 43 (Consistent Passenger Routing) *TN 's set \mathcal{Y} of all trip profiles constitutes a consistent passenger routing in TN iff for $y \in \mathcal{Y}$, for pair (t_b, t_a) of a boarding corridor $t_b \in \bigcup_{s \in \mathcal{S}} T_{bc}^{(s)}$ and an alighting corridor $t_a \in \bigcup_{s \in \mathcal{S}} T_{ac}^{(s)}$, and any subset of vehicle missions $\mathcal{W}_b \subseteq \mathcal{W}^{(g)}$ specified for transportation grid $g \in \mathcal{G}$; the fact that $(t_b, \mathcal{W}_b) t_a$ is a letter of y implies for every $w \in \mathcal{W}_b$ that*

- *w defines a transportation service in g according to Def. 28 from p. 51,*
- *w specifies a stop at $(t_b)^\star$ in that w defines a positive minimum dwell time at $^\star(t_b)$, and*
- *w specifies a stop at $^\star(t_a)$ in that w defines a positive minimum dwell time at $^\star(t_a)$.*

Graphical Representation. Every trip profile $y \in \mathcal{Y}$ can be visualized in the graphical representation of TN 's infrastructure e.g. for its convenient specification: First of all, label all tracks in TN 's transportations grids with the path and stop specifications of all vehicle missions $w \in \mathcal{W}$, as described at the end of Sec. 3.3.1. Next label every corridor $t \in T_c^{(s)}$ in some station $s \in \mathcal{S}$ with the ID with y iff t or $t\mathcal{W}'$ for $\mathcal{W}' \subseteq \mathcal{W}$ is a letter of y . Assuming that $t\mathcal{W}'$ is a letter of y , then write y 's ID behind every $w \in \mathcal{W}'$.

Look at Fig. 3.4.1 below, which depicts the sample infrastructure from Fig. 3.2.3 on p. 50 together with the specification of three different trip profiles 1, 2, and 3.

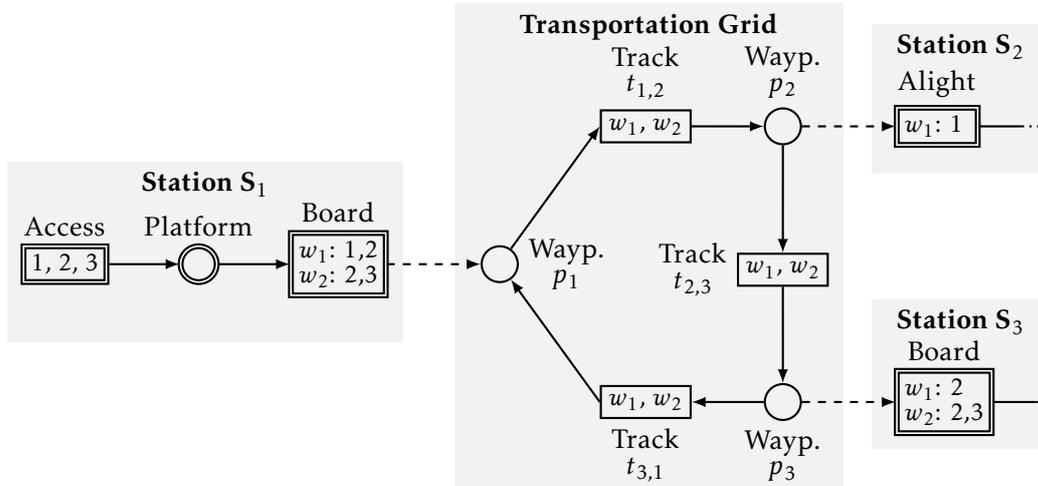


Figure 3.4.1: Sample infrastructure for TN labelled with the path specifications of two vehicle missions, w_1 and w_2 , and the route specifications of three trip profiles (TPs): In order to go from station S_1 to S_3 , all passengers of TP 3 prefer to board a vehicle that executes w_2 over a vehicle that executes w_1 ; although a vehicle that executes w_1 provides the same transportation service but with an additional stop at S_2 .

All passengers who travel according 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 w_1 , which passes through and stops at all three waypoints p_1 , p_2 , and p_3 . On-board such a vehicle stopped at p_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 according to the second or third trip profile would like to alight from a vehicle that is stopped at p_3 to a GP in the station S_3 (not depicted). Then note that, in principle, both vehicle missions w_1 and w_2 provide a transportation service from S_1 to S_3 . However in contrast to w_1 , a vehicle that executes w_2 is supposed to skip p_2 and thus provides an express service from S_1 to S_3 . Now the difference between the passengers who travel according to the second trip profile from the passengers who travel according to the third trip profile is that the latter show a preference for w_2 ; whereas all passengers of the second trip profile are indifferent to both vehicle missions and thus board whatever vehicle mission is available.

Passenger Re-Routing

The fact that a finite set of trip profiles underlies the complete passenger routing in $T-SHA$, does not contradict the possibility that some passengers change their trip profiles while they travel in $T-SHA$. In practice, reasons to do so might be manifold such as a massive perturbation of $T-SHA$'s vehicle operation which is communicated to all passengers waiting at the platforms for the vehicles to arrive via loudspeaker announcements. Hence, also in $T-SHA$ passengers are allowed to

change their trip profiles; however iff they cross some corridor so as to e.g. board a vehicle from a platform next.

Definition 44 (Passenger Re-Routing) *Every passenger re-routing $r \in \mathcal{R}$ in T -SHA is specified in form of a tuple $r := (y, s, b, u)$ of*

- some trip profile $y \in \mathcal{Y}$,
- some station $s \in \mathcal{S}$
- some transfer corridor $t \in T_{tc}^{(s)}$ in s , which implies $b = t$, or some tuple (t', w) of a boarding or an alighting corridor $t' \in T_{fc}^{(s)}$ from some station $s' \in \mathcal{S}$ and a vehicle mission $w \in \mathcal{W}$, which then implies $b = (t', w)$, and
- some column vector

$$u \in \left\{ k \in (\mathbb{R}_{\geq 0})^{|\mathcal{Y}|} : \sum_{i \in \mathcal{Y}} k[i] = 1 \right\}$$

which specifies the relative amount $u[j]$, for trip profile $j \in \mathcal{Y}$, of passengers who enter t - respectively t' - according to trip profile y and who leave t - respectively t' - according to j .

Derivation of Routing Matrices

Meaning. We map the specification of all trip profiles and the specification of the passenger re-routing among these trip profiles in T -SHA to a finite set of so-called routing matrices

$$R^{(s,b)} \in \left\{ K \in [0, 1]^{|\mathcal{Y}| \times |\mathcal{Y}|} : \sum_{i \in \mathcal{Y}} K[i, j] \in \{0, 1\}, \forall j \in \mathcal{Y} \right\},$$

for all

$$(s, b) \in \mathcal{S} \times \left((T^{(s)} \setminus T_{fc}^{(s)}) \cup (T_{fc}^{(s)} \times \mathcal{W}) \right).$$

We will use them for the simple (algebraic) verification of the proper routing of all passenger flows in T -SHA (see Sec. 3.4.4 on p. 64ff.), and the specification of all balance equations for the continuous passenger-flow dynamics in any mode $q \in \mathcal{Q}$ of T -SHA.

The i -th row and the j -th column of $R^{(s,b)}$, with $i, j \in \mathcal{Y}$, assigned to

- some entrance corridor $t \in T_{ec}^{(s)}$ in some station $s \in \mathcal{S}$, gives the relative amount of passengers who enter t from TN 's exterior according to trip profile j , and who leave t to t^* according to trip profile i ,
- some transfer corridor $t \in T_{tc}^{(s)}$ in some station $s \in \mathcal{S}$, gives the relative amount of passengers who enter t from *t according to trip profile j , and who leave t to t^* according to trip profile i ,

- some pair of boarding corridors $t \in T_{bc}^{(s)}$ in some station $s \in \mathcal{S}$ and a vehicle mission $w \in \mathcal{W}$, gives the relative amount of passengers who enter t according to trip profile j , and who leave t on-board a vehicle that (i) is stopped at t^* and (ii) executes the mission w according to trip profile i ,
- some pair of alighting corridors $t \in T_{ac}^{(s)}$ in some station $s \in \mathcal{S}$ and a vehicle mission w , that gives the relative amount of passengers on-board some vehicle, which (i) is stopped at t^* and (ii) executes the mission w , who enter t according to trip profile j , and who leave t to t^* according to trip profile i , and
- some exit corridor $t \in T_{xc}^{(s)}$ in some station $s \in \mathcal{S}$, gives the relative amount of passengers, who enter t from t^* , and who leave t to TN 's exterior according to trip profile i .

Example. Fig. 3.4.2 below illustrates the principle of all routing matrices.

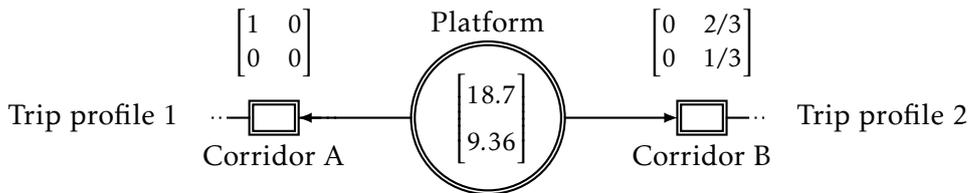


Figure 3.4.2: Principle of routing matrices: All 18.7 passengers with trip profile 1 at the platform want to take the corridor A, and all 9.36 passengers with the trip profile 2 want to take the corridor B. Among all passengers who enter corridor B according to trip profile 2, two thirds leave corridor B according to trip profile 2, and the remaining one third according to trip profile 1.

Depicted is a platform in some station that is connected to two corridors. Inscribed to this platform is a particular realization for all passenger numbers: 18.7 passengers travel according to trip profile 1, and 9.36 passengers travel according to trip profile 2. From the two routing matrices written above the two corridors A and B, we know that all passengers with trip profile 1 want to take corridor A, and all passengers with trip profile 2 want to take corridor B. More precisely, we even know that among all passengers who enter corridor B according to trip profile 2, two thirds leave corridor B according to trip profile 1. The remaining one third of passengers leaves corridor B according to trip profile 2.

Computation. We compute all $R^{(\cdot)}$ in five steps according to Alg. 1 below. Note that in order to keep the notational effort to a minimum, this algorithm computes more routing matrices than necessary, since it does not account for the missions' specifications and their actual assignment to the different transportation grids. In theory, we do only have to compute a transition matrix for some pair of a boarding corridor, say t , and a mission, say w , iff w specifies a stop at t^* .

Algorithm 1 Computation of all routing matrices $R(\cdot)$ for T -SHA

Step 1: Instantiate an empty routing matrix for every possible passenger transfer flow within a station, every passenger flow entering a station from TN 's exterior, and every passenger flow leaving a station to TN 's exterior.

1: **for**
 all $(s, t) \in \mathcal{S} \times (T_c^{(s)} \setminus T_{fc}^{(s)})$

2: **do**

3: $R^{(s,t)} := 0^{|\mathcal{Y}| \times |\mathcal{Y}|}$

4: **end for**

Step 2: Instantiate an empty routing matrix for every possible boarding & alighting passenger flow between a platform and a vehicle stopped in front.

5: **for**

 all $(s, t) \in \mathcal{S} \times T_{bc}^{(s)}$ and $w \in \mathcal{W}$

6: **do**

7: $R^{(s,(t,w))} := 0^{|\mathcal{Y}| \times |\mathcal{Y}|}$

8: **end for**

Step 3: Read in the specification of all trip profiles in the empty routing matrices that were instantiated for all passenger transfer flows within stations, passenger flows leaving stations to TN 's exterior, and passenger flows entering stations from TN 's exterior.

9: **for**

 all $(s, t) \in \mathcal{S} \times (T_c^{(s)} \setminus T_{fc}^{(s)})$ and trip profiles $i \in \mathcal{Y}$

10: **do**

11: **if**

t is an action in i

12: **then**

13: $R^{(s,t)}[i, i] := 1$

14: **end if**

15: **end for**

Step 4: Read in the specification of all trip profiles in the empty routing matrices that were instantiated for all boarding & alighting passenger flows.

16: **for**

all $(s, t) \in \mathcal{S} \times T_{bc}^{(s)}$, subsets of vehicle missions $\mathcal{W}_b \subseteq \mathcal{W}$, and trip profiles $i \in \mathcal{Y}$

```

17: do
18:   if
       $t\mathcal{W}_b$  is an action in  $i$ 
19:   then
20:     for
      all  $w \in \mathcal{W}_b$ 
21:     do
22:        $R^{(s,(t,w))}[i, i] := 1$ 
23:     end for
24:   end if
25: end for

```

Step 5: Read in the specification of the passenger re-routing situation.

```

26: for
      every passenger re-routing  $r \in \mathcal{R}$ , with  $r := (y, s, k_1, k_2)$ 
27: do
28:    $R^{(s,k_1)}[\cdot, y] := k_2$ .
29: end for

```

Verification of the Proper Routing of All Passenger Flows

We demand the same behaviour/intended action, from all passengers with the same trip profile; at any gathering point in any station and on-board any vehicle docked to some station for the purpose of boarding & alighting: Either all passengers want to stay where they are, or they want to take the same corridor so as to either 1) transfer to another gathering point within the same station, 2) board some stopped vehicle, 3) alight from some stopped vehicle, or 4) leave the station to TN 's exterior.

Definition 45 (Proper Passenger Routing) *All passengers are routed properly in T-SHA iff in any mode $q \in \mathcal{Q}$ of T-SHA,*

$$\left\{ \left\{ t \in (\bullet p \cup p \bullet) \cap (T_c^{(s)} \setminus T_{fc}^{(s)}) : \sum_{i \in \mathcal{Y}} R^{(s,t)}[i, y] = 1 \right\} + \right. \quad (3.14)$$

$$\left. \left\{ \left\{ t' \in (\bullet p \cup p \bullet) \cap T_{fc}^{(s)}, w \in \mathcal{W} : \sum_{k \in \mathcal{Y}} R^{(s,(t,w))}[k, y] = 1 \right\} \right\} \leq 1$$

for every gathering point $p \in P_s$ in any station $s \in \mathcal{S}$, and any trip profile $y \in \mathcal{Y}$.

Specification of All Passenger Flows

Overview

We distinguish five different types of passenger flows in *T-SHA* depending on their relative orientation w.r.t. the system boundary of the considered transportation network; namely network inflows, transfer flows, boarding flows, alighting flows, and network outflows. Among all these five different types of passenger flows, it is only the network inflow which might inject uncertainty into the continuous passenger flow-dynamics of *T-SHA*.

Table 3.6: Different types of passenger flows in *T-SHA* together with properties (A: autonomous, C: capacity-sensitive, D: demand-sensitive) which have to be enforced during their specification

Name	Orientation	A	C	D
Network Inflow	Network's exterior to some gathering point	✓	✓	
Transfer Flow	Between two gathering points in same station	✓	✓	✓
Boarding Flow	Platform to stopped vehicle	✓	✓	✓
Alighting Flow	Stopped vehicle to platform	✓	✓	✓
Network Outflow	Gathering point to network's exterior	✓		✓

We cannot provide some stringent set of universal rules for the specification of all above different types of passenger flows here; e.g. obtained from statistical observations. These rules would inevitably depend on the considered use case, including - but not limited to - the passengers' orientation and wayfinding capabilities in the complex geometries at hand such as the stations' interiors. However, what we can and will do in Sec. 3.5.2 to Sec. 3.5.5 below is to define qualitative properties for all passenger flows that have to be enforced during their specification so as to ensure the applicability of all computational bricks which we will introduce in the rest of this report. We will moreover provide some simple ansatz function for every transfer flow, every boarding & alighting flow, and every network outflow.

Passenger Flow Properties. We will assume from every passenger flow that it is demand- and/or capacity-sensitive: If the passenger compartment of some stopped vehicle is full, then no more passengers can board it. Similarly, if no passenger is on-board some stopped vehicle, then no passenger can alight from it. It is as simple as that (both in practice and in *T-SHA*). We moreover will demand from every passenger flow that it is autonomous: Its magnitude and its composition

w.r.t. the passengers' different trip profiles are a function of the passenger number at some gather point p iff the considered flow originates or ends in p . Similarly, the flow's magnitude and composition are a function of the passenger number on-board some vehicle v iff this flow originates or ends in v . No flow depends on the valuation of any clock associated with the operation of some vehicle.

Proposed Ansatz Functions. A basic assumption which will underlie all our ansatz functions is the fact that the corresponding passenger flows are fair: If $n_1 > 0$ passengers of some trip profile y_1 want to cross some corridor t at some time $\tau \geq 0$, and at the same time $n_2 > 0$ passengers of some other trip profile $y_2 \neq y_1$ want to cross t as well, then the number of passengers who cross t at τ with y_1 relative to the number of passengers who cross t at τ with y_2 is n_1/n_2 . All what then remains for the end user of our ansatz function to do, is to parametrize the relation between the magnitude of the considered flow and the number of passengers who want to be part of it, the number of passengers who might stand in its way, and so on. We believe that this use case-dependent (parametrizable) knowledge can be *created* in an automated way from simulations of sophisticated existing pedestrian flow models. Beside generalized cellular automata [Zhang 2008, Burstedde 2001], especially physical force models seem to be a good choice. They can capture not all but many important phenomena inherent to crowd dynamics when many passengers come together, i.e. situations typical for transportation infrastructures (refer to e.g. [Duives 2013] for an overview of different pedestrian flow modelling & simulation approaches and their capabilities in terms of complex crowd phenomena which they manage to replicate). Moreover they have a rigorous mathematical formulation as opposed to most of the commercial crowd simulation tools which can be purchased in the year 2016. However, here note that not every expert in the domain of passenger crowd simulation & analysis promotes the use of mathematical models [Still 2000, p. 16]:

“Understanding crowd dynamics is essential for understanding crowd safety. In 1992, at gate C, a small flow of people, coming up along the edge of the queue via a grass embankment, had an effect on the operation of the turnstiles. This small flow effectively blocked the forward motion of the crowd. No such phenomenon occurs in fluids, so there have to be some fundamental differences between crowd and fluid flows. The laws of crowd dynamics have to include the fact that people do not follow the laws of physics, they have a choice in their direction, have no conservation of momentum and can stop and start at will. They cannot be reduced to equations which are appropriate for the movement of ball bearings through viscous fluids.”

Anyway we will look very briefly at both physical- and social force models in the next paragraph, where we will also provide some references for further studies. However, before we will do so let us introduce two useful definitions first, which we will employ in the specification of our ansatz functions.

Definition 46 (Cumulative Passenger Number at Gathering Point) *The cumulative passenger number at some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ at some time $\tau \geq 0$ is*

$$m^{(s,p)}(\tau) := \sum_{y \in \mathcal{Y}} M^{(s,p)}(\tau)[y],$$

with

$$m^{(s,p)} : \mathcal{M}^{(s,p)} \rightarrow [0, \lambda_c^{(s,p)}].$$

Definition 47 (Cumulative Passenger Number On-board Vehicle) *The cumulative passenger number on-board some vehicle $v \in \mathcal{V}$ at some time $\tau \geq 0$ is*

$$u^{(v)}(\tau) := \sum_{y \in \mathcal{Y}} U^{(v)}(\tau)[y],$$

with

$$u^{(v)} : \mathcal{U}^{(v)} \rightarrow [0, \lambda_c^{(v)}].$$

Moreover, note that for the specification of some ansatz function data from so-called psycho-physical experiments may exist. For instance, [Fujiyama 2014] provides data for the specification of boarding & alighting passenger flows of/from from stopped metro vehicles for different door- and cabin layout configurations.

Physical Force Models. These models map the dynamics of a passenger crowd moving in some physical geometry to a system of ordinary differential equations (ODEs); one ODE for every individual passenger which has the shape

$$\text{Mass of passenger } i \frac{d}{d\tau} \text{Walking speed of } i = \sum \text{Forces applied to } i$$

of Newton's second law of motion: All passengers are modelled as point masses in some inertial reference frame who can exhibit accelerating and decelerating forces. These forces push them away from other passengers or obstacles, and pull them to their destinations or some other local points of comfort such as handrails. The way all forces are set up, groups all physical force models into either a) magnetic force models or b) social force models. See [Teknomo 2002, p. 14 - 19] for an overview of both.

Magnetic force models are inspired by the interaction of electrically charged bodies observed in nature. See e.g. [Okazaki 1993] for some graphical illustration of this model, and the way it is used to analyse the passenger movements in a railway station: Two oppositely charged bodies attract each other, whereas two positively charged or two negatively charged bodies practice repulsive forces on each other. Whether these forces are repulsive or not, they decay with the square of the distance between both bodies according to Coloumb's inverse square law. In some magnetic force model, this means that all passengers, all infrastructure obstacles, and the passengers' destinations are charged. For instance, all passengers are positively charged (and thus try to stay away from each other), all obstacles are also positively charged, but the passengers' destinations are negatively

charged. The main concern with magnetic force models is the good choice of all initial charges; some problem which does not exist in this form for social force models: All passengers have a desire to arrive at their destinations on-time which latter goal either motivates them to walk faster towards this destination or to slow them down. Moreover all passengers try to keep some distance to all other passengers so as to ensure privacy, their desired free walking speeds, etc. Since many different (social) factors defining the interaction between walking passengers can be mapped very easily to either attracting or repulsive forces, it is not a bit surprise that social force models manage to replicate many complex dynamics such as bonding effects [Xu 2010] much better than other pedestrian flow models; and this although their mathematical framework (= coupled ODEs) is straightforward.

Network Outflows

A network outflow is any passenger flow from some gathering point $\star t$ in some station $s \in \mathcal{S}$ to T -SHA's exterior via some exit corridor $t \in T_{xc}^{(s)}$. By assumption, this flow is not a source of uncertainty in T -SHA's dynamics: Passengers at $\star t$ either want to leave T -SHA through t , or they do not; which latter choice only depends on the specification of their trip profiles. Moreover, t is not prone to any technical failure that might affect its passenger throughput.

We map the network outflow through any exit corridor $t \in T_{xc}^{(s)}$ at any time $\tau \geq 0$ in any station $s \in \mathcal{S}$ to the vector $\phi_{xc}^{(s,t)}(\tau)$, with

$$\phi_{xc}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow (\mathbb{R}_{\geq 0})^{|\mathcal{Y}|}.$$

This vector captures the flow's magnitude

$$\phi_{xc,m}^{(s,t)}(\tau) := \sum_{i \in \mathcal{Y}} \phi_{xc}^{(s,t)}(\tau)[i], \quad (3.15)$$

with

$$\phi_{xc,m}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0},$$

and its composition w.r.t. the passengers' different trip profiles: $\phi_{xc}^{(s,t)}(\tau)[i]$ gives the number of passengers per second, who pass through t at τ according to trip profile $i \in \mathcal{Y}$.

Flow Properties. Independently of the particular use case, we demand that the network outflow through any exit corridor $t \in T_{xc}^{(s)}$ in any station $s \in \mathcal{S}$ of T -SHA is autonomous, in that it is completely defined by the passenger number at $\star t$ at any time $\tau \geq 0$.

Definition 48 (Autonomous Network Outflow) *A network outflow through some exit corridor $t \in T_{xc}^{(s)}$ in some station $s \in \mathcal{S}$ of T -SHA is autonomous iff*

$$\phi_{xc}^{(s,t)}(\tau) = f\left(M^{(s,\star t)}(\tau)\right)$$

for time $\tau \geq 0$.

Thus, we could also write

$$\phi_{xc}^{(s,t)}(M^{(s,*t)}(\tau))$$

instead of $\phi_{tc}^{(s,t)}(\tau)$ for time $\tau \geq 0$. We moreover demand that the network outflow through t is demand-sensitive, in that its i -th component $\phi_{xc}^{(s,t)}(\tau)[i]$, for trip profile $i \in \mathcal{Y}$, is zero at any time $\tau \geq 0$ if the number

$$D_{xc}^{(s,t)}(\tau)[i] := R^{(s,t)}[i, \cdot] M^{(s,*t)}(\tau), \quad (3.16)$$

with

$$D_{xc}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow \mathcal{M}^{(s,*t)},$$

of passengers at $*t$, who want to leave T -SHA from $*t$ via t so as to join T -SHA's exterior with i , is zero. In this context, note that - by model assumption - T -SHA's exterior can absorb infinitely many passengers.

Definition 49 (Demand-sensitive Network Outflow) *A network outflow through some exit corridor $t \in T_{xc}^{(s)}$ in some station $s \in \mathcal{S}$ of T -SHA is demand-sensitive iff*

$$D_{xc}^{(s,t)}(\tau)[i] = 0$$

implies

$$\phi_{xc}^{(s,t)}(\tau)[i] = 0$$

for all trip profiles $i \in \mathcal{Y}$ at any time $\tau \geq 0$.

Ansatz Function. Assume that the network outflow through any exit corridor $t \in T_{tc}^{(s)}$ in any station $s \in \mathcal{S}$ is fair, in that it does not privilege any trip profile.

Definition 50 (Fair Network Outflow) *A network outflow through some exit corridor $t \in T_{xc}^{(s)}$ in some station $s \in \mathcal{S}$ of T -SHA is fair iff*

$$D_{xc}^{(s,t)}(\tau)[i] > 0$$

and

$$D_{xc}^{(s,t)}(\tau)[j] > 0$$

implies

$$\frac{\phi_{xc}^{(s,t)}(\tau)[i]}{\phi_{xc}^{(s,t)}(\tau)[j]} = \frac{D_{xc}^{(s,t)}(\tau)[i]}{D_{xc}^{(s,t)}(\tau)[j]} \quad (3.17)$$

for any pair of two trip profiles $i, j \in \mathcal{Y}$ at any time $\tau \geq 0$.

We then do only have to specify its magnitude: Introduce the cumulative number

$$d_{xc}^{(s,t)}(\tau) := \sum_{i \in \mathcal{Y}} D_{xc}^{(s,t)}(\tau)[i],$$

with

$$d_{xc}^{(s,t)}(\tau) : \bigcup_{s' \in \mathcal{S}} T_{cxc,s'} \times \mathbb{R}_{\geq 0} \rightarrow [0, \lambda_c^{(s,*t)}]$$

of passengers at $\star t$, who want to leave T -SHA through t . Next define

$$\phi_{xc}^{(s,t)}(\tau) := 0$$

iff

$$d_{xc}^{(s,t)}(\tau) = 0.$$

Otherwise, define

$$\phi_{xc}^{(s,t)}(\tau)[i] := \frac{D_{xc}^{(s,t)}(\tau)[i]}{d_{xc}^{(s,t)}(\tau)} \phi_{xc,m}^{(s,t)}(\tau). \quad (3.18)$$

Without providing any empirical justification here, it should be possible to specify the magnitude $\phi_{xc,m}^{(s,t)}$ of $\phi_{xc}^{(s,t)}$ therein, as a function of two normalized parameters only: (i) the total number of passengers at $\star t$ who want to leave T -SHA via t related to the capacity of $\star t$, and (ii) the cumulative passenger number at $\star t$ related to the capacity of $\star t$; which implies

$$\phi_{xc,m}^{(s,t)}(\tau) := f\left(\frac{d_{xc}^{(s,t)}(\tau)}{\lambda_c^{(s,\star t)}}, \frac{m^{(s,\star t)}(\tau)}{\lambda_c^{(s,\star t)}}\right), \forall \tau \geq 0. \quad (3.19)$$

The use of the first normalized parameter is obvious: Beside the fact that no demand implies no flow, it can be used to capture more complicated observations such as saturations that occur when too many passengers try to proceed in the same direction at the same time; which impedes their free walking speeds. The second normalized parameter could be used to capture the following observation: It takes more time for the passengers at $\star t$ to struggle their way to t when $\star t$ is more crowded.

Transfer Flows

The specification of transfer flows is similar to the specification of network outflows: We do not consider the flow from a gathering point in a station to the exterior of the considered network, but the flow of passengers between two (capacity-limited) gathering points which are connected by some transfer corridor in the same station. By model assumption, all passengers disappear once they enter the network's exterior. They thus cannot block any new arriving passengers. This property of course does not hold for the passengers at some gather point who might block other transferring passengers. Having this mind, the specification of any transfer flow reads as follows: A transfer flow in T -SHA is any passenger flow from a gathering point $\star t$ to another gathering point t^\star within the same station $s \in \mathcal{S}$ through some transfer corridor $t \in T_{tc}^{(s)}$. By assumption, this transfer flow is not a source of uncertainty in T -SHA's dynamics: Passengers at $\star t$ either want to leave T -SHA through t , or they do not; which latter choice only depends on the specification of their trip profiles. Moreover, we also assume that t is not prone to any technical failure which might affect its passenger throughput.

We map the transfer flow through any transfer corridor $t \in T_{tc}^{(s)}$ at any time $\tau \geq 0$ in any station $s \in \mathcal{S}$ to the vector $\phi_{tc}^{(s,t)}(\tau)$, with

$$\phi_{tc}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow (\mathbb{R}_{\geq 0})^{|\mathcal{Y}|}.$$

This vector captures the flow's magnitude

$$\phi_{tc,m}^{(s,t)}(\tau) := \sum_{i \in \mathcal{Y}} \phi_{xc}^{(s,t)}(\tau)[i], \quad (3.20)$$

with

$$\phi_{tc,m}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0},$$

and its composition w.r.t. the passengers' different trip profiles: $\phi_{tc}^{(s,t)}(\tau)[i]$ gives the number of passengers per second, who pass through t at τ according to trip profile $i \in \mathcal{Y}$.

Flow Properties. Independently of the particular use case, we demand that the network outflow through any exit corridor $t \in T_{tc}^{(s)}$ in any station $s \in \mathcal{S}$ of T -SHA is autonomous, in that it is completely defined by the Passenger Number at $\star t$ at any time $\tau \geq 0$.

Definition 51 (Autonomous Transfer Flow) *A passenger flow from a gathering point $\star t$ to another gathering point t^\star in some station $s \in \mathcal{S}$ through some corridor $t \in T_{tc}^{(s)}$ is autonomous iff*

$$\phi_{tc}^{(s,t)}(\tau) = f\left(M^{(s,\star t)}(\tau), M^{(s,t^\star)}(\tau)\right)$$

at any time $\tau \geq 0$.

Thus, instead of writing $\phi_{tc}^{(s,t)}(\tau)$ for time $\tau \geq 0$, we can also write

$$\phi_{tc}^{(s,t)}\left(M^{(s,\star t)}(\tau), M^{(s,t^\star)}(\tau)\right).$$

We moreover demand that the transfer flow through t is demand-sensitive, in that its i -th component $\phi_{tc}^{(s,t)}(\tau)[i]$, for trip profile $i \in \mathcal{Y}$, is zero at any time $\tau \geq 0$ if the number

$$D_{tc}^{(s,t)}(\tau)[i] := R^{(s,t)}[i, \cdot] M^{(s,\star t)}(\tau), \quad (3.21)$$

with

$$D_{tc}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow \mathcal{M}^{(s,\star t)},$$

of passengers at $\star t$, who want to take t from $\star t$ so as to join t^\star with i , is zero.

Definition 52 (Demand-sensitive Transfer Flow) *A passenger flow from a gathering point $\star t$ to another gathering point t^\star through some corridor $t \in T_{tc}^{(s)}$ in some station $s \in \mathcal{S}$ is demand-sensitive iff*

$$D_{tc}^{(s,t)}(\tau)[i] = 0$$

implies

$$\phi_{tc}^{(s,t)}(\tau)[i] = 0$$

for trip profile $i \in \mathcal{Y}$ at any time $\tau \geq 0$.

Finally, we demand that every transfer flow is capacity-sensitive, in that its magnitude is zero if t^* is full.

Definition 53 (Capacity-sensitive Transfer Flow) *A passenger flow from a gathering point *t to another gathering point t^* through some corridor $t \in T_{tc}^{(s)}$ in some station $s \in \mathcal{S}$ is capacity-sensitive iff*

$$m^{(s,t^*)}(\tau) = \lambda_c^{(s,t^*)}(\tau)$$

implies

$$\phi_{tc}^{(s,t)}(\tau) = 0$$

at any time $\tau \geq 0$.

Ansatz Function. Assume that any transfer flow from *t to t^* through some corridor $t \in T_{tc}^{(s)}$ in some station $s \in \mathcal{S}$ is fair, in that it does not privilege any trip profile.

Definition 54 (Fair Transfer Flow) *A passenger flow from a gathering point *t to another gathering point t^* through some corridor $t \in T_{tc}^{(s)}$ in some station $s \in \mathcal{S}$ is fair iff*

$$D_{tc}^{(s,t)}(\tau)[i] > 0$$

and

$$D_{tc}^{(s,t)}(\tau)[j] > 0$$

implies

$$\frac{\phi_{tc}^{(s,t)}(\tau)[i]}{\phi_{tc}^{(s,t)}(\tau)[j]} = \frac{D_{tc}^{(s,t)}(\tau)[i]}{D_{tc}^{(s,t)}(\tau)[j]} \quad (3.22)$$

for any pair of two trip profiles $i, j \in \mathcal{Y}$ at any time $\tau \geq 0$.

This assumption implies that we only have to specify the transfer flow's magnitude: Define

$$\phi_{tc,m}^{(s,t)}(\tau) := 0$$

iff the cumulative number

$$d_{tc}^{(s,t)}(\tau) := \sum_{i \in \mathcal{Y}} D_{tc}^{(s,t)}(\tau)[i], \quad (3.23)$$

with

$$d_{tc}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow [0, \lambda_c^{(s,t^*)}],$$

of passengers at *t , who want to transfer from *t to t^* via t at some time $\tau \geq 0$, is zero. Otherwise, define

$$\phi_{tc}^{(s,t)}(\tau)[i] := \frac{D_{tc}^{(s,t)}(\tau)[i]}{d_{tc}^{(s,t)}(\tau)} \phi_{tc,m}^{(s,t)}(\tau) \quad (3.24)$$

for trip profile $i \in \mathcal{Y}$. Now distinguish two different situations:

1. The transfer flow through t is not confronted with an oppositely directed transfer flow from t^* to *t through some other corridor connecting t^* to *t . The magnitude of this passenger flow should thus be completely deducible from three normalized parameters at any time $\tau \geq 0$: (i) the total number of passengers at *t who want to transfer to t^* via t related to the capacity of *t ; (ii) the cumulative passenger number at *t related to the capacity of *t , which can be used to capture e.g. the observation that it is more complicated for the passengers at *t to transfer to t^* when it is more crowded at *t ; and (iii) the cumulative passenger number at t^* related to the capacity of t^* which is needed to account for the free capacity of t^* , and which in addition to that can be used to capture e.g. the observation that it is more complicated and thus takes more time for the passengers to go from *t to t^* when t^* is more crowded.

$$\phi_{\text{tc,m}}^{(s,t)}(\tau) := f\left(\frac{d_{\text{tc}}^{(s,t)}(\tau)}{\lambda_c^{(s,^*t)}}, \frac{m^{(s,^*t)}(\tau)}{\lambda_c^{(s,^*t)}}, \frac{m^{(s,t^*)}(\tau)}{\lambda_c^{(s,t^*)}}\right). \quad (3.25)$$

2. The transfer flow through t is confronted with an oppositely directed transfer flow from t^* to *t through some other corridor t' connecting t^* to *t ; which implies $^*t = (t')^*$ and $t^* = ^*(t')$. Compared to the first situation, we thus might have to account for another normalized parameter: the total number of passengers at t^* who want to transfer to *t via t' , related to the capacity of t^* .

$$\phi_{\text{tc,m}}^{(s,t)}(\tau) := f\left(\frac{d_{\text{tc}}^{(s,^*t)}(\tau)}{\lambda_c^{(s,^*t)}}, \frac{m^{(s,^*t)}(\tau)}{\lambda_c^{(s,^*t)}}, \frac{m^{(s,t^*)}(\tau)}{\lambda_c^{(s,t^*)}}, \frac{d_{\text{tc}}^{(s,t')}(\tau)}{\lambda_c^{(s,t^*)}}\right). \quad (3.26)$$

Boarding & Alighting Flows

Boarding and alighting flows are very similar to transfer flows in terms of their dynamics. A boarding flow is any passenger flow from a platform in some station (refer to Def. 25 on p. 50 for the meaning of “platform”) to a vehicle stopped in front (and not to another gathering point). Similarly, an alighting flow is a passenger flow originating on-board some stopped vehicle (and not at some gathering point) to a nearby platform. Where the gathering point in a station has a limited capacity, so does every stopped vehicle. Accordingly, where a passenger’s trip profile defines whether she wants to transfer from some gathering to another, so does this passenger’s trip profile define whether she wants to board some vehicle (which of course depends on the vehicle’s mission; cf. the specification of all trip profiles in Sec. 3.4.1 on p. 57ff.) or alight from it (which does not depend on its mission).

Because of this big similarity of both types of passenger flows, i.e. all transfer flows on the one hand, and all boarding & alighting flows on the other hand, we will copy and paste all equations including the ansatz functions from the previous Sec. 3.5.3 here. Everything the reader has to do, is to (i) replace the subscript “tc” in all variables by “bc” for boarding flows and by “ac” for alighting flows. Moreover s/he has to account for the mode $q \in \mathcal{Q}$ of our SHA model in the subscripts of

all variables since it is this mode which defines which vehicle is stopped at which waypoint together with its mission.

Note that the boarding & alighting process of passengers is of major interest to the operator of almost any transportation mode. This circumstance is reflected in the published experiments which were conducted with the goal to e.g. analyse the flow rates between the stopped vehicles and the station platforms as a function of many different parameters. See e.g. [Rowe 2012], where the authors thereof conducted a so-called psycho physical experiment to study the impact of the door width, the platform height, and some other parameters on the passenger flow rate into and out of a stopped commuter train. Of course, we can reuse the results of these experiments for the parametrization of our model's ansatz functions.

Network Inflows

Network inflows are very different to all other types of passenger flows which we have already introduced and discussed in the previous sections: Any passenger flow from the exterior of the considered transportation network into some gathering point t^* in some station $s \in \mathcal{S}$ through some entrance corridor $t \in T_{ec}^{(s)}$ is a network inflow. By assumption, and thus as opposed to all other types of considered passenger flows, this inflow might introduce some uncertainty into the continuous-time passenger flow-dynamics of T -SHA. In fact, we do not know at which time passengers arrive at the considered network. Neither do we know their trip profiles. We map our ignorance of this uncertainty to two functions specified for mode $q \in \mathcal{Q}$: To put it carefully, the i -th component of the first function,

$$\phi_{ec}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow (\mathbb{R}_{\geq 0})^{|\mathcal{Y}|},$$

with $i \in \mathcal{Y}$, defines the average number $\phi_{ec}^{(s,t)}(\tau)$ of passengers per second who enter T -SHA through t at some time $\tau \geq 0$ from with trip profile i in some short time interval. The i -th component of the second function,

$$\delta_{ec}^{(s,t)} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{|\mathcal{Y}|},$$

captures the variance $\delta_{ec}^{(s,t)}$ around this average in the same very short time interval. Now more mathematically speaking, $\phi_{ec}^{(s,t)}(\tau)$ defines the impact of the passenger flow through t in s on the drift of p 's passenger numbers, where $p := t^*$; and $\delta_{ec}^{(s,t)}$ defines the flow's invariant impact on the diffusion of p 's passenger numbers.

Flow Properties. Independently of its particular specification, we demand that the network inflow through any $t \in T_{ec}^{(s)}$ from the network's exterior to t^* in any station $s \in \mathcal{S}$ is autonomous, in that its drift term is a function of the passenger number at t^* if it depends on T -SHA's hybrid state at all.

Definition 55 (Autonomous Transfer Flow) A network inflow into some gathering point t^* in some station $s \in \mathcal{S}$ through some corridor $t \in T_{\text{ec}}^{(s)}$ is autonomous iff

$$\phi_{\text{ec}}^{(s,t)}(\tau) = f\left(M^{(s,t^*)}(\tau)\right)$$

at any time $\tau \geq 0$.

Moreover, this network inflow through t must be capacity-sensitive, in that its drift term is zero if t^* is full.

Definition 56 (Capacity-Sensitive Network Inflow) A passenger flow from TN 's exterior to a gathering point t^* through some corridor $t \in T_{\text{ec}}^{(s)}$ in some station $s \in \mathcal{S}$ is capacity-sensitive iff $\bar{m}^{(s,t^*)}(\tau) = 0$ implies $\phi_{\text{ec}}^{(s,t)}(\tau) = 0$ at any time $\tau \geq 0$.

Specification of All Balance Equations

By assumption, all passenger flows are autonomous, which implies that the system of Itô-SDEs from Eqn. 3.27 on p. 75 can be replaced by a set of lower-dimensional decoupled Itô-SDEs: Remember that Eqn. 3.27 stacks the balance equations for all passenger numbers in some mode $q \in \mathcal{Q}$ into one single (high-dimensional) system. However, the imposition of Def. 19 on p. 47 on the specification of TN 's infrastructure implies that stopped vehicles cannot establish bridges between stations in any $q \in \mathcal{Q}$ in that the passengers could board some vehicle/alight from some vehicle from/to the platforms of two or more different stations. Every passenger flow into, circulating within, or out of some station can thus only impact/depend on the passenger numbers of all gathering points in that station as well as the passenger numbers of all vehicles docked to it. This important property allows us to replace Eqn. 3.27 by

$$dX^{(s,q)}(\tau) := \alpha^{(s,q)}(\tau) d\tau + \beta^{(s,q)}(\tau) dW(\tau), \quad (3.27)$$

for station $s \in \mathcal{S}$ in any mode $q \in \mathcal{Q}$, where

$$X^{(s,q)} := \left[\overbrace{M^{(s,1)}; M^{(s,2)}; \dots; M^{(s,|P_{\text{gp}}^{(s)}|)}}^{\text{Gathering points in } s}; \overbrace{U^{(v_1)}; U^{(v_2)}; \dots; U^{(v_{n^{(s,q)}})}}^{\text{Vehicles docked to } s} \right] \quad (3.28)$$

is a $k^{(s,q)}$ -dimensional column vector, with

$$k^{(s,q)} := \left(|P_{\text{gp}}^{(s)}| + n^{(s,q)} \right) |\mathcal{Y}|. \quad (3.29)$$

This column vector groups together $m^{(s)} + n^{(s,q)}$ balance equations: For every gathering point in s and for every vehicle docked to s in q , we have to set up one balance equation which defines the time evolution of the corresponding passenger number in q . In this context, we say that some vehicle $v \in \mathcal{V}$ is docked to s in q iff according to q (i) v is stopped at some waypoint, and (ii) $v \in \mathcal{V}_{\text{db}}^{(q,s,t_1)}$, for $t_1 \in T_{\text{bc}}^{(s)}$, with

$$\mathcal{V}_{\text{db}}^{(q,s,t_1)} := \left\{ \bar{v} \in \mathcal{V}_s^{(q)} : (t_1)^* = \lambda_p(\bar{v}) \right\}, \quad (3.30)$$

which implies that passengers can board v from the platform $\star(t_1)$ in s via the boarding corridor t_1 , or $v \in \mathcal{V}_{\text{da}}^{(q,s,t_2)}$, for $t_2 \in T_{\text{ac}}^{(s)}$, with

$$\mathcal{V}_{\text{da}}^{(q,s,t_2)} := \{\bar{v} \in \mathcal{V}_s^{(q)} : \star(t_2) = \lambda_p(\bar{v})\}, \quad (3.31)$$

which implies that passengers on-board v can alight from it to the platform $(t_2)^\star$ in s via the alighting corridor t_2 . It then follows that v is docked to s in q iff $v \in \mathcal{V}_{\text{do}}^{(q,s)}$, with

$$\mathcal{V}_{\text{do}}^{(q,s)} := \bigcup_{t_1 \in T_{\text{bc}}^{(s)}} \mathcal{V}_{\text{db}}^{(q,s,t_1)} \cup \bigcup_{t_2 \in T_{\text{ac}}^{(s)}} \mathcal{V}_{\text{da}}^{(q,s,t_2)}. \quad (3.32)$$

Thus, $\mathcal{V}_{\text{do}}^{(q,s)} = \{v_1, v_2, \dots, v_{n^{(s,q)}}\}$ in Eqn. 3.28 above, and $n^{(s,q)} = |\mathcal{V}_{\text{do}}^{(q,s)}|$ in (3.29). Moreover, we demand $v_i < v_j$ for $i < j$ to avoid any ambiguity.

The balance equation set up for the passenger number of one particular gathering point in s or vehicle docked to s in q , relates this passenger number to the passenger flows into and out of it with the help of some routing matrices $R^{(\cdot)}$ that we have already introduced in Sec. 3.4.2. All this according to the informal law

$$\Delta \text{Passenger load of } X_i := \sum R^{(\cdot)} \times \text{Flows into } X_i - \sum \text{Flows out of } X_i,$$

where every X_i denotes the considered passenger number.

Gathering Point in a Station. The balance equation that defines the time evolution of the passenger number at any gathering point $p \in P_{\text{gp}}^{(s)}$ in any station $s \in \mathcal{S}$ at any time $\tau \geq 0$, when T -SHA is in mode $q \in \mathcal{Q}$, is

$$\begin{aligned} dM^{(s,p)}(\tau) := & \Delta M_{\text{ec}}^{(s,p)}(\tau) + \Delta M_{\text{tc},i}^{(s,p)}(\tau) + \Delta M_{\text{tc},o}^{(s,p)}(\tau) + \\ & \Delta M_{\text{bc}}^{(s,p,q)}(\tau) + \Delta M_{\text{ac}}^{(s,p,q)}(\tau) + \Delta M_{\text{xc}}^{(s,p)}(\tau). \end{aligned} \quad (3.33)$$

Therein,

- the term

$$\Delta M_{\text{ec}}^{(s,p)}(\tau) := \sum_{t \in \bullet p \cap T_{\text{ec}}^{(s)}} R^{(s,t)} \left[\phi_{\text{ec}}^{(s,t)}(\tau) d\tau + \delta_{\text{ec}}^{(s,t)}(\tau) dW(\tau) \right] \quad (3.34)$$

captures the impact of all passenger flows into p from TN 's exterior via some entrance corridor $t \in T_{\text{ec}}^{(s)}$ on $M^{(s,p)}(\tau + d\tau)$,

- the term

$$\Delta M_{\text{tc},i}^{(s,p)}(\tau) := \sum_{t \in \bullet p \cap T_{\text{tc}}^{(s)}} R^{(s,t)} \phi_{\text{tc}}^{(s,t)}(\tau) d\tau \quad (3.35)$$

captures the impact of all passenger flows into p from $\star t$ via some transfer corridor $t \in \bullet p \cap T_{\text{tc}}^{(s)}$ on $M^{(s,p)}(\tau + d\tau)$,

- the term

$$\Delta M_{tc,o}^{(s,p)}(\tau) := - \sum_{t \in p^\bullet \cap T_{tc}^{(s)}} \phi_{tc}^{(s,t)}(\tau) d\tau \quad (3.36)$$

captures the impact of all passenger flows out of p to t^\star via some transfer corridor $t \in p^\bullet \cap T_{tc}^{(s)}$ on $M^{(s,p)}(\tau + d\tau)$,

- the term

$$\Delta M_{bc}^{(s,p,q)}(\tau) := \sum_{t \in p^\bullet \cap T_{bc}^{(s)}} \sum_{v \in \mathcal{V}_{db}^{(q,s,t)}} \phi_{bc}^{(s,t,\lambda_m^{(q,v)})}(\tau) d\tau \quad (3.37)$$

captures the impact of all passenger flows from p to a vehicle stopped at t^\star via some boarding corridor $t \in p^\bullet \cap T_{bc}^{(s)}$ on $M^{(s,p)}(\tau + d\tau)$,

- the term

$$\Delta M_{ac}^{(s,p,q)}(\tau) := \sum_{t \in p^\bullet \cap T_{ac}^{(s)}} \sum_{v \in \mathcal{V}_{db}^{(q,s,t)}} \phi_{ac}^{(s,t,\lambda_m^{(q,v)})}(\tau) d\tau \quad (3.38)$$

captures the impact of all passenger flows into p from a vehicle stopped at t^\star via some alighting corridor $t \in p^\bullet \cap T_{ac}^{(s)}$ on $M^{(s,p)}(\tau + d\tau)$,

- the term

$$\Delta M_{xc}^{(s,p)}(\tau) := - \sum_{t \in p^\bullet \cap T_{xc}^{(s)}} \phi_{xc}^{(s,t)}(\tau) d\tau \quad (3.39)$$

captures the impact of all passenger flows from p to TN 's exterior via some exit corridor $t \in p^\bullet \cap T_{xc}^{(q)}$ on $M^{(s,p)}(\tau + d\tau)$,

where $d\tau$ denotes the infinitesimal time interval.

Stopped Vehicle. The (diffusion-free) balance equation that defines the time evolution of the passenger number on-board any vehicle $v \in \mathcal{V}_{do}^{(q,s)}$ at any time $\tau \geq 0$, which is docked to some station $s \in \mathcal{S}$ when T -SHA is in mode $q \in \mathcal{Q}$, is

$$dU^{(v)}(\tau) := \sum_{t_1 \in T_{bf}^{(q,s,v)}} R^{(s,t_1,\lambda_m^{(q,v)})} \underbrace{\phi_{bc}^{(s,t_1,\lambda_m^{(q,v)})}(\tau)}_{\text{Passenger flow into } v \text{ via } t_1 \text{ in } s} d\tau - \sum_{t_2 \in T_{af}^{(q,s,v)}} \underbrace{\phi_{ac}^{(s,t_2,\lambda_m^{(q,v)})}(\tau)}_{\text{Passenger flow out of } v \text{ via } t_2 \text{ in } s} d\tau, \quad (3.40)$$

where

- the set

$$T_{bf}^{(q,s,v)} := \{t' \in T_{bc}^{(s)} : (t')^\star = \lambda_m^{(q,v)}\}$$

comprises all those boarding corridors in station $s \in \mathcal{S}$ through which passengers can board v in q , and

- the set

$$T_{\text{af}}^{(q,s,v)} := \{t' \in T_{\text{ac}}^{(s)} : \star(t') = \lambda_m^{(q,v)}\}$$

comprises all those alighting corridors in station $s \in \mathcal{S}$ through which passengers can alight from v in q .

The passenger numbers of all other vehicles

$$v' \in \mathcal{V} \setminus \bigcup_{s' \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s')}$$

remain unchanged in q .

Specification of All Mode Transitions

Some Preliminary Considerations

From Atomic Actions to Mode Transitions. Probably no human modeller wants to define all possible mode transitions for *T-SHA* and constraints imposed on them for any non-trivial use case by hand. Fortunately there is no need to do so if we make use of the four simple assumptions A.1 to A.4 introduced below; something which we make here and which we will make in the rest of this report/our implementation: Behind every mode transition there is a subset of vehicles, which perform some atomic actions at the same time. These actions are atomic in a sense that any action $a \in \mathcal{A}$ performed by some vehicle $v \in \mathcal{V}$

- A.1: changes the position and/or the internal state of v but leaves the positions and the internal states of all other vehicles unchanged,
- A.2: may be conditioned on the passenger number on-board v if it is conditioned on the passenger number on-board any vehicle,
- A.3: may be conditioned on the valuation of the clock associated with v 's operation if it is conditioned on the valuation of any clock,
- A.4: may be conditioned on the passenger number at some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ iff v is docked to s and p is a platform from which passengers can board v via some boarding corridor in s and/or to which passengers can alight from v via some alighting corridor in s .

With these four assumptions in mind, all that we have to do then is to properly define the set of all possible actions only, whose choice obviously depends on the considered use case. This proper definition involves the rigorous specification of all preconditions for their performance, and the specification of their impacts on the vehicles' positions, driving conditions, and operational states upon performance. Assuming this to be done (refer to Sec. 3.7.2 to Sec. 3.7.4 below), we can take any mode of *T-SHA* such as its initial mode q_0 , and it is then this mode which (uniquely) defines which subset of actions might be performed by which subset of

vehicles at the same time; some combinatorial problem whose solution produces some - possibly empty - set of outgoing mode transitions together with constraints for their occurrences: Every mode transition corresponds to the fact that some subset of vehicles perform some actions; every vehicle in this subset one atomic action. The superposition of all actions' preconditions then defines the compound clock constraints and compound passenger number constraints for the considered mode transition. Similarly, the superposition of all actions' impacts defines *T-SHA* new mode.

Mutual Exclusion of Some Atomic Actions. We can group all atomic actions into either passenger number-dependent actions (elaborated in Sec. 3.7.3 on p. 83ff.) or passenger number-independent actions (elaborated in Sec. 3.7.4 on p. 89ff.), where every passenger number-dependent action - as its name suggests - must be conditioned on at least one passenger number at some gathering point or on-board some vehicle but every passenger number-independent action must not. This re-grouping of all actions into two disjoint unions does not seem to be a big deal. However, remember that all passenger numbers are continuous random variables whose evolutions are defined by stochastic differential balance equations. This dynamical property implies that passenger number-dependent actions do not occur at isolated points in time with some positive marginal probability, which in turn implies that two or more passenger number-dependent actions will not occur at the same time with some positive marginal probability.

Mode Graph. All possible sequential evolutions of *T-SHA*'s mode starting from its initial mode q_0 at time $\tau = 0$, with $\tau \geq 0$, can be represented in an edge-labelled directed graph; cf. 3.7.1 below: Every node in this (mode) graph corresponds to some mode $q \in \mathcal{Q}$ of *T-SHA*. Every edge connecting two modes, say mode $q_1 \in \mathcal{Q}$ to mode $q_2 \in \mathcal{Q}$, corresponds to the possibility that *T-SHA* transitions from q_1 to q_2 . Every label written next to some edge defines all constraints that are respected upon the corresponding mode transition occurs.

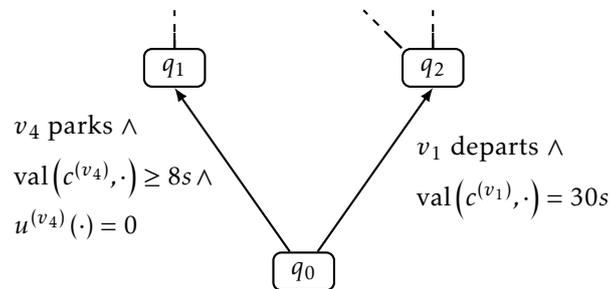


Figure 3.7.1: Extract of some possible mode graph for *T-SHA*, where the parking of vehicle v_4 illustrates the performance of a passenger number-dependent action, and the departure of vehicle v_1 illustrates the performance of a passenger number-independent action

Note that we will not provide some algorithm for its computation here. We only mention it, because it creates some important link to the existing literature on hybrid automata. In addition to this link, we believe that knowing the existence of this (mode) graph is some useful means in better understanding the continuous-time dynamics of *T-SHA* (see the next following paragraphs), and similarities/differences of this dynamics with the discrete-time dynamics that we will enforce in/employ for the computation of any forecast in the next chapter.

Cycles in Mode Graph. *T-SHA*'s mode graph may have cycles. To see that this is true, assume that at the beginning of some considered use case all vehicles are parked in depots, where every vehicle has its proper parking lot (= waypoint in some transportation grid). Some subset of these vehicles are then dispatched one after the other, and upon completion of their runs at the end of a day they are all parked in their proper parking lots once again.

Node Explosion of Mode Graph. Every node in *T-SHA*'s mode graph, say graph G_m , corresponds to some mode $q \in \mathcal{Q}$. By construction, \mathcal{Q} is finite which implies that G_m 's number of nodes is limited from above by $|\mathcal{Q}|$. However, for almost all conceivable use cases this upper limit can be considered as to be too conservative since it neglects the fact that only some *unfoldings* of all vehicles' positions, driving conditions, and operational states are feasible starting from *T-SHA*'s initial mode q_0 . A less conservative upper limit can be computed as follows: Starting from q_0 , unfold for every vehicle $v \in \mathcal{V}$ the path $\psi^{(v,h)}$: $\psi^{(v,h)}$ captures the possible/-maximum achievable evolution of v 's discrete state in some fixed time horizon, say $h > 0$, while neglecting the presence of all other vehicles; where v 's discrete state is defined by its position in some transportation grid (= waypoint), as well as its driving condition and operational state at this position. Say that this path has length $k^{(v,h)} \in \mathbb{N}_{>0}$. It then follows that G_m cannot have more than

$$\prod_{v \in \mathcal{V}} k^{(v,h)}$$

nodes; which upper limit is likely to explode with increasing h !

Missing Zeno Behaviour. In the analysis of continuous-time hybrid (dynamical) automata, zeno behaviour refers to the observation that some (hybrid) automaton changes infinitely many times its discrete state in finite time. Can this behaviour also be observed in any execution of *T-SHA*? The answer is no if (= non-exclusive if) all above four assumptions are respected during the specification of all atomic actions/corresponding mode transitions: We associate with the operation of every vehicle $v \in \mathcal{V}$ some clock $c^{(v)}$. More specifically, we use the valuation of this clock to measure/model the time which must elapse before the impact of any (physical) action $a \in \mathcal{A}$ performed by some v can take effect; in that we map this time consumption to a constraint with a positive time threshold for the valuation of $c^{(v)}$.

Upon mode transition caused by v performing a , $c^{(v)}$ is reset to zero. Thus, no more than one mode transition induced by at maximum $|\mathcal{V}|$ vehicles which simultaneously perform some actions can occur in some vanishing time interval.

Useful Definitions for the Specification of All Atomic Actions

The following three definitions tell us whether some vehicle $v \in \mathcal{V}$ is supposed to park, stop, or depart next when T -SHA is in some mode $q \in \mathcal{Q}$; given v 's current position, driving condition, and operational state in q .

Definition 57 (Planned Vehicle Parking) *Some vehicle $v \in \mathcal{V}_s^{(q)}$ which is stopped at some waypoint $p := \lambda_p^{(q,v)}$ is supposed to park at p when T -SHA is in some mode $q \in \mathcal{Q}$ iff*

- *its mission $w := \lambda_m^{(q,v)}$ is identical to the last mission (refer to Def. 37 on p. 56 for the meaning of “last mission”) in its run $\lambda_r^{(q,v)}$, and*
- *p is identical to the last waypoint (refer to Def. 31 on p. 53 for the meaning of “last waypoint”) in w .*

Definition 58 (Planned Vehicle Stop) *Some vehicle $v \in \mathcal{V}_d^{(q)}$ which moves towards some waypoint $p := \lambda_p^{(q,v)}$ when T -SHA is in some mode $q \in \mathcal{Q}$ is supposed to stop at p iff*

- *its mission $w := \lambda_m^{(q,v)}$ defines some positive minimum dwell time $\lambda_-^{(w)}(p)$ at p , or*
- *p is identical to the last waypoint (refer to Def. 31 on p. 53 for the meaning of “last waypoint”) in w .*

Definition 59 (Planned Vehicle Departure) *Some vehicle $v \in \mathcal{V}_s^{(q)}$ which is stopped at some waypoint $\lambda_p^{(q,v)}$ when T -SHA is in some mode $q \in \mathcal{Q}$ is supposed to depart from p iff*

- *p is not identical to the last waypoint (refer to Def. 31 on p. 53 for the meaning of “last waypoint”) in its mission $\lambda_m^{(q,v)}$, and*
- *v is not supposed to park at p according to Def. 57 on p. 81.*

If v moves towards some waypoint, say p , but is not supposed to stop at p , then we say that v is supposed to skip p ; so as to move towards some other waypoint immediately after its arrival at p .

Definition 60 (Planned Waypoint Skip) *Some vehicle $v \in \mathcal{V}_d^{(q)}$ which moves towards some waypoint $p := \lambda_p^{(q,v)}$ when T -SHA is in some mode $q \in \mathcal{Q}$ is supposed to move to some other waypoint immediately after its arrival at p iff v is not supposed to stop at p according to Def. 58 on p. 81.*

This next waypoint, say p' , is uniquely defined by v 's mission and position in q , given that v is not supposed to change its mission at p .

Definition 61 (Planned Mission Change) *Some vehicle $v \in \mathcal{V}_{io}^{(q)}$ which is either stopped at or moves towards some waypoint $p := \lambda_p^{(q,v)}$ is supposed to change its mission $w := \lambda_m^{(q,v)}$ when T-SHA is in some mode $q \in \mathcal{Q}$ iff*

- w is not identical to the last mission (refer to Def. 37 on p. 56 for the meaning of “last mission”) in its run $\lambda_r^{(q,v)}$, and
- p is identical to the last waypoint (refer to Def. 31 on p. 53 for the meaning of “last waypoint”) in w .

Otherwise, v 's next mission defines p' .

Definition 62 (Next Mission) *The next mission of some vehicle $v \in \mathcal{V}_s^{(q)} \cup \mathcal{V}_d^{(q)}$ which (i) either is stopped at or moves towards some waypoint $\lambda_p^{(q,v)}$ when T-SHA is in some mode $q \in \mathcal{Q}$, and (ii) is supposed to change its mission $w := \lambda_m^{(q,v)}$ according to Def. 61 on p. 82, is the mission $w' \in \mathcal{W}$ iff w' immediately follows w in the specification of its run $z := \lambda_r^{(q,v)}$.*

Definition 63 (Next Position) *The next position of some vehicle $v \in \mathcal{V}_s^{(q)} \cap \mathcal{V}_d^{(q)}$ which (i) is either stopped at or moves towards some waypoint $p := \lambda_p^{(q,v)}$ when T-SHA is in some mode $q \in \mathcal{Q}$, and (ii) is not supposed to park at p according to Def. 57 on p. 81, is the waypoint p' , where p'*

- immediately follows p in the path $\pi^{(w)}$ which is specified by v 's mission $w := \lambda_m^{(w)}(v)$ iff p is not the last waypoint in w (refer to Def. 31 on p. 53 for the meaning of “last waypoint”), or
- is the first waypoint in the mission w' (refer to Def. 30 on p. 52 for the meaning of “first waypoint”) iff w' is v 's next mission (refer to Def. 62 on p. 82 for the meaning of “next mission”).

We introduce three more definitions which we will use for the resolution of all conflicting vehicle movements: A conflict between two vehicles, say between vehicle $v_1 \in \mathcal{V}$ and vehicle $v_2 \in \mathcal{V} \setminus \{v_1\}$ in some transportation grid $g \in \mathcal{G}$, arises iff both vehicles try to access the same waypoint, say waypoint $p \in P_w^{(g)}$, at the same time from two different directions; say v_1 via the track $t_1 \in \bullet p$ and v_2 via the track $t_2 \in \bullet p \setminus \{t_1\}$. Note that v_1 and v_2 cannot do so if p is already blocked by some other vehicle.

Definition 64 (Blocked Waypoint) *A waypoint $p \in P_w^{(g)}$ is blocked in some transportation grid $g \in \mathcal{G}$ by some vehicle $v \in \mathcal{V}^{(g)}$ when T-SHA is in some mode $q \in \mathcal{Q}$ iff $p = \lambda_p^{(q,v)}$.*

Assuming this to be true, we use the enumeration $\lambda_p^{(g)}(\cdot)$ of all tracks in g to resolve this conflict in a deterministic manner; cf. Sec. 3.2.2 on p. 45ff.: v_1 wins the conflict and as a consequence is allowed to access p iff $\lambda_p^{(g)}(t_1) < \lambda_p^{(g)}(t_2)$.

Definition 65 (Priority for Waypoint Access) A vehicle $v \in \mathcal{V}^{(g)} \cap (\mathcal{V}_s^{(q)} \cup \mathcal{V}_d^{(q)})$ which (i) is either stopped at or moves towards some waypoint $p := \lambda_p^{(q,v)}$ in some transportation grid $g \in \mathcal{G}$ when T-SHA is in some mode $q \in \mathcal{Q}$, and (ii) wants to depart to or move towards some other waypoint $p' \in P_w^{(g)} \setminus \{p\}$ via some track $t \in \bullet(p') \cap p^\bullet$ upon its departure from or arrival at p , is $\lambda_p^{(g)}(t)$.

Definition 66 (Higher Priority for Waypoint Access) If two vehicles $v_1, v_2 \in \mathcal{V}^{(g)}$ are operated in the same transportation grid $g \in \mathcal{G}$, and they both want to access the same waypoint $p \in P_w^{(g)}$, say v_1 via the track $t_1 := \bullet p$ and v_2 via the track $t_2 := \bullet p \setminus \{t_1\}$, at the same time, then v_1 is doing so with a higher priority iff $\lambda_p^{(g)}(t_1) < \lambda_p^{(g)}(t_2)$.

Definition 67 (Permission to Access Waypoint) Some vehicle $v \in (\mathcal{V}_s^{(q)} \cup \mathcal{V}_d^{(v)})$ which (i) is either stopped at or moves towards some waypoint $p := \lambda_p^{(q,v)}$ when T-SHA is in some mode $q \in \mathcal{Q}$, and (ii) wants to move towards some other waypoint $p' \in P_w^{(g)} \setminus \{p\}$ upon its departure from or stop at p , is allowed to do so iff

- p' is not blocked by some other vehicle, and
- no other vehicle $v' \in \mathcal{V}_{io}^{(q)} \setminus \{v\}$ wants to access p' with a higher priority.

Specification of Passenger Number-Dependent Actions

Table 3.7 below provides an overview of all passenger number-dependent actions, which we employ in our implementations.

Table 3.7: Overview of considered passenger number-dependent actions (LD actions)

Name	Short Description
LD Late Departure	Some vehicle which is docked to some station departs late once no more passengers want to alight from it
LD Mission Change	Some vehicle executes a new mission upon completion of some previous one
LD On-Time Departure	Some vehicle which is docked to some station departs on time once no more passengers want to alight from and board it
LD Vehicle Parking	Some vehicle which is docked to some station parks once no more passengers want to alight from it

We elaborate conditions for their occurrences and impacts of their occurrences on the mode of T-SHA in alphabetical order next.

LD Late Departure. Some vehicle $v \in \mathcal{V}_{\text{do}}^{(q,s)}$ which is docked to some station $s \in \mathcal{S}$ departs from its position $p := \lambda_p^{(q,v)}$ late at some time $\tau \geq 0$ when $T\text{-SHA}$ is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LD_LATEDEPARTURE}(q, v, k_1, k_2) = \text{TRUE},$$

where

- the function $\text{GUARD_LD_LATEDEPARTURE}(\cdot)$ is defined in Alg. 6 below,
- $k_1 \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v , and
- $k_2 \geq 0$ denotes the number of passengers on-board v at τ who want to alight from v .

Algorithm 2 Specification of the function

$\text{GUARD_LD_LATEDEPARTURE} :$

$$(q, v, k_1, k_2) \in \mathcal{Q} \times \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LD_LATEDEPARTURE( $q, v, k_1, k_2$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $w \leftarrow \lambda_m^{(q,v)}$ 
4:   if
      •  $v$  is supposed to depart from  $p$  when  $T\text{-SHA}$  is in  $q$  according
        to Def. 59 on p. 81,
      •  $v$ 's next position (refer to Def. 63 on p. 82 for the meaning of
        "next position") is not blocked by another vehicle according to
        Def. 64 on p. 82
      •  $\lambda_-^{(w,p)} + \lambda_s^{(w,p)} \left[ \lambda_+^{(w,p)} - \lambda_-^{(w,p)} \right] \leq k_1 \leq \lambda_+^{(w,p)}$ , and
      •  $k_2 = 0$ .
5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then $T\text{-SHA}$ transitions to the mode $q' := \text{DEPART}(q, v)$, where $q' \in \mathcal{Q}$ is computed according to Alg. 3 below.

Algorithm 3 Specification of the function

$$\text{DEPART} : (q, v) \in \mathcal{Q} \times \mathcal{V}_s^{(q)} \rightarrow \mathcal{Q}$$

1: **function** DEPART(q, v)
2: $g \leftarrow$ transportation grid which accommodates v
3: $p \leftarrow \lambda_p^{(q,v)}$
4: $z \leftarrow \lambda_r^{(q,v)}$
5: $w \leftarrow \lambda_m^{(q,v)}$
6: $p' \leftarrow$ Next position of v according to Def. 63 on p. 82
7: $Q_s^{(q')} \leftarrow Q_s^{(q)} \setminus \{(g, p, v, z, w)\}$
8: $Q_d^{(q')} \leftarrow Q_d^{(q)} \cup \{(g, p', v, z, w)\}$
9: **return** $(Q_p^{(q)}, Q_s^{(q')}, Q_d^{(q')}, \mathcal{V}_f^{(q)})$
10: **end function**

LD Misson Change. Some vehicle $v \in \mathcal{V}_{\text{do}}^{(q,s)}$ which is docked to some station $s \in \mathcal{S}$ changes its mission at some time $\tau \geq 0$ when $T\text{-SHA}$ is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LD_MISSIONCHANGE}(q, v, k_1, k_2) = \text{TRUE},$$

where

- the function $\text{GUARD_LD_MISSIONCHANGE}(\cdot)$ is defined in Alg. 4 below,
- $k_1 \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v , and
- $k_2 \geq 0$ denotes the number of passengers on-board v at τ who want to alight from it.

Algorithm 4 Specification of the function

GUARD_LD_MISSIONCHANGE :

$$(q, v, k_1, k_2) \in \mathcal{Q} \times \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

1: **function** GUARD_LD_MISSIONCHANGE(q, v, k_1, k_2)
2: $p \leftarrow \lambda_p^{(q,v)}$
3: $w \leftarrow \lambda_m^{(q,v)}$
4: **if**

- v is supposed to change its mission when T -SHA is in q according to Def. 61 on p. 82,
- $k_1 \geq \lambda_{-}^{(w,p)}$, and
- $k_2 = 0$.

```

5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{CHANGEMISSION}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 5 below.

Algorithm 5 Specification of the function

$$\text{CHANGEMISSION} : (q, v) \in \mathcal{Q} \times \mathcal{V}_s^{(q)} \rightarrow \mathcal{Q}$$

```

1: function CHANGEMISSION( $q, g, v$ )
2:    $g \leftarrow$  transportation grid which accommodates  $v$ 
3:    $p \leftarrow \lambda_p^{(q,v)}$ 
4:    $z \leftarrow \lambda_r^{(q,v)}$ 
5:    $w \leftarrow \lambda_m^{(q,v)}$ 
6:    $w' \leftarrow$  Next mission of  $v$  according to Def. 62 on p. 82
7:    $Q_s^{(q')} \leftarrow (Q_s^{(q)} \setminus \{(g, p, v, z, w)\}) \cup \{(g, p, v, z, w')\}$ 
8:   return ( $Q_p^{(q)}, Q_s^{(q')}, Q_d^{(q)}, \mathcal{V}_f^{(q)}$ )
9: end function

```

LD On-Time Departure. Some vehicle $v \in \mathcal{V}_{\text{do}}^{(q,s)}$ which is docked to some station $s \in \mathcal{S}$ departs from its position $p := \lambda_p^{(q,v)}$ at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LD_ONTIMEDEPARTURE}(q, v, k_1, k_2, k_3) = \text{TRUE},$$

where

- the function $\text{GUARD_LD_ONTIMEDEPARTURE}(\cdot)$ is defined in Alg. 6 below,

- $k_1 \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v ,
- $k_2 \geq 0$ denotes the number of passengers on-board v at τ who want to alight from v , and
- $k_3 \geq 0$ denotes the number of passengers at some platform in s at τ who want to board v .

Algorithm 6 Specification of the function

GUARD_LD_ONTIMEDEPARTURE :

$$(q, v, k_1, k_2, k_3) \in \mathcal{Q} \times \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LD_ONTIMEDEPARTURE( $q, v, k_1, k_2, k_3$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $w \leftarrow \lambda_m^{(q,v)}$ 
4:   if
      •  $v$  is supposed to depart from  $p$  when  $T\text{-SHA}$  is in  $q$  according
        to Def. 59 on p. 81,
      •  $v$ 's next position (refer to Def. 63 on p. 82 for the meaning of
        "next position") is not blocked by another vehicle according to
        Def. 64 on p. 82
      •  $\lambda_-^{(w,p)} \leq k_1 \leq \lambda_-^{(w,p)} + \lambda_s^{(w,p)} [\lambda_+^{(w,p)} - \lambda_-^{(w,p)}]$ , and
      •  $k_2 = 0$ ,
      •  $k_3 = 0$ .
5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then $T\text{-SHA}$ transitions to the mode $q' := \text{DEPART}(q, v)$, where $q' \in \mathcal{Q}$ is computed according to Alg. 3 from p. 85 above.

LD Vehicle Parking. Some vehicle $v \in \mathcal{V}_{\text{do}}^{(q,s)}$ which is docked to some station $s \in \mathcal{S}$ parks at some time $\tau \geq 0$ when $T\text{-SHA}$ is in mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LD_VEHICLEPARKING}(q, v, k) = \text{TRUE},$$

where

- the function $\text{GUARD_LD_VEHICLEPARKING}(\cdot)$ is defined in Alg. 7 below, and
- $k \geq 0$ denotes the number of passengers on-board v at τ .

Algorithm 7 Specification of the function

$\text{GUARD_LD_VEHICLEPARKING} :$

$$(q, v, k) \in \mathcal{Q} \times \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LD_VEHICLEPARKING( $q, v, k$ )
2:   if
      •  $v$  is supposed to park when  $T$ -SHA is in  $q$  according to Def. 57
        on p. 81, and
      •  $k = 0$ 
3:   then
4:     return TRUE
5:   else
6:     return FALSE
7:   end if
8: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{PARK}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 8 below.

Algorithm 8 Specification of the function

$$\text{PARK} : (q, v) \in \mathcal{Q} \times \mathcal{V}_s^{(q)} \rightarrow \mathcal{Q}$$

```

1: function PARK( $q, g, v$ )
2:    $g \leftarrow$  transportation grid which accommodates  $v$ 
3:    $p \leftarrow \lambda_p^{(q,v)}$ 
4:    $z \leftarrow \lambda_I^{(q,v)}$ 
5:    $w \leftarrow \lambda_m^{(q,v)}$ 
6:    $Q_p^{(q')} \leftarrow Q_p^{(q)} \cup \{(g, p, v)\}$ 
7:    $Q_s^{(q')} \leftarrow Q_s^{(q)} \setminus \{(g, p, v, z, w)\}$ 

```

```

8:   return ( $Q_p^{(q')}$ ,  $Q_s^{(q')}$ ,  $Q_d^{(q)}$ ,  $\mathcal{V}_f^{(q)}$ )
9: end function

```

Specification of Passenger Number-Independent Actions

Table 3.8 below provides an overview of all passenger number-independent actions, which we employ in our implementations and which have related passenger number-dependent actions.

Table 3.8: Overview of considered passenger number-independent actions (LI actions), which have some related passenger number-dependent action

Name	Short Description
LI Mission Change	Some vehicle executes a new mission upon completion of some previous one
LI Vehicle Departure	Some stopped vehicle that is not docked to any station for the purpose of boarding or alighting departs
LI Vehicle Parking	Some vehicle that is stopped at a waypoint but not docked to any station parks

In addition to Tab. 3.8, Tab. 3.9 below lists all passenger number-independent actions, which we employ in our implementations but which have no related passenger number-dependent action.

Table 3.9: Overview of considered passenger number-independent actions (LI actions), which have no related passenger number-dependent action

Name	Short Description
LI Operational Freeze	Operation of some vehicle is frozen, because its dwell time at some waypoint exceeds some threshold
LI Operational Unfreeze	Frozen operation of some stopped vehicle is unfrozen
LI Planned Vehicle Stop	Some moving vehicle stops because its mission says so
LI Stop Skipping	Some moving vehicle does not stop at a waypoint upon its arrival there, but immediately moves to another waypoint
LI Vehicle Dispatch	Some parked vehicle starts processing a run as specified in TN 's dispatch plan
LI Unplanned Vehicle Stop	Some moving vehicle stops because it is not allowed to accounts the waypoint it shall move towards next

We elaborate conditions for their occurrences and impacts of their occurrences on the mode of T -SHA in alphabetical order next.

LI Misson Change. Some vehicle

$$v \in \mathcal{V}_s^{(q)} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{V}_{do}^{(q,s)}$$

which is stopped at some waypoint but not docked to some station $s \in \mathcal{S}$ changes its mission at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_MISSIONCHANGE}(q, v, k) = \text{TRUE},$$

where

- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v , and
- the function $\text{GUARD_LI_MISSIONCHANGE}(\cdot)$ is defined in Alg. 9 below.

Algorithm 9 Specification of the function

GUARD_LI_MISSIONCHANGE :

$$(q, v, k) \in \mathcal{Q} \times \left(\mathcal{V}_s^{(q)} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \right) \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LI_MISSIONCHANGE( $q, v, k$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $w \leftarrow \lambda_m^{(q,v)}$ 
4:   if
      •  $v$  is supposed to change its mission when  $T$ -SHA is in  $q$  according to Def. 61 on p. 82, and
      •  $k \geq \lambda_-(w)(p)$ 
5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{CHANGEMISSION}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 5 on p. 86.

LI Operational Freeze. The operation of some vehicle $v \in \mathcal{V}_{\text{do}}^{(q,s)}$ which is docked to some station $s \in \mathcal{S}$ is frozen at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_OPERATIONALFREEZE}(q, v, k) = \text{TRUE},$$

where

- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v , and
 - the function $\text{GUARD_LI_OPERATIONALFREEZE}(\cdot)$ is defined in Alg. 10 below.
-

Algorithm 10 Specification of the function

GUARD_LI_OPERATIONALFREEZE :

$$(q, v, k) \in \mathcal{Q} \times \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LI_OPERATIONALFREEZE( $q, v, k$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $w \leftarrow \lambda_m^{(q,v)}$ 
4:   if
      •  $v$  is supposed to change its mission when  $T$ -SHA is in  $q$  according to Def. 61 on p. 82, and
      •  $k > \lambda_+^{(w)}(p)$ 
5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{FREEZE}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 11 below.

Algorithm 11 Specification of the function

$$\text{FREEZE} : (q, v) \in \mathcal{Q} \times \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)} \rightarrow \mathcal{Q}$$

```

1: function FREEZE( $q, v$ )
2:    $\mathcal{V}_f^{(q')} \leftarrow \mathcal{V}_f^{(q)} \cup \{v\}$ 
3:   return ( $Q_p^{(q')}, Q_s^{(q')}, Q_d^{(q')}, \mathcal{V}_f^{(q')}$ )
4: end function

```

LI Operational Unfreeze. The operation of some vehicle $v \in \mathcal{V}_f^{(q)}$ whose operation was frozen is unfrozen at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_OPERATIONALUNFREEZE}(q, v, k) = \text{TRUE},$$

where

- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v , and
- the function $\text{GUARD_LI_OPERATIONALUNFREEZE}(\cdot)$ is defined in Alg. 10 below.

Algorithm 12 Specification of the function

$\text{GUARD_LI_OPERATIONALUNFREEZE} :$

$$(q, v, k) \in \mathcal{Q} \times \mathcal{V}_f^{(q)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LI_OPERATIONALUNFREEZE( $q, v, k$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $w \leftarrow \lambda_m^{(q,v)}$ 
4:   if
      •  $v$  is supposed to change its mission when  $T\text{-SHA}$  is in  $q$  according to Def. 61 on p. 82, and
      •  $k > \lambda_{\text{nf}}^{(w)}(p)$ 
5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then $T\text{-SHA}$ transitions to the mode

$$q' := \text{UNFREEZE}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 13 below.

Algorithm 13 Specification of the function

$$\text{UNFREEZE} : (q, v) \in \mathcal{Q} \times \mathcal{V}_f^{(q)} \rightarrow \mathcal{Q}$$

```

1: function UNFREEZE( $q, v$ )
2:    $\mathcal{V}_f^{(q')} \leftarrow \mathcal{V}_f^{(q)} \setminus \{v\}$ 
3:   return  $(Q_p^{(q)}, Q_s^{(q)}, Q_d^{(q)}, \mathcal{V}_f^{(q)})$ 
4: end function

```

LI Planned Vehicle Stop. Some vehicle $v \in \mathcal{V}_d^{(q)}$ which moves towards some waypoint stops at this waypoint because it is supposed to do so at some time $\tau \geq 0$ when $T\text{-SHA}$ in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_PLANNEDVEHICLESTOP}(q, v, k) = \text{TRUE},$$

where

- the function $\text{GUARD_LI_PLANNEDVEHICLESTOP}(\cdot)$ is defined in Alg. 14 below, and
- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v .

Algorithm 14 Specification of the function

$$\begin{aligned} &\text{GUARD_LI_PLANNEDVEHICLESTOP} : \\ &(q, v, k) \in \mathcal{Q} \times \mathcal{V}_d^{(q)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\} \end{aligned}$$

```

1: function GUARD_LI_PLANNEDVEHICLESTOP( $q, v, k$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $w \leftarrow \lambda_m^{(q,v)}$ 
4:   if
      •  $v$  is supposed to stop at  $p$  according to Def. 58 on p. 81, and
      •  $k \geq \lambda_{dr}^{(w)}(p)$ 
5:   then
6:     return TRUE
7:   else
8:     return FALSE
9:   end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then $T\text{-SHA}$ transitions to the mode

$$q' := \text{STOP}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 15 below.

Algorithm 15 Specification of the function

$$\text{STOP} : (q, v) \in \mathcal{Q} \times \mathcal{V}_d^{(q)} \rightarrow \mathcal{Q}$$

```

1: function STOP( $q, v$ )

```

```

2:    $g \leftarrow$  transportation grid which accommodates  $v$ 
3:    $p \leftarrow \lambda_p^{(q,v)}$ 
4:    $z \leftarrow \lambda_r^{(q,v)}$ 
5:    $w \leftarrow \lambda_m^{(q,v)}$ 
6:    $Q_s^{(q')} \leftarrow Q_s^{(q)} \cup \{(g, p, v, z, w)\}$ 
7:    $Q_d^{(q')} \leftarrow Q_d^{(q)} \setminus \{(g, p, v, z, w)\}$ 
8:    $\mathcal{V}_f^{(q')} \leftarrow \mathcal{V}_f^{(q)} \setminus \{v\}$ 
9:   return  $(Q_p^{(q)}, Q_s^{(q')}, Q_d^{(q')}, \mathcal{V}_f^{(q')})$ 
10: end function

```

LI Stop Skipping. Some vehicle $v \in \mathcal{V}_d^{(q)}$ which moves towards some waypoint, say waypoint p , immediately moves towards some other waypoint upon its arrival at p at some time $\tau \geq 0$ when T -SHA in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_STOPSKIPPING}(q, v, k) = \text{TRUE},$$

where

- the function $\text{GUARD_LI_STOPSKIPPING}(\cdot)$ is defined in Alg. 16 below, and
- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v .

Algorithm 16 Specification of the function

$$\text{GUARD_LI_STOPSKIPPING} : (q, v, k) \in \mathcal{Q} \times \mathcal{V}_d^{(q)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LI_STOPSKIPPING( $q, v, k$ )
2:    $p \leftarrow \lambda_p^{(q,v)}$ 
3:    $p' \leftarrow$  waypoint towards which  $v$  is supposed to move next when  $T$ -SHA
      is in  $q$  according to Def. 63 on p. 82
4:    $w := \lambda_m^{(q,v)}$ 
5:   if
      •  $v$  is supposed to skip  $p$  when  $T$ -SHA is in mode  $q$  according to
        Def. 58 on p. 81,
      •  $v$  is allowed to move towards  $p'$  when  $T$ -SHA is in mode  $q$  ac-
        cording to Def. 67 on p. 83, and
      •  $k \geq \lambda_{dr}^{(w)}(p)$ 
6:   then
7:     return TRUE
8:   else

```

```

9:     return FALSE
10:  end if
11: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{SKIP}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 17 below.

Algorithm 17 Specification of the function

$$\text{SKIP} : (q, v) \in \mathcal{Q} \times \mathcal{V}_d^{(q)} \rightarrow \mathcal{Q}$$

```

1: function SKIP( $q, v$ )
2:    $g \leftarrow$  transportation grid which accommodates  $v$ 
3:    $p \leftarrow \lambda_p^{(q,v)}$ 
4:    $z \leftarrow \lambda_r^{(q,v)}$ 
5:    $w \leftarrow \lambda_m^{(q,v)}$ 
6:    $p' \leftarrow$  Next position of  $v$  according to Def. 63 on p. 82
7:    $Q_d^{(q')} \leftarrow Q_d^{(q)} \setminus \{(g, p, v, z, w)\}$ 
8:    $Q_d^{(q')} \leftarrow Q_d^{(q')} \cup \{(g, p', v, z, w)\}$ 
9:   return ( $Q_p^{(q)}, Q_s^{(q)}, Q_d^{(q')}, \mathcal{V}_f^{(q)}$ )
10: end function

```

LI Vehicle Departure. Some vehicle

$$v \in \mathcal{V}_s^{(q)} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{V}_{do}^{(q,s)}$$

which is stopped at some waypoint but not docked any station $s \in \mathcal{S}$ departs from its position $p := \lambda_p^{(q,v)}$ at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_VEHICLEDEPARTURE}(q, v, k) = \text{TRUE},$$

where

- the function $\text{GUARD_LI_VEHICLEDEPARTURE}(\cdot)$ is defined in Alg. 18 below, and
- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v .

Algorithm 18 Specification of the function

GUARD_LI_VEHICLEDEPARTURE :

$$(q, v, k) \in \mathcal{Q} \times \left(\mathcal{V}_s^{(q)} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{V}_{do}^{(q,s)} \right) \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

1: **function** GUARD_LI_VEHICLEDEPARTURE(q, v, k)
2: $p \leftarrow \lambda_p^{(q,v)}$
3: $w \leftarrow \lambda_m^{(q,v)}$
4: **if**
 • v is supposed to depart from p when T -SHA is in q according to Def. 59 on p. 81,
 • v is allowed to move towards p' when T -SHA is in q according to Def. 67 on p. 83, and
 • $\lambda_-^{(w)}(p) \leq k \leq \lambda_+^{(w)}(p)$
5: **then**
6: **return** TRUE
7: **else**
8: **return** FALSE
9: **end if**
10: **end function**

Assuming that only this LDA occurs, then T -SHA transitions to the mode

$$q' := \text{DEPART}(q, v),$$

where $q' \in \mathcal{Q}$ is computed according to Alg. 3 on p. 85.

LI Vehicle Dispatch. Some parked vehicle $v \in \mathcal{V}_p^{(q)}$ is dispatched and thus starts processing a new run at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_VEHICLEDISPATCH}(q, v, \tau) = \text{TRUE},$$

where the function GUARD_LI_VEHICLEDISPATCH(\cdot) is defined in Alg. 19 below.

Algorithm 19 Specification of the function

GUARD_LI_VEHICLEDISPATCH :

$$(q, v, \tau) \in \mathcal{Q} \times \mathcal{V}_p^{(q)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

1: **function** GUARD_LI_VEHICLEDISPATCH(q, v, τ)
2: **if**

```

3:    $\exists d := (\tau', v, z, \Delta\tau) \in \mathcal{D}$  s.t.  $\tau \in [\tau', \tau' + \Delta\tau]$ 
4:   then
5:     return TRUE
6:   else
7:     return FALSE
8:   end if
9: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then *T-SHA* transitions to the mode

$$q' := \text{DISPATCH}(q, v, \tau)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 20 below.

Algorithm 20 Specification of the function

$$\text{DISPATCH} : (q, v, \tau) \in \mathcal{Q} \times \mathcal{V}_p^{(q)} \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{Q}$$

```

1: function DISPATCH( $q, v, \tau$ )
2:    $g \leftarrow$  transportation grid which accommodates  $v$ 
3:    $p \leftarrow \lambda_p^{(q,v)}$ 
4:    $d := (\tau', v, z, \Delta\tau) \leftarrow$  (unique) scheduled dispatch in  $\mathcal{D}$  at some time  $\tau \in [\tau', \tau' + \Delta\tau]$ 
5:    $w \leftarrow$  first mission of  $z$  according to Def. 36 on p. 36
6:    $Q_p^{(q')} \leftarrow Q_p^{(q)} \setminus \{(g, p, v)\}$ 
7:    $Q_s^{(q')} \leftarrow Q_s^{(q)} \cup \{(g, p, v, z, w)\}$ 
8:   return  $(Q_p^{(q')}, Q_s^{(q')}, Q_d^{(q)}, \mathcal{V}_f^{(q)})$ 
9: end function

```

LI Vehicle Parking. Some vehicle

$$v \in \mathcal{V}_s^{(q)} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{V}_{\text{do}}^{(q,s)}$$

which is stopped at some waypoint but not docked to any station $s \in \mathcal{S}$ parks at some time $\tau \geq 0$ when *T-SHA* is in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_VEHICLEPARKING}(q, v) = \text{TRUE},$$

where the function $\text{GUARD_LI_VEHICLEPARKING}(\cdot)$ is defined in Alg. 21 below.

Algorithm 21 Specification of the function

GUARD_LI_VEHICLEPARKING :

$$(q, v) \in \mathcal{Q} \times \left(\mathcal{V}_s^{(q)} \setminus \bigcup_{s \in \mathcal{S}} \mathcal{V}_{do}^{(q,s)} \right) \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LI_VEHICLEPARKING( $q, v$ )
2:   if
3:      $v$  is supposed to park according to Def. 57 on p. 81 when  $T$ -SHA is
       in  $q$ 
4:   then
5:     return TRUE
6:   else
7:     return FALSE
8:   end if
9: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{PARK}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 8 on p. 88.

LI Unplanned Vehicle Stop. Some vehicle $v \in \mathcal{V}_d^{(q)}$ which moves towards some waypoint stops at this waypoint although it is not supposed to do so at some time $\tau \geq 0$ when T -SHA in some mode $q \in \mathcal{Q}$ iff

$$\text{GUARD_LI_UNPLANNEDVEHICLEDSTOP}(q, v, k) = \text{TRUE},$$

where

- the function $\text{GUARD_LI_UNPLANNEDVEHICLESTOP}(\cdot)$ is defined in Alg. 22 below, and
- $k \geq 0$ denotes the valuation of the clock $c^{(v)}$ at τ which is associated with the operation of v .

Algorithm 22 Specification of the function

GUARD_LI_UNPLANNEDVEHICLESTOP :

$$(q, v, k) \in \mathcal{Q} \times \mathcal{V}_d^{(q)} \times \mathbb{R}_{\geq 0} \rightarrow \{\text{TRUE}, \text{FALSE}\}$$

```

1: function GUARD_LI_UNPLANNEDVEHICLESTOP( $q, v, k$ )

```

```

2:   $p \leftarrow \lambda_p(q, v)$ 
3:   $w \leftarrow \lambda_m(q, v)$ 
4:  if
      •  $v$  is supposed to skip  $p$  when  $T$ -SHA is in  $q$  according to Def.
        60 on p. 81,
      • the waypoint towards which  $v$  is supposed to move next ac-
        cording to Def. 63 on p. 82 is blocked by another vehicle ac-
        cording to Def. 64 on p. 82, and
      •  $k \geq \lambda_{dr}^{(w)}(p)$ 
5:  then
6:    return TRUE
7:  else
8:    return FALSE
9:  end if
10: end function

```

Assuming that no other vehicle $v' \in \mathcal{V} \setminus \{v\}$ performs this or some other action at the same time, then T -SHA transitions to the mode

$$q' := \text{STOP}(q, v)$$

at τ , where $q' \in \mathcal{Q}$ is computed according to Alg. 15 on p. 94.

OBTAINING FORECASTS

This chapter introduces a discrete time simulation scheme for the computation of passenger number forecasts in multimodal and monomodal transportation networks alike, as well as forecasts for the positions of all vehicles, their driving conditions, and their operational states therein; cf. the table of contents below: In Sec. 4.1 we will briefly discuss the possible scope of any set of simultaneously considered forecast objectives, as well as the proper formulation of any initial state. In Sec. 4.2 we will then introduce our discrete time scheme, before we will look at all computational bottlenecks inherent to it in Sec. 4.3. We will next introduce some algorithmic bricks which aim at the these bottlenecks. Finally, we integrate all of algorithmic bricks into one single forecast algorithm which we will introduce in Sec. 4.5.

Contents

4.1	Preliminary Considerations	102
4.1.1	Forecast Objective	102
4.1.2	Specification of <i>T-SHA</i> 's Initial State	103
4.2	Discrete Time Traversal of Complete Forecast Horizon	104
4.2.1	Idea of Discrete Time Computation Scheme	104
4.2.2	Introducing Propagation DAGs	105
4.2.3	Propagation of Probability Densities Along Propagation DAGs	113
4.3	Bottlenecks	114
4.3.1	Node Explosion of Propagation DAG	114
4.3.2	High-dimensionality of Decoupled SDEs	116
4.4	Bricks for the Efficient Computation of Forecasts	117
4.4.1	Overview	117
4.4.2	Layer-By-Layer Computation of a Forecast	122
4.4.3	Proper Specification of Initial Passenger Loads	123
4.4.4	Macro Modes	124
4.4.5	Static Clock Valuations for All Parked Vehicles	126
4.4.6	Canonical Decoupling of All Passenger Flows	127
4.4.7	Disconnect Platforms in Stations	140
4.4.8	Unfold Vehicle Trajectories	144
4.4.9	Offline Computation of Transition Probability Matrices . . .	146
4.5	Forecast Algorithm	148

Preliminary Considerations

Forecast Objective

Starting point of every new forecast is the formulation of some set of objectives. For instance, we might want to compute the probability according to which the cumulative passenger number at some platform exceeds a certain threshold. Another forecast objective might be to compute the probability according to which the deviations of the vehicles' arrival and departure times at/from some major hub remain within some tight intervals. Whatever the set of simultaneously considered forecast objectives is, it must be tied to the time evolution of *T-SHA*'s hybrid state in some finite time horizon $h > 0$. We cannot use *T-SHA* to compute forecasts for infinite time horizons since these kind of forecasts would require us to look for/-take advantage of some special properties which are inherent to *T-SHA*'s structure and/or dynamics given its initial state; rather than traversing *T-SHA*'s state space forward in time. In this context, also note that we will not provide some use case specific forecast algorithm here. Instead, it is the end user of our forecast algorithm who must define which part of *T-SHA*'s hybrid state (space) s/he wants to include in some post-processing steps in that s/he specifies the following sets:

- A set $\mathcal{V}_t \subseteq \mathcal{V}$ of so-called target vehicles: If $v \in \mathcal{V}_t$ for vehicle $v \in \mathcal{V}$, then no algorithmic brick is allowed to neglect v during the computation of the considered forecast. Instead, the passenger number on-board v , as well as v 's position, driving condition, and operational state must be computed for the complete forecast horizon.
- For station $s \in \mathcal{S}$, a set $P_t^{(s)} \subseteq P^{(s)}$ of so-called target gathering points in s : If $p \neq P_t^{(s)}$ for gathering point $p \in P_{gp}^{(s)}$ in station s , then it might be the case that our algorithms will not compute a forecast which considers the passenger number at p over the complete forecast horizon.

We moreover expect the end user of our forecast algorithm to be able to specify some stop criterion. This stop criterion must take the form of (i) some compound boolean expression which relates *T-SHA*'s mode to be in some subset of admissible modes, to the cumulative passenger numbers at some gathering points or on-board some vehicles to be less than, bigger than, etc. some thresholds; and (ii) some threshold for the probability associated with the evaluation of this boolean expression. Our forecast algorithm then terminates all computations before it has computed the time evolution of *T-SHA*'s hybrid state over the complete forecast horizon when it realizes that the probability according to which the above boolean expression evaluates to `TRUE` exceeds its threshold.

From all above inputs specified by the end user plus the complete parametrization of *T-SHA* (also specified by this end user), all that our forecast algorithm can and will do once it is called, is to compute the evolution of the considered part of *T-SHA*'s hybrid state forward in time; starting from the initial (simulation) time

$\tau = 0$ until a) it reaches the terminal time $\tau = h$ or b) prior to $\tau = h$ because the considered stop criterion is fulfilled at some time $\tau' \in [0, h)$ before; cf. Fig. 4.1.1 below.

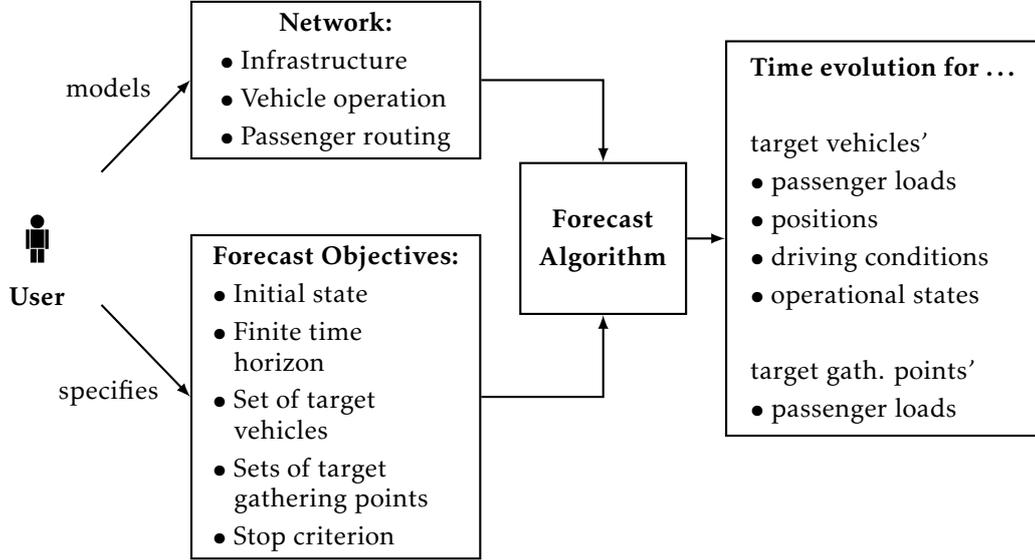


Figure 4.1.1: Use case diagram for our forecast algorithm: the specification of additional parameters by the end user such as the discretization of the forecast horizon are not depicted here

Specification of *T-SHA*'s Initial State

By assumption, we have an exact knowledge of the complete vehicle operational state in the considered multimodal transportation network at simulation/prediction time $\tau = 0$ with $\tau \in [0, h]$, from which we can deduce some initial mode $q_0 \in \mathcal{Q}$ s.t.

$$\mathbb{P}(q(0) = q_0) = 1,$$

and some valuation $\varphi_0[v] \geq 0$ for the clock $c^{(v)}$ associated with the operation of every vehicle $v \in \mathcal{V}$ s.t.

$$\mathbb{P}(\text{val}(c^{(v)}, 0) = \varphi_0[v]) = 1; \quad (4.1)$$

where - here and in the rest of this chapter - $q(\tau) \in \mathcal{Q}$ denotes the mode of *T-SHA* at τ ,

$$\mathbb{P}(q(\tau) = q')$$

denotes the marginal probability for *T-SHA* to be in some mode $q' \in \mathcal{Q}$ at τ , $\text{val}(c^{(v)}, \tau)$ denotes the valuation of $c^{(v)}$ for vehicle $v \in \mathcal{V}$ at τ , and

$$\mathbb{P}(\text{val}(c^{(v)}, \tau) = k^{(v)})$$

denotes the marginal probability for the valuation of some $c^{(v)}$ to be $k^{(v)} = 0$ at τ .

Estimations of all passenger numbers complement the specification of T -SHA's initial state at time $\tau = 0$: We either manage to specify the joint probability pdf₀ for all (vectorial) passenger numbers at $\tau = 0$ from scratch s.t. $\text{pdf}(X(0)) = \text{pdf}_0$, or we make a small but probably more practical detour: Assuming that all initial passenger number vectors are independent random variables, we map our estimation of the passenger number at every gathering point $p \in P_{\text{gp}}^{(s)}$ in any station $s \in \mathcal{S}$ to some (marginal) probability density function

$$\text{pdf}_{\text{gp}}^{(s,p)} : \mathcal{M}^{(s,p)} \rightarrow \mathbb{R}_{\geq 0}.$$

Similarly, we map our estimation of the passenger number on-board every vehicle $v \in \mathcal{V}$ to some (marginal) probability density function

$$\text{pdf}_{\text{vh}}^{(v)} : \mathcal{U}^{(v)} \rightarrow \mathbb{R}_{\geq 0}.$$

From these many individual (marginal) probability density functions, we then compute pdf₀ according to

$$\text{pdf}_0(x) := \prod_{(s,p) \in \mathcal{S} \times P_{\text{gp}}^{(s)}} \text{pdf}_{\text{gp}}^{(s,p)}(m^{(s,p)}) \prod_{v \in \mathcal{V}} \text{pdf}_{\text{vh}}^{(v)}(u^{(v)}) \quad (4.2)$$

for all

$$x := \left[m^{(1,1)}; \dots; m^{(1,|P_{\text{gp}}^{(1)}|)}; m^{(2,1)}; \dots; m^{(|\mathcal{S}|,|P_{\text{gp}}^{(|\mathcal{S}|)}|)}; u^{(1)}; \dots; u^{(|\mathcal{V}|)} \right]^t,$$

where $m^{(s,p)} \in \mathcal{M}^{(s,p)}$ and $u^{(v)} \in \mathcal{U}^{(v)}$.

Discrete Time Traversal of Complete Forecast Horizon

Idea of Discrete Time Computation Scheme

Stochastic differential balance equations govern the continuous time evolution of all uncertain passenger numbers in any mode $q \in \mathcal{Q}$ of T -SHA. Thus, starting from one particular continuous state for all passenger numbers either none or uncountably many passenger number trajectories hit some well-defined target region (= closed convex polytope) in finite time; which in turn implies that atomic passenger number-dependent actions performed by individual vehicles either cannot occur in q at all, or their occurrence times are continuous random variables in some half-open or closed intervals. The problem with this particularity of T -SHA's continuous passenger-flow dynamics is that it renders the exhaustive computation of its (hybrid) state space intractable. However, this inconvenient fact does not prevent us to employ our discrete time scheme for the computation of any forecast:

We let all vehicles perform any atomic action from Tab. 3.7 on p. 83, Tab. 3.8 on p. 89, and Tab. 3.9 on p. 90 only at fixed discrete points in time which equidistantly cover the complete forecast horizon.

Introducing Propagation DAGs

As outlined above, we confine the occurrences of all atomic actions performed by all vehicles $v \in \mathcal{V}$ in T -SHA to a sequence $\tau_0, \tau_1, \dots, \tau_k$ of some $k \in \mathbb{N}_{>0}$ equidistantly-spaced discrete points in time, which covers the complete time horizon $h > 0$ of the considered forecast. This intervention implies that the occurrence times of passenger number-dependent actions are no longer continuous- but discrete random variables; which property in turn implies that two or more different passenger number-dependent actions might be performed by some vehicles at the same time with some positive marginal probability. That being said, note that the fact that the vehicles' internal clocks (= clocks associated with their operation) progress as linear functions of T -SHA's simulation time in any mode $q \in \mathcal{Q}$, implies that T -SHA is in one particular *timed* mode $\sigma \in \Sigma$ at any discrete simulation time τ_i with $i \in \mathcal{H} := \{0, 1, \dots, k\}$.

Definition 68 (Timed Mode) *Every timed mode $\sigma \in \Sigma$ of T -SHA defines a particular realization of its mode $q \in \mathcal{Q}$ and a particular valuation $\varphi[v]$, with $\varphi \in (\mathbb{R}_{\geq 0})^{|\mathcal{V}|}$, of the clock $c^{(v)}$ associated with the operation of every vehicle $v \in \mathcal{V}$ in form of some tuple $\sigma := (q, \varphi)$.*

We can map all possible evolutions of T -SHA's timed mode to some directed acyclic graph (DAG), which we call propagation DAG.

Definition 69 (Propagation DAG) *A propagation DAG for T -SHA is a weakly-connected edge-labelled directed acyclic graph*

$$G := \left(N, E, \Psi := \left\{ \Psi^{(e)} : e \in E \right\} \right),$$

where

- every node $n \in N$, with $N \subseteq \Sigma$, corresponds to some particular candidate for T -SHA's timed mode at the discrete time step $\mu^{(G)}(n)$, in which
 - $\mu^{(G)}(n)$, with $\mu^{(G)} : N \rightarrow \mathbb{N}_{\geq 0}$, denotes the height of n in G ,
 - there is only one node $n_0 \in N$ with zero height in G ,
 - $n_0 = (q_0, \varphi_0)$, and
 - every leaf node in G has the same height,
- every edge $(n_1, n_2) \in E$, with $E \subseteq N \times N$, corresponds to the possibility of T -SHA to transition from n_1 to n_2 at the discrete time step $\mu^{(G)}(n_2)$, and
- the set

$$\Psi^{(e)} \in \left\{ e \in 2^{\mathcal{V} \times \mathcal{A}} : (v, a_1) \in e \wedge (v, a_2) \in e \text{ implies } a_1 = a_2, \text{ for } v \in \mathcal{V} \right\}$$

for $e \in E$, with $e := (n_1, n_2)$, stores the subset of all vehicles $v \in \mathcal{V}$ together with actions performed by these vehicles, which make T -SHA to transition from n_1 to n_2 at the discrete time step $\mu^{(G)}(n_2)$.

Nodes with the same height $i \in \mathcal{H}$ in G correspond to (all possible) alternative candidates for T -SHA's timed mode at the discrete time $\tau = i \Delta\tau$, where

$$\Delta\tau := \frac{h}{k} \quad (4.3)$$

is the fixed duration of the time step that separates τ_i from τ_{i+1} for $i \in \mathcal{H}$. This constructed property of G implies

$$\sum_{n \in (\mu^{(G)})^{-1}(i)} \mathbb{P}(\sigma(i \Delta\tau) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0) = 1 \quad (4.4)$$

for $i \in \mathcal{H}$, where $\sigma(i \Delta\tau) \in \Sigma$ denotes the timed mode of T -SHA at time $\tau = i \Delta\tau$. Moreover, it implies

$$\begin{aligned} \mathbb{P}(\sigma(\mu^{(G)}(n) \Delta\tau) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0) = \\ \sum_{n' \in {}^\circ n} \mathbb{P}(\sigma(\mu^{(G)}(n) \Delta\tau) = n \mid \sigma(\mu^{(G)}(n') \Delta\tau) = n') \\ \mathbb{P}(\sigma(\mu^{(G)}(n') \Delta\tau) = n' \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0) \end{aligned} \quad (4.5)$$

for $n \in N$ s.t. $\mu^{(G)}(n) > 0$, where

$${}^\circ n := \{m \in N : (m, n) \in E\}$$

denotes the set of all parent nodes n has in G , and

$$n^\circ := \{m \in N : (n, m) \in E\}$$

denotes the set of all child nodes n has in G .

Computation. Before we come to the main algorithm (Alg. 25 on p. 112 below) for the computation of any propagation DAG G over some finite time horizon $h > 0$ in $k \in \mathbb{N}_{>0}$ fixed time steps of equal length $\Delta\tau := h/k$ each, we introduce two auxiliary algorithms first: Once called, the first auxiliary algorithm, Alg. 23 below, returns T -SHA's mode $q' \in \mathcal{Q}$ given the fact that some vehicle $v \in \mathcal{V}$ performs some action $a \in \mathcal{A}$ at some time $\tau > 0$ when T -SHA is in some mode $q \in \mathcal{Q}$.

Algorithm 23 Specification of the function

$$\text{PERFORMACTION} : \mathcal{Q} \times \mathcal{V} \times \mathcal{A} \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{Q}$$

```

1: function PERFORMACTION( $q, v, a, \tau$ )
2:   switch  $a$ 
3:     case LD Mission Change:
4:       // Function CHANGEMISSION( $\cdot$ ) from Alg. 5 on p. 86:
5:        $q' \leftarrow \text{CHANGEMISSION}(q, v)$ 

```

```
6:     case LD Vehicle Parking:
7:         // Function PARK(·) from Alg. 8 on p. 88:
8:          $q' \leftarrow \text{PARK}(q, v)$ 

9:     case LD Vehicle Departure:
10:        // Function DEPART(·) from Alg. 3 on p. 85:
11:         $q' \leftarrow \text{DEPART}(q, v)$ 

12:    case LI Mission Change:
13:        // Function CHANGEMISSION(·) from Alg. 5 on p. 86:
14:         $q' \leftarrow \text{CHANGEMISSION}(q, v)$ 

15:    case LI Operational Freeze:
16:        // Function FREEZE(·) from Alg. 11 on p. 92:
17:         $q' \leftarrow \text{FREEZE}(q, v)$ 

18:    case LI Operational Unfreeze:
19:        // Function UNFREEZE(·) from Alg. 13 on p. 93:
20:         $q' \leftarrow \text{UNFREEZE}(q, v)$ 

21:    case LI Planned Vehicle Stop:
22:        // Function STOP(·) from Alg. 15 on p. 94:
23:         $q' \leftarrow \text{STOP}(q, v)$ 

24:    case LI Vehicle Departure:
25:        // Function DEPART(·) from Alg. 3 on p. 85:
26:         $q' \leftarrow \text{DEPART}(q, v)$ 

27:    case LI Vehicle Dispatch:
28:        // Function DISPATCH(·) from Alg. 20 on p. 98:
29:         $q' \leftarrow \text{DISPATCH}(q, v, \tau)$ 

30:    case LI Vehicle Parking:
31:        // Function PARK(·) from Alg. 8 on p. 88:
32:         $q' \leftarrow \text{PARK}(q, v)$ 

33:    case LI Unplanned Vehicle Stop:
34:        // Function STOP(·) from Alg. 15 on p. 94:
35:         $q' \leftarrow \text{STOP}(q, v)$ 

36:    end switch
37:    return  $q'$ 
38: end function
```

The second auxiliary algorithm, Alg. 24 below, appends to some non-empty propagation DAG G with height $h \in \mathbb{N}_{\geq 0}$ a (complete) new layer of timed modes when being called. In order to achieve so, it must compute all conceivable children of every leaf node $n := (q, \varphi) \in N$ in G . In this context, note the following: We say that some vehicle $v \in \mathcal{V}$ performs some passenger number-independent action $a \in \mathcal{A}$ at some time $\tau \geq 0$ when T -SHA is in n iff the guard function which we have specified for the occurrence of a in Sec. 3.7.4 on p. 89ff. returns the boolean value `TRUE` when it is evaluated for v as vehicle, a as action, $\varphi[v]$ as valuation for the clock $c^{(v)}$, and possibly τ as time. Similarly, we say that v *might* perform some passenger number-dependent action $a' \in \mathcal{A}$ at τ iff the guard function which we have specified for the occurrence of a' in Sec. 3.7.3 on p. 83ff. returns the boolean value `TRUE` for some arbitrary choice of all passenger numbers when it is evaluated for v as vehicle, a' as action, and $\varphi[v]$ as valuation for the clock $c^{(v)}$. That being said, we have to distinguish three different situations:

1. No vehicle $v \in \mathcal{V}$ performs any action at time $\tau = h_{\Delta} \tau$ when the timed mode of T -SHA is n . Assuming this to be true, the lines 6 to 13 in Alg. 24 below append one single child node $n' \in N$ to n in G , where compared to n the valuations of all clocks in n' are incremented by $\Delta \tau$.
2. Every vehicle $v' \in \mathcal{V}'$ of some non-empty subset $\mathcal{V}' \subseteq \mathcal{V}$ of vehicles performs some passenger number-independent action at time $\tau = i_{\Delta} \tau$ when the timed mode of T -SHA is n . However, every other vehicle $v \in \mathcal{V} \setminus \mathcal{V}'$ does not perform any action at all. Assuming this to be true, the lines 15 to 36 in Alg. 24 below append one single child node $n' \in N$ to n in G , where compared to n (i) the valuations of all clocks associated with the operation of any $v \in \mathcal{V} \setminus \mathcal{V}'$ are incremented by $\Delta \tau$, (ii) the valuations of all clocks associated with the operation of any vehicle $v' \in \mathcal{V}'$ are reset to zero, and (iii) the positions, driving conditions, and operational states of all vehicles $v' \in \mathcal{V}$ are updated corresponding to the impacts of their atomic actions.
3. Every vehicle $v_1 \in \mathcal{V}_1$ of some non-empty subset $\mathcal{V}_1 \subseteq \mathcal{V}$ of vehicles performs some passenger number-dependent action. Every vehicle $v_2 \in \mathcal{V}_2$ of some (possibly empty) subset $\mathcal{V}_2 \subseteq \mathcal{V}$ of vehicles performs some passenger number-independent action, and every remaining vehicle $v_3 \in \mathcal{V}_3$ of some (possibly empty) subset $\mathcal{V}_3 \subseteq \mathcal{V}$ of vehicles performs no action at all. Assuming this to be true, the lines 38 to 72 in Alg. 24 below append $2^{|\mathcal{V}_1|}$ different child nodes to n in G : Every child node $n' \in n^{\circ}$ corresponds to one particular subset (including the empty subset!) of vehicles $v_1 \in \mathcal{V}_1$ which perform some passenger number-dependent action together at time τ .

For reasons that will be clear only later when we will have introduced our forecast algorithm in Sec. 4.2.2 on p. 149ff., Alg. 24 below has a third input argument; apart from some existing propagation DAG G and the fixed time step $\Delta \tau > 0$ which we use for the discretization of the complete forecast horizon: A subset $\bar{\mathcal{V}} \subseteq \mathcal{V}$ of

all vehicles whose operation will be ignored in the computation of a complete new layer of nodes for G . More specifically, Alg. 24 ignores the possibility that a vehicle $v \in \bar{\mathcal{V}}$ performs some action when T -SHA makes one discrete time step into the future.

Algorithm 24 Specification of the function

$$\text{APPENDLAYER} : \mathbb{G} \times \mathbb{R}_{>0} \times 2^{\mathcal{V}} \rightarrow \mathbb{G},$$

with

$\mathbb{G} := \{G : G \text{ is some propagation DAG for } T\text{-SHA according to Def. 69 on p. 69}\},$

which appends to some existent propagation DAG G computed for T -SHA some complete new layer of nodes. For the incrementation of all clocks $\Delta\tau > 0$ is used. If $v \in \bar{\mathcal{V}}$ for vehicle $v \in \mathcal{V}$, then the possibility that v performs any action will be ignored.

```

1: function APPENDLAYER( $G := (N, E, \Psi), \Delta\tau, \bar{\mathcal{V}}$ )
2:    $h \leftarrow$  height of  $G$ 
3:   for
4:     every node  $n := (q, \varphi) \in N$  s.t.  $\mu^{(G)}(n) = h$ 
5:   do
6:     // Situation 1:
7:     if
8:       No vehicle  $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$  performs any action at time  $\tau := (h + 1) \Delta\tau$ 
       when the timed mode of  $T$ -SHA is  $n$ 
9:     then
10:       $n' \leftarrow \left( q, [\varphi[1] + \Delta\tau, \varphi[2] + \Delta\tau, \dots, \varphi[|\mathcal{V}|] + \Delta\tau]^t \right)$ 
11:       $N \leftarrow N \cup \{n'\}$ 
12:       $e \leftarrow (n, n')$ 
13:       $E \leftarrow E \cup \{e\}$ 
14:       $\Psi^{(e)} \leftarrow \emptyset$ 
15:       $\Psi \leftarrow \Psi \cup \{\Psi^{(e)}\}$ 
16:     end if
17:     // Situation 2:
18:     if

```

```

19:      Some vehicles  $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$  perform some passenger load-
      independent actions at time  $\tau := (h+1)_{\Delta}\tau$  when the timed mode
      of T-SHA is  $n$ , whereas all other vehicles perform no actions at
      all

20:  then
21:     $q' \leftarrow q$ 
22:     $k \leftarrow \varphi$ 
23:     $K \leftarrow \emptyset$ 
24:  for
25:    every  $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$ 
26:  do
27:    if
28:       $v$  performs some passenger load-independent action  $a \in \mathcal{A}$ 
29:    then
30:       $k[v] \leftarrow 0$ 
31:       $q' \leftarrow \text{PERFORMACTION}(q', v, a, \tau)$ 
32:       $K \leftarrow K \cup \{(v, a)\}$ 
33:    else
34:       $k[v] \leftarrow k[v] + \Delta\tau$ 
35:    end if
36:  end for
37:   $n' \leftarrow (q', k)$ 
38:   $N \leftarrow N \cup \{n'\}$ 
39:   $e \leftarrow (n, n')$ 
40:   $E \leftarrow E \cup \{e\}$ 
41:   $\Psi^{(e)} \leftarrow K$ 
42:   $\Psi \leftarrow \Psi \cup \{\Psi^{(e)}\}$ 
43: end if

44:  // Situation 3:
45:  if
46:    Some vehicles  $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$  might perform some passenger load-
    dependent actions at time  $\tau = (h+1)_{\Delta}\tau$  when the timed mode
    of T-SHA is  $n$ 
47:  then

```

```

48:      $\mathcal{V}_1 \leftarrow$  subset of all vehicles  $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$  which might perform some
      passenger load-dependent action at time  $\tau = (h + 1) \Delta\tau$ 
49:      $\mathcal{V}_2 \leftarrow$  subset of all vehicles  $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$  which perform some pas-
      senger load-independent action at time  $\tau = (h + 1) \Delta\tau$ 
50:      $q' \leftarrow q$ 
51:      $k \leftarrow \varphi$ 
52:      $K \leftarrow \emptyset$ 
53:     for
54:         every  $v \in \mathcal{V} \setminus (\mathcal{V}_1 \cup \mathcal{V}_2)$ 
55:     do
56:          $k[v] \leftarrow k[v] + \Delta\tau$ 
57:     end for
58:     for
59:         every  $v \in \mathcal{V}_2$ 
60:     do
61:          $k[v] \leftarrow 0$ 
62:          $a \leftarrow$  passenger load-independent action performed by  $v$ 
63:         // Function PERFORMACTION( $\cdot$ ) according to Def. 23 on p. 106:
64:          $q' \leftarrow$  PERFORMACTION( $q', v, a, \tau$ )
65:          $K \leftarrow K \cup \{(v, a)\}$ 
66:     end for
67:     for
68:         every subset  $\mathcal{V}' \subseteq \mathcal{V}_1$ 
69:     do
70:         for
71:             every  $v \in \mathcal{V}'$ 
72:         do
73:              $k[v] \leftarrow 0$ 
74:              $a \leftarrow$  passenger load-dependent action which might be
              performed by  $v$ 
75:             // Function PERFORMACTION( $\cdot$ ) according to Def. 23 on p. 106:
76:              $q' \leftarrow$  PERFORMACTION( $q', v, a, \tau$ )
77:              $K \leftarrow K \cup \{(v, a)\}$ 
78:         end for
79:      $n' \leftarrow (q', k)$ 
80:      $N \leftarrow N \cup \{n'\}$ 

```

```

81:          $e \leftarrow (n, n')$ 
82:          $E \leftarrow E \cup \{e\}$ 
83:          $\Psi^{(e)} \leftarrow K$ 
84:          $\Psi \leftarrow \Psi \cup \{\Psi^{(e)}\}$ 
85:     end for
86: end if
87: end for
88: return  $(N, E, \Psi)$ 
89: end function

```

We are now ready to introduce Alg. 25; see below: In the first two lines of its (pseudo) code, we instantiate G with one single node, which captures T -SHA's initial vehicle operational state. In the remaining lines of code, we then append to G one layer of nodes with common height $i \in \{1, \dots, k\}$ in G after the other.

Algorithm 25 Compute the propagation DAG

$$G := (N, E, \Psi)$$

for T -SHA with some fixed time step $\Delta\tau := \frac{h}{k}$ for $k \in \mathbb{N}_{>0}$ s.t. it completely covers the finite forecast horizon $h > 0$ for some considered use case.

Step 1: Instantiate G with one single node which captures T -SHA's initial vehicle operational state.

```

1:  $N \leftarrow n_0 := (q_0, \varphi_0)$ 
2:  $E \leftarrow \emptyset$ 
3:  $\Psi \leftarrow \emptyset$ 
4:  $G \leftarrow (N, E, \Psi)$ 

```

Step 2: Compute G layer by layer starting from the single node in its root.

```

5:  $i \leftarrow 0$ 
6: while
7:      $i < k$ 
8: do
9:     // Function APPENDLAYER( $\cdot$ ) according to Alg. 24 on p. 109:
10:     $G \leftarrow \text{APPENDLAYER}(G, \Delta\tau, \emptyset)$ 
11:     $i \leftarrow i + 1$ 
12: end for

```

Propagation of Probability Densities Along Propagation DAGs

Our proposed discrete time scheme for the computation of any forecast implies that the systems of stochastic differential balance equations which we assign to any mode $q' \in \mathcal{Q}$ of *T-SHA*, defines the time evolution of all passenger numbers stacked in the vector $X(\cdot)$ in form of some aggregated drift term $\alpha^{(q')}$ and some aggregated diffusion term $\beta^{(q')}$ in the (half-open) time interval $[i_{\Delta\tau}, (i+1)_{\Delta\tau})$ for $i \in \mathcal{H} := \{0, 1, 2, \dots, k\}$ with marginal probability

$$\mathbb{P}(q(i_{\Delta\tau}) = q' \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0),$$

where $k \in \mathbb{N}_{>0}$ denotes the number of time steps chosen which equidistantly discretize the complete considered forecast horizon. We can thus compute

$$\text{pdf}\left(X((i+1)_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right)$$

from

$$\text{pdf}\left(X(i_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right)$$

as follows: First of all, note that several nodes with the same height in some forecasting graph

$$G := (N, E, \Psi := \{\Psi^{(e)} : e \in E\}),$$

might have the same mode $q' \in \mathcal{Q}$ in common, which implies

$$\begin{aligned} \mathbb{P}(q(i_{\Delta\tau}) = q' \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0) = \\ \sum_{n \in \{n'' := (q'', k'') \in N : \mu^{(G)}(n'') = i, q'' = q'\}} \mathbb{P}(\sigma(i_{\Delta\tau}) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0) \end{aligned} \quad (4.6)$$

for all $i \in \mathcal{H}$. That being said, we first compute the marginal conditional probability density

$$\text{pdf}\left(X((i+1)_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q(i_{\Delta\tau}) = q'\right)$$

for every (different) mode $q' \in \mathcal{Q}$ which underlies some node $n \in (\mu^G)^{-1}(i)$ with height $i \in \mathcal{H} \setminus \{k\}$ in G , as solution to the Fokker–Planck equation, (3.2) on p. 37, with reflecting boundary conditions according to (3.10) on p. 38, and initial probability density

$$\text{pdf}\left(X(i_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right).$$

We then compute

$$\begin{aligned} \text{pdf}\left(X((i+1)_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right) = \\ \sum_{q' \in \mathcal{Q}} \text{pdf}\left(X((i+1)_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q(i_{\Delta\tau}) = q'\right) \quad (4.7) \\ \mathbb{P}(q(i_{\Delta\tau}) = q' \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0). \end{aligned}$$

Note that (4.6) and (4.7) above do not fully describe the propagation of X 's density along the complete forecast horizon: Only for time step $i = 0$,

$$\mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right)$$

in (4.6) is known in that it is one for $n = n_0$ but zero otherwise. However for all other $i \in \mathcal{H} \setminus \{0\}$ it follows from (4.5) on p. 106 that

$$\begin{aligned} \mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right) = \\ \sum_{n' \in {}^\circ n} \mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma((i-1)_{\Delta\tau}) = n'\right) \\ \mathbb{P}\left(\sigma((i-1)_{\Delta\tau}) = n' \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right). \end{aligned} \quad (4.8)$$

Next note that a transition of T -SHA at time $\tau = i_{\Delta\tau}$, for $i \in \mathcal{H} \setminus \{0\}$, to some timed mode $n \in (\mu^{(G)})^{-1}(i)$ in G from some timed mode $n' := (q', k') \in {}^\circ n$ (e.g. caused by some vehicle departure) might be conditioned on some non-trivial constraint on the state of $X \in \mathcal{X}$ at τ . We can extract this passenger number-constraint from the set $\Psi^{(e)}$ which we have assigned to the edge $e := (n', n)$ from n' to n in G ; and translate it to some (closed) subset $\rho^{(G)}(e)$, with

$$\rho^{(G)} : E \rightarrow \{K \subset \mathcal{X} : K \text{ is a closed subset of } \mathcal{X}\},$$

of \mathcal{X} . This then allows us to compute

$$\mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma((i-1)_{\Delta\tau}) = n'\right)$$

in (4.8) above according to

$$\begin{aligned} \mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma((i-1)_{\Delta\tau}) = n'\right) = \\ \int_{\rho^{(G)}(e)} \text{pdf}\left(X(i_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q((i-1)_{\Delta\tau}) = q'; x\right) dx. \end{aligned} \quad (4.9)$$

Bottlenecks

Node Explosion of Propagation DAG

If at all, we are confronted with some finite propagation DAG G since we assume that all considered forecasts have finite time horizons. However, finite in this context does not necessarily mean manageable. The fact that many vehicles might perform some passenger number-dependent actions at the same time during the discrete time traversal of some forecast's complete time horizon, can explode the number of nodes in the different layers of G . This node explosion of course has many disadvantages; among which notably is the problem to store G for some simulation's a-posteriori analysis. That is why, we look at G 's branching in more detail next, before we will try to compute an upper bound for its number of nodes. Finally, we will provide some numerical example so as to illustrate our findings.

Branching. Branches in any propagation DAG result from the possibility of passenger number-dependent actions, which implies that some propagation DAG reduces to a simple path iff passenger number-dependent actions cannot be performed by the vehicles in the considered time horizon. More specifically, The number of different branches that may leave some node $n \in N$ in some propagation DAG G corresponds to the number $|n^\circ|$ of child nodes n has in G . This number is zero iff n is a leaf node of G . On the other hand, at least one branch must leave any internal node $n' \in \bar{N}$, with

$$\bar{N} := \left\{ n'' \in N : \mu^{(G)}(n'') < \max_{m \in N} \mu^{(G)}(m) \right\}, \quad (4.10)$$

due to the continuity of the considered dynamical process. At some simulation time, T -SHA might not transition to another mode any more. However, the valuations of all clocks then still progress in time.

Upper Bound for Number of Nodes. Prior to the actual computation of any propagation DAG G , we can compute an upper limit for its cardinality, i.e. for the number of nodes in G . A precondition for the occurrence of any passenger number-dependent action at some time $\tau \geq 0$ when T -SHA is in some mode $q \in \mathcal{Q}$ is the circumstance that some vehicle $v \in \mathcal{V}^{(g)}$, which is operated in some transportation grid $g \in \mathcal{G}$ and docked to some station $s \in \mathcal{S}$ in q for the purpose of boarding & alighting, either departs, parks, or changes its mission at τ . Now v can be docked to some station iff it is hold at some stop in g (refer to Def. 26 on p. 50 for the meaning of “stop”). The invariant number of stops $n^{(g)} \in \mathbb{N}_{\geq 0}$ in g in turn is completely defined by the specification of TN 's infrastructure. Thus, from the number $n^{(g)}$ of stops in g and the number $|\mathcal{V}^{(g)}|$ of vehicles operated in g taken alone, we know that no more than

$$m^{(g)} := \min(n^{(g)}, |\mathcal{V}^{(g)}|) + 1$$

passenger number-dependent actions can be performed in g at the same time (where “+1” refers to the situation where no vehicle . Then given the possibility that two or more passenger number dependent actions occur at the same time in both lines (with zero marginal probability in the original model dynamics at some isolated point in time; however with some zero or non-zero marginal probability now), we know that no more than 2^k , with $k := \sum_{g \in \mathcal{G}} m^{(g)}$, branches can leave any internal node $n \in \bar{N}$ in G . Finally, assuming that G has a tree structure, we can claim that

$$|N| \leq \sum_{i=0}^h 2^{ki}, \quad (4.11)$$

where

$$h := \max_{n \in N} \mu^{(G)}(n)$$

shall denote the height of G . This upper limit for the cardinality of G exponentially grows with rate 2^k with the number of discrete time steps chosen. However, note

that a tree structure for G is the most conservative assumption which we could have made here. Examples which show that G can have some more general and compact non-tree structures, where two or more nodes with the same height in G have some same node as child in common, can be easily created; and we leave it to the reader to create these examples as some kind of exercise.

Numerical Example. Assume that we would like to forecast the vehicle operational state of some simplistic transportation network, where two lines, l_1 and l_2 , connect three stations; cf. Fig. 4.3.1 below. The circular commuter train line l_1 connects the station s_1 with the station s_2 , and the circular line l_2 of a people mover connects the station s_2 with the station s_3 . Suppose that 13 vehicles are operated in l_1 , and one vehicle in l_2 . Moreover, suppose that one waypoint in l_1 is connected to s_1 for the purpose of boarding & alighting, and another waypoint to s_2 for the same purpose. Similarly, one waypoint in l_2 is connected to s_2 for the purpose of boarding & alighting, and another to s_3 for the same purpose. Thus, in total three vehicles can be docked to some station at the same time; two vehicles in l_1 and one vehicle in l_2 .

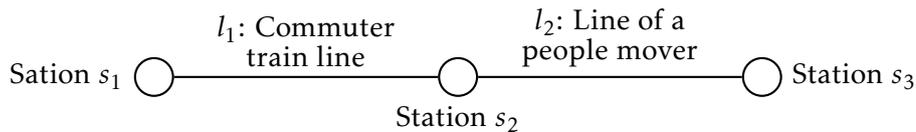


Figure 4.3.1: Schematic representation of some simplistic two-lines transportation network

If the complete forecast horizon covers 20 minutes and is discretized into fixed time intervals of 10 seconds each, then any propagation DAG G computed for some considered use case (initial state of T -SHA etc.) cannot have more than

$$|N| \leq \left(\sum_{i=0}^{120} 2^{(2+1)i} \right) \approx 10^{108}$$

nodes. Although a very conservative upper limit, this huge number underpins the necessity of sophisticated strategies, which we will introduce in Sec. 4.4 in form of some algorithmic bricks.

High-dimensionality of Decoupled SDEs

Fokker–Planck equations (FPEs) with reflective boundary conditions describe the time evolution of all passenger numbers/densities in all stations and vehicles docked to them in every mode of T -SHA; see Sec. 3.1.4 on p. 34. The numerical integration of these partial differential equations (PDEs) is everything else than straightforward assuming that the systems of stochastic differential balance equation from

which they were derived from have more than three dimensions; which is the situation in our model for every non-trivial use case: First of all note that not all numerical integration schemes (including the very popular Finite Difference method; see e.g. [Causon 2010] for a brief introduction to this method, and [Pichler 2013] for its application to the numerical solution of FPEs) can explicitly enforce the conservation of the considered probability flux; at least in their classical set ups. They must be run with very fine-granular spatial discretizations of the considered domain so as to approximately ensure all reflective boundary conditions within some acceptable error threshold. In theory, this error threshold can be chosen arbitrarily small. However, in practice very fine-granular discretizations mean that the numerical integrations become very quickly intractable. Further advancements e.g. in the development of multigrid methods (see e.g. [Zubair 2007] as illustration for the application of this method to the numerical integration of high-dimensional PDEs) may find a remedy in the future. Secondly, note that those numerical integration schemes such as the Finite Volume method (see e.g. [Causon 2011] for a brief introduction to this method) which manage to do so, are not easily extendible from common two or three dimensional applications to higher-dimensional problems due to the use of special operators such as the cross product in their present implementations.

A popular alternative to the numerical integration of any Fokker–Planck equation is to (i) sample realizations of the uncertain initial state of the considered continuous random variable (RV), and to (ii) simulate different state trajectories of the governing Itô process starting from every sampled initial state (while still enforcing all reflective boundary conditions): Whereas the simulation of the governing Itô process - or more specifically the simulation of all (discretized) Wiener paths by means of some number generator - starting from one particular initial state is relatively straightforward (see e.g. [Higham 2001] for some quick and practical introduction to this issue), sampling the initial state from some arbitrary density seems to be a non-trivial task that is active and still an open problem; at least for multi-dimensional RVs: For one-dimensional RVs this sampling might be achieved by employing the inverse transform sampling. Among the algorithms proposed for the sampling of high-dimensional RVs thus far, we mention the Metropolis-Hastings and the Gibbs sampler (also referred to as Monte Carlo methods; see e.g. [MacKay 1998] for a brief overview of/introduction to these numerical approaches), which can be integrated into what is called a Markov Chain Monte Carlo simulation [Brooks 2011]. Other more exotic sampling techniques might involve e.g. neural networks [Hoogerheide 2002].

Bricks for the Efficient Computation of Forecasts

Overview

In the preceding sections we have introduced a discrete time scheme for the propagation of *T-SHA*'s hybrid state along some so-called propagation DAG; in that

we confined the occurrences of all passenger number-dependent actions to some sequence of equidistantly-spaced discrete points in time which completely covers the considered forecast horizon. Although this discrete time scheme renders the computation of some forecast in principle tractable, it does not necessarily render this computation feasible in some reasonable amount of time given some limited computational (processing) constraints. In fact, for any non-trivial use case, we must expect that the number of nodes in our propagation DAG G exponentially grows as a function of the number of simulation time steps which discretize the considered forecast horizon. On the other hand, we are confronted with many high-dimensional SDEs which we have to numerically solve on one way or the other; or more specifically the Fokker–Planck equations derived thereof. That is why we will elaborate some bricks in the following sections which all aim at the above mentioned bottlenecks. Some of these bricks seem to be a natural consequence of the way we have specified T -SHA’s dynamics, and the way any forecast has to be computed by means of our discrete time scheme. They do not further simplify T -SHA’s dynamics, and we thus introduce them first. In particular we discuss

- the layer by layer computation of any forecast in Sec. 4.4.2 on p. 122ff.,
- the proper specification of all initial passenger numbers in Sec. 4.4.3 on p. 123, and
- the offline computation and utilization of so-called macro modes in Sec. 4.4.4 on p. 124ff.

As opposed to these first four bricks, the next two bricks require some further simplifications of T -SHA’s dynamics. In particular, we propose to ignore the time evolution of the valuations of all clocks which are associated with the operation of any parked vehicle in Sec. 4.4.5 on p. 126; some straightforward modification. The second brick that we will introduce in Sec. 4.4.6 on p. 127ff. however is far less trivial and its application might have much bigger impact on the feasibility of any forecast computation. More specifically, we propose to decouple all passenger flows in any mode $q \in Q$ of T -SHA between two consecutive discrete simulation time steps. As a consequence of this (canonical) decoupling approach, passenger number disturbances can no longer spread between any pair of two gathering points in some station and vehicles docked to this station in zero time. This latter dynamical restriction enables us to introduce three more bricks. The first two additional bricks propose the computation of some subset $\bar{\mathcal{G}} \subseteq \mathcal{G}$ of transportation grids whenever a new layer of nodes is appended to G . The presence of every vehicle $v \in \mathcal{V}^{(g)}$ operated in any $g \in \bar{\mathcal{G}}$ can be ignored in the computation of all remaining discrete time layers of the considered forecast since their ignorance can no longer affect the discrete state of/continuous passenger number on-board any target vehicle, nor the passenger number at any target gathering point.

For the computation of $\bar{\mathcal{G}}$, both bricks modify one and the same instantiation

of T -SHA's infrastructure graph G_i ; where this latter instantiation must be created before the actual forward propagation of T -SHA's hybrid state begins.

Definition 70 (Infrastructure Graph) T -SHA's infrastructure graph is a directed graph

$$G_i := (N_i, E_i),$$

with node set

$$N_i := \bigcup_{s \in \mathcal{S}} \left(s \times (P_{\text{gp}}^{(s)} \cup T_c^{(s)}) \right) \cup \bigcup_{g \in \mathcal{G}} \left(g \times (P_w^{(g)} \cup T_{\text{tr}}^{(g)}) \right),$$

and edge set

$$E := \mathcal{I} \cup \bigcup_{s \in \mathcal{S}} \left\{ ((s, a), (s, b)) : (a, b) \in E_{\text{st}}^{(s)} \right\} \cup \bigcup_{g \in \mathcal{G}} \left\{ ((g, a), (g, b)) : (a, b) \in E_{\text{gr}}^{(g)} \right\}.$$

The first additional brick (“disconnect platforms in stations”; see Sec. 4.4.7 on p. 140ff.) removes all transfer corridors from any station in G_i , whenever this station cannot transmit passenger number disturbances between all transportation grids which are connected to it in the remaining forecast horizon. On the other hand, the second additional brick (“unfold vehicle trajectories”; see Sec. 4.4.8 on p. 144) removes any stop from any transportation grid in G_i , whenever this stop cannot be served by any vehicle in the remaining forecast horizon. As a consequence of this removal of transfer corridors from stations and stops from transportation grids, some transportation grids (namely all $g \in \bar{\mathcal{G}}$) become disconnected from any station which comprises some target gathering point and from any transportation grid which accommodates some target vehicle.

For some subgraph G_i of T -SHA's infrastructure graph (refer to Sec. 1.5 on p. 1.5ff. for the meaning of “subgraph”), Alg. 26 below computes $\bar{\mathcal{G}}$.

Algorithm 26 Specification of the function

TARGETSEARCH :

$$\left\{ G_i : G_i \text{ is some subgraph of } T\text{-SHA's infrastructure graph} \right\} \rightarrow 2^{\mathcal{G}},$$

which returns a subset $\bar{\mathcal{G}} \subseteq \mathcal{G}$ of transportation grids, with

$$\bar{\mathcal{G}} := \text{TARGETSEARCH}(G_i),$$

s.t. all $g \in \bar{\mathcal{G}}$ are disconnected from any $g' \in \mathcal{G}_t$ and from any $s' \in \mathcal{S}_t$ in G .

1: **function** TARGETSEARCH(G_i)

 // Step 1:

2: $\bar{\mathcal{G}} \leftarrow \emptyset$

3: $\bar{\mathcal{S}} \leftarrow \emptyset$

4: $\mathcal{S}_t \leftarrow \left\{ s \in \mathcal{S} : \exists p \in P_{\text{gp}}^{(s)} \text{ s.t. } p \in P_t^{(s)} \right\}$

```

5:   $\mathcal{G}_t \leftarrow \{g' \in \mathcal{G} : \exists v \in \mathcal{V}^{(g)} \text{ s.t. } v \in \mathcal{V}_t\}$ 
    // Step 2:
6:  for
    every  $s \in \mathcal{S} \setminus \mathcal{S}_t$ 
7:  do
8:     $\Psi \leftarrow \emptyset$ 
9:    for
    every platform  $p \in P_p^{(s)}$ 
10:   do
11:     for
    every  $s' \in \mathcal{S}_t$ 
12:    do
13:       $p_1 \leftarrow$  some arbitrary gathering point in  $s'$ 
14:      if
     $p$  is weakly connected to  $p_1$  in  $G_i$ 
15:      then
16:         $\Psi \leftarrow \Psi \cup p_1$ 
17:      end if
18:    end for
19:    for
    every  $g \in \mathcal{G}_t$ 
20:    do
21:       $p_2 \leftarrow$  some arbitrary waypoint in  $g_i$ 
22:      if
     $p$  is weakly connected to  $p_2$  in  $G_i$ 
23:      then
24:         $\Psi \leftarrow \Psi \cup p_2$ 
25:      end if
26:    end for
27:  end for
28:  if
     $\Psi = \emptyset$ 
29:  then
30:     $\bar{\mathcal{S}} \leftarrow \bar{\mathcal{S}} \cup \{s\}$ 
31:     $N_i \leftarrow N_i \setminus \bigcup_{s \in \mathcal{S}} (s \times T_{tc}^{(s)})$ 

```

```

32:          $E_i \leftarrow E_i \cap (N_i)^2$ 
33:          $G_i \leftarrow (N_i, E_i)$ 
34:     end if
35: end for
    // Step 3:
36: for
    every  $g \in \mathcal{G} \setminus \mathcal{G}_t$ 
37: do
38:      $\Psi \leftarrow \emptyset$ 
39:      $p \leftarrow$  some arbitrary waypoint in  $g$ 
40:     for
    every  $s \in \mathcal{S}_t$ 
41:     do
42:          $p_1 \leftarrow$  some arbitrary gathering point in  $s$ 
43:         if
     $p$  is weakly connected to  $p_1$  in  $G_i$ 
44:         then
45:              $\Psi \leftarrow \Psi \cup p_1$ 
46:         end if
47:     end for
48:     for
    every  $g' \in \mathcal{G}_t$ 
49:     do
50:          $p_2 \leftarrow$  some arbitrary waypoint in  $g'$ 
51:         if
     $p$  is weakly connected to  $p_2$  in  $G_i$ 
52:         then
53:              $\Psi \leftarrow \Psi \cup p_2$ 
54:         end if
55:     end for
56:     if
     $\Psi = \emptyset$ 
57:     then
58:          $\bar{\mathcal{G}} \leftarrow \bar{\mathcal{G}} \cup g$ 
59:     end if
60: end for

```

```

61:   return ( $\bar{G}$ ,  $\bar{S}$ ,  $G_i$ )
62: end function

```

Finally the last brick, which we will introduce in Sec. 4.4.9 on p. 146ff., promotes the discretization of all passenger number vectors in the considered transportation networks; which last measure becomes possible since the above mentioned canonical decoupling of all passenger flows in all modes $q \in \mathcal{Q}$ of *T-SHA* produces many but relatively low-dimensional balance equations.

Note that we will not discuss heuristics here, which e.g. propose to synchronize different vehicle departures from stops, or to implement constant and thus passenger number-independent dwell times at stops. In general, these heuristics are hard to justify and the applicability of most of them depends on the particular use case. In this context, note that each of our above-mentioned brick employs some rigorous mathematical reasoning.

Layer-By-Layer Computation of a Forecast

In practice, there are two very different approaches to the computation of any forecast, assuming that our proposed discrete time scheme is chosen: In the first approach, the propagation DAG G is computed in its entirety prior to the actual forward propagation of all passenger number densities. The primary benefit of this approach might be the fact that it clearly separates two fundamentally different computations, i.e. the computation of G on the one hand, and amongst others the numerical integrations coming along with the forward propagation of all densities in G on the other hand; beneficial here in a sense that it is in principle possible to implement both computations as two separate but communicating black boxes which can be maintained and/or optimized in isolation given the specification of their interface. The major drawback of this first approach however is, that - regarded as a whole - it most likely computes more than it has to do: Prior to the computation of *T-SHA*'s (hybrid) state at the end of the considered forecast horizon, some stop criterion deduced from the set of considered forecast objectives might already be fulfilled. For instance, at simulation time step 20 out of 120 simulation time steps it is already known that the cumulative passenger number of some platform exceeds some critical number with some positive marginal probability. Moreover, it might be the situation that some possible transitions of *T-SHA*'s timed mode have zero marginal probabilities, in which case we can neglect them. Besides the above two points, it might also be the case that we do not want to store G in its entirety for some a-posteriori analysis. Instead, it might be enough to compute some values of interest whenever some new layer is appended to G , and to disregard/delete all preceding layers of nodes once they are not needed any more.

Proper Specification of Initial Passenger Loads

In *T-SHA* all passengers stick to their trip profiles while travelling in the considered network, which does not mean that they cannot change their trip profiles though. In this context, it is more than consequent to specify some initial non-zero numbers of passengers who travel according to some particular trip profile $y \in \mathcal{Y}$ at some gathering point, say p , with some positive marginal probability iff these passengers could have accounted p before (e.g. from some stopped vehicle, and not fallen from the sky). In this context, we say that the initial passenger number at some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ is specified properly by some marginal density $\text{pdf}_{\text{gp}}^{(s,t)}(\cdot)$, with

$$\text{pdf}_{\text{gp}}^{(s,p)} : \mathcal{M}^{(s,p)} \rightarrow \mathbb{R}_{\geq 0},$$

iff the following implication holds (refer to Tab. 3.3 on p. 49 for an overview of all different corridor types, and to Sec. 3.4.3 on p. 61ff. for the meaning and correct interpretation of all routing matrices): The fact that

$$\sum_{i \in \mathcal{Y}} R^{(s,t)}[y, i] = 0$$

for $t \in (\bullet p \setminus T_{\text{ac}}^{(s)})$, and

$$\sum_{i \in \mathcal{Y}} R^{(s,(t',w))}[i, y] = 0$$

for $t' \in (\bullet p \cap T_{\text{ac}}^{(s)})$ as well as all vehicle missions $w \in \mathcal{W}^{(g)}$ set up for the vehicle operation in some transportation grid $g \in \mathcal{G}$, must imply

$$\int_{x \in \{x' \in \mathcal{M}^{(s,p)} : x'[y] > 0\}} \text{pdf}_{\text{gp}}^{(s,p)}(x) \, dx = 0.$$

Similarly, we say that the initial passenger number on-board some vehicle $v \in \mathcal{V}^{(g)}$ which is operated in some transportation grid $g \in \mathcal{G}$ is properly specified by some density $\text{pdf}_{\text{vh}}^{(v)}$, with

$$\text{pdf}_{\text{vh}}^{(v)} : \mathcal{U}^{(v)} \rightarrow \mathbb{R}_{\geq 0},$$

iff the following two implications hold: The fact that v is initially parked according to the specification of *T-SHA*'s initial mode q_0 must imply

$$\int_{x \in \{x' \in \mathcal{U}^{(v)} : x' > k, k > 0\}} \text{pdf}_{\text{vh}}^{(s,p)}(x) \, dx = 0. \quad (4.12)$$

Moreover, the fact there does not exist some boarding corridor $t \in T_{\text{bc}}^{(s)}$ in some station $s \in \mathcal{S}$ s.t. (i) t is connected to g according to *TN*'s interface specification, and (ii)

$$\sum_{y \in i} R^{(s,(t,w))}[i, y] > 0$$

for vehicle mission $w \in \mathcal{W}^{(g)}$, must also imply (4.12) above if v initially executes w according to q_0 .

Now if we can assume that all initial passenger numbers are *specified properly* as described above, and we moreover know that all passengers stick to their trip profiles while travelling in the considered transportation network, then there is absolutely no need to consider every trip profile everywhere in our model of this network: We include only those trip profiles in the domain specification of some particular passenger number (at some gathering point or on-board some vehicle) iff some passenger of this trip profile can contribute to it due to her presence. This adjustment of all passenger number-domains immediately impacts the complexity of all computations since it results into the specification of lower-dimensional systems of stochastic differential balance equations; compared to the original situation where we had to consider every trip profile everywhere (refer to Sec. 3.6 on p. 75ff. for the specification of all balance equations).

Macro Modes

The passenger routing in *T-SHA* is static, which does not imply that passengers cannot change their trip profiles. However if they do so, then this re-routing neither depends on *T-SHA*'s (hybrid) state nor on its simulation time. Moreover, note that all passenger flows in some mode $q \in \mathcal{Q}$ of *T-SHA* are completely defined by the subset of all vehicles which are docked to some station for the purpose of boarding & alighting, and the missions these vehicles execute. It then follows that we assign to several modes the same set of balance equations and thus associate with them the same passenger flow-dynamics. More specifically, we assign to all modes $q \in \mathcal{Q}$ of *T-SHA* with a belonging to the same macro mode $\delta \in \Delta$ the same set of balance equations, which define the time evolution of all passenger numbers at all gathering points in all stations and on-board all vehicles which are docked to them as long as *T-SHA* does not transition to another macro mode $\delta' \in \Delta$, with $\delta' \neq \delta$.

Definition 71 (Macro Mode of *T-SHA*) Every macro mode $\delta \in \Delta$, with

$$\Delta := \left\{ A \subseteq 2^{\Delta} : (v, p_1, w_1) \in A \wedge (v, p_2, w_2) \in A \text{ implies } p_1 = p_2 \wedge w_1 = w_2 \right\}$$

and

$$\underline{\Delta} := \bigcup_{g \in \mathcal{G}} \mathcal{V}^{(g)} \times \left\{ p \in P_w^{(g)} : p \text{ is a stop in } g \text{ according to Def. 26 on p. 50} \right\} \times \mathcal{W}^{(g)},$$

of *T-SHA* is a set that defines

- which vehicle is docked to some station,
- the position of every vehicle which is docked to some station, and
- the mission of every vehicle which is docked to some station,

when T -SHA is in δ . More specifically, some vehicle $v \in \mathcal{V}^{(g)}$ operated in some transportation grid $g \in \mathcal{G}$ is docked to some station $s \in \mathcal{S}$ when T -SHA is in δ iff there is some pair of a waypoint $p \in P_w^{(g)}$ and a mission $w \in \mathcal{W}^{(g)}$ s.t. $(v, p, w) \in \delta$. Assuming this to be true, then p corresponds to v 's position, and w corresponds to v 's mission in δ .

With the above introduction of macro modes, it then follows that we do not have to compute the marginal conditional probability density

$$\text{pdf}\left(X\left((i+1)\Delta\tau\right) \middle| \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q(i\Delta\tau) = q'\right)$$

for every (different) mode $q' \in \mathcal{Q}$ so as to compute in (4.7) on p. 113 the marginal density

$$\text{pdf}\left(X\left((i+1)\Delta\tau\right) \middle| \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right)$$

for $X(i\Delta\tau)$. Instead we do only have to compute this density if it was not already computed (in the same time layer) for some other mode which has the same macro mode in common before.

Upper Limit. Def. 71 for a macro mode of T -SHA above, implies that the number of different macro modes $|\Delta|$ is limited from above by

$$|\Delta| \leq \prod_{g \in \mathcal{G}} \left(\underbrace{\text{No vehicle is docked to any station}}_1 + \sum_{i=1}^{\min(n^{(g)}, |\mathcal{V}^{(g)}|)} \underbrace{\frac{n^{(g)}!}{(n^{(g)} - i)!} \left(|\mathcal{W}^{(g)}|\right)^i}_{i \text{ vehicles are docked to some station}} \right) \quad (4.13)$$

where $\mathcal{V}^{(g)}$ is the finite set of vehicles which are operated in some transportation grid $g \in \mathcal{G}$, $\mathcal{W}^{(g)}$ is the finite set of missions defined for the vehicle operation in g , and

$$n^{(g)} := \left| \left\{ p \in P_w^{(g)} : p \text{ is a stop in } g \text{ according to Def. 26 on p. 50} \right\} \right|$$

denotes the invariant number of stops in g . Every stop in g is either empty or occupied by at maximum one vehicle at the same time. Assuming that this stop is occupied by some stopped vehicle for the purpose of boarding & alighting, then the mission of this vehicle defines which passengers want to board it, and which passengers want to alight from it. Thus, although the number of nodes in any propagation DAG G might explode as a function of T -SHA's discrete simulation time step, the number of different balance equations that have to be solved during the layer by layer forward propagation of all densities in G cannot.

Numerical Example. Look at the simplistic transportation network from Fig. 4.3.1 on p. 116, once again, and assume that

- the commuter train line l_1 accommodates 13 vehicles in operation, whereas only one people mover is in operation in line l_2 ,
- one stop in l_1 is connected to the station s_1 and another stop in l_1 to the station s_2 , and
- one stop in l_2 is connected to the station s_2 and another stop in l_2 to the station s_3 .

In addition, suppose that four different missions have been specified for the complete vehicle operation in both lines; two missions for the commuter train line, and another two missions for the people mover line. From Eqn. 4.13 above, we then know that

$$|\Delta| \leq \overbrace{\left(1 + \sum_{i=1}^2 \frac{2!}{(2-i)!} 2^i \right)}^{\text{Commuter train line}} \overbrace{\left(1 + \sum_{j=1}^1 \frac{2!}{(2-i)!} 2^i \right)}^{\text{People mover line}} = 65,$$

which looks much less alarming than the 10^{108} nodes that we might encounter during the computation of the propagation DAG.

Distributed Computing. The upper limit for the different number of macro modes for *T-SHA* in Eqn. 4.13 above, can be used as a starting point for the parallelization of any forecast algorithm: This upper limit defines the maximum number of different balance equations that have to be resolved during the layer by layer traversal of some propagation DAG G . One might thus want to allocate the numerical integration of any particular system of balance equations to one dedicated process/computing machine, plus another dedicated process/computing machine for the communication of any numerical integration result and the computation of any new layer of nodes in G . Moreover note that Eqn. 4.13 is very conservative in that it neglects the fact that not all vehicles might execute all missions; something which can be easily checked offline by analysing the specification of *T-SHA*'s dispatch plan given its initial state. An unfolding of all vehicle trajectories (considered in isolation, while disregarding all other vehicles; also computed offline) can even give some less conservative upper bound when the finite time horizon of the forecast is also taken into account.

Static Clock Valuations for All Parked Vehicles

In practice, the operation of some transportation line might define some minimum turnaround times for all vehicles that park at one of its ends upon their arrivals there so as to e.g. enforce some labour laws (time to go to wash rooms, etc.). Now in

the dynamics of *T-SHA* we might do not want/have to include these fine granular particularities, which latter ignorance means that we do not couple the dispatch time of any vehicle to the valuation of the clock (= elapsed parking time) which we associate with its operation. Assuming this to be true, it makes sense to slightly alter *T-SHA*'s dynamics with the goal to reduce the possible number of alternative timed modes in any computed propagation DAG: We do not let the valuation of the clock associated with any parked vehicle linearly progress in simulation time but keep it constant instead. In this context, note that we also did not include the clock valuation of any parked vehicle in its departure condition in Alg. 19 on p. 97 since we do not consider dynamic minimum parking times in our use cases; but assume a strict adherence to a static time table (= dispatch plan).

Canonical Decoupling of All Passenger Flows

Preliminary Considerations. Our *T-SHA* does not capture the geometry of any physical space in the considered transportation network such as the length and the curvature of some platform edge, or the spatial distribution of the passengers therein. Thus, if we want to include the (spatial) distribution of all passengers at some gathering point or on-board some vehicle's passenger compartment in our computations, then we have to make some idealized assumptions; such as the following one which we employ here for the canonical decoupling of all passenger flows in any mode $q \in Q$ of *T-SHA*. Look at Fig. 4.4.1 below: We assume that the geometry of every gathering point in every station resembles a circular area which accommodates all passengers. Similarly, we assume that the geometry of the passenger compartment of every vehicle resembles a circular area. We moreover assume that all passengers are spatially equally distributed in these circular areas w.r.t. their different trip profiles at every discrete simulation time step $i \in \mathbb{N}_{\geq 0}$ when we let *T-SHA* change its mode.

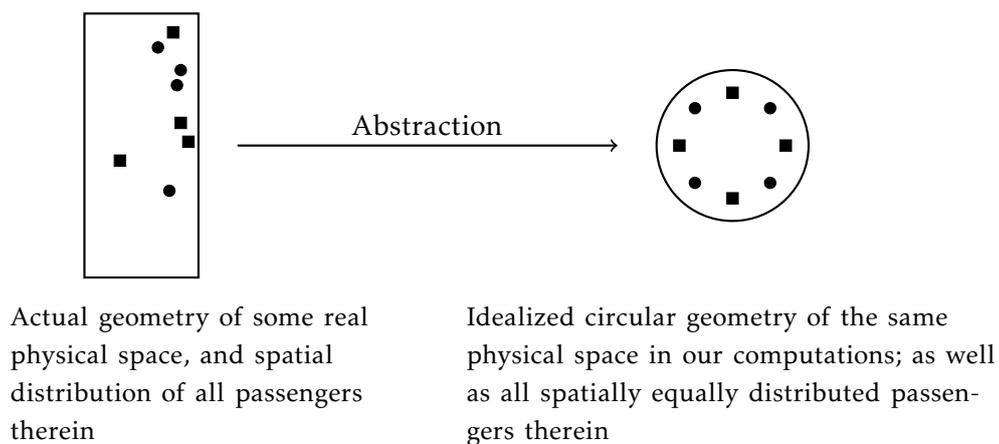


Figure 4.4.1: Comparison: geometry of some physical space and the spatial distribution of all passengers therein in 1) the real network and 2) our computations

Our above assumptions imply that at any simulation time step i the number of passengers, as well as the distribution of this number w.r.t. the passengers' different trip profiles, in any (pie) slice taken away from the idealized circular geometry of some gathering point or some vehicle's passenger compartment is completely defined by the relative size of this slice w.r.t. the geometry's circular area; cf. Fig. 4.4.2 below.

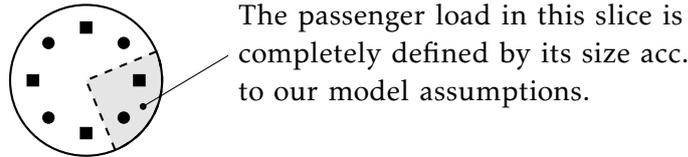


Figure 4.4.2: If the geometries of all gathering points in all stations and of all vehicles' passenger compartments resemble circular areas, and the passengers are spatially equally distributed in these circular areas w.r.t. their different trip profiles, then the number of passengers in any slice taken away from such a circular area is completely defined by its relative size.

Basic Principle. We assume that the geometry of every gathering point $p \in P_{gp}^{(s)}$ in every station $s \in \mathcal{S}$ resembles a circular area, which is divided into as many non-overlapping slices of equal size each, as there are corridors connected to it. For corridor $t \in \bullet p \cup p \bullet$ connected to p we thus cut away from p one slice of relative size

$$\gamma^{(s,p)} := \frac{1}{|\bullet p \cup p \bullet|} \tag{4.14}$$

w.r.t. p 's circular area. We will refer to this slice in all our further considerations by the triple (s, p, t) as it is illustrated in Fig. 4.4.3 below.

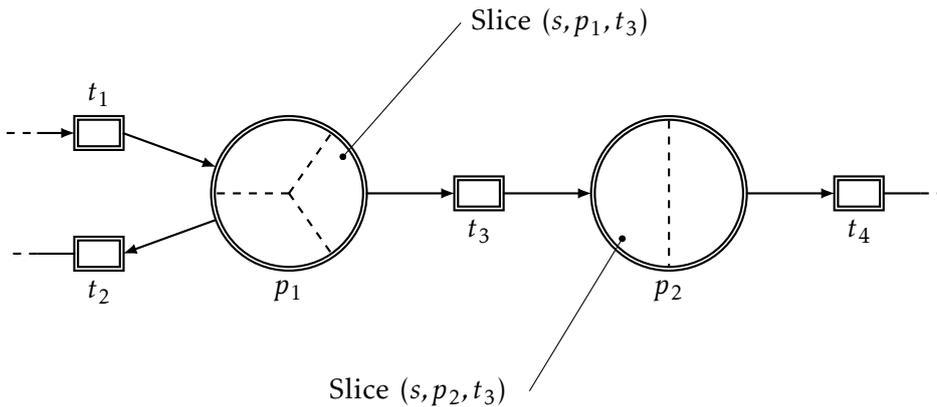


Figure 4.4.3: Extract of some station $s \in \mathcal{S}$: Every gathering point has an idealized circular geometry, which is divided into as many non-overlapping slices of equal size each as there are corridors connected to it.

We also assume that the geometry of the passenger compartment of every ve-

hicle $v \in \mathcal{V}$ resembles a circular area. This circular area is divided into two non-overlapping slices of relative size of one half each: one slice is dedicated to the boarding of passengers (we will use the subscript “b” to refer to it), and the other slice is dedicated to the alighting of passengers (we will use the subscript “a” to refer to it).

At every simulation time step $i \in \mathbb{N}_{\geq 0}$ during our discrete computation of some forecast, we assume that all passengers are equally distributed among the different slices at every gathering point in every station and on-board every vehicle as it is illustrated in Fig. 4.4.4 below.

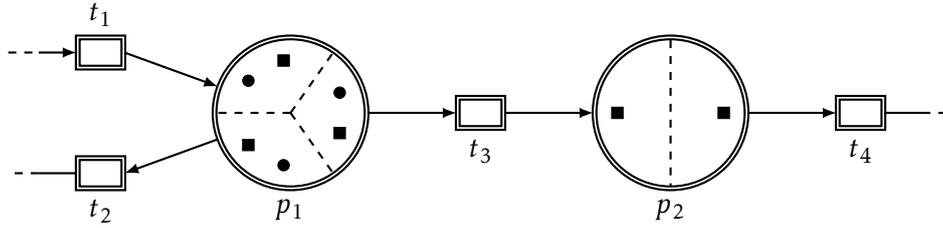


Figure 4.4.4: Extract of some station $s \in S$ adopted from Fig. 4.4.3 above: All passengers at some gathering point (on-board some vehicle) are equally distributed among all slices taken away from the circular geometry of this gathering point (vehicle’s passenger compartment) whenever T -SHA may change its mode.

Thus, if we denote by $\vec{M}^{(s,p,t)}(\tau)[y]$, with

$$\vec{M}^{(s,p,t)} : \mathbb{R}_{\geq 0} \rightarrow \vec{\mathcal{M}}^{(s,p)} := \left\{ M \in \mathcal{M}^{(s,p)} : \sum_{y \in \mathcal{Y}} M[y] \leq \gamma^{(s,p)} \lambda_c^{(s,p)} \right\},$$

the number of passenger in (s,p,t) with trip profile $y \in \mathcal{Y}$ at simulation time $\tau \geq 0$, we then assume that

$$\vec{M}^{(s,p,t)}(i_{\Delta\tau}) = \gamma^{(s,p)} M^{(s,p)}(i_{\Delta\tau}) \quad (4.15)$$

for all discrete simulation time steps $i \in \mathbb{N}_{\geq 0}$; where $\Delta\tau > 0$ denotes the length of the fixed time interval which separates i from $i+1$. Note that this simple projection implies

$$\text{pdf}\left(\vec{M}^{(s,p,t)}(i_{\Delta\tau}); \gamma^{(s,p)} k\right) = \text{pdf}\left(M^{(s,p)}(i_{\Delta\tau}); k\right), \forall k \in \mathcal{M}^{(s,p)}. \quad (4.16)$$

Similarly, if we denote by $\vec{U}_a^{(v)}(\tau)[y]$ and $\vec{U}_b^{(v)}(\tau)[y]$, with

$$\vec{U}_a^{(v)}, \vec{U}_b^{(v)} : \mathbb{R}_{\geq 0} \rightarrow \vec{\mathcal{U}}^{(v)} := \left\{ U \in \mathcal{U}^{(v)} : \sum_{y \in \mathcal{Y}} U[y] \leq 0.5 \lambda_c^{(v)} \right\},$$

the number of passengers with trip profile y in the slice which we (i) take away from the circular passenger compartment of some vehicle $v \in \mathcal{V}$, and (ii) dedicate to the alighting (\vec{U}_a) or boarding (\vec{U}_b) of all passengers, we then assume that

$$\vec{U}_a^{(v)}(i_{\Delta\tau}) = 0.5 U^{(v)}(i_{\Delta\tau}), \quad (4.17)$$

and

$$\vec{U}_b^{(v)}(i_{\Delta\tau}) = 0.5 U^{(v)}(i_{\Delta\tau}). \quad (4.18)$$

These two above projections in turn imply

$$\text{pdf}\left(\vec{U}_a^{(v)}(i_{\Delta\tau}); 0.5k\right) = \text{pdf}\left(U_a^{(v)}(i_{\Delta\tau}); k\right), \forall k \in \mathcal{U}^{(v)}, \quad (4.19)$$

and

$$\text{pdf}\left(\vec{U}_b^{(v)}(i_{\Delta\tau}); 0.5k\right) = \text{pdf}\left(U_b^{(v)}(i_{\Delta\tau}); k\right), \forall k \in \mathcal{U}^{(v)}. \quad (4.20)$$

Now figuratively speaking we install impenetrable walls between the different slices in every circular area pertaining to some gathering point or some vehicle's passenger compartment during the complete open time interval

$$\Gamma_i := (i_{\Delta\tau}, (i+1)_{\Delta\tau}),$$

which separates every discrete simulation time step $i \in \mathbb{N}_{\geq 0}$ from its successor $i+1$. This isolation of all slices from neighbouring slices within the same circular area in Γ_i has some important consequence: It implies that the passenger numbers in all slices with a belonging to the same circular area evolve independently from each other in Γ_i if we assume that for station $s \in \mathcal{S}$ (i)

- every alighting corridor $t \in T_{ac}^{(s)}$ in s does not connect the passenger compartment of some vehicle $v \in \mathcal{V}$ which is stopped at the waypoint *t in some mode $q \in \mathcal{Q}$ of T -SHA to the gathering point t^* , but the unique slice from v which is dedicated to the alighting of all passengers to the slice (s, t^*, t) from t^* ;
- every boarding corridor $t \in T_{bc}^{(s)}$ in s does not connect the gathering point *t to the passenger compartment of some vehicle $v \in \mathcal{V}$ which is stopped at the waypoint t^* in some mode $q \in \mathcal{Q}$ of T -SHA, but the slice $(s, {}^*t, t)$ from *t to the unique slice from v which is dedicated to the boarding of all passengers;
- every entrance corridor $t \in T_{ec}^{(s)}$ in s does not connect the exterior of the considered network to the gathering point t^* , but this exterior to the slice (s, t^*, t) from t^* ;
- every exit corridor $t \in T_{xc}^{(s)}$ in s does not connect the gathering point *t to the exterior of the considered network, but the slice $(s, {}^*t, t)$ from *t to this exterior;
- every transfer corridor $t \in T_{tc}^{(s)}$ in s does not connect the gathering point *t to the gathering point t^* , but the slice $(s, {}^*t, t)$ from *t to the slice (s, t^*, t) from t^* ;

and (ii) every (autonomous) passenger flow into, out of, or circulating within s through

- any alighting corridor $t \in T_{ac}^{(s)}$ is not a function of $M^{(s,t^*)}$ and $U^{(v)}$ for vehicle $v \in \mathcal{V}$, assuming that v is stopped at t^* for the purpose of boarding & alighting in some mode $q \in \mathcal{Q}$ of T -SHA, but of $\vec{M}^{(s,t^*,t)}$ and $\vec{U}_a^{(v)}$;
- any boarding corridor $t \in T_{bc}^{(s)}$ is not a function of $M^{(s,t^*)}$ and $U^{(v)}$ for vehicle $v \in \mathcal{V}$, assuming that v is stopped at t^* for the purpose of boarding & alighting in some mode $q \in \mathcal{Q}$ of T -SHA, but of $\vec{M}^{(s,t^*,t)}$ and $\vec{U}_b^{(v)}$;
- any entrance corridor $t \in T_{ec}^{(s)}$ is not a function of $M^{(s,t^*)}$, but of $\vec{M}^{(s,t^*,t)}$;
- any exit corridor $t \in T_{xc}^{(s)}$ is not a function of $M^{(s,t^*)}$, but of $\vec{M}^{(s,t^*,t)}$;
- any transfer corridor $t \in T_{tc}^{(s)}$ is not a function of $M^{(s,t^*)}$ and $M^{(s,t^*)}$, but of $\vec{M}^{(s,t^*,t)}$ and $\vec{M}^{(s,t^*,t)}$.

Note that there is some disaccord in our argumentation so far: On the one hand, we demand in Eqn. 4.15, 4.17, and 4.18 above, that at every discrete simulation time step $i \in \mathbb{N}_{\geq 0}$ all passengers at every gathering point in all stations and on-board every vehicle are equally distributed among the different slices on their idealized circular areas. On the other hand, we install impenetrable walls which shall isolate all neighbouring slices with a belonging to the same circular area in the open time intervals that separate every pair of two succeeding discrete simulation time steps. We then state that the consequence of this isolation is that the passenger numbers in some slice evolve independently from the passenger numbers in all neighbouring slices taken away from the same circular area; some observation which is illustrated in Fig. 4.4.5 below.

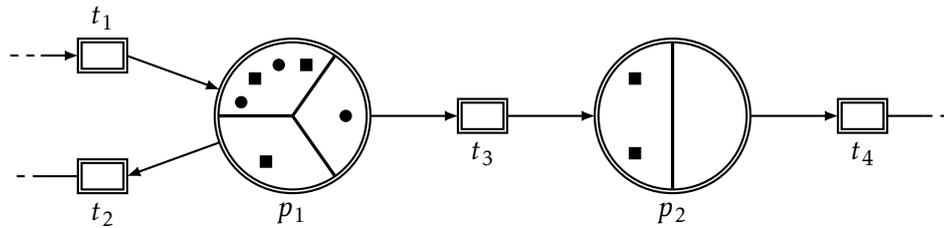


Figure 4.4.5: Extract from some station $s \in \mathcal{S}$: Compared to the situation depicted in Fig. 4.4.4 above, impenetrable walls now separate all slices from neighbouring slices within the idealized circular geometry of the same gathering point (vehicle's passenger compartment). This major difference indicated by solid lines implies that the passenger numbers in the different slices may evolve independently from each other; starting from the same initial distribution.

We fix this above inconsistency in that we literally speaking redistribute all passengers at every circular area among its different slices at every discrete simulation time step $i \in \mathbb{N}_{\geq 0}$ by hand; in that, we remove all impenetrable walls at every simulation time $\tau = i_{\Delta} \tau$ which latter intervention in T -SHA's passenger flow-dynamics then initiates - at least when following our imagination - some infinitely

fast running diffusion process (of independent random variables) which starts and ends at τ . In other words, we define at every discrete simulation time step $i \in \mathbb{N}_{\geq 0}$:

$$\text{pdf}\left(M^{(s,p)}(i_{\Delta\tau}); k\right) := \int_{\underline{\mathcal{M}}^{(s,p,k)}} \prod_{t \in (\bullet p \cup p \bullet)} \text{pdf}\left(\vec{M}^{(s,p,t)}(i_{\Delta\tau}); [k_1; k_2; \dots; k_{|\bullet p \cup p \bullet|}]\right) d(k_1, k_2, \dots, k_{|\bullet p \cup p \bullet|}) \quad (4.21)$$

for gathering point $p \in P_{\text{gp}}^{(s)}$ in every station $s \in \mathcal{S}$ and all $k \in \mathcal{M}^{(s,p)}$; where

$$\underline{\mathcal{M}}^{(s,p,k)} := \left\{ (k_1, k_2, \dots, k_{|\bullet p \cup p \bullet|}) \in (\mathbb{R}_{\geq 0})^{|\bullet p \cup p \bullet|} : \sum_{j=1}^{|\bullet p \cup p \bullet|} k_j = k \right\}. \quad (4.22)$$

Similarly, we define at every i :

$$\text{pdf}\left(U^{(v)}(i_{\Delta\tau}); k'\right) := \int_{\underline{\mathcal{U}}^{(v,k')}} \text{pdf}\left(\vec{U}_a^{(v)}(i_{\Delta\tau}); k_1\right) \text{pdf}\left(\vec{U}_b^{(v)}(i_{\Delta\tau}); k_2\right) d(k_1, k_2) \quad (4.23)$$

for vehicle $v \in \mathcal{V}$ and all $k' \in \mathcal{U}^{(v)}$; where

$$\underline{\mathcal{U}}^{(v,k')} := \{(k_1, k_2) \in (\mathbb{R}_{\geq 0})^2 : k_1 + k_2 = k'\}. \quad (4.24)$$

Note that the above timeless re-distribution of passengers in some confined physical space cannot be observed in practice. Even if it is observed that passengers distribute themselves equally e.g. on some platform, then this re-distribution is a continuous process which consumes time and probably never reaches some final stable state due the fact that passenger enter and leave the considered space.

Once the above re-distribution step is done, we again equally distribute $M^{(s,p)}$ among all slices from p according to Eqn. 4.15; $U^{(v)}(i_{\Delta\tau})$ among $\vec{U}_a^{(v)}(i_{\Delta\tau})$ according to Eqn. 4.17 among and $\vec{U}_b^{(v)}(i_{\Delta\tau})$ according to Eqn. 4.18. From now on everything repeats; which brings us to the proper execution of our canonical decoupling approach next: At every new simulation time step $i \in \mathbb{N}_{\geq 0}$ do:

- For all gathering points $p \in P_{\text{gp}}^{(s)}$ in all stations $s \in \mathcal{S}$:

1. Distribute: compute

$$\text{pdf}\left(\vec{M}^{(s,p,t)}(i_{\Delta\tau})\right)$$

from

$$\text{pdf}\left(M^{(s,p)}(i_{\Delta\tau})\right)$$

according to (4.16) for all corridors $t \in \bullet p \cup p \bullet$

2. Propagate: compute

$$\text{pdf}\left(\vec{M}^{(s,p,t)}\left((i+1)\Delta\tau\right)\right)$$

from

$$\text{pdf}\left(\vec{M}^{(s,p,t)}\left(i\Delta\tau\right)\right);$$

see specification of corresponding balance equation below; replace $\partial\mathcal{X}$ by $\partial\vec{M}^{(s,p)}$ in Eqn. 3.10 on p. 38 for the specification of the reflective boundary condition

3. Re-distribute: compute

$$\text{pdf}\left(M^{(s,p)}\left((i+1)\Delta\tau\right)\right)$$

according to Eqn. 4.21

• For all vehicles $v \in \mathcal{V}$:

1. Distribute:

– compute

$$\text{pdf}\left(\vec{U}_a^{(v)}\left(i\Delta\tau\right)\right)$$

from

$$\text{pdf}\left(U^{(v)}\left(i\Delta\tau\right)\right)$$

according to Eqn. 4.19

– compute

$$\text{pdf}\left(\vec{U}_b^{(v)}\left(i\Delta\tau\right)\right)$$

from

$$\text{pdf}\left(U^{(v)}\left(i\Delta\tau\right)\right)$$

according to Eqn. 4.20

2. Propagate:

– compute

$$\text{pdf}\left(\vec{U}_a^{(v)}\left((i+1)\Delta\tau\right)\right)$$

from

$$\text{pdf}\left(\vec{U}_a^{(v)}\left(i\Delta\tau\right)\right);$$

see specification of corresponding balance equation below; replace $\partial\mathcal{X}$ by $\partial\vec{U}^{(v)}$ in Eqn. 3.10 on p. 38 for the specification of the reflective boundary condition

– compute

$$\text{pdf}\left(\vec{U}_b^{(v)}((i+1)\Delta\tau)\right)$$

from

$$\text{pdf}\left(\vec{U}_b^{(v)}(i\Delta\tau)\right);$$

see specification of corresponding balance equation below; replace $\partial\mathcal{X}$ by $\partial\vec{U}^{(v)}$ in (3.10) on p. 38 for the specification of the reflective boundary condition

3. Re-distribute: compute

$$\text{pdf}\left(U^{(v)}((i+1)\Delta\tau)\right)$$

according to Eqn. 4.23

Note that $\text{pdf}(M^{(s,p)}(0))$ can be computed from pdf_0 for all gathering points $p \in P_{\text{gp}}^{(s)}$ in all stations $s \in \mathcal{S}$, where pdf_0 is part of *T-SHA*'s specification according to Def. 11 on p. 34. Similarly, $\text{pdf}(U^{(v)}(0))$ can be computed from pdf_0 for all vehicles $v \in \mathcal{V}$.

Claim. We claim here that our canonical decoupling approach of all passenger flows as described above approximates *T-SHA*'s original non-decoupled passenger flow-dynamics arbitrarily well for vanishing fixed time intervals which separate all discrete simulation time steps if we correctly project the dynamics of all non-decoupled passenger flows onto the dynamics of all decoupled passenger flows, i.e. the passenger flows assigned to all corridors in our above approach: We define all decoupled passenger flows (indicated by some arc written on top of the original flow symbols) into (out of) any station $s \in \mathcal{S}$ from (to) the exterior of the considered network, as well as all passenger transfer flows within s as they are specified in Tab. 4.1 below. Note that this linearly-scaled projection ensures all flow properties (demand-sensitiveness etc.) which we have introduced in Sec. 3.5 on p. 65ff. Moreover note that every network inflow might have some non-zero diffusion term, which - by model assumption - however is constant. We thus do not scale it.

Table 4.1: Specification of all decoupled passengers flows through any inflow-, outflow-, or transfer corridor $t \in T_c^{(s)}$ in any station $s \in \mathcal{S}$

Network Inflow	$\vec{\phi}_{ec}^{(s,t)}(\tau) := \phi_{ec}^{(s,t)} \left(\frac{\vec{M}^{(s,t^*)}(\tau)}{\gamma^{(s,t^*)}} \right)$
	$\vec{\delta}_{ec}^{(s,t)} := \delta_{ec}^{(s,t)}$
Transfer Flow	$\vec{\phi}_{tc}^{(s,t)}(\tau) := \phi_{tc}^{(s,t)} \left(\frac{\vec{M}^{(s,t^*)}(\tau)}{\gamma^{(s,t^*)}}, \frac{\vec{M}^{(s,t^*)}(\tau)}{\gamma^{(s,t^*)}} \right)$
Network Outflow	$\vec{\phi}_{xc}^{(s,t)}(\tau) := \phi_{xc}^{(s,t)} \left(\frac{\vec{M}^{(s,t^*)}(\tau)}{\gamma^{(s,t^*)}} \right)$

Similarly, we define all decoupled boarding & alighting passenger flows (again indicated by some arc written on top of the original flow symbols) between every pair of some vehicle $v \in \mathcal{V}$ which is stopped in front of some platform $p \in P_{gp}^{(s)}$ in some station $s \in \mathcal{S}$, when T -SHA is in some mode $q \in \mathcal{Q}$, as they are specified in Tab. 4.2 below.

Table 4.2: Specification of all decoupled boarding & alighting flows through some corridor $t \in T_c^{(s)}$ in some station s between some platform in s and some vehicle $v \in \mathcal{V}$ docked to this platform when T -SHA is in some mode $q \in \mathcal{Q}$

Boarding Flow	$\vec{\phi}_{bc}^{(s,t,\lambda_m^{(q,v)})}(\tau) := \phi_{bc}^{(s,t,\lambda_m^{(q,v)})} \left(\frac{\vec{M}^{(s,t^*)}(\tau)}{\gamma^{(s,t^*)}}, 0.5 \vec{U}_b^{(v)}(\tau) \right)$
Alighting Flow	$\vec{\phi}_{ac}^{(s,t,\lambda_m^{(q,v)})}(\tau) := \phi_{ac}^{(s,t,\lambda_m^{(q,v)})} \left(\frac{\vec{M}^{(s,t^*)}(\tau)}{\gamma^{(s,t^*)}}, 0.5 \vec{U}_a^{(v)}(\tau) \right)$

In what follows next, we (i) look at the set up of the balance equations for all decoupled passenger flows (what we did not do so far), (ii) prove that our above claim is correct, and (iii) discuss the impact of our canonical decoupling approach on the complexity of any forecast computation.

Set Up of Replacing Balance Equations. The systems of balance equations which define the time evolution of the passenger numbers in all slices that are connected to some inflow-, outflow-, or transfer corridor $t \in T_c^{(s)}$ in some station $s \in \mathcal{S}$ do not depend on T -SHA's mode $q \in \mathcal{Q}$ since the passenger flows through these corridors do not do so. At any simulation time $\tau \geq 0$, these systems of balance equations have the general form

$$d\vec{X}^{(s,t)}(\tau) := \vec{\alpha}^{(s,t)} \left(\vec{X}^{(s,t)}(\tau) \right) d\tau + \vec{\beta}^{(s,t)} dW(\tau), \quad (4.25)$$

which is very similar to the system of balance equations Eqn. 3.27 on p. 75 that we have set up for all (coupled) passenger flows pertaining to s in q . Table 4.3 below lists their specifications.

Table 4.3: Specification of the system of balance equations set up for any decoupled passenger flow through some entrance-, transfer-, or exit corridor $t \in T_c^{(s)}$ in some station $s \in \mathcal{S}$

	Network Inflow	Transfer Flow	Network Outflow
Schematic structure			
$\vec{X}^{(s,t)}(\tau)$	$\vec{M}^{(s,t^*,t)}(\tau)$	$\begin{bmatrix} \vec{M}^{(s,*t,t)}(\tau) \\ \vec{M}^{(s,t^*,t)}(\tau) \end{bmatrix}$	$\vec{M}^{(s,*t,t)}(\tau)$
$\vec{\alpha}^{(s,t)}(\tau)$	$\mathbf{R}^{(s,t)} \vec{\phi}_{ec}^{(s,t)}(\tau)$	$\begin{bmatrix} -\vec{\phi}_{tc}^{(s,t)}(\tau) \\ \mathbf{R}^{(s,t)} \vec{\phi}_{tc}^{(s,t)}(\tau) \end{bmatrix}$	$-\vec{\phi}_{xc}^{(s,t)}(\tau)$
$\vec{\beta}^{(s,t)}$	$\vec{\delta}_{ec}^{(s,t)}$	0	0

On the contrary, the system of balance equations set up for any decoupled boarding or alighting flow between some platform $p \in P_{gp}^{(s)}$ in some station s and a vehicle docked to p in some mode $q \in Q$ of T -SHA, depend on q since it depends on the mission of v (what passengers want to alight from/board v , etc.) which in turn is specified by q . Thus, in general it has the form

$$d\vec{X}^{(s,t,q)}(\tau) := \vec{\alpha}^{(s,t,q)}(\vec{X}^{(s,q)}(\tau)) d\tau, \quad (4.26)$$

which - by model assumption - has no non-zero diffusion term. Table 4.4 below lists its specification.

Table 4.4: Specification of balance equations set up for all decoupled boarding & alighting flows between some platform $p \in P_{gp}^{(s)}$ in some station $s \in \mathcal{S}$ and some vehicle $v \in \mathcal{V}$ which is docked to p when T -SHA is in some mode $q \in Q$

	Boarding Flow	Alighting Flow
Schematic structure		
$\vec{X}^{(s,t)}(\tau)$	$\begin{bmatrix} \vec{M}^{(s,*t,t)}(\tau) \\ \vec{U}_b^{(v)}(\tau) \end{bmatrix}$	$\begin{bmatrix} \vec{M}^{(s,t^*,t)}(\tau) \\ \vec{U}_a^{(v)}(\tau) \end{bmatrix}$
$\vec{\alpha}^{(s,t,q)}(\tau)$	$\begin{bmatrix} -\vec{\phi}_{bc}^{(s,t,q)}(\tau) \\ \mathbf{R}^{(s,t)} \vec{\phi}_{bc}^{(s,t,q)}(\tau) \end{bmatrix}$	$\begin{bmatrix} \mathbf{R}^{(s,t)} \vec{\phi}_{ac}^{(s,t,q)}(\tau) \\ -\vec{\phi}_{ac}^{(s,t,q)}(\tau) \end{bmatrix}$

Correctness of Our Approach. In order to proof the correctness of our above canonical decoupling approach, we must show that all passenger numbers in the different modes of *T-SHA* evolve according to the same dynamics starting from the same initial condition in both coupled and decoupled situations. We will proof such for the limiting case of vanishing fixed time intervals which separate every discrete simulation time step when we let *T-SHA* change its mode. Note that this dynamical equivalence, which we demand in the limiting case of very close discrete simulation time steps, implies that all mode transitions of *T-SHA* which occur at some simulation time step with some marginal probability in the original model dynamics also occur at the same simulation time step with the same marginal probability in the approximating decoupled dynamics. Based on these considerations, assume that we want to compute the probability

$$\mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma((i-1)_{\Delta\tau}) = n'\right)$$

from Eqn. 4.8 on p. 114 according to which *T-SHA* transitions from some timed mode $n' \in \Sigma$ to some other timed mode $n \in \Sigma$ at some discrete simulation time step $i \in \mathbb{N}_{>0}$. Since we do want to unnecessarily complicate things here, moreover assume that this transition is only conditioned on the fact that the (vectorial) number of passengers at some single gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ adopts some value from $K \subseteq \mathcal{M}^{(s,p)}$. Thus,

$$\begin{aligned} & \mathbb{P}\left(\sigma(i_{\Delta\tau}) = n \mid \sigma((i-1)_{\Delta\tau}) = n'\right) \\ &= \mathbb{P}\left(M^{(s,p)}(i_{\Delta\tau}) \in K\right) \\ &= \int_K \text{pdf}\left(M^{(s,p)}(i_{\Delta\tau}) \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q((i-1)_{\Delta\tau}) = q'; k\right) dk. \end{aligned} \quad (4.27)$$

according to Eqn. 4.9 on p. 114 if we assume that $q' \in \mathcal{Q}$ is *T-SHA*'s mode in n' . However, let us compute

$$\begin{aligned} & \mathbb{P}\left(\sum_{t \in \bullet p \cup p \bullet} \vec{M}^{(s,p,t)}(i_{\Delta\tau}) \in K\right) = \\ & \int_K \text{pdf}\left(\sum_{t \in \bullet p \cup p \bullet} \vec{M}^{(s,p,t)}(i_{\Delta\tau}) \mid \right. \\ & \left. \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q((i-1)_{\Delta\tau}) = q'; k\right) dk \end{aligned} \quad (4.28)$$

instead, which is the conditional probability that the sum of the passenger numbers in all different slices from p 's circular area is in K at i ; supposing that we employ our canonical decoupling approach. Next, denote the number of corridors which are connected to p by

$$l := |\bullet p \cup p \bullet|, \quad (4.29)$$

and define $\{t_1, t_2, \dots, t_l\} := \bullet p \cup p^\bullet$. Then, rewrite Eqn. 4.28 in form of

$$\mathbb{P}\left(\sum_{t \in \bullet p \cup p^\bullet} \vec{M}^{(s,p,t)}(i_{\Delta\tau}) \in K\right) = \int_K \int_{\underline{\mathcal{M}}^{(s,p,k)}} \text{pdf}\left(\left[\vec{M}^{(s,p,t_1)}(i_{\Delta\tau}); \vec{M}^{(s,p,t_2)}(i_{\Delta\tau}); \dots; \vec{M}^{(s,p,t_l)}(i_{\Delta\tau})\right] \middle| \sigma(0) = n_0, \quad (4.30)\right. \\ \left. X(0) \sim \text{pdf}_0, q\left((i-1)_{\Delta\tau} = q'; [k_1; k_2; \dots; k_l]\right) d(k_1, k_2, \dots, k_l) dk,\right.$$

with $\underline{\mathcal{M}}^{(s,p,k)}$ according to Eqn. 4.22. Therein, note that $M^{(s,p,t_1)}(i_{\Delta\tau}), \dots, M^{(s,p,t_l)}(i_{\Delta\tau})$ are all independent random variables: according to the proper execution of our decoupling approach as described above, they were computed from $M^{(s,p,t_1)}((i-1)_{\Delta\tau}), \dots, M^{(s,p,t_l)}((i-1)_{\Delta\tau})$ through the numerical integration of decoupled systems of balance equations. Thus, Eqn. 4.30 simplifies to

$$\mathbb{P}\left(\sum_{t \in \bullet p \cup p^\bullet} \vec{M}^{(s,p,t)}(i_{\Delta\tau}) \in K\right) = \int_K \int_{\underline{\mathcal{M}}^{(s,p,k)}} \prod_{t_j \in \bullet p \cup p^\bullet} \text{pdf}\left(\vec{M}^{(s,p,t_j)}(i_{\Delta\tau}) \middle| \sigma(0) = n_0, \quad (4.31)\right. \\ \left. X(0) \sim \text{pdf}_0, q\left((i-1)_{\Delta\tau} = q'; k_j\right) d(k_1, k_2, \dots, k_l) dk\right.$$

This insight brings us to the following theorem which underlines the correctness of our canonical decoupling approach.

Theorem 1 *The integral*

$$\int_{\underline{\mathcal{M}}^{(s,p,k)}} \prod_{t_j \in \bullet p \cup p^\bullet} \text{pdf}\left(\vec{M}^{(s,p,t_j)}(i_{\Delta\tau}) \middle| \sigma(0) = n_0, X(0) \sim \text{pdf}_0, \quad (4.31)\right. \\ \left. q\left((i-1)_{\Delta\tau} = q'; k_j\right) d(k_1, k_2, \dots, k_l)\right.$$

in Eqn. 4.31 above converges to

$$\text{pdf}\left(M^{(s,p)}(i_{\Delta\tau}) \middle| \sigma(0) = n_0, X(0) \sim \text{pdf}_0, q\left((i-1)_{\Delta\tau} = q'; k\right)\right)$$

from Eqn. 4.27 for $\Delta\tau \xrightarrow{\Delta\tau > 0} 0$.

Proof of Theorem 1.

- *Common Initial State:*

From Eqn. 4.15, note that

$$\sum_{t \in \bullet p \cup p^\bullet} \vec{M}^{(s,p,t)}(i_{\Delta\tau}) = \sum_{t \in \bullet p \cup p^\bullet} \gamma^{(s,p)} M^{(s,p)}(i_{\Delta\tau}) \\ = M^{(s,p)}(i_{\Delta\tau}) \sum_{t \in \bullet p \cup p^\bullet} \gamma^{(s,p)}. \quad (4.32)$$

From Eqn. 4.14 follows

$$\sum_{t \in \bullet p \cup p \bullet} \gamma^{(s,p)} = 1, \quad (4.33)$$

which in turn implies

$$\sum_{t \in \bullet p \cup p \bullet} \vec{M}^{(s,p,t)}(i_{\Delta\tau}) = M^{(s,p)}(i_{\Delta\tau}). \quad (4.34)$$

- *Common Differential Dynamics:*

The continuous time evolution of

$$\sum_{t \in \bullet p \cup p \bullet} \vec{M}^{(s,p,t)}(\tau)$$

in the time interval $\tau \in [i_{\Delta\tau}, (i+1)_{\Delta\tau})$ is defined by

$$d \left(\sum_{t \in \bullet p \cup p \bullet} \vec{M}^{(s,p,t)}(\tau) \right) = \sum_{t \in \bullet p \cup p \bullet} d\vec{M}^{(s,p,t)}(\tau), \quad (4.35)$$

with initial state

$$\vec{M}^{(s,p,t)}(i_{\Delta\tau})$$

for $t \in \bullet p \cup p \bullet$. This compound balance equation is identical to Eqn. 4.25 which is the balance equation for $M^{(s,p)}$ in the limiting case of $\Delta\tau \rightarrow 0$; given the specification of Eqn. 4.25 in Tab. 4.3; q.e.d.

Computational Impact. In the original discrete time computation of any forecast, we were confronted with one system of coupled stochastic differential equations (SDEs) for every station $s \in \mathcal{S}$ in every mode of *T-SHA*. The dimension of this system was $n := (n_{s,1} + n_{s,2}) n_y$, where $n_{s,1}$ is the number of different gathering points in s , $n_{s,2}$ is the number of vehicles docked to s in q , and $n_y := |\mathcal{Y}|$ is the number of different trip profiles considered in *T-SHA*. Our decoupling approach replaces this n -dimensional system of coupled SDEs by a set of probably much smaller systems of SDEs, where every system of SDEs from this set is decoupled from all other systems of SDEs from the same set: Every of this new/replacing system of equations has at maximum $2n_y$ dimensions:

- $2n_y$ for all decoupled transfer-, boarding-, and alighting flows; and
- n_y dimensions for all remaining types of decoupled passenger flows, i.e. decoupled network inflows & outflows.

Very likely it thus has much less dimensions than the single system of balance equations in the original coupled setting before. However, compared to this original coupled setting we now have to perform many distribution and re-distribution steps of all passenger numbers from the idealized circular areas of all gathering points and vehicle compartments to the different slices therein, and vice versa.

Disconnect Platforms in Stations

The decoupling of all passenger flows in any mode $q \in \mathcal{Q}$ of *T-SHA* as proposed in the previous section implicates a finite speed for the propagation of all passenger number disturbances in all (modelled) stations; cf. Fig. 4.4.6 below: In one single discrete (simulation) time step, a change in the passenger number of p_1 can affect the passenger number of p_2 caused e.g. by a passenger flow from p_1 to p_2 through t_{12} . However, it cannot affect the passenger number at p_3 , which latter disturbance takes at least two discrete time steps in order to occur: originating from p_1 the disturbance must first arrive at p_2 which takes one discrete time step, and from p_2 it must further spread to p_3 which takes another discrete time step. Similarly, a change in the passenger number of p_3 can affect the passenger numbers of p_1 and p_2 only after some discrete time steps have elapsed. However, the chain of coupled reactions is more complicated here: We assume that all passenger flows are capacity- and demand-sensitive. This means that a passenger flow from p_2 to p_3 via t_{23} depends on the cumulative passenger number at p_3 . Disturbing the (cumulative) passenger number at p_3 thus may affect the passenger flow through t_{23} , which in turn disturbs the passenger number at p_2 in at minimum one discrete time step. In the same way, the latter disturbance of p_2 's passenger number may affect the passenger number of p_1 caused by a passenger flow through t_{12} in another discrete time step; giving us at minimum two discrete time steps that are necessary for a disturbance of p_3 's passenger number to affect p_1 's passenger number.

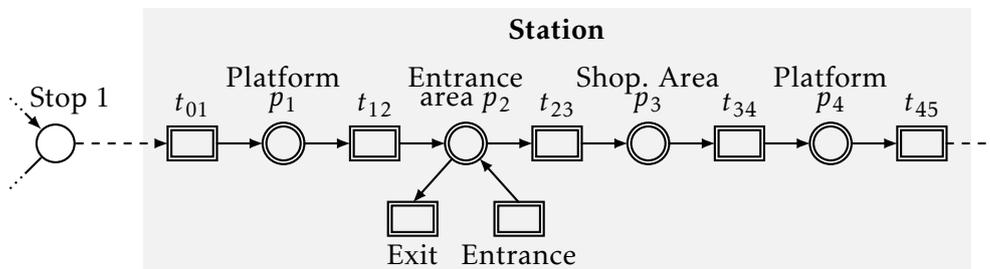


Figure 4.4.6: Extract of a station which we use to explain the impact of our canonical decoupling approach in terms of the finite speed for the propagation of passenger number disturbances in all stations: In one discrete simulation time step, a disturbance of p_2 's passenger number can affect the passenger numbers at p_1 and p_3 . However, it cannot affect the passenger number at p_4 , which takes at least two discrete time steps.

Assuming that we employ our canonical decoupling approach from Sec. 4.4.6 on p. 127ff., we can thus make the following statement: Independently of the structure of the considered station, it takes at least one discrete (simulation) time step until a change in the passenger number at one of its gathering points, say gathering point p , may affect the passenger number at some neighbouring gathering point, i.e., a gathering point which is connected to p via some corridor. This

dynamical limitation/property brings us to the next definition.

Definition 72 (Catchment Area) *The catchment area $\sigma(s, p, r)$, with*

$$\sigma : (s, p, r) \in \mathcal{S} \times P_{\text{gp}}^{(s)} \times \mathbb{N}_{>0} \rightarrow 2^{P_{\text{gp}}^{(s)} \cup T_c^{(s)}},$$

of some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$ of some radius $r \in \mathbb{N}_{>0}$ comprises all those gathering points and corridors in s whose shortest paths in s to p have lengths equal to or smaller than r if we assign a cost/weight of one half to every edge $e \in E_{\text{st}}^{(s)}$ in s .

Acc. to the above definition, every catchment area of some radius $r \in \mathbb{N}_{>0}$ computed for some gathering point $p \in P_{\text{gp}}^{(s)}$ in some station $s \in \mathcal{S}$, tells us the disturbance of which gathering point in s might affect p 's passenger number in r discrete time steps: a disturbance of every gathering $p' \in P_{\text{gp}}^{(s)}$ s.t. $p' \in \sigma(s, p, r)$. It also tells us the set of all corridors which must carry these disturbances, namely every corridor $t \in T_c^{(s)}$ s.t. $t \in \sigma(s, p, r)$. For instance in Fig. 4.4.6 above, the catchment area of the platform p_1 with radius 2 comprises everything in the depicted station apart from t_{34} , p_4 , and t_{45} .

Now what do/can we do with the above knowledge? Well, first of all note that we might be able to compute some (possibly zero) maximum radius $r_m \in \mathbb{N}_{\geq 0}$ which is common to the catchment areas computed for all platforms in some station $s \in \mathcal{S}$ s.t. the catchment area computed for some particular platform in s does not comprise any other platform in s than the platform it is computed for. This job of computing r_m is done by Alg. 27 below.

Algorithm 27 Specification of the function

$$\text{COMPUTEMAXRADIUS} : \mathcal{S} \rightarrow \mathbb{N}_{\geq 0} \cup \{-1\}$$

which for station $s \in \mathcal{S}$ computes the maximum radius $r_m := \text{COMPUTEMAXRADIUS}(s)$ common to the catchment areas for all platforms in s s.t. every catchment area computed for some platform in s does not comprise any other platform than the platform it is computed for. If s has less than two platforms, then the returned value for r_m is minus one and has no meaning.

```

1: function COMPUTEMAXRADIUS( $s$ )
   // Refer to Def. 25 on p. 50 for the meaning of "platform":
2:    $P \leftarrow$  set of all platforms in  $s$ 
3:   if
        $|P| < 2$ 
4:   then
5:      $r_m \leftarrow -1$ 
6:   else

```

```

// Compute minimum shortest path between any pair of two platforms
in s:
7:   for
      every  $(p_1, p_2) \in \Sigma := \{(a, b) \in P^2 : a \neq b\}$ 
8:   do
      // Ignore orientation of all edge labels:
9:      $k_{p_1, p_2} \leftarrow$  shortest path connecting  $p_1$  with  $p_2$  in  $s$ 
10:  end for
11:   $k \leftarrow \min_{(p_1, p_2) \in \Sigma} k_{p_1, p_2}$ 

// Compute  $r_m$ ; use  $k$  as stop criterion:
12:   $arg \leftarrow$  TRUE
13:   $r_m \leftarrow 1$ 
14:  while
       $arg =$  TRUE
15:  do
16:    if
       $r_m \leq k$ 
17:    then
18:       $flag =$  FALSE
19:      for
        every  $p \in P$ 
20:      do
21:         $\sigma(s, p, r_m) \leftarrow$  catchment area of  $p$  in  $s$  with radius  $r_m$  accord-
        ing to Def. 72 on p. 141
22:        if
           $\sigma(s, p, r_m) \cap (P \setminus \{p\}) \neq \emptyset$ 
23:        then
24:           $flag \leftarrow$  TRUE
25:        end if
26:      end for
27:      if
28:         $flag =$  TRUE
29:      then
30:         $arg \leftarrow$  FALSE
31:      else
32:         $r_m \leftarrow r_m + 1$ 

```

```

33:         end if
34:     else
35:         arg = FALSE
36:     end if
37: end while
38: end if
39: return r_m
40: end function

```

Assuming that $r_m > 0$, we know that within r_m discrete time steps, passenger number disturbances cannot propagate between the platforms in s and thus between all transportation grids connected to s ; which implies that s becomes an impenetrable barrier to the propagation of all passenger number disturbances between the different transportation grids connect to it. Now this is where Alg. 28 below comes into play, which for some positive integer $r \in \mathbb{N}_{>0}$ (= remaining discrete simulation time steps until the end of the considered forecast horizon) removes from T -SHA's infrastructure graph or some subgraph thereof all transfer corridors in every station where passenger number disturbances cannot propagate between its different platforms in r time steps.

Algorithm 28 Specification of the function

$$\text{REMOVETRANSFERCORRIDORS} : \mathbb{G} \times \mathbb{N}_{>0} \rightarrow \mathbb{G},$$

with

$$\mathbb{G} := \left\{ G_i : G_i \text{ is a subgraph of } T\text{-SHA's} \right. \\ \left. \text{infrastructure graph from Def. 70 on p. 119} \right\},$$

which for some positive integer $r \in \mathbb{N}_{>0}$ removes from some subgraph $G_i = (N_i, E_i)$ of T -SHA's infrastructure graph all transfer corridors in every station which 1) does not comprise any target gathering point, and 2) cannot spread passenger number perturbations between the different transportation grids connected to it in r discrete simulation time steps

```

1: function REMOVETRANSFERCORRIDORS( $G_i, r$ )
   // Step 1:
2:    $\mathcal{S}_t \leftarrow \{s \in \mathcal{S} : \exists p \in P_{\text{gp}}^{(s)} \text{ s.t. } p \in P_t^{(s)}\}$ 
   // Step 2:
3:   for
     every  $s \in \mathcal{S} \setminus \mathcal{S}_t$ 
4:   do

```

```

// Refer to Def. 25 on p. 50 for the meaning of "platform":
5:    $P^{(s)} \leftarrow$  set of all platforms in  $s$ 
6:   for
      every  $p \in P^{(s)}$ 
7:   do
8:      $\sigma(s, p, r) \leftarrow$  catchment area of  $p$  in  $s$  with radius  $r$  according to
      Def. 72 on p. 141
9:   end for
10:  if
       $\sigma(s, p, r_m) \cap (P^{(s)} \setminus \{p\}) = \emptyset, \forall p \in P^{(s)}$ 
11:  then
12:     $N_i \leftarrow N_i \setminus \bigcup_{s \in \mathcal{S}} (s \times T_{tc}^{(s)})$ 
13:     $E_i \leftarrow E_i \cap (N_i)^2$ 
14:     $G_i \leftarrow (N_i, E_i)$ 
15:  end if
16: end for
17: return  $G_i$ 
18: end function

```

Unfold Vehicle Trajectories

When called at some discrete simulation time step $i \in \{0, 1, \dots, k\}$, with $k \in \mathbb{N}_{>0}$, Alg. 29 below checks for every leaf node of T -SHA's propagation DAG (= possible timed mode for T -SHA at i) and every pair of a transportation grid and a stop, say stop p , in this transportation grid, whether there is a chance that some vehicle can hold at p for the purpose of boarding & alighting in the remaining $k - i$ simulation time steps. If this not the case, then Alg. 29 removes p from G_i which must be some subgraph of T -SHA's infrastructure graph. In order to achieve so, Alg. 29 unfolds the trajectories of all vehicles operated in the considered transportation grids, where we assume some ideal conditions: minimum dwell times at all possible stops, and no interactions with other (possibly blocking) vehicles.

Algorithm 29 Specification of the function

REMOVESTOPS : $\{G : G \text{ is some propagation DAG for } T\text{-SHA according to Def. 70 on p. 119}\} \times \mathbb{G} \times \mathbb{N}_{>0} \rightarrow \mathbb{G}$

with

$$\mathbb{G} := \left\{ G_i : G_i \text{ is a subgraph of } T\text{-SHA's infrastructure graph from Def. 70 on p. 119} \right\},$$

which for some propagation DAG G and some number $r \in \mathbb{N}_{>0}$ removes from some subgraph G_i of $T\text{-SHA}$'s infrastructure graph all stops in every transportation grid, where 1) this transportation grid must not comprise any target vehicle, and 2) these stops are not served by any vehicle in r discrete time steps.

```

1: function REMOVESTOPS(  $G, G_i, r$  )
2:    $h \leftarrow$  height of  $G$ 
3:    $N_h \leftarrow$  all nodes in  $G$  with height  $h$ 
4:    $\mathcal{G}_t \leftarrow \{g \in \mathcal{G} : \exists v \in \mathcal{V}^{(g)} \text{ s.t. } v \in \mathcal{V}_t\}$ 
5:   for
     every  $n \in N_h$ 
6:   do
7:     for
       every  $g \in \mathcal{G} \setminus \mathcal{G}_t$ 
8:     do
       // Refer to Def. 26 on p. 50 for the meaning of "stop":
9:        $P_s \leftarrow$  set of all stops in  $g$ 
10:      for
        every  $v \in \mathcal{V}^{(g)}$ 
11:      do
        // Ignore the presence of all other vehicles, assume minimum
        dwell times:
12:         $\overline{P}_s^{(n)} \leftarrow$  subset of all stops in  $g$  where  $v$  might hold for the
        purpose of boarding & alighting
13:      end for
14:    end for
15:  end for

```

```

16:    $N_i \leftarrow N_i \setminus \left( P_s \setminus \bigcup_{v \in \mathcal{V}(s)} \bar{P}_s(v) \right)$ 
17:    $E_i \leftarrow E_i \cap (N_i)^2$ 
18:   return  $(N_i, E_i)$ 
19: end function

```

Offline Computation of Transition Probability Matrices

Our canonical decoupling approach from Sec. 4.4.6 on p. 127 replaces all original (high-dimensional) systems of coupled balance equations set up for the different stations in the different modes/macro modes of *T-SHA* by some set of lower-dimensional systems of coupled balance equations set up for all decoupled passenger flows for the same station in the same mode/macro mode. These new lower-dimensional systems of balance equations might still have too many dimensions in that we could numerically integrate all Fokker–Planck equations (FPEs) derived from them in reasonable amounts of time; when traditional numerical integration schemes are to be employed. Even worse, we are not confronted with one single FPE but with many FPEs in our forecast computations! However remember that not every FPE which shows up during some forecast computation is new/was not encountered before. In fact, we face the same system of balance equations for every decoupled network inflow, network outflow, and transfer flow at every discrete simulation time step if we employ our canonical decoupling approach. This insight brings us to the following question.

How can we (algorithmically) take advantage of the fact that we have to solve one and the same FPE again and again if we employ our canonical decoupling approach?

Well, first of all note that we face indeed many different FPEs again and again during some forecast computation, namely any FPE which was derived from the system of SDEs set up for some decoupled network inflow, network outflow, or transfer flow. However, the fact that we consider one and the same FPE several times (and possibly at every discrete simulation time step), does not mean that we will always perform the same numerical integration since we have to assume that we start our computations with different initial conditions/densities. And this is a problem since all numerical methods (Finite Volume method, etc.) which are known to us cannot take advantage of the fact that we have to solve many times the same partial differential equations systems but with different initial states. That is why, we have developed our last brick which proposes to not only discretize the simulation time in the computation of any forecast, but also all passenger numbers in *T-SHA*. This new brick requires from the end user of our model to

1. discretize all passenger numbers/the local state spaces thereof,
2. project all initial passenger number densities to approximating probability mass functions,

3. compute (Markov-like) transition matrices for all decoupled passenger flows, and
4. replace the systems of balance equations computed for all decoupled passenger flows by these latter matrices in our discrete time discrete space forecast computations.

We next take a closer look at the first and third point. However, we will not discuss the simple but cumbersome collection of all equations. Instead, we will only highlight all essential ideas in an informal manner.

Discretization of Passenger Numbers. In principle, all passenger numbers in *T-SHA* can be discretized independently from each other. However, in practice most probably some human modeller has to do this nasty job. Thus having this human modeller in mind, it sounds reasonable to us to only introduce one parameter, and to use this single parameter for the discretization of all passenger numbers. More specifically, this additional parameter $d \in \{2, 3, 4, \dots\}$ completely defines the equidistant discretization of every vectorial passenger number along all of its dimensions. For instance, if (i) $d = 3$, (ii) the capacity-limit of the cabin on-board some vehicle v is $c_v \in \mathbb{R}_{>0}$ passengers, and (iii) these passengers can have $y_v \in \mathbb{N}_{>0}$ different trip profiles, then the vector k which completely defines the discrete number of passengers on-board v can adopt all values in the set

$$\left\{ \left[k_1, k_2, \dots, k_{y_v} \right]^t : \frac{k_1}{g_v}, \frac{k_2}{g_v}, \dots, \frac{k_{y_v}}{g_v} \in \{0, 1, \dots, d-1\} \right. \\ \left. \text{and } (k_1 + k_2 + \dots + k_{y_v}) g_v \leq c_v \right\},$$

where $g_v := \frac{c_v}{y_v}$.

Meaning and Computation of Transition Matrices. First of all, notice that the above mentioned discretization of all passenger numbers does not contradict our canonical decoupling approach, which can also be written down with discrete passenger numbers instead of continuous ones. In this case, all integral must be replaced by finite sums. Secondly, notice that a discretization of all passenger numbers implies that these passenger numbers are no longer defined by probability density functions, but rather by probability mass functions. Thirdly, notice that we can enumerate the discretized state spaces of all passenger numbers, which in turn allows us to uniquely map every probability mass function to some vector of fixed length. So if we discretize all passenger numbers, then we have to specify update rules for the probability mass functions for all passenger numbers; where we use the plural form of mass function here since we assume that our canonical decoupling approach is to be employed. We can map these update rules to square

transition matrices; one for every decoupled passenger flow. Every transition matrix Π computed for some particular decoupled passenger flow defines how the cumulative probability mass function pmf for the passenger numbers at the slices connected to its corridor changes from some discrete simulation time step $i \in \mathbb{N}_{>0}$ to $i + 1$ according to

$$\text{pmf}_{i+1} := \text{pmf}_i \Pi;$$

and thus reminds us of a Markov chain.

The computation of Π is straightforward. All that we have to do is to take the corresponding system of balance equations from Tab. 4.1 on p. 135 or Tab. 4.2 on p. 135, and to numerically integrate it from time zero one fixed simulation time step into the future for all discrete initial states. Once this is done, we then have to count how many times the numerically computed values lie in which container of some kind of a histogram which we have defined before. From the counting of all *balls* in the different bins of our histogram and the arbitration of all balls among the discrete passenger numbers which adjoin their bin, we then can compute all entries of Π ; where all elements in the same column of Π must sum up to one. In this context, note that only the balance equations for network inflows can have non-zero diffusion terms, which implies that only for these flows we might have to numerically integrate their systems of balance equations many times for different realizations of a discretized Wiener path (Euler-Maruyama method).

Forecast Algorithm

Here it is, our forecast algorithm: When being called, Alg. 30 below first does some preprocessing in the steps 1 to 4, before it propagates in step 5 *T-SHA*'s hybrid state forward in time. In particular, it computes in step 1 the set Δ of all possible macro modes for *T-SHA* in the considered time horizon of the forecast. We discussed this computation already in Sec. 4.4.4 on p. 124ff. In step 1, Alg. 30 also sets up all possible balance equations which we might encounter in the considered forecast horizon; one balance equation for every possible decoupled passenger flow. Note that this set up of balance equations has to be done only once (and here), and not during the forward propagation of *T-SHA*'s hybrid state in step 5 again and again: Every macro mode for *T-SHA* defines which passenger flow between some station and some vehicle stopped in front of this station is possible when *T-SHA* is in this macro mode. In doing so, it *picks* some subset of balance equations from the pool of all possible balance equations. Now the fact that we might ignore some vehicles during the forward propagation of *T-SHA*'s hybrid state, does not imply that we might face some unprecedented macro modes with unprecedented passenger flows. Remember that we discussed this issue already at the end of Sec. 4.4.6, where we elaborated our canonical decoupling approach. Anyway, in the following steps 2 to 4, Alg. 30 creates a new instantiation of *T-SHA*'s infrastructure graph, and a new instantiation of its propagation DAG with one single node which captures *T-SHA*'s initial timed mode. Moreover, it computes here the

first time step when it will try to make use of the algorithmic brick “disconnect platforms in stations” from Sec. 4.4.7. All lines of pseudo code which describe the actual forward propagation of *T-SHA*’s hybrid state in step 5 implement the following actions in the given order for every new simulation time step; until the end of the considered forecast horizon (at some time step $k > 0$) assuming that the end user-specified stop criterion does not abort the forward propagation of *T-SHA*’s hybrid state before: (i) propagate *T-SHA*’s continuous state from time step i to time step $i + 1$, (ii) compute the set \bar{V} of all vehicles whose operation can be ignored in the remaining $k - i$ time steps of the forecast horizon, (iii) append a new layer of timed modes to *T-SHA*’s propagation DAG G while taking \bar{V} into account, (iv) increment the simulation time step, (v) for every leaf node in G compute the marginal probability that *T-SHA* finds itself in the corresponding timed mode at i , and (vi) removes all leaf nodes from G which have zero marginal probabilities to be adopted by *T-SHA* at i .

Algorithm 30 Our forecast algorithm which computes the evolution of *T-SHA*’s hybrid state over the considered time horizon $h > 0$, where this time horizon is discretized into some $k > 0$ equidistantly-distributed points in time

Step 1: Prepare the propagation of *T-SHA*’s state.

- 1: Compute the set Δ of all possible macro modes in $[0, h]$
- 2: Set up the balance equation for every possible decoupled passenger flow in $[0, h]$ from the knowledge of Δ

Step 2: Instantiate a new infrastructure graph for *T-SHA*.

- 3: $G_i \leftarrow$ infrastructure graph for *T-SHA* according to Def. 70 on p. 119

Step 3: Compute the first discrete simulation time step, when the function `REMOVETRANSFERCORRIDORS()` from Alg. 28 on p. 143ff. shall be called.

4: // Function `COMPUTEMAXRADIUS(·)` from Alg. 27 on p. 141:

- 5: $i_f \leftarrow k - \min_{s \in \mathcal{S}} \text{COMPUTEMAXRADIUS}(s, k)$

Step 4: Instantiate a new propagation DAG G with one single node which captures *T-SHA*’s initial timed mode.

- 6: $N \leftarrow (q_0, \varphi_0)$
- 7: $E \leftarrow \emptyset$
- 8: $\Psi \leftarrow \emptyset$
- 9: $G \leftarrow (N, E, \Psi)$

Step 5: Propagate *T-SHA*'s hybrid state k steps forward in time; assuming that the end user-specified stop criterion does not cause any abort before.

```

10:  $arg \leftarrow \text{TRUE}$ 
11:  $i \leftarrow 0$ 
12:  $\bar{\mathcal{G}} \leftarrow \emptyset$ 
13: while
14:    $(i < k) \wedge (arg = \text{TRUE})$ 
15: do
16:   if
17:     Stop criterion is fulfilled
18:   then
19:      $arg \leftarrow \text{FALSE}$ 
20:   else
21:     // Propagate T-SHA's passenger load density to the next simulation time
     step:
22:     Compute
     
$$\text{pdf}\left(X\left((i+1)\Delta\tau\right) \middle| \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right)$$

     from
     
$$\text{pdf}\left(X\left(i\Delta\tau\right) \middle| \sigma(0) = n_0, X(0) \sim \text{pdf}_0\right)$$

     according to Eqn. 4.7 on p. 113
23:     // Remove from  $G_i$  all transfer corridors in some stations according to
     Alg. 28 on p. 143:
24:     if
25:        $i \geq i_f$ 
26:     then
27:        $G_i \leftarrow \text{REMOVE\_TRANSFER\_CORRIDORS}(G_i, k - i)$ 
28:     end if
29:     // Remove from  $G_i$  some stops in some transportation grids according to
     Alg. 29 on p. 145:
30:      $G_i \leftarrow \text{REMOVE\_STOPS}(G_i, k - i)$ 
31:     // Compute the subset  $\bar{\mathcal{G}} \subseteq \mathcal{G}$  of all transportation grids in T-SHA ac-
     cording to Alg. 26 on p. 119, where the operation of every vehicle
      $v \in \mathcal{V}^{(g)}$  in any  $g \in \bar{\mathcal{G}}$  can be ignored in the remaining  $k - i$  simulation
     time steps:

```

```

32:    $\bar{G} \leftarrow \text{TARGETSEARCH}(G_i)$ 
33:    $\bar{V} \leftarrow \bigcup_{g \in \bar{G}} \mathcal{V}^{(g)}$ 
34:   // Append a new layer of nodes to  $G$ , with  $\text{APPENDLAYER}(\cdot)$  from Alg. 24
   // on p. 109; ignore the presence of all  $v \in \mathcal{V}^{(g)}$  in any  $g \in \bar{G}$ :
35:    $G := (N, E, \Psi) \leftarrow \text{APPENDLAYER}(G, \Delta\tau, \bar{V})$ 
36:   // Increment the counter variable  $i$  already here so as to ease in the lines
   // of code below the referencing to all equations on previous pages:
37:    $i \leftarrow i + 1$ 
38:   // Compute the marginal probability for T-SHA to be in any leaf node
   // of  $G$  (= timed mode) at time step  $i$ ; remove all leaf nodes with zero
   // marginal probabilities:
39:   for
40:        $n \in N$  s.t.  $\mu^{(G)}(n) = i$ 
41:   do
42:       Compute
           
$$\mathbb{P}(\sigma(i_{\Delta\tau}) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0)$$

           according to Eqn. 4.8 on p. 114
43:       if
44:            $\mathbb{P}(\sigma(i_{\Delta\tau}) = n \mid \sigma(0) = n_0, X(0) \sim \text{pdf}_0) = 0$ 
45:       then
46:            $N \leftarrow N \setminus \{n\}$ 
47:            $E \leftarrow E \cap N^2$ 
48:            $\Psi \leftarrow \Psi \setminus \Psi^{(e)}$ 
49:            $G \leftarrow (N, E, \Psi)$ 
50:       end if
51:   end for
52:   end if
53: end while

```

IMPLEMENTATION & TEST

This chapter provides a rough overview of our software module in Sec. 5.1, which implements our forecasting algorithm and some more supplementary functions in form of one single executable; cf. the table of contents below: We will not only provide an overview of our software module in this chapter, but we will also analyse its performance/the performance of our algorithms. In order to achieve so, we will introduce a simple test case in Sec. 5.2, and use it to benchmark the efficiency of our algorithmic bricks that try to reduce the size of any computed propagation DAG. Finally in Sec. 5.3, we will go through some more realistic use case step by step, which requires us to also propagate passenger numbers/densities along propagation DAGs in form of complete forecast computations. It is this use case which we will use to show how our software module or some similar implementation of our model/algorithms can be used to evaluate different strategies in degraded situations of real transportation networks with the aim to ensure the smoothness and thus the safety of all considered passenger flows.

Contents

5.1	Our Forecast Engine for Transportation Networks	154
5.1.1	Naming, Licensing, and I/O	154
5.1.2	Functional Decomposition	155
5.1.3	Implementation	157
5.2	A Quantitative Analysis of the Computation of Propagation DAGs . .	158
5.2.1	Preface	158
5.2.2	Layout of the Considered Transportation Network	160
5.2.3	Specification of All Simulation Runs	163
5.2.4	Computed Results	165
5.2.5	Conclusions	173
5.3	Ensure Smooth Passenger Transfer Flows At Massy	174
5.3.1	Motivation	174
5.3.2	Scenario Description & Objectives	175
5.3.3	Specification of Transportation Network	177
5.3.4	Simulation Results	182
5.3.5	Conclusions	187

Our Forecast Engine for Transportation Networks

Naming, Licensing, and I/O

There are plenty ways to become familiar with a complex software implementation. In the next few paragraphs we approach the software implementation of our algorithms by first giving them some common name, then by briefly looking at the targeted end users, and then by looking at the module's input/output (I/O) specification.

Naming. We call our implemented software module

Forecast Engine for Transportation Networks .

Of course, we could have come up with another name such as “Intelligent Transportation Forecast Engine” or “Smart Forecast-Producing Black Box for Complex Multimodal Transportation Networks”. However, FETN does not have a brain and neither do we thus like to call it smart nor intelligent. Instead, we note that FETN was developed to do its job. Secondly, FETN's internal structure and functioning is well-documented, which is not a characteristic of a black box. Thirdly, complex is almost everything if regarded from different perspectives. Fourthly, the application of FETN is not limited to multimodal transportation networks, but it can also be applied to monomodal transportation networks as well.

Ownership & End Users. FETN was developed at the premises of the Technological Research Institute IRT SystemX with money from the French government and some industrial partners (refer to Sec. 1.3 on p. 3 for some more information on IRT SystemX and the context of this work). It is integrated in a bigger platform, which provides some more functionalities such as the computation of key performance indicators; where all functionalities are related to the analysis of multimodal transport use cases in urban agglomerations. As of mid 2016, only partners involved in some industrial/academic projects at IRT SystemX - amongst whom there are e.g. experts in the development of passenger information systems - can account FETN or the platform it hosts. However, this very restricted accounts to FETN does not prevent us from illustrating its I/O specification and development process in the next two paragraphs, discussing its functional decomposition in Sec. 5.1.2, and saying a few words on its implementation in Sec. 5.1.3.

I/O. Primary inputs to FETN are the specification of some infrastructure, some vehicle operation and passenger routing therein, the specification of all initial passenger loads, and the specification of the vehicles' initial positions, initial driving conditions, and initial operational states of the considered use case. Other secondary inputs parametrize the simulation of this use case and the forecast objective associated with it in form of some finite forecast horizon, some temporal

discretization of this forecasting horizon, some spatial discretization of all passenger loads, and some target sets of gathering points in station, vehicles, and trip profiles. All these inputs are communicated to FETN in the Extensible Markup Language (XML); a language for the formal specification of structured data with the goal to be human- and machine-readable alike.

Listing 5.1: Some code snippet illustrating the XML specification of a use case which can be read in by our forecasting module FETN

```
1 <GatheringPoints>
2   <GatheringPoint ID=" ... ">
3     <Capacity>
4       Positive real number
5     </Capacity>
6   </GatheringPoint>
7   ...
8 </GatheringPoints>
```

Upon completion of all computations, FETN outputs the time evolution of the passenger loads at all target gathering points and on-board all target vehicles; which are vectors storing the computed probabilities. Another output is the computed propagation DAG and the marginal probabilities computed for the different modes in its different layers.

Development Process. Apart from the mathematical framework, the rest of FETN's functionality was developed according to the agile software development process Scrum [Schwaber 2012], which in practice means many incremental but executable implementation steps towards some accepted level of maturity in a group of relatively autonomous working software developers with frequent regular meetings.

Functional Decomposition

Figure 5.1.1 below depicts a high-level overview of FETN's functional decomposition: All functions contribute to either (i) the verification of the correct use case specification, (ii) the feasibility check of all computations pertaining to the actual forward propagation of all initial estimations, (iii) the computation of all possible balance equations which may be encountered during this forward propagation, or (iv) the forward propagation itself.

Verification. All functions with a belonging to this block check the correct specification of the considered infrastructure, as well as the passenger routing and the vehicle operation therein; in that the XML specifications of all modelling objects such as the specifications for all stations and for all trip profiles are confronted with the their corresponding definitions from chapter 3. This preliminary check

may be of great interest if the efficiency of some strategies applied to the vehicle operation for the considered use case shall be analyzed/optimized from one simulation run/prediction to the next in some automated way; where every proposed modification must not violate any operational/physical constraint.

Feasibility Check. All functions allocated to this block check the feasibility of the computation of the actual forecast prior to its computation. In order to achieve so, they compute amongst others the minimum number of trip profiles which must be considered at every discrete point in the considered network from the specification of all trip profiles. From this latter knowledge together with a) the user-specified spatial discretization of all passenger loads and b) thresholds gathered from experience (We can compute forecasts up to X dimensions given a spatial discretization of y on our target machine, etc.), i.e., the previous computation of forecasts, they then either let the software module proceed with the computation of the forecast or they do not.

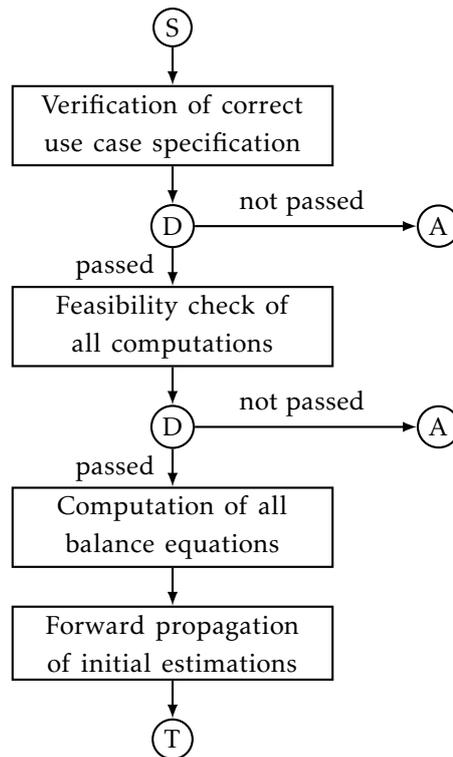


Figure 5.1.1: A high-level overview of our software module's functional decomposition, with the abbreviations S for start, D for decision, A for abort, and T for successful termination

Computation of Balance Equations. This step is key to the success of our software module. It is a consequence of our canonical decoupling approach of all passenger flows. Functions with a belonging to this block, compute all macro

modes which may be encountered in the considered forecast horizon. Based on the knowledge of these macro modes, they moreover compute all routing matrices for the specification of all decoupled balance equations, and transition matrices which approximate these balance equations/the continuous passenger flow dynamics by a discrete passenger flow dynamics as described in Sec. 4.4.9 on p. 146.

Computation of Forecast. Functions pertaining to this last block perform the actual discrete time layer by layer computation of the forecast according to Alg. 30 on p. 149.

Implementation

The computation of all transition matrices which approximate the balance equations for all decoupled passenger flows/the continuous passenger flow dynamics is very math-intensive. Moreover, note that these matrices are computed prior to the actual forward propagation of all initial estimations. That is why we decided to implement their computations in some “less-performing” interpreted numerical oriented programming language.

Upon successful computation, every matrix is written to an own text file in form of some comma-separated values: During the analysis of some use case, the end user may want to execute different simulation runs, where every individual simulation run considers e.g. some minor modifications to some common reference dispatch plan. If this modification does not alter the set of possible passenger flows and their dynamics in the considered forecast horizon, then all simulation runs share the same set of transition matrices. We thus do not have to recompute them for every different simulation run. Instead it is sufficient to compute them once, i.e. for the first simulation run, and to check for every further simulation run whether the set of macro modes it may encounter is identical to the set of macro modes the previous simulation run may have encountered (yes: use the same set of transition matrices; no: recompute the set of all possible transition matrices).

The implementation of all other functional blocks from Fig. 5.1.1, which involves the verification of the correct use case specification, the feasibility check of the forward propagation of all initial estimations, and the forward propagation itself was implemented in some other high-level platform-independent programming language, where we made extensive use of hash tables where ever it was possible so as to e.g. reduce the memory space which is necessary to store the forecasting graph/the different modes in its different layers.

A Quantitative Analysis of the Computation of Propagation DAGs

Preface

In the previous chapter, we have introduced two algorithmic bricks which target the node explosion of any propagation DAG during its layer by layer computation in our forecast algorithm, i.e. Alg. 30 on p. 149. Remember that the first algorithmic brick focuses on the passenger transfer flows in all stations of *T-SHA*. Whenever it realizes that passengers can no longer transfer between any pair of two platforms in some station in the remaining simulation time of the considered forecast horizon, then this first algorithmic brick (hereafter referred to as B_1) removes all transfer corridors in this station from a copy of *T-SHA*'s infrastructure graph. This removal of transfer corridors from *T-SHA*'s infrastructure graph in turn might disconnect some transportation grids from all target gathering points and target vehicles in the considered network, which implies that we can ignore all vehicles operated in these transportation grids in all our computations in the remaining simulation time. Refer to Sec. 4.4.7 on p. 140ff. for more information on the functional principle of B_1 , and to Sec. 4.4.1 on p. 117 for its integration into the layer by layer computation of any propagation DAG. On the other hand, the second algorithmic brick (hereafter referred to as B_2) unfolds for every new layer of nodes which is appended to some propagation DAG, the trajectories of all vehicles in the remaining simulation time of the considered forecast horizon. Its goal is to identify all stops in all transportation grids which cannot be served by any vehicle in this remaining time horizon, and to remove them from *T-SHA*'s infrastructure graph. This latter removal might also disconnect some transportation grids from all target gathering points and all target vehicles in the considered network. Refer to Sec. 4.4.8 on p. 144ff. for the functional principle of B_2 .

Now in this section of our report, we like to analyse the effectiveness and efficiency of both algorithmic bricks. For this purpose we will model some simplistic two-lines transportation network in Sec. 5.2.2. We will then specify seven different simulation runs for this network model (= *T-SHA*) in Sec. 5.2.3. Here note that although we primarily only want to analyze the impact of B_1 and B_2 on the computation of propagation DAGs, we specify seven different simulation runs so as to compare them with two other non-algorithmic means of computational simplifications, namely a) increasing discrete simulation time steps and b) increasing freezing times for the operation of all vehicles. In this context, note that zero freezing times in *T-SHA* imply that we might consider situations during the computation of some forecast which are very unlikely in real transportation networks; in that e.g. some vehicle never departs from a stop because some passengers still want to alight from/or board it but nevertheless this vehicle tries to depart again and again. Thus, specifying non-zero freezing times does not only target the node explosion during the computation of some propagation DAG, but shall also better model the dynamics of the considered network. That being said, we will compare

four parameters (P_1 to P_4) computed for all simulation runs as functions of the model's simulation time in Sec. 5.2.4; where one simulation run makes use of B_1 , another does not but specifies some non-zero freezing time, and so on:

P_1 : Number of nodes in the computed propagation DAG

P_2 : Number of modes in the computed propagation DAG

P_3 : Elapsed computation time of the considered propagation DAG

P_4 : Number of macro modes in the different layers of the computed propagation DAG

Amongst others, P_1 tells us how much memory we must reserve for the execution of our forecast algorithm since we must store one marginal probability for every node (= timed mode of *T-SHA*) in the computed propagation DAG if we want to include it in any post-processing computation; whereas an upper limit for the number of probability densities/mass functions is known prior to the actual simulation given the finite forecast horizon and its temporal discretization. In addition, P_2 tells us how much memory we have to reserve for the propagation DAG itself given the fact that several nodes might have the same mode in common; some useful property which we exploit in our implementation of FETN in form of hash tables. More specifically, we exploit this fact in that if - all things being equal - more different modes underlie all nodes in some propagation DAG G_1 than they underlie in some propagation DAG G_2 , then FETN needs less memory to store G_1 than it needs memory to store G_2 . On the other hand, the correct interpretation of the parameter P_3 should be self explanatory in that it gives us some quantitative measure of the complexity of all computations; no more no less. In this context, note that all computations were performed on a Dell Vostro portable with Intel Core i5 processor (3rd generation, 2×2.5 GHz), 4 GB DDR3, and a 64 bit Windows operating system. Neither should the reader be too much concerned about the details of this workstation here, nor care too much about the absolute numbers. Instead s/he should compare all computed numbers between the different simulation runs (Do we talk about minutes here, but seconds there?). Finally coming to P_4 , note that the number of macro modes in the different layers of any computed propagation DAG equals the number of decoupled balance equations which have to be numerically solved by FETN in these (simulation time) layers during the forward propagation of all passenger load densities along this graph; where we assume that branches in this graph do not disappear because some marginal probabilities of some (mode) transitions are zero. Optimizing this parameter, i.e. reducing it, should thus be of utmost interest to us.

Once we are done with the analysis of all computed results, we draw some conclusions in Sec. 5.2.5.

Layout of the Considered Transportation Network

Figure 5.2.1 below depicts the layout of our simplistic two-lines transportation network for our quantitative analysis which we perform next. Therein, two vehicles are operated in a commuter train line which connects the station S_1 with the station S_2 , and another two vehicles are operated in the people mover line which connects the station S_2 with the station S_3 . These two vehicles operated in the commuter train line move with an average speed of 60 km/h between s_1 and s_2 . While moving from S_1 to S_2 , and vice versa, they must cover a distance of 12 km in both directions. The two vehicles operated in the people mover line on the other hand, move with an average speed of 30 km/h between s_2 and s_3 . They cover a distance of 1.5 km when moving from s_2 to s_3 , and vice versa.

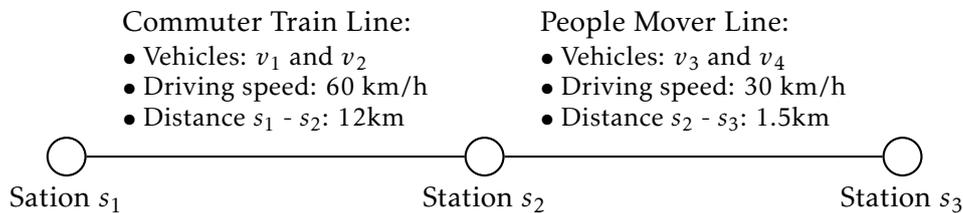


Figure 5.2.1: Set up of a simplistic two-lines transportation network for all considered test cases

We next look at the infrastructure of this two-lines transportation network in more detail, as well as the vehicle operation and the passenger routing therein.

Infrastructure. The infrastructure of our simplistic two-lines transportation network comprises three stations s_1 , s_2 , and s_3 ; one transportation grid, say g_1 , which accommodates the commuter train line; another transportation grid, say g_2 , which accommodates the people mover line; and an interface which defines all possible passenger flows between the vehicles stopped in the transportation lines and the stations.

Fixed-block systems ensure the safe spacing of v_1 and v_2 in the commuter train line, and v_3 and v_4 in the people mover line: Both lines are divided into 12 non-overlapping blocks, where each block is either empty or occupied by at maximum one vehicle at the same time. The alignment and orientation of every block w.r.t. all stations is depicted in Fig. 5.2.2 below.

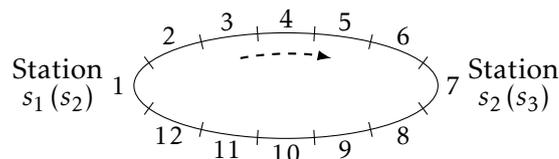


Figure 5.2.2: Alignment and orientation of all blocks in the commuter train line (people mover line) w.r.t. the neighbouring stations

We map all blocks and all connections between these blocks for the commuter train line to g_1 and for the people mover line to g_2 : We represent every block $i \in \{1, 2, \dots, 12\}$ by a waypoint $w_i^{(g)}$, where $g \in \mathcal{G} := \{g_1, g_2\}$. Moreover, we map the possibility to move from one block to the next by a track which connects the corresponding pair of two waypoints in the given order. Thus, the set of all waypoints in g is

$$P_w^{(g)} := \{w_1^{(g)}, w_2^{(g)}, \dots, w_{12}^{(g)}\},$$

the set of all tracks is

$$T_{\text{tr}}^{(g)} := \{t_1^{(g)}, t_2^{(g)}, \dots, t_{12}^{(g)}\},$$

and the set of all edges connecting these waypoints and tracks is

$$E_{\text{gr}}^{(g)} := \left\{ \left(p_k^{(g)}, t_k^{(g)} \right), \left(t_k^{(g)}, p_{k+1}^{(g)} \right) : k = 1, 2, \dots, 11 \right\} \cup \left\{ \left(t_{12}^{(g)}, p_1^{(g)} \right) \right\}.$$

Fig. 5.2.3 below depicts the interior of s_1 , and the way it is connected to g_1 . Passengers can enter s_1 from the network's exterior via the corridor t_1 so as to arrive at s_1 's single platform p_1 next. From p_1 all passengers can board v_1 (v_2) via the corridor t_2 iff v_1 (v_2) is stopped at $w_1^{(g_1)}$. Thus, the set of all gathering points in s_1 is $P_{\text{gp}}^{(s_1)} := \{p_1\}$, the set of all corridors is $T_c^{(s_1)} := \{t_1, t_2\}$, and the set of edges connecting all gathering points and corridors is $E_{\text{st}}^{(s_1)} := \{(t_1, p_1), (p_1, t_2)\}$;

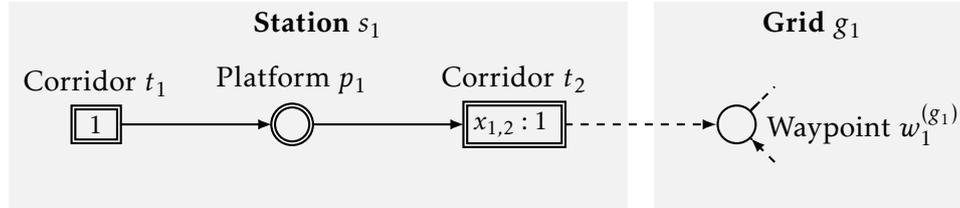


Figure 5.2.3: Interior of station s_1 and its connection to the grid g_1 which accommodates the commuter train line

Fig. 5.2.4 below depicts the interior of s_2 , and the way it is connected to g_1 and g_2 . Passengers can enter s_2 via the corridor t_3 from vehicle v_1 (v_2) iff v_1 (v_2) is stopped at the waypoint $w_7^{(g_1)}$. From p_2 all passengers can transfer to some shopping area from where they can further transfer to some ticketing area and then to the platform p_4 . From p_4 all passengers can board v_3 (v_4) via the corridor t_6 iff v_3 (v_4) is stopped at $w_1^{(g_2)}$. Thus, the set of all gathering points in s_2 is $P_{\text{gp}}^{(s_2)} := \{p_2, p_3, p_4, p_5\}$, the set of all corridors is $T_c^{(s_2)} := \{t_3, t_4, \dots, t_7\}$, and the set of edges connecting all gathering points and corridors is $E_{\text{st}}^{(s_2)} := \{(t_3, p_2), (p_2, t_4), (t_4, p_3), (p_3, t_5), (t_5, p_4), (p_4, t_6), (t_6, p_5), (p_5, t_7)\}$.

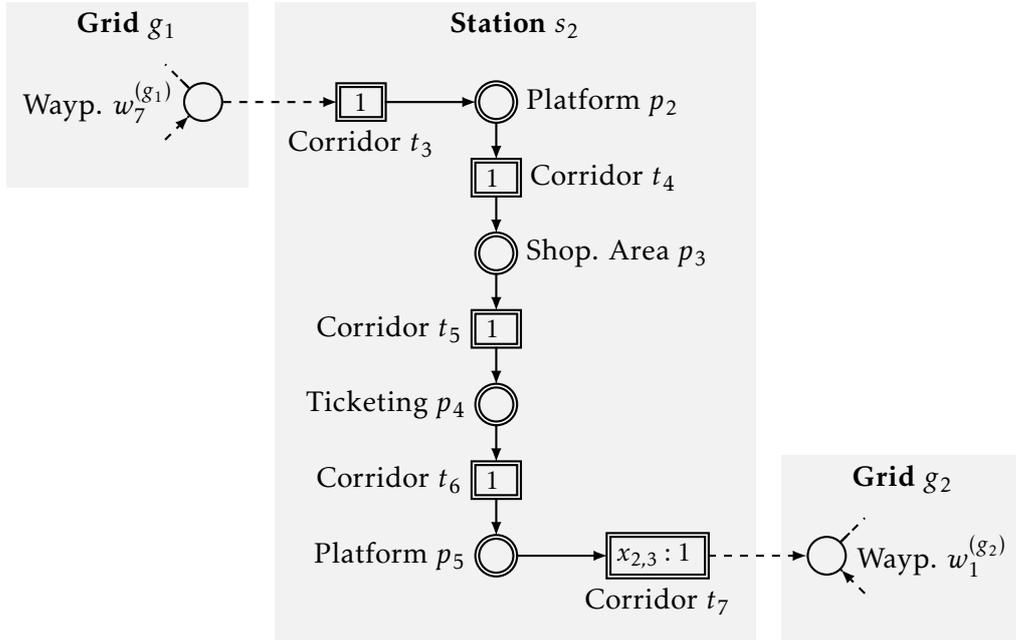


Figure 5.2.4: Interior of station s_2 and its connection to the grid g_1 (g_2) which accommodates the commuter train line (people mover line)

Last but not least, Fig. 5.2.3 below depicts the interior of s_3 , and the way it is connected to g_2 . According to this graphical specification, passengers can enter s_3 via the corridor t_8 from vehicle v_3 (v_4) iff v_3 (v_4) is stopped at the waypoint $w_7^{(g_2)}$. From p_5 all passengers can leave s_3 to the exterior of the considered transportation network through the corridor t_9 . Thus, the set of all gathering points in s_2 is $P_{gp}^{(s_3)} := \{p_6\}$, the set of all corridors is $T_c^{(s_3)} := \{t_8, t_9\}$, and the set of edges connecting all gathering points and corridors is $E_{st}^{(s_3)} := \{(t_8, p_6), (p_6, t_9)\}$.

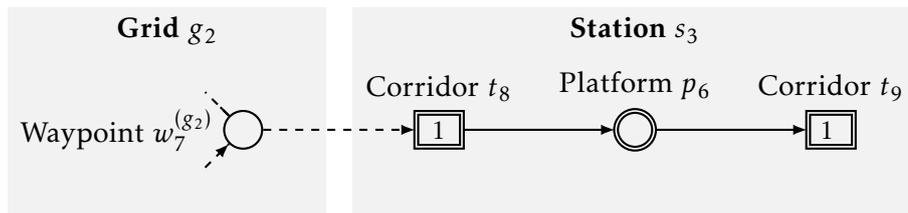


Figure 5.2.5: Interior of station s_3 and its connection to the grid g_2 which accommodates the people mover line

Vehicle Operation. The two vehicles v_1 and v_2 execute a transportation service $x_{1,2}$ from s_1 to s_2 , and a dead-heading from s_2 back to s_1 in an endless loop. Similarly, the two vehicles v_3 and v_4 execute a transportation service from s_2 to s_3 and a dead-heading from s_3 back to s_2 . The minimum dwell time imposed on v_1 and v_2 at $p_1^{(g_1)}$ and $p_7^{(g_1)}$ is 30 seconds, and the corresponding maximum dwell time at

both stops is 120 seconds. We choose

$$\lambda_s \left(g_1, p_1^{(g_1)} \right) = \lambda_s \left(g_1, p_7^{(g_1)} \right) = 1,$$

which implies that every departure from $p_1^{(g_1)}$ and $p_7^{(g_1)}$ of v_1 and v_2 is considered to be an on-time departure (refer to the elaboration of a vehicle mission in Sec. 3.3.1 on p. 50 for the meaning of the parameter λ_s). Similarly, the minimum dwell time imposed on v_3 and v_4 at $p_1^{(g_2)}$ and $p_7^{(g_2)}$ is 30 seconds, but the corresponding maximum dwell time at both stops is only 60 seconds. Moreover we choose

$$\lambda_s \left(g_2, p_1^{(g_2)} \right) = \lambda_s \left(g_2, p_7^{(g_2)} \right) = 1.$$

Passenger Routing. We have to specify some passenger routing here, although we do not want to compute any forecast. In fact, we want to evaluate the efficiency of two of our algorithmic bricks (referred to as B_1 and B_2 ; see preface) which try to reduce the size of any computed propagation DAG. However, no passenger routing means no passenger number-dependent actions which in turn implies no alternative mode transitions but propagation DAGs which resemble simple paths instead. That is why, we specify some very simple passenger routing here. More specifically, all passengers have the same trip profile; cf. the annotation of all corridors in Fig. 5.2.3 - Fig. 5.2.5. They enter our two-lines transportation network through the corridor t_1 in s_1 . From p_1 in s_1 they want to board some vehicle which provides the transportation service $x_{1,2}$ from s_1 to s_2 . Arrived at the platform p_2 in s_2 , all passengers want to transfer to the shopping area p_3 , from p_3 to the ticketing area p_4 , and from p_4 to the platform p_5 . From p_5 , all passengers want to board some vehicle which provides the transportation service $x_{2,3}$ from s_2 to s_3 . Finally, arrived at the platform p_6 in s_3 , all passengers want to leave s_3 to the network's exterior.

Specification of All Simulation Runs

We assume the same initial state of our two-lines transportation network for all simulation runs, where

- v_1 is stopped at $w_1^{(g_1)}$,
- v_2 moves towards $w_8^{(g_1)}$,
- v_3 moves towards $w_2^{(g_2)}$, and
- v_4 moves towards $w_9^{(g_2)}$.

Moreover, we assume that the initial valuations of all four clocks $c^{(v_i)}$, with $i \in \{1, 2, 3, 4\}$, associated with the operation of v_i are zero.

Note that we have to specify some non-empty set of target vehicles or some non-empty set of target gathering points here (refer to Sec. 4.1.1 on p. 102ff. for the meanings of “target vehicles” and “target gathering points”) for the computation of all propagation DAG. Otherwise, the activation of our two algorithmic bricks, B_1 and B_2 , cannot show any positive effect. In this context, note that for all simulation runs we consider the platform p_6 in station S_3 as the only target gathering point. We do not mark any vehicle as target vehicle, though.

Table 5.1 below specifies seven different simulation runs $R_{n,x}$. Every $R_{n,x}$ defines some fixed time step which is used for the incrementation of all clocks that are associated with the operation of all considered vehicles. This parameter thus defines the number of layers in/height of the to be computed propagation DAG. Thus, a fixed discrete simulation time step of (i) 60 seconds implies 21 time layers of nodes, (ii) 30 seconds implies 41 layers of nodes, and (iii) of 15 seconds implies 81 layers of nodes. Moreover, every $R_{n,x}$ defines some default value for the fixed lengths of all operational freezing times in both lines if they occur/will be enforced; because some vehicle’s dwell at a stop exceeds its maximum admissible dwell time at this stop. Last but not least, every $R_{n,x}$ defines whether the propagation DAG shall be computed with B_1 and/or B_2 being activated.

Table 5.1: Overview of all simulation runs, where B_1 refers to our algorithmic brick from Sec. 4.4.7 on p. 140ff. and B_2 refers to our our algorithmic brick from Sec. 4.4.8 on p. 144ff.

Sim. Run	Time Step [s]	Freezing Time [s]	B_1	B_2
$R_{1,a}$	60	0		
$R_{1,b}$	30	0		
$R_{1,c}$	15	0		
$R_{2,a}$	30	60		
$R_{2,b}$	30	120		
$R_{3,a}$	30	0	✓	
$R_{3,b}$	30	0		✓

The first five simulation runs $R_{1,x}$ and $R_{2,x}$ have to be executed without B_1 and B_2 being activated. More particularly, all $R_{1,x}$ also disregard the possibility that the operation of some vehicle is frozen after its dwell time at some stop has exceeded the vehicle’s maximum admissible dwell time at this stop. All first three simulation runs $R_{1,x}$ shall thus enable us to analyze how the number of nodes, modes, etc. in the computed propagation DAG change as a function of the discrete simulation time step chosen. Compared to all $R_{1,x}$, all $R_{2,x}$ shall enable us to analyze the impact of decreasing operational freezing times on the computation of the considered propagation DAG. Finally, all $R_{3,x}$ shall enable us to analyze the

effectiveness and efficiency of our two algorithm bricks, B_1 and B_2 .

We will use simulation run $R_{1,b}$ as common reference for the evaluation of all other simulation runs.

Computed Results

Before we depict all diagrams which illustrate the evolution of all considered parameters (P_1 to P_4) for all simulation runs as a function of our model’s simulation time below, we first state here what we already know. From Eqn. 4.11 on p. 115 we know that the number of nodes in any propagation DAG computed for some simulation run with a discrete time step of (i) 60 seconds cannot exceed 10^{24} , (ii) 30 seconds cannot exceed 10^{48} , and (iii) 15 seconds cannot exceed 10^{96} ; which latter upper limits will hopefully prove to be far too conservative. Acc. to (4.13) on p. 125, we moreover know that we may not encounter more than

$$|\Delta| \leq \left(1 + \sum_{i=1}^2 \frac{2!}{(2-i)!} 2^i \right)^2 = 169$$

macro modes. In addition we know - independently of the considered simulation run - the first discrete time step when the use of our algorithmic brick B_1 should show some first effect, namely two discrete simulation time steps prior to the terminal simulation step since it takes a disturbance of the passenger load at platform p_2 in station s_2 at least three discrete simulation time steps before it can affect the passenger load at platform p_5 . Finally, we know which set of atomic actions our vehicles might perform during the execution of the different simulation runs; cf Tab. 5.2 below: Branches in any computed propagation DAG either result from the fact that some vehicles perform some “LD Mission Change”, or that some vehicles perform some “LD On-Time Departure”, or that two or more vehicles perform both at the same time.

Table 5.2: Overview of all possible atomic actions which may be performed by some vehicle during the execution of the considered simulation runs

Atomic Action	$R_{1,a}$	$R_{1,b}$	$R_{1,c}$	$R_{2,a}$	$R_{2,b}$	$R_{3,a}$	$R_{3,b}$
LD Mission Change	✓	✓	✓	✓	✓	✓	✓
LD On-Time Departure	✓	✓	✓	✓	✓	✓	✓
LI Intended Vehicle Stop	✓	✓	✓	✓	✓	✓	✓
LI Mission Change	✓	✓	✓	✓	✓	✓	✓
LI Operational Freeze				✓	✓		
LI Operational Unfreeze				✓	✓		
LI Safety Vehicle Stop	✓	✓	✓	✓	✓	✓	✓
LI Stop Skipping	✓	✓	✓	✓	✓	✓	✓

Impact of Number of Simulation Time Steps. Fig. 5.2.6 below depicts the evolution of the number of nodes in the propagation DAGs computed for all simulation runs $R_{1,x}$, which latter simulation runs differ from each other in the number of simulation time steps, which equidistantly discretize the same forecast horizon; thus producing propagation DAGs with different heights. Notice that the number of nodes in all propagation DAGs increase exponentially as a function of the simulation time. As expected, increasing the number of simulation time steps further increases the number of nodes. Also notice that due to an overflow of the DDR3 memory of our test platform we did not manage to compute the complete propagation DAG for $R_{1,c}$.

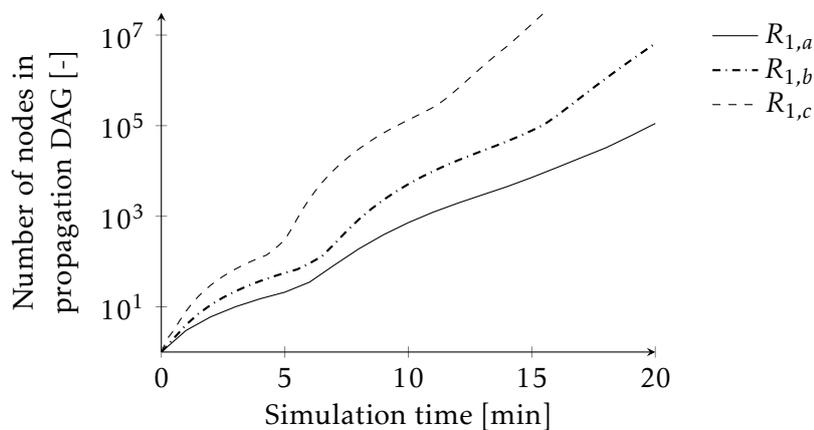


Figure 5.2.6: Number of nodes in propagation DAG for all simulation runs $R_{1,x}$

As can be seen in the next graph, Fig. 5.2.7 below, we cannot claim that an increase in the number of simulation time steps tremendously increases or decreases the number of different modes in the computed propagation DAG.

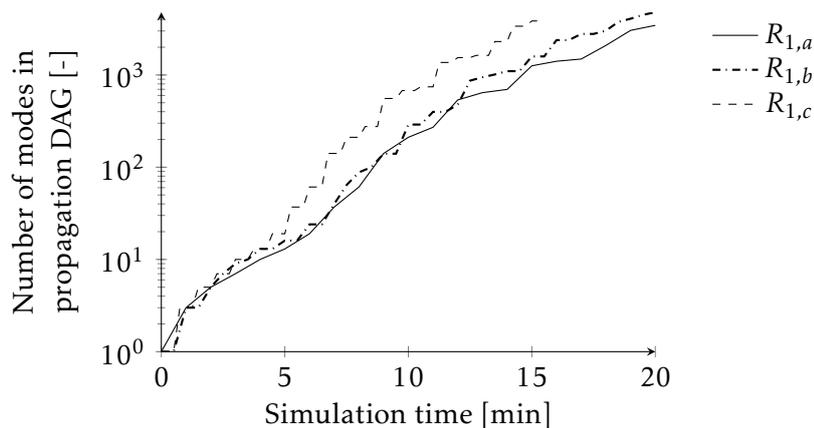


Figure 5.2.7: Number of modes in propagation DAG for all simulation runs $R_{1,x}$

This latter observation of nearly identical curves corresponds to our expectations.

In fact, every mode defines a particular realization of all vehicle's (discrete) positions, their (discrete) driving conditions, and their (discrete) operational states. Increasing the number of simulation time steps which equidistantly discretizes the same forecast horizon does not alter any time threshold for the vehicles' atomic actions, but makes our automaton approach these thresholds more softly. We thus expect our automaton to approach some particular mode transition in less simulation time if the increments of all clocks associated with the operation of all vehicles are smaller.

Next, Fig. 5.2.8 below indicates the circumstance that increasing the number of simulation time steps which discretize the complete forecast horizon might over-exponentially increase the computation time of the propagation DAG in the long run; at least when using our software module FETN. This phenomenon might be explained as follows. Normally we would expect the elapsed computation time to grow exponentially since the numbers of nodes in the different layers of the considered propagation DAGs grow exponentially as a function of the simulation time; and the number of overall computation steps linearly grows with the number of nodes. However, the computation is performed on some real capacity-limited workstation and not on some e.g. idealized Turing machine. Thus increasing the number of nodes above some threshold might cause some kind of a competition for shared memory between all different processes which run on the same simulation platform (our workstation).

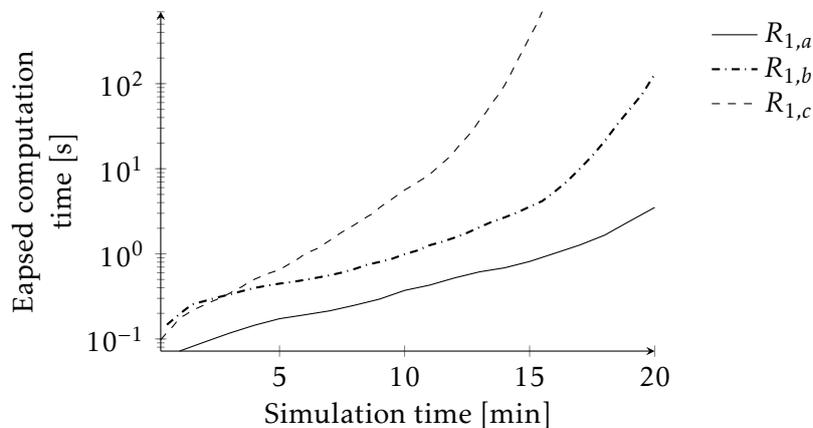


Figure 5.2.8: Elapsed computation time of propagation DAG for all simulation runs $R_{1,x}$

Figure 5.2.9 below illustrates that the number of macro modes computed in the different layers of the considered propagation DAG as a function of our model's simulation time does not significantly differ between all simulation runs $R_{1,x}$. Again this observation is not something unexpected. Indeed, every macro mode defines which vehicle is stopped in front of which platform for the purpose of boarding & alighting, together with the mission of this vehicle and thus the transportation service it provides. It follows that T -SHA's macro mode changes iff some

vehicle stops or departs; where this latter atomic action must consume some minimum amount of time.

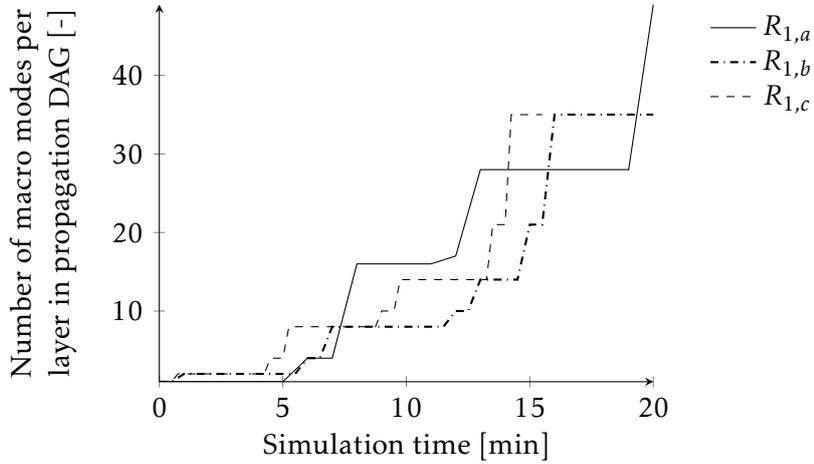


Figure 5.2.9: Number of macro modes in different layers of propagation DAG for all simulation runs $R_{1,x}$

This minimum time-consumption in turn is independent of the length of the discrete simulation time step chosen. However, increasing the number of simulation time steps in the considered forecast horizon means that *T-SHA* approaches any transition between two modes which belong to two different macro modes in more fine-granular steps. We thus expect our automaton to approach some new macro mode in less simulation time if the increments of all clocks associated with the operation of all vehicles are smaller.

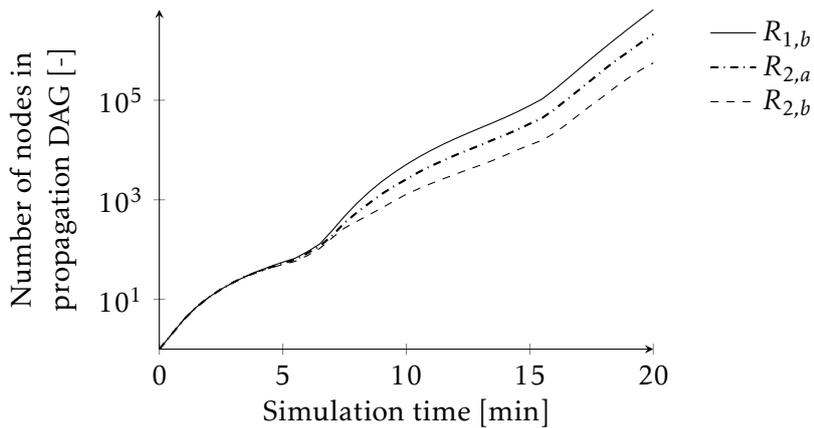


Figure 5.2.10: Number of nodes in propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference

Impact of Non-zero Freezing Times. Figure 5.2.10 above depicts the evolution of the number of nodes in the propagation DAGs computed for all simulation runs

$R_{2,x}$, which latter simulation runs differ from each other in the way they proceed with vehicles whose dwell times exceed preselected values (= maximum dwell times) at stops. More specifically, in $R_{2,a}$, the operation of vehicles with excess dwell times at stops is frozen for one minute before said vehicles can try to depart once again, whereas it is frozen for two minutes in $R_{2,b}$ (cf. Tab. 5.1 on p. 164). In $R_{1,b}$, which serves as reference, the operation of those vehicles is not frozen at all.

Figure 5.2.10 suggests that simulation runs with non-zero freezing times tend to produce more compact propagation DAGs. More specifically, it suggests that the higher the freezing times are chosen, the more compact the computed propagation DAGs are. Note that this observation fully corresponds to our understanding of the model dynamics. In fact, the larger the value for the freezing time of some vehicle v with an excess dwell time at some stop w is specified, the lower the number of discrete simulation time steps when we allow v to depart from w in the considered forecast horizon is. This reduced number of admissible departure times of v from w in turn implies that we consider fewer alternative evolutions of v 's discrete position, -driving condition, and -operational state.

On the other hand, Fig. 5.2.11 below suggests that the particular choice of the vehicles' freezing times does not have any significant impact on the number of modes in the computed propagation DAGs; at least for our trivial test case. However, note that this observation cannot be projected to any more complex test case, where the introduction of non-zero freezing times might either increases or decreases the number of modes in the computed propagation DAGs.

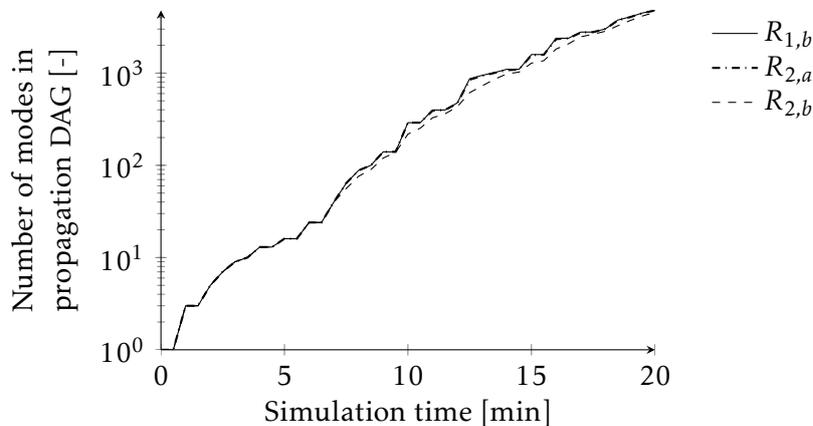


Figure 5.2.11: Number of modes in propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference

Figure 5.2.12 below depicts the evolution of the elapsed computation times for the propagation DAGs of $R_{1,b}$, $R_{2,a}$, and $R_{2,b}$ as a function of the model's simulation time. We can see that all elapsed simulation times exponentially increase until the model's simulation time exceeds around 15 minutes, from which time point on they over-exponentially increase for all $R_{1,b}$, $R_{2,a}$, and $R_{2,b}$. Moreover, we can see that the computation of the propagation DAG for $R_{2,b}$ consumes less time than it

does for $R_{2,a}$, and the computation of the propagation DAG for $R_{2,a}$ consumes less time than it does for $R_{1,b}$ over the complete simulation time horizon.

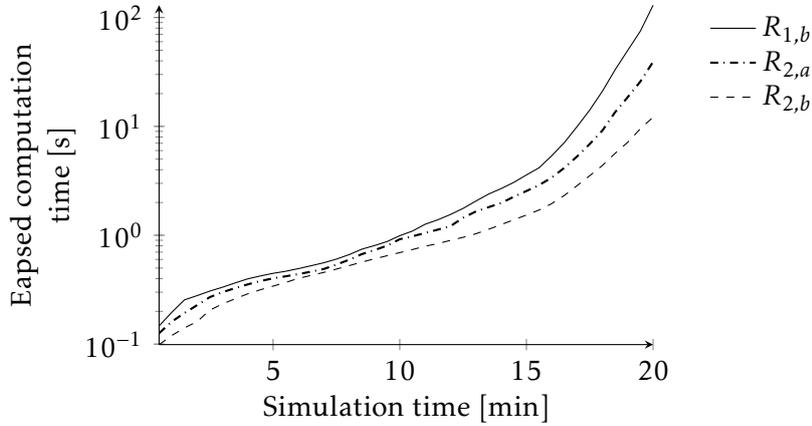


Figure 5.2.12: Elapsed computation time of propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference

Figure 5.2.13 below shows that the specification of non-zero freezing times in our simple example has no impact on the evolution of the number of macro modes in the computed propagation DAG.

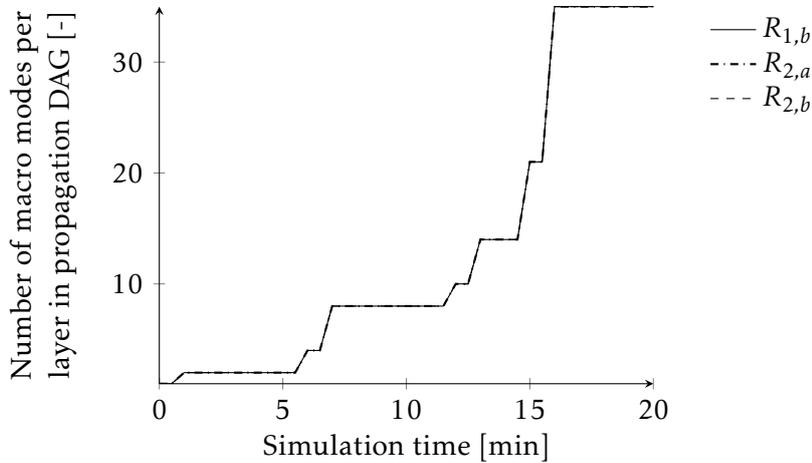


Figure 5.2.13: Number of macro modes in different layers of propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference

However, note that this observation cannot always hold for all more complex test cases. To see that this is true, imagine some test case, where two vehicles, say v_1 and v_2 , are initially stopped. Moreover, suppose that v_1 cannot depart as long as v_2 has not departed before, because v_2 blocks the next position of v_1 . Finally, suppose that v_2 departs after v_1 's dwell time exceeds some maximum dwell time. Now if v_1 's operation were about to be frozen due to this involuntary excess of dwell time,

then the distance v_1 could cover in the considered transportation infrastructure before the end of the specified forecast horizon would significantly differ from the distance v_1 could cover if its operation is not frozen. The different distances in turn could imply that v_1 manages to stop at some station for the purpose of boarding & alighting in one configuration of some freezing time, whereas it is not in some other.

Impact of Our Algorithmic Bricks. Figure 5.2.14 below depicts the evolution of the number of nodes in the propagation DAGs computed for the simulation run $R_{3,a}$ (with algorithmic brick B_1 being activated; cf. Tab. 5.1 on p. 164), $R_{3,b}$ (with algorithmic brick B_2 being activated), and $R_{1,b}$ (with neither B_1 nor B_2 being activated). We can see that B_2 manages to reduce the number of nodes in the propagation DAG computed for $R_{3,b}$ by almost ten to the power of two w.r.t. the number of nodes in the propagation DAG computed for $R_{1,b}$. We moreover see that, although less effective than B_2 , B_1 still manages to significantly reduce the number of nodes in the computed propagation DAG.

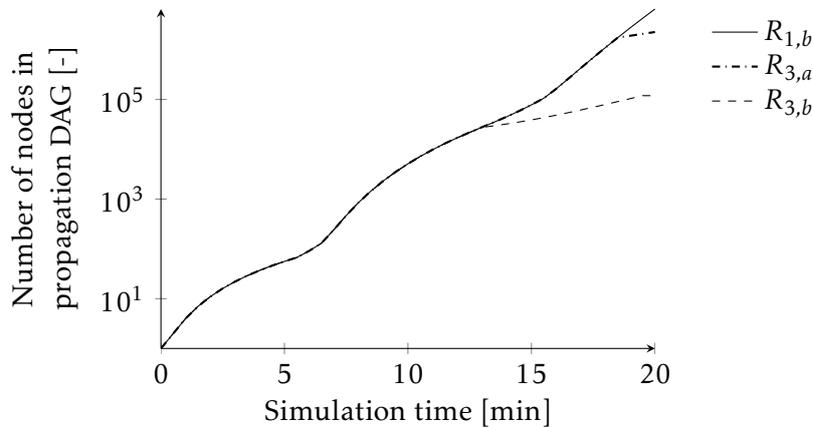


Figure 5.2.14: Number of nodes in propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference

Next, Fig. 5.2.15 below depicts the number of different nodes in the propagation DAGs computed for $R_{1,b}$, $R_{3,a}$, and $R_{3,b}$. We can see from this figure that the both B_1 in $R_{3,a}$ and B_2 in $R_{3,b}$ reduce this number. Moreover, we can see that the activation of B_2 in $R_{3,b}$ begins to show effect on the number of nodes in the computed propagation DAG around simulation time 9 minutes before it affects the number of nodes therein around simulation time 12 minutes. Note that this observation is not intuitive.

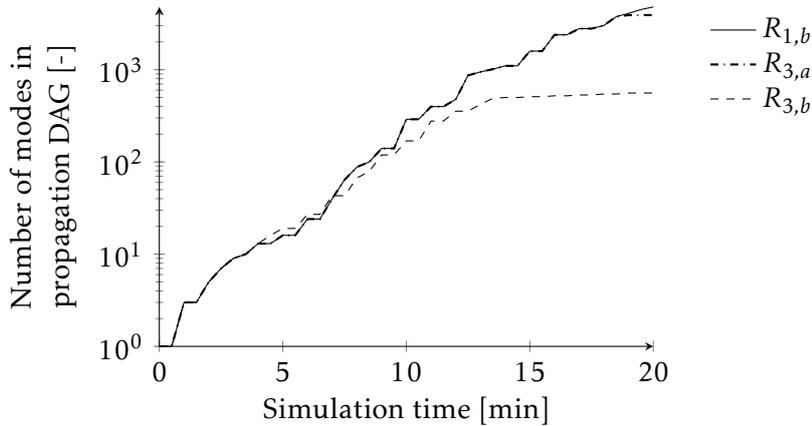


Figure 5.2.15: Number of modes in propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference

From Fig. 5.2.16 below we can see that the activation of B_2 in $R_{3,b}$ has a cost, which has to amortize. More specifically, we can see that the elapsed computation time for the propagation DAG in $R_{3,b}$ between simulation time 5 minutes and simulation time 10 minutes is strictly bigger than it is for the propagation DAG in $R_{1,b}$. Only then, when the simulation time of our model exceeds around 10 minutes, the elapsed simulation time in $R_{3,b}$ stays strictly below the elapsed computation time in $R_{1,b}$.

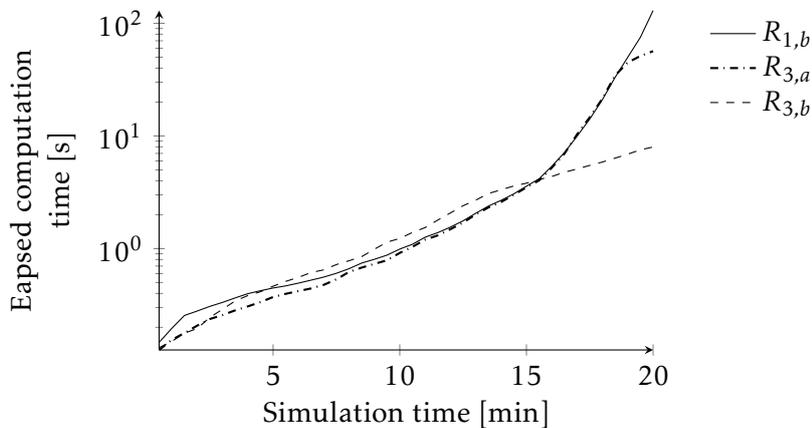


Figure 5.2.16: Elapsed computation time of propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference

Moreover, note that the graphs for the elapsed computation times of the propagation DAGs computed for $R_{1,b}$ and $R_{3,a}$ almost overlap until when B_1 starts to show effect on the computed number of nodes and modes in Fig. 5.2.14 and respectively Fig. 5.2.15 above. However, this observation is not a surprise at all. In fact, FETN computes the first simulation time step, say i , when B_1 may show effect prior to the actual execution of $R_{3,a}$. Before i , FETN does not compute any operation else

than it already does in $R_{1,b}$. In this context, note that deviations in the five simulation time steps must result from the execution of some other - possibly operating system-related - processes.

Figure 5.2.17 below depicts the impact of B_1 and B_2 on the number of different macro modes in the propagation DAGs computed for $R_{1,b}$, $R_{3,a}$ and $R_{3,b}$.

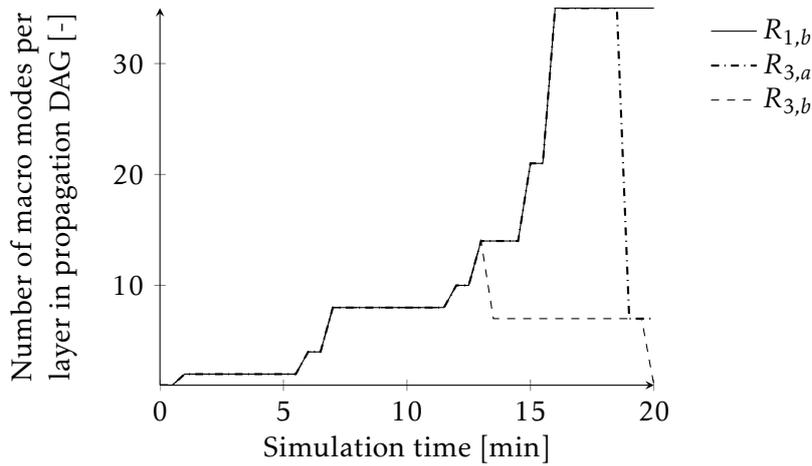


Figure 5.2.17: Number of macro modes in different layers of propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference

It can be seen, that B_2 not only shows effect much earlier than B_1 , but also to a higher extend. In fact, whereas the number of macro modes in the propagation DAG computed for $R_{3,a}$ drops to around 8, it drops to 1 in the propagation DAG computed for $R_{3,b}$.

Conclusions

Table 5.3 depicts some major numerical indicators of our above simulation runs.

Table 5.3: Overview of all computed parameters

	$R_{1,b}$	$\frac{R_{1,a}}{R_{1,b}}$	$\frac{R_{2,a}}{R_{1,b}}$	$\frac{R_{2,b}}{R_{1,b}}$	$\frac{R_{3,a}}{R_{1,b}}$	$\frac{R_{3,b}}{R_{1,b}}$
Total number of nodes	6.54×10^6	0.02	0.32	0.09	0.35	0.02
Total number of modes	4784	0.72	1.00	0.95	0.82	0.12
Complete computation time	129.72	0.03	0.30	0.09	0.44	0.06
Average number of macro modes in every layer	13.24	1.22	1.00	1.00	0.85	0.43

More particularly, Tab. 5.3 lists the total number of nodes and modes in the propagation DAG computed for simulation run $R_{2,b}$, as well as the complete computa-

tion time for this graph. It moreover depicts the average number of macro modes in the DAG's different layers. Next to these parameter values for $R_{1,b}$, we have written down how these values change for $R_{1,a}$, all simulation runs $R_{2,x}$, and all simulation runs $R_{3,x}$. Note that simulation run $R_{1,c}$ is missing here since we did not manage to compute a propagation DAG for this simulation run on our workstation, which covers the complete forecast horizon of 20 minutes given the run's relatively small simulation time step of 15 seconds; as opposed to a fixed time step of 60 seconds for $R_{1,a}$, and a fixed time step of 30 seconds for all other simulation runs.

From this above table it can be seen that B_2 outperforms (only activated in $R_{3,b}$) all other measures (B_1 , less-granular simulation time steps, non-zero freezing times) in our very simple test case, in terms of the number of nodes and modes the computed propagation DAG has, and the average number of macro modes in its different layers of nodes. Note that it is doing so without deteriorating the result of any computed forecast in that it e.g. requires us to reduce the temporal discretization of the model's simulation. Also note that the computed results for B_1 (only activated in $R_{3,a}$) are remarkable. Although this brick first engages into the computation of the propagation DAG two simulation steps away from the end of the considered forecast horizon, its intervention reduces the total number of nodes in the complete propagation DAG by almost one third and decreases the computation time thereof by more than 50 percent. Finally, note that non-zero freezing times, which very likely better capture the dynamics observed in the real transportation network, may also significantly reduce the computation time of the propagation DAG.

Of course, the above results cannot be easily projected to more complex test cases, where more vehicles may perform more passenger number-dependent actions at the same time. Nevertheless, our simple test case shows that decreasing the number of simulation time steps in the considered forecast horizon should not be the first option we run for if we want to confine the node explosion of some propagation DAG during its layer by layer computation. It inevitably diminishes the accuracy of the computed forecast since we let the state of the considered transportation network evolve at a slower pace, which also includes the redistribution of all passenger number densities. However, we do not necessarily have to pay this price if we manage to develop algorithms which are very similar to B_1 and B_2 .

Ensure Smooth Passenger Transfer Flows At Massy

Motivation

As outlined in the introduction to this report, the primary goal of our work is to provide to transportation operators and -authorities a tool which allows them to anticipate the impact of different operational strategies (introduce some additional vehicle into service, delay the dispatch of some parked vehicle, provide

some re-routing information to all passengers, and so on) when being applied to their transportation lines and modes in degraded situations (some vehicle blocks some line due to an engine failure, etc.) of operation, where

- the degraded operational state of the considered network is known (where are all vehicles, what are they doing, and so on),
- the passenger numbers on-board all vehicles and at the various gathering points in all considered stations can be estimated, and
- estimations for all passenger arrivals flows to the considered (part of the) network exist.

In this context, we will consider in the use case below a transportation incident and its resolution which employs our SHA model.

It is Monday 8.05am. We are sitting in a control center somewhere in Paris, where we supervise all passenger transfer flows between the two commuter train lines RER B and RER C. Suddenly, an alarm pops up which turns the huge control monitor in front of us into some symbiosis of red flashing lights. Apparently, some person is on the tracks of line RER B in between the two stations “Les Baconnets” and “Fontaine Michalon” south of Paris. Our experience tells us that - given the day, the time, and the affected part of the transportation line - this person is very likely some young adult who likes to share with us his/her graffiti artwork. We thus expect the track section between Les Baconnets and Fontaine Michalon to be blocked in both directions for at most the next 20 minutes; i.e. until 8.25am. We now try to find out whether the line RER C can absorb all additional re-routed passengers who transfer from line RER B to line RER C at Massy–Palaiseau in the next twenty minutes, so as to continue their trip to Paris; without making any modification to line RER C’s operation, and with doubling the maximum dwell times at Massy–Palaiseau of all vehicles which (i) are operated on line RER C, (ii) move towards Massy–Palaiseau in the direction to Paris, (iii) are supposed to stop at Massy–Palaiseau, and in doing so (iv) establish a critical link for all stressed commuters to Paris.

Following the above narrative introduction to our use case, we will look in Sec. 5.3.2 at the scenario of this use case and its objectives in more detail. We will then discuss very briefly in Sec. 5.3.3 the specification of the considered transportation network (its infrastructure, as well as the vehicle operation and the passenger routing therein), before we will illustrate and analyse some computed forecasts in Sec. 5.3.4. Finally, we will draw some conclusions in Sec. 5.3.5.

Scenario Description & Objectives

The Network. The commuter train lines RER B and RER C which we consider in our use case here, define two major arteries in the Parisian transportation network.

The line RER B connects the south-west of the greater Parisian area with its north-east. The line RER C on the other hand, connects the south-west with the north-west. Both lines serve Massy–Palaiseau, which major transportation hub is located around 15 kilometres south of the center of Paris. The connection of this hub with the two lines is illustrated in Fig. 5.3.1 below.

Figure 5.3.1: Passenger transfer possibility between the two commuter train lines RER B (dark gray line) and RER C (light gray line) in the south of Paris at Massy–Palaiseau



The two lines RER B and RER C are haunted by frequent interruptions with many different causes for these interruptions. This low-operational performance of both lines is reported e.g. in STIF's (Syndicat des transports d'Île-de-France) trimestrial published reports on the lines' service punctualities (see e.g. p. 3-4 in nr. 24 of "Bulletin de la qualité de service trimestriel", published in August 2016). Fortunately however, major incidents which interrupt the operation of both lines at the same time occur less often given the physical separation of their rail tracks. Amongst others, this means that passengers who commute to the centre of Paris might transfer at Massy-Palaiseau from RER B to RER C, and vice versa, if some incident blocks one line somewhere in between Massy–Palaiseau and Paris but not the other. In this context, note that our use case does not portray some artificial situation, but the sad truth many commuters travelling in and out of Paris are confronted with a lot of times every year.

The Forecast Objective. The objective of our use case is to anticipate the impact of two different maximum dwell times at Massy–Palaiseau for all vehicles that are operated on line RER C and provide a transportation service from Versailles to Paris; namely 80 seconds as stated for their nominal operation, and 160 seconds which means a doubling of this nominal value. More specifically, we want to find out if such an increase of the vehicles' maximum dwell times from 80 to 160 seconds may better ensure the safety of all transferring passengers in the station Massy–Palaiseau in the next 20 next minutes, i.e. in the time interval [8.05am, 8.25am]; where this safety of all passengers is considered to be not given

if passengers cannot leave this station in case e.g. of a fire or some panic attack because its exit area is overcrowded. Of secondary interest to us, is the impact of the extended maximum dwell times at Massy–Palaiseau on the passenger numbers on-board all vehicles operated on line RER C since too many passengers on-board these vehicle may cause bunching effects. Although of minor interest, we also want to know the passenger numbers on-board all other vehicles.

Specification of Transportation Network

Very similar to the specification of the simplistic two lines transportation network for our test case in Sec. 5.2.2 on p. 160ff., we will look at the specification of the infrastructure of the considered network for our use case, as well as the vehicle operation and the passenger routing therein in the given order next. Finally, we will look at the specification of all passenger flows and initial passenger numbers.

Infrastructure. Figure 5.3.2 below depicts our very simplified model of Massy–Palaiseau.

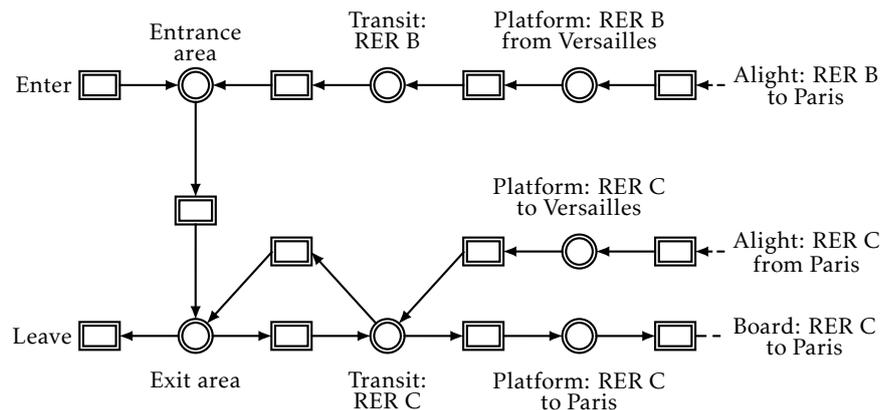


Figure 5.3.2: Layout of transportation hub Massy–Palaiseau

This model (= station) contains all gathering points in the transfer hub Massy–Palaiseau which may be occupied by passengers who are re-routed from the line RER B to the line RER C on their way to Paris, as well as one additional gathering point (= a platform) so as to account for the interaction of this transfer flow with all oppositely-directed passenger flows. More specifically, the model's set of gathering points comprises a platform which accommodates all passengers who

- alight from a vehicle at Massy–Palaiseau which provides a transportation service on line RER B from Orsay to Paris,
- alight from a vehicle at Massy–Palaiseau which provides a transportation service on line RER C from Versailles–Chantiers to Paris, and
- board a vehicle at Massy–Palaiseau which provides a transportation service on line RER C from Versailles–Chantiers to Paris.

In addition, the model's set of gathering points comprises one entrance area which accommodates all passengers who start their trips in the considered network at Massy–Palaiseau, one exit area for all passengers who leave the considered transportation network at Massy–Palaiseau, and two transit areas which are located in between all modelled platforms on the one hand, and Massy–Palaiseau's modelled exit- and entrance area on the other hand. Since all vehicles which are operated on line RER B cannot provide a transportation service to Paris in the considered time frame, we have not included another platform in our model for Massy–Palaiseau which is dedicated to all passengers who board some vehicle which provides a transportation service on this line to Paris. In other words, we ignore all possible passenger flows in Massy–Palaiseau from the depicted entrance area to this additional platform. Moreover, we assume that every gathering point can accommodate at maximum 200 passengers.

Apart from Massy–Palaiseau we will not consider boarding & alighting passenger flows at any other station along the two lines RER B and RER C. We thus do not model them.

We model the line RER B from the station “Orsay” which is located south to Massy–Palaiseau, to the station “Saint-Michel - Notre-Dame” which is in Paris, in form of a simple path. Every waypoint along this path (= transportation grid) corresponds to a stop at Orsay, at Saint-Michel - Notre-Dame, or at any station in between; or it corresponds to a vehicle's position along this line between two consecutive stops, where in our model exactly two waypoints separate every pair of two consecutive stops. We moreover extend this path by a sequence of some additional waypoints and tracks which we append to the stop at Massy–Palaiseau. These latter waypoints model parking positions.

Our model for the line RER C in both directions from Paris to Versailles–Chantiers, and vice versa, is limited to the section from Saint-Michel - Notre-Dame in Paris to Versailles–Chantier south to Massy–Palaiseau. Again we model both directions as simple paths, with as many waypoints as there are stations in the considered section plus two additional waypoints between every pair of two consecutive stops.

Vehicle Operation. In total, we consider the operation of six vehicles $v \in \mathcal{V}$, with $\mathcal{V} := \{v_1, v_3, v_4, v_6, v_7, v_{14}\}$, on both lines. Every $v \in \mathcal{V}$ can accommodate up to 400 passengers on-board its cabin and executes one particular mission in the considered forecast horizon. This mission w_i , with $i \in \{1, 2, 3\}$, specifies a transportation service from or to Paris w.r.t. Massy–Palaiseau along either line RER B or line RER C. See Tab. 5.4 below.

Table 5.4: Initial vehicle operational state of our use case at time 8.05am, when the fictitious passenger incident occurs

Mission	Line	Direction	Executing Vehicles
w_1	RER B	To Paris	v_6, v_7
w_2	RER C	To Paris	v_1, v_4, v_{14}
w_3	RER C	From Paris	v_3

Note that all vehicles which execute w_1 , and thus are supposed to provide a transportation service from Orsay to Paris on line RER B, with a stop at Massy–Palaiseau, enter the above mentioned depot at Massy–Palaiseau and park there once all passengers have alighted from their cabins. In doing so, they clear the stop along line RER B and in front of Massy–Palaiseau for a vehicle which may arrive on the same line and in the same direction next.

Table 5.5 below lists the scheduled arrival times of all vehicles at Massy–Palaiseau w.r.t. time 8.05am, which we have adopted from published timetables for both lines.

Table 5.5: Scheduled arrival times in seconds/minutes of all vehicles at Massy–Palaiseau w.r.t. time 8.05am, when the fictitious passenger incident occurs in our use case

v_1	v_3	v_4	v_6	v_7	v_{14}
300/5	660/11	0/0	210/3.5	480/8	930/15.5

We have used the same timetables to specify the initial positions, driving conditions, and elapsed dwell- and driving times of $v \in \mathcal{V}$ at time 8.05am, where v_4 is the only vehicle which is initially stopped at Massy–Palaiseau. By assumption, its initial dwell time is zero. Moreover, we derived the vehicles' driving times from these timetables; or more specifically from all indicated arrival- and consecutive departure times at/from the stops.

Passenger Routing. We group all passengers into five different trip profiles $\gamma \in \mathcal{Y}$, with $\mathcal{Y} = \{1, 2, 3, 4, 6\}$; cf. Fig. 5.3.3 below.

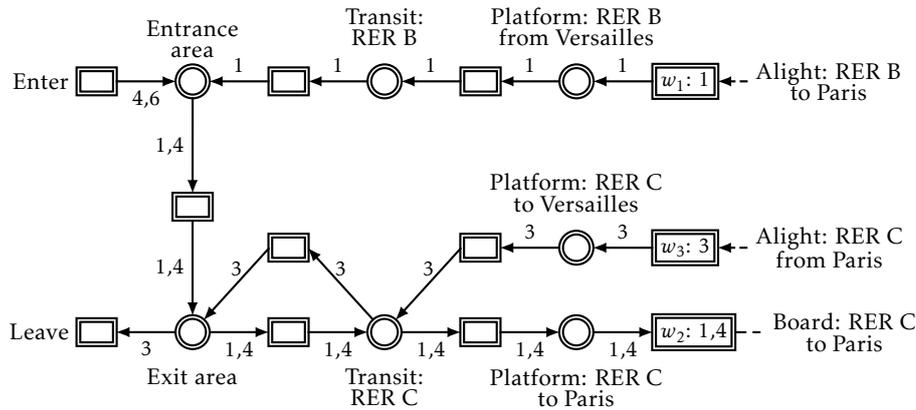


Figure 5.3.3: Layout of transportation hub Massy–Palaiseau together with the annotation of the routing of all passengers in this station

Note that we leave out trip profile 5 here. We introduced this trip profile to capture the routing of all passengers who board a vehicle at Orsay which provides a transportation service on line RER B to Paris according to w_1 . We assume here that all passengers on-board such a vehicle are immediately informed about the passenger incident upon its occurrence; and as a consequence of this passenger information instantaneously adopt trip profile 1 at (the initial simulation) time 8.05am.

All passengers of trip profile 4 enter the considered transportation network at Massy–Palaiseau, where they want to board a vehicle which executes w_2 ; i.e. a vehicle which provides a transportation service to Paris on line RER C. All passengers with trip profile 6 also enter the considered transportation network at Massy–Palaiseau. However compared to all passengers with trip profile 4, passengers with trip profile 6 want to board a vehicle which executes w_1 , i.e. a vehicle which provides a transportation service to Paris on line RER B and not on line RER C. Upon arrival at Massy–Palaiseau, these passengers with trip profile 6 learn from some loudspeaker announcement about the passenger incident which blocks the operation of line RER B in the direction to Paris. As a consequence of this passenger announcement, they decide to take line RER C instead; which latter re-routing decision implies that all of them enter Massy–Palaiseau’s entrance area with trip profile 1 and not 6 (= a one hundred percent re-routing of a network inflow). All passengers with trip profile 1 (i) enter the considered transportation network at Orsay (not modelled), board some vehicle operated on line RER B which provides a transportation service to Massy–Palaiseau as specified in mission w_1 , (ii) alight from this vehicle at Massy–Palaiseau, and (iii) transfer to another vehicle which provides a transportation service on line RER C to Paris as specified in mission w_2 . All passengers with trip profile 2 board a vehicle which executes w_2 at Versailles–Chantiers so as to go to Paris. They do not want to alight at Massy–Palaiseau but simply block some physical space on-board this vehicle’s cabin which is not available to all passengers who want to board the same vehicle at Massy–Palaiseau. Last

but not least, all passengers with trip profile 3 travel from Paris to Massy–Palaiseau on-board some vehicle which executes w_3 , i.e. a which provides a transportation service to Paris on line RER C.

Passenger Flows. We assume that all transfer-, boarding & alighting-, and network outflows are fair. As upper limits for their magnitudes, we choose

- 8 passengers per second and vehicle for all boarding & alighting flows,
- 6 passengers/second for all transfer flows, and
- 6 passengers/second for the network outflow via the corridor “Leave” in Fig. 5.3.3 above.

These values were defined by experts from Alstom Transport. They are not based on any measurements or publicly available data. The same holds for the specification of the magnitudes’ shapes: Eqn. 3.19 on p. 70 defines the shape of the above network outflow if the function $f(\cdot)$ therein is replaced by $\min(x_1, x_2)$, wherein (i) x_1 equals the number of passengers who want to leave the network from the exit area in Massy–Palaiseau related to the capacity of this exit area, and (ii) x_2 denotes the above upper limit for the network outflow. Similarly, Eqn. 3.25 on p. 73 defines the shape of all transfer and boarding & alighting flows if the function $f(\cdot)$ therein is replaced by $\min(y_1, y_2, y_3)$, wherein a) y_1 equals the number of passengers who want to transfer to a platform or board/alight from a vehicle to the capacity of the place/vehicle where/on-board which these passengers are currently staying, b) y_2 denotes the respective upper limit as defined above, and c) y_3 equals the free capacity of the destination place/vehicle related to its total capacity; with the particularity that no passengers board any stopped vehicle (zero magnitude of corresponding boarding flow) as long as passengers on-board this vehicle still want to alight from it.

Their exact passenger number-dependent shapes are not important for the comprehension of this use case, and we thus skip their specifications here.

Coming to the network inflow into Massy–Palaiseau through its corridor “Enter” next, note that we assume that in average 0.4 passengers per second with trip profile 4 and another 0.4 passengers per second with trip profile 6 arrive at Massy–Palaiseau and want to enter it. Again these values are not based on any statistics. We choose a variance of 1 for the network inflow w.r.t. both trip profiles (due to lack of/access to statistical data also this value is chosen arbitrarily). Moreover we assume that both inflows (passengers w.r.t. trip profile 4, and passengers w.r.t. trip profile 6) are uncorrelated, which latter assumption completely specifies the diffusion terms of all balance equations.

Initial Passenger Numbers. For demonstration purposes, we assume that we have an exact knowledge of all passenger numbers; so that we do not have to provide the possibly complex specifications of all densities or probability mass

functions (we use FETN for all simulations, which discretizes all continuous passenger numbers) here. However, note that one of the strengths of our model is indeed the fact that it can be initialized with uncertain initial passenger numbers. That being said, Tab. 5.7 below, specifies the initial passenger numbers on-board all vehicles for our use case.

Table 5.6: Initial passenger numbers on-board all vehicles at time 8.05am in the format $y : n$, where $y \in \mathcal{Y}$ is some trip profile and n the number of on-board passengers with y

Alight: RER C from Paris	Alight: RER C from Paris	Board: RER C to Paris	Transit RER B	Transit RER C	Entrance Area	Exit Area
1:70	3:20	4:56	1:78	3:57, 4:37	4:55, 1:35	3:30, 4:50

Correspondingly, Tab. 5.7 below specifies the initial passenger numbers at all gathering points in Massy–Palaiseau.

Table 5.7: Initial passenger numbers at all gathering points considered in Massy–Palaiseau in the format $y : n$, where $y \in \mathcal{Y}$ is some trip profile and n the number of on-board passengers with y

Alight: RER C from Paris	Alight: RER C from Paris	Board: RER C to Paris	Transit RER B	Transit RER C	Entrance Area	Exit Area
1:70	3:20	4:56	1:78	3:57, 4:37	4:55, 1:35	3:30, 4:50

Simulation Results

Before we will look at the simulated passenger numbers on-board all vehicles and at all gathering points in Massy–Palaiseau in the given order next, we first provide an overview of all major simulation results.

Simulation Parameters. We have simulated the time evolutions of the passenger numbers on-board all vehicles and at all gathering points in Massy–Palaiseau over the complete forecast horizon [8.05am, 8.25am] for two different scenarios, which we refer to S_{80} and S_{120} in the following. In S_{80} , the same maximum dwell time of 80 seconds is applied to all vehicles at Massy–Palaiseau. In S_{160} , this dwell time is extended to 160 seconds for v_1 , v_4 , and v_{14} ; i.e. for all vehicles which provide a transportation service from Massy–Palaiseau to Paris on line RER C.

We have run all simulations with a fixed time step of 15 seconds each. Hence, the driving- and dwell times of all vehicles evolve in quarters of a minute starting from the initial simulation time 8.05am until the final simulation time 8.25am.

Moreover we have discretized every dimension of every passenger number (real-valued vector, where every component refers to some particular trip profile), into five equally distributed points starting from zero (= no passenger of this trip profile) until the capacity limit of the considered vehicle or gathering point. We have chosen this rough discretization and not some more fine-granular one in order to ensure that all pre-computed transition matrices (computation is described in Sec. 4.4.9 on p. 146) can be loaded into the DDR3 memory of our simulation platform before the actual layer-by-layer forecast computation starts. However, we acknowledge that future experiments should analyze the appropriateness of this rough discretization.

Passenger Numbers On-board Vehicles

Both S_{80} and S_{160} produce the same time evolution of the passenger numbers on-board all vehicles which move towards Massy–Palaiseau in the considered forecast horizon [8.05am, 8.25am] on either line RER B or C. All of them, i.e. vehicles v_3 , v_6 , and v_7 , arrive at Massy–Palaiseau as scheduled. This observation is replicated in Fig 5.3.4 below, which illustrates the minute by minute evolution of the expected numbers of passengers on-board their cabins starting from 8.05am.

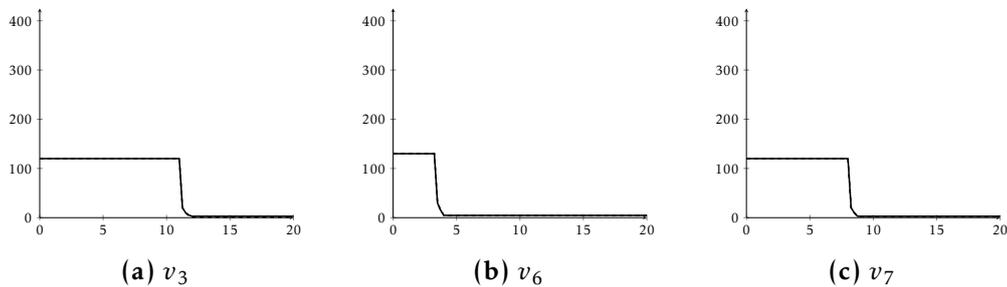


Figure 5.3.4: Minute by minute evolution of expected number of passengers on-board vehicles v_3 , v_6 , and v_7 over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Vehicles v_1 , v_4 , and v_{14} also arrive at Massy–Palaiseau as scheduled, which can be observed in the minute by minute evolution of the expected passenger numbers on-board their cabins in Fig. 5.3.5 below. Note that this predicted punctuality is no surprise at all since a) we have not modelled any other passenger stop and passenger number-dependent departure for these vehicles prior to Massy–Palaiseau, and b) the initial spacing of all vehicles, their constant driving times, and their maximum dwell times at Massy–Palaiseau do not permit any bunching effects. We can moreover observe that the expected passenger numbers on-board all vehicles v_1 , v_4 , and v_{14} increase from S_{80} to S_{160} ; given the fact that in S_{160} the vehicles' dwell- and thus possible boarding times last twice as much as they do in S_{80} .

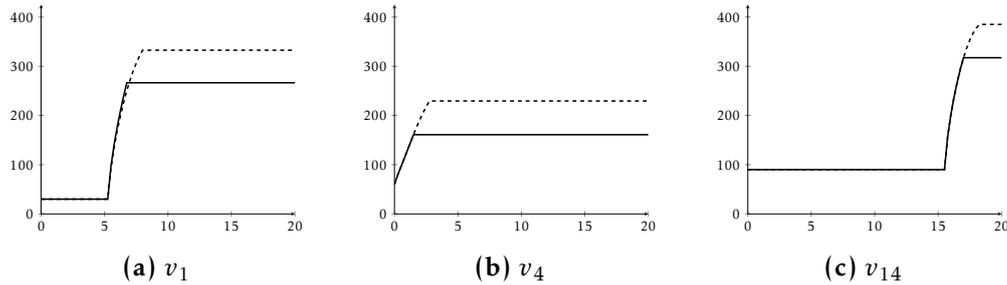


Figure 5.3.5: Minute by minute evolution of expected number of passengers on-board vehicles v_1 , v_4 , and v_{14} over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

The fact that Fig. 5.3.5 above supposes that the expected number of passengers on-board v_4 does not exceed 300 passengers (75 percent of the maximum number of passengers v_4 can accommodate on-board its cabin) for both S_{80} and S_{160} does not imply that v_4 's cabin cannot be overcrowded; in that passengers occupy more than e.g. 75 percent of its available space. This simulation result is illustrated in Fig. 5.3.6 below.

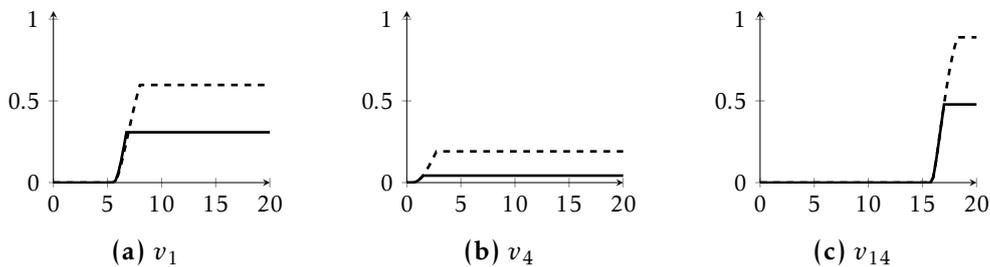


Figure 5.3.6: Minute by minute evolution of probability according to which passengers on-board v_1 , v_4 , and v_{14} occupy more than 75 percent of the physical space available in their cabins over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Passenger Numbers At Gathering Points in Massy–Palaiseau. Figure 5.3.7 below depicts the time evolutions of the expected passenger numbers at all (modelled) platforms in Massy–Palaiseau. It looks like the increased maximum dwell times for v_1 , v_4 , and v_{14} at Massy–Palaiseau has no impact on the passengers who alight at Massy–Palaiseau from a vehicle which provides a transportation service on-line RER B. However, it seems like the increased maximum dwell times are likely to considerably decrease the passenger numbers at both platforms which are connected to line RER C.

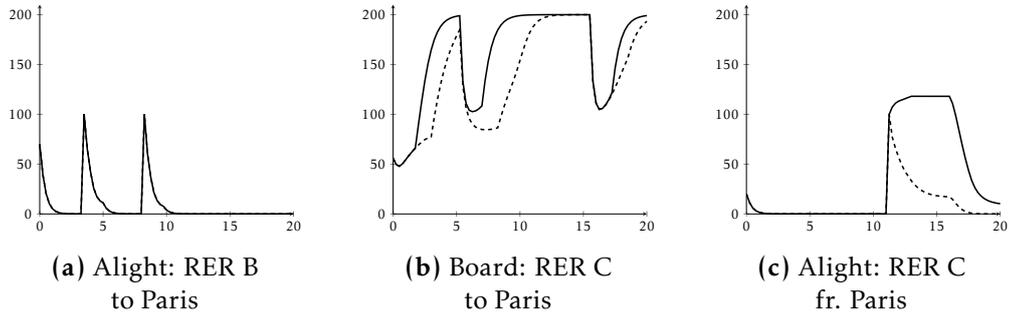


Figure 5.3.7: Minute by minute evolution of expected number of passengers at all modelled platforms in Massy–Palaiseau over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

This improved situation of reduced passenger numbers at the above mentioned platforms for S_{160} can also be observed in Fig. 5.3.8 below. This same figure also indicates that it is almost sure that the platforms which accommodate all passengers who alight at Massy–Palaiseau from a vehicle operated on line RER B or on line RER C, are not overcrowded if a 75 percent occupancy of their maximum available spaces is chosen as reference.

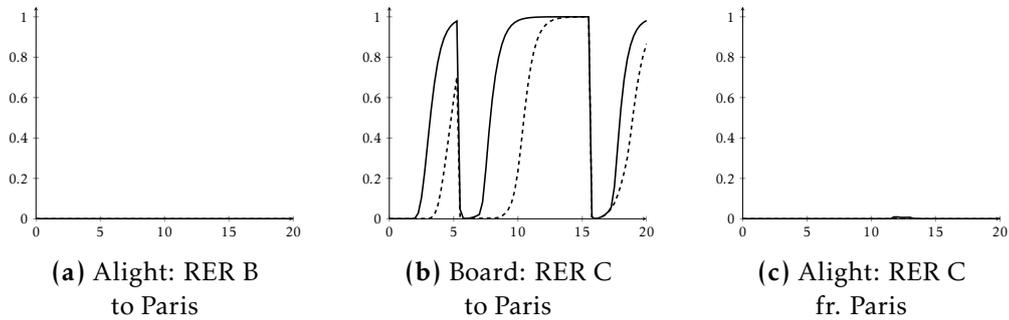


Figure 5.3.8: Minute by minute evolution of probability according to which passengers at some platform in Massy–Palaiseau occupy more than 75 percent of the available space over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Note that the graphs for the expected number of passengers at all platforms in Fig. 5.3.7 above capture some qualitative properties in the time evolution of our transportation network’s hybrid state trajectory which is less visible or invisible at all in Fig. 5.3.8, namely the arrivals and departures of all vehicles; and thus the (expected) mode transition times of our automaton.

Next Fig. 5.3.9 below depicts the time evolution of the expected number of passengers at the transit areas RER B and RER C in Massy–Palaiseau.

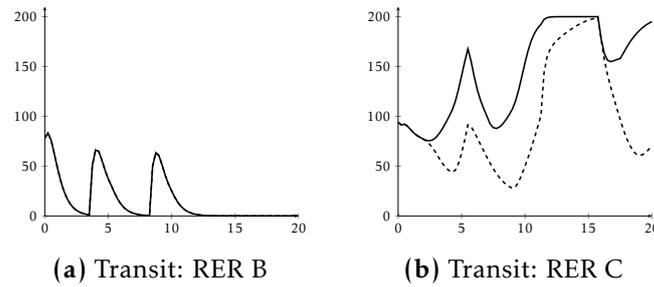


Figure 5.3.9: Minute by minute evolution of expected number of passengers at transit area RER B/RER C in Massy-Palaiseau over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Whereas S_{160} reduces the number of passengers at the transit area RER C compared to S_{80} , the transition from S_{80} to S_{160} has no visible impact on the occupancy of the transit area RER B. This observation is highlighted in Fig. 5.3.10 below.

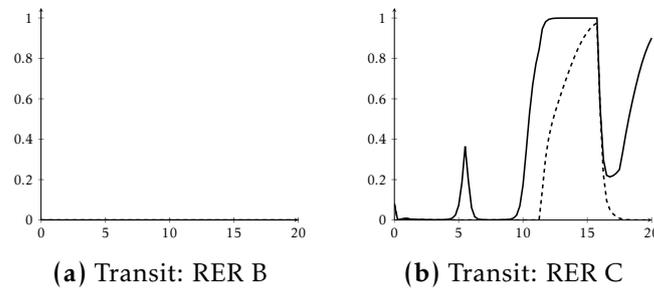


Figure 5.3.10: Minute by minute evolution of probability according to which passengers at transit area RER B/RER C in Massy-Palaiseau occupy more than 75 percent of the available space over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Remember that of primary interest to us is the number of passengers at Massy-Palaiseau's exit area since this location is the common bottleneck of all passenger flows in every safety-analysis. Figure 5.3.12 illustrates that a doubling of the vehicles dwell times at Massay-Palaiseau keeps the expected number of passengers at this exit area well below 300 passengers and thus two thirds of this location's available space.

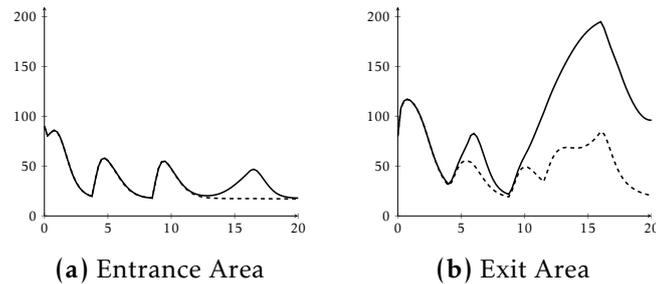


Figure 5.3.11: Minute by minute evolution of expected number of passengers at entrance/exit area in Massy–Palaiseau over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Even more important, Fig. 5.3.12 below indicates that the above increase of the vehicles' dwell times eliminates the critical accumulation of passengers at Massy–Palaiseau's exit area at around simulation 8.20am (= 8.05am plus 15 minutes of simulation time).

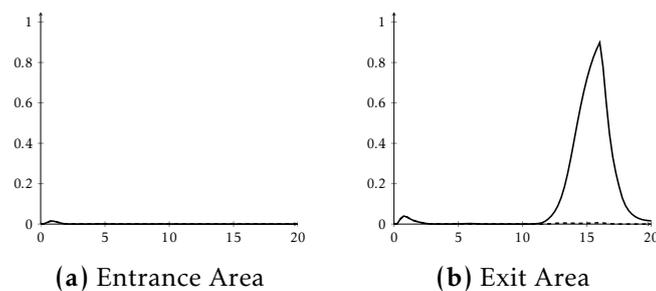


Figure 5.3.12: Minute by minute evolution of probability according to which passengers at entrance/exit area in Massy–Palaiseau occupy more than 75 percent of the available space over complete forecast horizon starting from time 8.05am for S_{80} (continuous line) and S_{160} (dashed line)

Conclusions

The fictitious supervisor in our use case wanted to know whether a relatively small modification (increase of maximum dwell times at Massy–Palaiseau) of line RER C's operation, is likely to improve the safety of all passenger flows in Massy–Palaiseau. The simulation results have shown that this is the case. The same simulation results moreover have shown that the delayed departures of all vehicles operated on line RER C, which provide a transportation service to Paris and in doing so absorb all re-routed passengers from line RER B, are likely to increase the passenger numbers on-board these vehicles. However, this increase of passenger numbers is likely to stay below some acceptable threshold (cf. time evolution of 75 percent cabin occupancies), which means that is very unlikely that the delayed departure times/increased dwell times from/at Massy–Palaiseau cause some additional bunching effects along line RER C.

SUMMARY & OUTLOOK

This chapter concludes our report; cf. the table of contents below: In Sec. 6.1, we will briefly discuss our contributions to the transportation community, the automata community, and the community of computer sciences/applied mathematics in more general. We will then review in Sec. 6.2 all major results of our numerical experiments. Last but not least, we will consolidate all major open problems in Sec. 6.3, where we will also provide an overview of possible future prospects.

Contents

6.1	Contributions	189
6.2	Obtained Results From Numerical Experiments	190
6.3	Open Problems & Future Prospects	191
6.3.1	Parametrization of Our SHA Model	191
6.3.2	Computation of Forecasts	192
6.3.3	Exploitation of Our SHA Model	193
6.3.4	Miscellaneous	194

Contributions

In the beginning of this report we have outlined the prospective benefit and thus the importance of model-predictive supervision to the operation of modern multimodal transportation systems. In fact, the upgrade or completely newly design of these systems with the goal to account for every downgrading operational eventuality is very costly if not impossible. Quite the contrary, the operators of the different modes and lines in such systems (i) must accept the possibility of incident occurrences to some extent, and (ii) are likely to invest their money in soft solutions, such as model-predictive supervision approaches, which assist them in the resolution of these incident occurrences. For this reason, we have developed a new stochastic hybrid automaton (SHA) model, and a discrete time simulation scheme for this model which allows us to compute forecasts.

Our new model can capture the structure and the dynamics of a real transportation system. Compared to existing models, it explicitly accounts for all individual vehicle movements and their interactions with the more fluid-like moving passenger flows in these highly-populated systems. Note that this latter vehicle movement-dynamical property of our model is crucial to the impact analysis of

many actions applied to the operation of the considered network such as the willingly delayed departure of some vehicle from a stop. In addition, neither do we assume that the passengers follow cost-efficient paths, nor do we assume that all passenger arrival flows or initial passenger numbers are known. Compared to classical non Petri net-styled SHA models, our new SHA model moreover has a convenient graphical specification which reduces the risk of man-made specification errors, which may find their ways into the computation of any forecast. More specifically, our Petri net-styled SHA model does not require the end user to set up any differential equations system by hand, because the structure of these equation systems is encoded into our model's graphical specification. Instead, everything that has to be done by the end user, is to specify the parameters of some very general ansatz functions.

The discrete time simulation scheme, which we have developed for our model, pinpoints the changes in the discrete positions, -driving conditions and -operational states of all vehicles to a finite number of time points. These user-specified isolated time points equidistantly cover the complete finite time horizon of the considered forecast. Thus compared to approaches developed for other hybrid automata models, neither does our discrete time approach envisage the computation of some more or less arbitrarily-chosen state trajectories, nor is its application limited to some trivial problems which try to take advantage of their structures given their initial states, so as to compute e.g. their steady state behaviours. However, like all other simulation/computation approaches known to us, also our approach is prone to many bottlenecks. These bottlenecks either result from the high-dimensionality of the differential equations systems which define all passenger flows in our SHA model, or they result from the combinatorial explosion of our model's compound discrete state in the considered forecast horizon. In order to attack these nasty bottlenecks, we thus have developed several algorithmic bricks and incorporated all of them in one common forecast algorithm. Amongst others, one algorithmic brick decouples all passenger flows between every pair of two consecutive simulation time steps, and in doing replaces all high-dimensional differential equations by a set of possibly much lower-dimensional equations systems. It is especially this algorithmic brick which might be of great interest to other domains which also require the numerical integration of high-dimensional differential equations models.

Apart from the specification of our new SHA model and the discrete time computation of forecast which pertain to this model, we have also demonstrated the feasibility and usefulness of our model-predictive supervision in some simple but still realistic use case.

Obtained Results From Numerical Experiments

In Sec. 5.2 on p. 158ff., we have demonstrated the effectiveness and efficiency of two of our algorithmic bricks which target the combinatorial explosion of any com-

puted propagation DAG, in that they repeatedly unfold the ideal time evolution of all vehicle positions and analyse how far passenger flows can spread in stations between their different gathering points. We did so by means of some simplistic two lines-transportation grid, without propagating any passenger numbers along the computed propagation DAG. What was very special about this demonstration, is the fact that both algorithmic bricks do not manipulate the result of any computed forecast. They are thus in stark contrast to other simple approaches, which reduce the computational burden by e.g. proposing less granular discretizations of the simulation time. Moreover note that our two algorithmic bricks do not look *into the past*, which latter property ensures the semi Markov-behaviour of our computation process, and does not render our very general forecast algorithm obsolete. Last but not least, we have shown that (i) both our algorithmic bricks are likely to not negatively affect the computation time of a forecast, meaning that their use is a win win situation; and (ii) similar approaches which do not manipulate the computed forecast should be easy to develop and to implement.

Next in Sec. 5.3 on p. 174ff., we have demonstrated the prospective benefits of our model-predictive supervision approach. More specifically, we looked at an extract of a real transportation network in some degraded state, and managed to anticipate the impact of some small intervention (delay of some vehicle departures) in the operation of this network so as to ensure the smooth and safe flow of all passenger therein. The elaboration of this use case also demonstrated the importance of the graphical specification of our model's infrastructure together with the annotation of the passenger routing. Whereas one single graph quickly explained the complete passenger routing situation in the considered transportation hub, several difficult to read text sentences were needed to do the same job in a less formal manner.

Open Problems & Future Prospects

We have decomposed all major open problems and future prospects of our work into four different topics, namely the parametrization of our new SHA model in Sec. 6.3.1, our discrete time simulation scheme of this model for the computation of forecasts in Sec. 6.3.2, the exploitation of this discrete time simulation scheme beyond the computation of single forecasts in Sec. 6.3.3, and everything else which does not fit very well to the first three topics in Sec. 6.3.4.

Parametrization of Our SHA Model

Our SHA model is a complex model with many many parameters for every non-trivial use case, which have to be specified by its end user. The specification of these many parameters not only takes a lot of time if it is done by hand, but is also a non-negligible source of possible errors which has to be minimized. In this context, it is of utmost importance that some of the future works on our SHA model focus on its automated parametrization. In this context, remember that we have

already discussed the parameter specification of all passenger flows from results computed from the simulation of existing pedestrian flow models. However, the parameters for the passenger flows' ansatz functions represent only a small part of all model parameters. Most other parameters specify the infrastructure of the considered transportation network and the vehicle operation therein. However, note that today many operators publish a lot of data which contains amongst others the specification of all timetables and the platform layouts in the stations in some standardized format. This standardized format allows the end user of our model to develop some software interface which can read in this data, extracts all relevant information, and uses this information for the automated specification of e.g. our model's dispatch plan.

Computation of Forecasts

Our proposed discrete time simulation scheme for the computation of any forecast is prone to many bottlenecks, which either result from the high-dimensionality of the stochastic differential equations systems in the discrete modes of our SHA model or the combinatorial explosion of this mode in the considered forecast horizon. Note that although all bricks which we have introduced in Sec. 4.3 successfully target both types of bottlenecks for relatively small use cases, i.e. use cases where the underlying transportation networks only have very few lines, few trip profiles, and vehicles performing very few passenger number-dependent actions; the same bricks might not be sufficient to analyse more complex use cases. For instance, our canonical decoupling approach produces a replacing set of differential equations systems which defines the passenger flows and their impact on all passenger numbers in the different modes of our SHA model. Every of those replacing systems has at maximum 2γ dimensions, where γ is the overall number of different trip profiles which we consider in the network. Of course, this number is very likely to be big for every more complex use case, which then again renders the numerical integration of all balance equations or more precisely the numerical integration of all corresponding Fokker-Planck equations intractable. As another example which demonstrates the insufficiency of the collection of our bricks when being applied to more complex use cases, note that the repeatedly unfolding of all vehicle trajectories with the goal to ignore some vehicles and thus to reduce the branching of the propagation DAG in the considered forecast horizon, may only take effect too late when the number of nodes in the propagation DAG has already grown beyond any feasible threshold. That is why, future work related to our new SHA model should concentrate on new bricks, which better scale with the complexity of the considered use case. Heuristics which e.g. propose to synchronize several vehicle departures and thus to reduce alternative mode transitions might be one place for the end user of our SHA model to look at. However, most often heuristics are difficult to justify if not employed in very specific situations. Hence, we do not promote them. Instead, the interesting reader of our report should try to develop more mathematically rigorous algorithms first; which may realize one

of the below ideas.

Lump Together Passengers Along Common Last Miles. As it is often the situation in real transportation networks, it might also be the situation in *T-SHA*: the routes of passengers with different trip profiles and probably different origins collide at some point in the considered network, where starting from this point on their future sequences of actions (which mission to board, where to alight, etc.) are identical. In *T-SHA* this means, that two or more trip profiles (= words concatenating actions) might have the same sequence of actions as common endings. Assuming this to be true, we might thus want to lump together all passengers of these trip profiles along their common last mile (of identical actions) so as to reduce the number of different trip profiles which we have to consider in the set up of our stochastic differential balance equations. How do we realize this lumping process in some *system optimal* manner?

Decouple Vehicle Movements in Different Transportation Grids. What we propose here is to develop a computation scheme which temporarily decouples all vehicle movements in one transportation grid, from all vehicle movements in every other transportation grid; with the goal to replace one single huge propagation DAG by some much smaller propagation DAGs; namely one for every different transportation grid. According to our ideas, the execution of this new computation scheme is then very similar to the execution of our canonical decoupling approach of the passenger flows in our SHA model. More specifically, we let the vehicle positions, driving conditions, and vehicle operational states evolve independently in the different transportation grids. However, at fixed time steps we somehow *average* all modifications made to the different passenger numbers at the platforms and at other gathering points in the considered stations, which have been caused by the boarding & alighting processes of/from vehicles with a belonging to different transportation grids.

Exploitation of Our SHA Model

Many studies such as [Coffey 2012] suggest that small modifications to the vehicle operation of a transportation network might be more constructive than the act of confusing passengers with different route propositions; especially in perturbed modes of operation when the nominal passenger seeks stability, safety, and *route continuity*. In this context, small modifications to planned vehicle dispatches, the introduction of some additional time points along existent routes, and similar local actions might produce better and *cheaper* results than abandoning all published timetables. Now the approximate computation of our SHA model's hybrid state for one particular realization of vehicle operational rules is already complex, i.e. time consuming due to the acyclic unfolding of the model's discrete state. Computing forecasts for different configurations of operational rules seems to be impossible if results computed for one parameter realization cannot be adopted

to the model's computation with another parameter realization. In this context, it might be very interesting to develop algorithms which incrementally push the network's anticipated state in the right direction by slightly modifying the operational parameters based on knowledge obtained from previous forecast computations.

Miscellaneous

Here, we could write down a long list of bullet points, where every single bullet point picks up some particular block of our SHA model or brick developed for the discrete time computation of a forecast, and proposes to use this block or brick in some other problem area. However, we do not do so. Instead, we replace this imaginary list by one single proposal. More specifically, we believe that our fluidification approach of all passenger flows could be also applied to any nets-within-nets model if (i) all net tokens therein represent autonomous moving entities in some common networked environment, and (ii) different token types implement different routes between the discrete positions in this environment. Moreover this nets-within-nets model must be highly-populated, in that (iii) many net tokens move around at the same time. Assuming this to be true, one could - very similar to the transition from our original purely discrete multi-agent system model in [Theissing 2016b] to our new SHA model for multi-modal transportation systems - map the routing of all moving entities in the nets-within-nets model to routing matrices; set up stochastic differential balance equations which relate the vectorial number of tokens at every location, to all token flows into and out of this location; and propagate all initial token numbers forward in time through the numerical integration of all balance equations, which latter numerical integration employs our canonical approach for the decoupling of all passengers (= moving entities) flows. Of course, the great benefit of this proposed approach is that the quantitative analysis of this highly-populated nets-within-nets model is not limited any more to the computation of single state trajectories.

Bibliography

- [Abbas-Turki 2002] A. Abbas-Turki, O. Grunder and A. Elmoudni. *Public transportation systems: modeling and analysis based on a new Petri net approach*. In IEEE International Conference on Systems, Man and Cybernetics. Institute of Electrical and Electronics Engineers (IEEE), 2002. (Cited on page 19.)
- [Bednarczyk 2005] Marek A. Bednarczyk, Luca Bernardinello, Wieslaw Pawlowski and Lucia Pomello. *Modelling Mobility with Petri Hypernets*. In Proceedings of the 17th International Conference on Recent Trends in Algebraic Development Techniques, 2005. (Cited on page 20.)
- [Board 2013] Transportation Research Board. Transit capacity and quality of service manual. National Academy of Sciences, 3rd édition, 2013. (Cited on page 9.)
- [Brooks 2011] Steve et al. Brooks. Handbook of markov chain monte carlo (chapman & hall/crc handbooks of modern statistical methods). Chapman and Hall/CRC, 2011. (Cited on page 117.)
- [Burstedde 2001] C. Burstedde, K. Klauck, A. Schadschneider and J. Zittartz. *Simulation of pedestrian dynamics using a two-dimensional cellular automaton*. Physica A: Statistical Mechanics and its Applications, vol. 295, no. 3-4, pages 507–525, 2001. (Cited on page 66.)
- [Carrier 2015] Aurélien Carrier, Alix Munier-Kordon and Witold Klaudel. *Mathematical Model for the Study of Relocation Strategies in One-way Carsharing Systems*. Transportation Research Procedia, vol. 10, pages 374–383, 2015. (Cited on page 4.)
- [Castelain 2002] E. Castelain and K. Mesghouni. *Regulation of a public transport network with consideration of the passenger flow: modeling of the system with high-level Petri nets*. In Systems, Man and Cybernetics, 2002 IEEE International Conference on, volume 6, 2002. (Cited on page 21.)
- [Causon 2010] D. M. Causon and C. G. Mingham. Introductory finite difference methods for pdes. Ventus Publishing ApS, 1 édition, 2010. (Cited on page 117.)

- [Causon 2011] D. M. Causon and C. G. Mingham. Introductory finite volume methods for pdes. Ventus Publishing ApS, 2011. (Cited on page 117.)
- [Coffey 2012] Cathal Coffey, Rahul Nair, Fabio Pinelli, Alexei Pozdnoukhov and Francesco Calabrese. *Missed Connections: Quantifying and Optimizing Multi-modal Interconnectivity in Cities*. In Proceedings of the 5th ACM SIGSPATIAL International Workshop on Computational Transportation Science, 2012. (Cited on page 193.)
- [David 2010] René David and Hassane Alla. Discrete, continuous, and hybrid petri nets. Springer Nature, 2010. (Cited on page 21.)
- [De Schutter 2008] B. De Schutter and T. van den Boom. *Max-plus algebra and max-plus linear discrete event systems: An introduction*. In Proceedings of the 9th International Workshop on Discrete Event Systems (WODES'08), 2008. (Cited on page 22.)
- [Derrible 2012] Sybil Derrible. *Network Centrality of Metro Systems*. PLoS ONE, vol. 7, no. 7, pages 1–10, 07 2012. (Cited on page 10.)
- [Desel 1995] Jörg Desel and Javier Esparza. Free choice petri nets. Cambridge University Press, New York, NY, USA, 1995. (Cited on page 17.)
- [Dib 2015] Omar Dib, Marie-Ange Manier and Alexandre Caminada. *Memetic algorithm for computing shortest paths in multimodal transportation networks*. Transportation Research Procedia, vol. 10, pages 745 – 755, 2015. (Cited on page 4.)
- [Duives 2013] Dorine C. Duives, Winnie Daamen and Serge P. Hoogendoorn. *State-of-the-art crowd motion simulation models*. Transportation Research Part C: Emerging Technologies, vol. 37, pages 193 – 209, 2013. (Cited on page 66.)
- [Febbraro 2004] A. Di Febbraro, D. Giglio and N. Saccording. *Urban traffic control structure based on hybrid Petri nets*. IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 4, pages 224–237, Dec 2004. (Cited on page 21.)
- [Fu 2012] Qian Fu, Ronghui Liu and Stephane Hess. *A Review on Transit Assignment Modelling Approaches to Congested Networks: A New Perspective*. Procedia - Social and Behavioral Sciences, 2012. (Cited on page 14.)
- [Fujiyama 2014] Taku Fujiyama, Roselle Thoreau and Nick Tyler. The effects of the design factors of the train-platform interface on pedestrian flow rates, pages 1163–1173. Springer International Publishing, 2014. (Cited on page 67.)
- [Gardiner 1985] C.W. Gardiner. Handbook of stochastic methods. Springer Publishing Company, 2nd édition, 1985. (Cited on pages 37 and 38.)

- [Giua 2008] A. Giua and C. Seatzu. *Modeling and Supervisory Control of Railway Networks Using Petri Nets*. IEEE Transactions on Automation Science and Engineering, 2008. (Cited on page 19.)
- [Hamdouch 2008] Younes Hamdouch and Siriphong Lawphongpanich. *Schedule-based transit assignment model with travel strategies and capacity constraints*. Transportation Research Part B: Methodological, 2008. (Cited on page 14.)
- [Heidergott 2005] Bernd Heidergott, Geert Jan Olsder and Jacob van der Woude. *Max plus at work: Modeling and analysis of synchronized systems: A course on max-plus algebra and its applications*. Princeton University Press, 1st édition, 2005. (Cited on page 22.)
- [Heye 2003] Corinna Heye and Sabine Timpf. *Factors influencing the physical complexity of routes in public transportation networks*. In Proceedings of the 10th International Conference on Travel Behaviour Research, 2003. (Cited on page 12.)
- [Higham 2001] Desmond J. Higham. *An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations*. SIAM Review, vol. 43, no. 3, pages 525–546, 2001. (Cited on page 117.)
- [Hoogerheide 2002] Lennart Hoogerheide, Johan Kaashoek and Herman van Dijk. *Functional approximations to posterior densities: a neural network approach to efficient sampling*. Technical report EI 2002-48, Erasmus University Rotterdam, December 2002. (Cited on page 117.)
- [Horton 1998] G. Horton, 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. (Cited on page 21.)
- [Hu 2000] Jianghai Hu, John Lygeros and Shankar Sastry. *Towards a Theory of Stochastic Hybrid Systems*. In Hybrid Systems: Computation and Control. Springer Berlin Heidelberg, 2000. (Cited on page 21.)
- [Jiang 2010] Tao Jiang, Ming Li, Bala Ravikumar and Kenneth W. Regan. *Formal Grammars and Languages*. In Mikhail J. Atallah and Marina Blanton, editors, Algorithms and Theory of Computation Handbook. Chapman & Hall/CRC, 2010. (Cited on page 58.)
- [Köhler 2003] Michael Köhler, Daniel Moldt and Heiko Rölke. *Modelling Mobility and Mobile Agents Using Nets Within Nets*. In Proceedings of the 24th International Conference on Applications and Theory of Petri Nets, 2003. (Cited on page 20.)
- [Kummer 2004] Olaf Kummer, Frank Wienberg, Michael Duvigneau, Jörn Schumacher, Michael Köhler, Daniel Moldt, Heiko Rölke and Rüdiger Valk. *An*

- Extensible Editor and Simulation Engine for Petri Nets: Renew*. In Applications and Theory of Petri Nets 2004, 2004. (Cited on page 42.)
- [Lopez 2011] D. F. Lopez, A. M. Triana and H. R. Chamorro. *Simulation model of public transportation system using multiagent approach by means of Petri Nets: Bogotá; study case*. In Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC), 2011. (Cited on page 19.)
- [MacKay 1998] D. J. C. MacKay. *Introduction to Monte Carlo Methods*. In Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models, 1998. (Cited on page 117.)
- [Mahi 2012] Faiza Mahi, Ahmed Nait-Sidi-Moh, Fatima Debbat and Mohamed-Faycal Khelfi. *A multimodal transportation system: Hybrid Petri net-based modeling and simulation*. In 2012 IEEE International Conference on Complex Systems (ICCS), 2012. (Cited on page 21.)
- [Marsan 1994] Marco Ajmone Marsan, G. Balbo, Gianni Conte, S. Donatelli and G. Franceschinis. *Modelling with generalized stochastic petri nets*. John Wiley & Sons, Inc., New York, NY, USA, 1st édition, 1994. (Cited on page 18.)
- [Nait-Sidi-Moh 2002a] A. Nait-Sidi-Moh, M.-A. Manier, A. El-Moudni and H. Manier. *A (max, plus) modelling approach for the evaluation of travelling times in a public transportation system*. In Systems, Man and Cybernetics, 2002 IEEE International Conference on, 2002. (Cited on page 22.)
- [Nait-Sidi-Moh 2002b] A. Nait-Sidi-Moh, M. A. Manier, A. El Moudni and H. Manier. *A (max, plus) modelling approach for the evaluation of travelling times in a public transportation system*. In Systems, Man and Cybernetics, 2002 IEEE International Conference on, 2002. (Cited on page 22.)
- [Okazakia 1993] Shigeyuki Okazakia and Satoshi Matsushitaa. *A Study of Simulation Model for Pedestrian Movement with Evacuation and Queuing*. In Proceedings of the International Conference on Engineering for Crowd Safety, 1993. (Cited on page 67.)
- [Pichler 2013] L. Pichler, A. Masud and L. A. Bergman. Numerical solution of the Fokker–Planck equation by finite difference and finite element methods—a comparative study, pages 69–85. Springer Netherlands, 2013. (Cited on page 117.)
- [Prandini 2007] M. Prandini and Jianghai 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. (Cited on page 21.)

- [Rowe 2012] Ian Rowe and Nick Tyler. *High density boarding and alighting*. In *Impacts on and of People for Successful Rail Operations*, pages 835–843. Informa UK Limited, sep 2012. (Cited on page 74.)
- [Schwaber 2012] Ken Schwaber and Jeff Sutherland. *Software in 30 days*. Wiley-Blackwell, jan 2012. (Cited on page 155.)
- [Silva 2004] Manuel Silva and Laura Recalde. *On fluidification of Petri Nets: from discrete to hybrid and continuous models*. *Annual Reviews in Control*, vol. 28, no. 2, pages 253–266, 2004. (Cited on page 21.)
- [Sosoe 2015] K.S. Sosoe and J-P. Lebacque. *A multiclass vehicular dynamic traffic flow model for main roads and dedicated lanes/roads of multimodal transport network*. In *AIP Conference Proceedings*, 2015. (Cited on page 4.)
- [Still 2000] G. Keith Still. *Crowd Dynamics*. PhD thesis, University of Warwick, 2000. (Cited on page 66.)
- [Szabados 2010] Tamás Szabados. *An elementary introduction to the Wiener process and stochastic integrals*. ArXiv e-prints, 2010. (Cited on page 37.)
- [Teknomo 2002] Kardi Teknomo. *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*. PhD thesis, Tohoku University: Department of Human Social Information Sciences Graduate School of Information Sciences, 2002. (Cited on page 67.)
- [Theissing 2015] Simon Theissing and Stefan Haar. *A Hybrid-Dynamical Model for Passenger-flow in Transportation Systems*. In Bengt Lennartson and Paulo Tabuada, editors, *Proceedings of the 5th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'15)*, volume 48 of *IFAC-PapersOnLine*, pages 236–241, Atlanta, Georgia, USA, October 2015. Elsevier Science Publishers. (Cited on pages 4, 42 and 43.)
- [Theissing 2016a] S. Theissing and S. Haar. *Predicting traffic load in public transportation networks*. In *2016 American Control Conference (ACC)*, pages 821–826, July 2016. (Cited on pages 4 and 43.)
- [Theissing 2016b] Simon Theissing and Stefan Haar. *A Passenger-centric Multi-agent System Model for Multimodal Public Transportation*. Research Report hal-01322956, HAL-inria, May 2016. 12 pages. (Cited on pages 4, 41 and 194.)
- [Theissing 2016c] Simon Theissing and Stefan Haar. *Decoupling passenger flows for improved load prediction*, pages 364–379. Springer International Publishing, 2016. (Cited on page 4.)
- [Theissing 2016d] Simon Theissing and Stefan Haar. *Forecasting Passenger Loads in Transportation Networks*. *Electronic Notes in Theoretical Computer Science*,

- vol. 327, pages 49 – 69, 2016. The 8th International Workshop on Practical Application of Stochastic Modeling, {PASM} 2016. (Cited on page 4.)
- [Trivedi 1993] K. S. Trivedi 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. (Cited on page 21.)
- [Tuffin 2001] Bruno Tuffin, Dong S. Chen and Kishor S. Trivedi. *Comparison of Hybrid Systems and Fluid Stochastic Petri Nets*. Discrete Event Dynamic Systems, vol. 11, no. 1/2, pages 77–95, 2001. (Cited on page 21.)
- [Valk 1998] Rüdiger Valk. *Petri Nets as Token Objects*. In Application and Theory of Petri Nets 1998, 1998. (Cited on pages 20 and 41.)
- [Wolter 2000] Katinka Wolter. *Modelling hybrid systems with fluid stochastic Petri nets*. In Proceedings of the 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems, 2000. (Cited on page 21.)
- [Xu 2010] S. Xu and H. B. L. Duh. *A Simulation of Bonding Effects and Their Impacts on Pedestrian Dynamics*. IEEE Transactions on Intelligent Transportation Systems, vol. 11, pages 153–161, March 2010. (Cited on page 68.)
- [Zhang 2008] Qi Zhang, Baoming Han and Dewei Li. *Modeling and simulation of passenger alighting and boarding movement in Beijing metro stations*. Transportation Research Part C: Emerging Technologies, vol. 16, no. 5, pages 635–649, oct 2008. (Cited on page 66.)
- [Zijal 1994] Reinhard Zijal Robertand German. A new approach to discrete time stochastic petri nets, pages 198–204. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994. (Cited on page 18.)
- [Zubair 2007] H. bin Zubair, C. W. Oosterlee and R. Wienands. *Multigrid for High-Dimensional Elliptic Partial Differential Equations on Non-equidistant Grids*. SIAM J. Sci. Comput., 2007. (Cited on page 117.)

LIST OF FIGURES

2	Simulating Transportation Networks	
2.3.1	Reference Petri net with marking before transition firing	15
2.3.2	Reference Petri net with marking after transition firing	17
2.3.3	A simple nets-within-nets model	20
3	A New SHA Model for Multimodal Transport Networks	
3.1.1	Illustration of mode transitions	27
3.1.2	Illustration of fluid vectorial passenger numbers	31
3.1.3	Illustration of our Petri nets-styled multi-agent system model . . .	41
3.1.4	Illustration of sequential mode transitions in our DHA model . . .	43
3.2.1	Graphical specification of a sample station	45
3.2.2	Graphical specification of a sample transportation grid	46
3.2.3	Connection of stations and transportation grids to complete trans- portation network infrastructures	50
3.3.1	Graphical representation of all transportation services	52
3.3.2	Decomposition of some vehicle's driving time	55
3.4.1	Graphical specification of trip profiles	60
3.4.2	Principle of routing matrices	62
3.7.1	Graphical specification of mode transitions	79
4	Obtaining Forecasts	
4.1.1	Use case diagram of our forecast algorithm	103
4.3.1	Schematic representation of some simplistic two-lines transporta- tion network	116
4.4.1	Comparison: geometry of some physical space and the spatial dis- tribution of all passengers therein in 1) the real network and 2) our computations	127
4.4.2	If the geometries of all gathering points in all stations and of all vehicles' passenger compartments resemble circular areas, and the passengers are spatially equally distributed in these circular areas w.r.t. their different trip profiles, then the number of passengers in any slice taken away from such a circular area is completely de- fined by its relative size.	128
4.4.3	Extract of some station $s \in \mathcal{S}$: Every gathering point has an ideal- ized circular geometry, which is divided into as many non-overlapping slices of equal size each as there are corridors connected to it. . . .	128

4.4.4	Extract of some station $s \in \mathcal{S}$ adopted from Fig. 4.4.3 above: All passengers at some gathering point (on-board some vehicle) are equally distributed among all slices taken away from the circular geometry of this gathering point (vehicle's passenger compartment) whenever <i>T-SHA</i> may change its mode.	129
4.4.5	Extract from some station $s \in \mathcal{S}$: Compared to the situation depicted in Fig. 4.4.4 above, impenetrable walls now separate all slices from neighbouring slices within the idealized circular geometry of the same gathering point (vehicle's passenger compartment). This major difference indicated by solid lines implies that the passenger numbers in the different slices may evolve independently from each other; starting from the same initial distribution.	131
4.4.6	Extract of a station which we use to explain the impact of our canonical decoupling approach in terms of the finite speed for the propagation of passenger number disturbances in all stations: In one discrete simulation time step, a disturbance of p_2 's passenger number can affect the passenger numbers at p_1 and p_3 . However, it cannot affect the passenger number at p_4 , which takes at least two discrete time steps.	140
5	Implementation & Test	
<i>Our Forecast Engine for Transportation Networks:</i>		
5.1.1	A high-level overview of our software module's functional decomposition, with the abbreviations <i>S</i> for start, <i>D</i> for decision, <i>A</i> for abort, and <i>T</i> for successful termination	156
<i>A Quantitative Analysis of the Computation of Propagation DAGs:</i>		
5.2.1	Set up of a simplistic two-lines transportation network for all considered test cases	160
5.2.2	Alignment and orientation of all blocks in the commuter train line (people mover line) w.r.t. the neighbouring stations	160
5.2.3	Interior of station s_1 and its connection to the grid g_1 which accommodates the commuter train line	161
5.2.4	Interior of station s_2 and its connection to the grid g_1 (g_2) which accommodates the commuter train line (people mover line)	162
5.2.5	Interior of station s_3 and its connection to the grid g_2 which accommodates the people mover line	162
5.2.6	Number of nodes in propagation DAG for all simulation runs $R_{1,x}$	166
5.2.7	Number of modes in propagation DAG for all simulation runs $R_{1,x}$	166
5.2.8	Elapsed computation time of propagation DAG for all simulation runs $R_{1,x}$	167

5.2.9	Number of macro modes in different layers of propagation DAG for all simulation runs $R_{1,x}$	168
5.2.10	Number of nodes in propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference	168
5.2.11	Number of modes in propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference	169
5.2.12	Elapsed computation time of propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference	170
5.2.13	Number of macro modes in different layers of propagation DAG for all simulation runs $R_{2,x}$ with simulation run $R_{1,b}$ as reference	170
5.2.14	Number of nodes in propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference	171
5.2.15	Number of modes in propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference	172
5.2.16	Elapsed computation time of propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference	172
5.2.17	Number of macro modes in different layers of propagation DAG for all simulation runs $R_{3,x}$ with simulation run $R_{1,b}$ as reference	173

Ensure Smooth Passenger Transfer Flows At Massy-Palaiseau:

5.3.1	Passenger transfer possibility between the two commuter train lines RER B (dark gray line) and RER C (light gray line) in the south of Paris at Massy-Palaiseau	176
5.3.2	Layout of transportation hub Massy-Palaiseau	177
5.3.3	Layout of transportation hub Massy-Palaiseau together with annotation of passenger routing	180
5.3.4	Time evolution of expected number of passengers on-board vehicles v_3 , v_6 , and v_7	183
5.3.5	Expected number of passengers on-board v_1 , v_4 , and v_{14}	184
5.3.6	Probability that passengers on-board v_1 , v_4 , and v_{14} occupy more than 75 percent of the physical space available in their cabins	184
5.3.7	Time evolution of expected number of passengers at all modelled platforms in Massy-Palaiseau	185
5.3.8	Probability that passengers at a platform in Massy-Palaiseau occupy more than 75 percent of the available space	185
5.3.9	Time evolution of expected number of passengers at transit area RER B/RER C in Massy-Palaiseau	186
5.3.10	Probability that passengers at a transit area in Massy-Palaiseau occupy more than 75 percent of the available space	186
5.3.11	Time evolution of expected number of passengers at entrance/exit area in Massy-Palaiseau	187
5.3.12	Probability that passengers at a entrance/exit area in Massy-Palaiseau occupy more than 75 percent of the available space	187

LIST OF TABLES

1.1	List of our publications	4
2	Simulating Transportation Networks	
2.1	Requirements for supervision model	12
3	A New SHA Model for Multimodal Transport Networks	
3.1	Frequently used sets for the referencing of all vehicles in any mode .	33
3.2	Frequently used shorthands for the positions, driving conditions, runs, and missions of all vehicles in any mode	33
3.3	Different corridor types in stations	49
3.4	Illustration of some vehicle run	53
3.5	Illustration of some dispatch plan	54
3.6	Overview of different passenger flow types	65
3.7	Overview of passenger number-dependent actions	83
3.8	Overview of passenger number-independent actions, which have some related passenger number-dependent action	89
3.9	Overview of passenger number-independent actions, which have no related passenger number-dependent action	90
4	Obtaining Forecasts	
4.1	Specification of all decoupled network inflows, network outflows, and transfer flows	135
4.2	Specification of all decoupled boarding & alighting flows	135
4.3	Specification of the system of balance equations set up for any de- coupled network inflow, network outflow, or transfer flow	136
4.4	Specification of balance equations set up for all decoupled boarding & alighting flows	136
5	Implementation & Test	
	<i>A Quantitative Analysis of the Computation of Propagation DAGs:</i>	
5.1	Overview of all simulation runs	164
5.2	Overview of all possible actions	165
5.3	Overview of all computed parameters	173
	<i>Ensure Smooth Passenger Transfer Flows At Massy-Palaiseau:</i>	
5.4	Initial vehicle operational state of our use case	179
5.5	Scheduled arrival times of all vehicles at Massy-Palaiseau in the considered use case	179
5.6	Initial passenger numbers on-board all vehicles	182

5.7	Initial passenger numbers at all gathering points considered in Massy–Palaiseau	182
6	Summary & Outlook	

Titre : Supervision en Transport Multimodal

Mots clés : Automates stochastiques avec une dynamique hybride, Réseaux de Petri, Réseaux de transport multimodales, Systèmes des équations différentielles et stochastiques

Résumé : Nous introduisons un nouveau modèle stochastique, un automate hybride avec une dynamique probabiliste, et nous montrons comment ce modèle peut prédire le nombre de passagers dans et l'état de fonctionnement du véhicule en question du réseau de transport, ignorons en plus la quantité et la composition des arrivées successives de passagers. Dans notre approche, l'utilisateur final définit l'instant du changement de positions des véhicules, les conditions de conduite et les états opérationnels, et en d'abord par de simples estimations du nombre de tous les passagers et la connaissance exacte de l'état du véhicule au moment de l'incident. Ce nouvel automate réunit sous un même regard les passagers demandeurs de services de transport à parcours fixes ainsi que les véhicules capables de les assurer. Il prend explicitement en compte la

capacité maximale et le fait que les passagers n'empruntent pas nécessairement des chemins efficaces, dont la représentation sous la forme d'une fonction de coût facilement compréhensible devient nécessaire. Chaque passager possède plutôt son propre profil de voyage qui définit un chemin fixe dans l'infrastructure du réseau de transport, et une préférence pour les différents services de transport sur ce chemin. Nous démontrons l'effectivité et l'efficacité de nos éléments constitutifs, ainsi que la faisabilité des calculs de toute notre approche sur la supervision prévoyant le modèle dans des cas type simplistes et dans des cas plus réalistes d'utilisation. Cette dernière occurrence souligne la facilité d'utilisation et les bénéfices prospectifs de notre approche de supervision prospective modèle en vue de systèmes de transport multimodaux futur.

Title : Supervision in Multimodal Transportation Systems

Keywords : Stochastic hybrid automata, Petri nets, Transportation networks, Differential equations systems

Abstract : We introduce a new stochastic hybrid automaton model for multimodal transportation systems. Our new automaton model brings under the same roof, all passengers who demand fixed-route transportation services, and all vehicles which provide them. It explicitly accounts for all capacity-limits and the fact that passengers do not necessarily follow efficient paths which must be mapped to some simple to understand cost function. Instead, every passenger has a trip profile which defines a fixed route in the infrastructure of the transportation system, and a preference for the different transportation services along this route. Moreover, our model does not abstract away from all vehicle movements but explicitly includes them in its

dynamics, which latter property is crucial to the impact analysis of any vehicle movement-related action. In addition our model accounts for uncertainty; resulting from unknown initial Besides introducing our new model, we also develop in this report some algorithmic bricks which target two major bottlenecks which are likely to occur during its forecast-producing simulation, namely the numerical integration of the many high-dimensional systems of stochastic differential equations and the combinatorial explosion of its discrete state. Moreover, we proof the computational feasibility and show the prospective benefits of our approach in form of some simplistic test-and some more realistic use case.

