Stochastic Modelling and Control of Road Traffic Congestion

VRIJE UNIVERSITEIT

Stochastic Modelling and Control of Road Traffic Congestion

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van Doctor
aan de Vrije Universiteit Amsterdam
op gezag van de rector magnificus
prof.dr. V. Subramaniam
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op woensdag 23 januari 2019 om 09.45 uur
in de aula van de universiteit
De Boelelaan 1105

door Daphne van Leeuwen
geboren te Haarlemmermeer

promotoren:	prof.dr. R.D. van der Mei prof.dr. R. Núñez-Queija
copromotor:	prof.dr. S. Bhulai

Contents

Page

1 Introduction 1

1.1 Research context 2

1.2 Traffic flow theory 3

1.3 User behaviour models 6

1.4 Network analysis 9

1.5 Stochastic models 11

1.6 Research objectives 20

1.7 Overview of the dissertation 21

Part I Actuator Control 23

2 Modelling Two-stage Systems with Sequential Processing 25

2.1 Introduction 26

2.2 Single service model 29

2.3 Batch transition model 36

3 Tandem Queue with Fixed Threshold Strategies 41

3.1 Introduction 42

3.2 Model description 42

3.3 Approximation method 44

3.4 Simulation experiments 50

3.5 Conclusion 53

4 Tandem Queue with Dynamic Threshold Strategies 57

4.1 Introduction 58

4.2 Model description 59

4.3 Experimental results 65

4.4 Conclusion 72

Part II User Behaviour 77

5 Modelling User Interaction at a Stochastic Traffic Bottleneck 79

5.1 Introduction 80

5.2 Deterministic bottleneck model 82
5.3 Stochastic bottleneck model 83
5.4 Conclusion 100

6 Modelling of Arrival Time Uncertainty at a Bottleneck 103
6.1 Introduction 104
6.2 Model description 105
6.3 Preliminary analysis 107
6.4 Optimal responses 109
6.5 Conclusion 123

7 Coordinated Scheduling to Enforce Demand Spreading 125
7.1 Introduction 125
7.2 Model description 129
7.3 Model analysis 131
7.4 Experimental setup 139
7.5 Results 141
7.6 Implementation 147
7.7 Conclusion 153

Part III Network Modelling **159**

8 Network Partitioning on Origin-Destination Traces 161
8.1 Introduction 162
8.2 Data analysis 164
8.3 Model description 170
8.4 Preliminary cluster results 175
8.5 Robustness of communities 178
8.6 Consistency of communities 180
8.7 Conclusion 190

Summary 193

Samenvatting 197

Publications of the author 203

References 205

Chapter 1
Introduction

Over the past few decades, road traffic congestion has become a major issue in our modern society. A large part of our daily travel movements is considered to be necessary. Not only do we prefer to move fast from A to B, we also want the trip to be reliable in terms of travel time. In several studies, it has been observed that the *uncertainty* in travel time is considered to be a larger discomfort than the travel time itself [109]. An understanding of the mechanisms that cause traffic congestion is the key to alleviate the impact of congestion. Current technological developments, such as advances in automated driving and developments in sensor technology, allow for better regulation of road traffic in transportation systems. Accurate monitoring of the network state combined with the coordination and cooperation of individuals, paves the way for more reliable and efficient usage of the infrastructure.

The research in this dissertation combines the area of road traffic models with stochastic models in operations research. Specifically, we describe modelling and control techniques of road traffic congestion in which the main focus lies on incorporating the impact of uncertainty by means of quantitative stochastic methods.

This chapter gives an overview of methods and challenges to alleviate road traffic congestion. We describe the current state-of-the-art in modelling and optimisation of transportation systems in relation to the models developed in this dissertation. An introduction to the underlying stochastic models and methods with their applicability to the subject is presented subsequently.

1.1 Research context

All over the world cities are expanding. Currently, more than half of the world's population is living in urban areas. In Europe, around three quarters of the population lives in urban areas, and in Latin America and the United States around 80% of the total population lives in urban areas [42]. According to the United Nations, urbanisation is expected to increase the average city population by 66% by 2050, pushing the existing infrastructure systems beyond its limits [17]. As a by-result of increasing prosperity, the number of individual travel movements increases together with the expansion of cities. Both lead to a tremendous rise in travel demand, causing road congestion to become an even bigger problem. The search for solutions to reduce traffic congestion is a necessity to keep a liveable situation.

Traffic congestion significantly contributes to business economical damages and problems related to energy efficiency and pollution. According to Net!Works European Technology [94], road transportation accounts for 83% of the total energy consumption within the transportation sector and 85% of the total CO_2-emissions.

Congestion in road traffic systems occurs when the travel demand exceeds the available road capacity, e.g., the maximum number of vehicles that can pass a certain stretch of road per time unit. This results in lower driving speeds, longer trip times, and the formation of queues. We distinguish three major approaches for dealing with congestion: *expansion*, *technological development*, and *traffic management*.

For decades, expansion of the infrastructure was a 'good', maintainable solution to tackle the congestion problem. However, this is not a sustainable solution as space and resources become limited. Ongoing developments, such as stagnating population growth, e-business and telecommunication, and technological developments regarding autonomous driving and emerging technologies such as drones, make the future travel demand uncertain. Therefore, it is important to examine temporary solutions that alleviate current congestion problems which are inexpensive and easy to implement within a short-term time horizon.

Technological developments potentially resolve current capacity problems. This provides great opportunities to optimise and exploit the current city assets such as roadside equipment, sensor data and individual GPS traces. Integration of these data sources creates a wealth of opportunities to tackle the complexity of the congestion problem. A drawback of both these developments is that they require time to be integrated into the current transportation system.

Due to the limitations of expansion and the long-time horizon of technological advances we focus on traffic management solutions. The effectiveness of *traffic management* depends on the ability to accurately model travel behaviour. Travel behaviour can be studied from different perspectives. We focus on three main areas:

1. Modelling of *traffic flow dynamics*, i.e., modelling of the interacting behaviour on a stretch of road.
2. Modelling of *user behaviour* decisions regarding timing, modality and location: The mechanisms that drive these decisions, and the interaction due to the decisions of others.
3. Modelling of *network performance* with respect to high-level travel movements across the infrastructure and the infrastructural topology.

An introduction to the research in each of these areas are given in Sections 1.2-1.4, respectively. The dynamics in each of these areas is subject to uncertainty, which is for a large part due to the stochastic nature of individual travel behaviour. We provide an introduction to stochastic models and methods relevant for the research in this dissertation in Section 1.5.

1.2 Traffic flow theory

Traffic flow theory is the study of *interactions* between *individuals* and the *infrastructure*, where an understanding of these movements and interactions is captured in a mathematical framework. Models in transportation research are developed to capture the traffic flow dynamics and to mimic the performance of the transportation network. Traffic has an intrinsic irreproducibility, mainly due to uncertainty in individual behaviour and their interactions, it behaves in a complex, highly nonlinear manner. Fortunately, this behaviour is not entirely random. Individuals tend to have some notion of repetition and regularity, and their driving behaviour is bounded by a reasonably consistent range, allowing flow to be described mathematically.

The development of traffic flow theory and modelling started in the 1930s, pioneered by Greenshields [57], illustrated in Figure 1.1b. To represent the traffic flow mathematically, relationships have been established between its three main characteristics: (1) *flow*, (2) *density*, and (3) *velocity*:

$$q = v \cdot \rho, \tag{1.1}$$

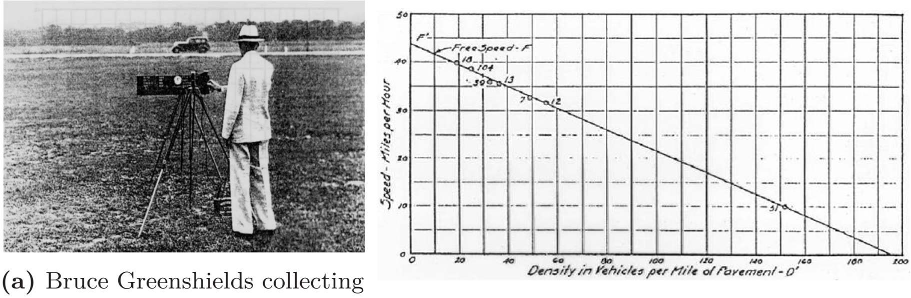
where q is the traffic flow (vehicles/hour), ρ the traffic density (vehicles/km) and v the velocity (km/hour). The values of these three characteristics are bounded by the road capacity. Back in 1933, Greenshields described the relationship among flow and density by a linear model:

$$v = v_0 - q \cdot c, \tag{1.2}$$

where c represents the decay rate in velocity, and v_0 the maximum velocity determined by the road capacity. Both are constants that were determined from field observations as shown in Figure 1.1a. An interesting observation from these experiments is the connection between traffic density and vehicle velocity: The more vehicles there are on a road, the lower their velocity will be. This results in a split into two regimes in the q-v-diagram: a *stable* and an *unstable* regime, meaning that we can have two values for the density ρ for the same traffic flow value q.

The relation between the three characteristics of Equation (1.1) is displayed by the so-called *fundamental diagram.* This fundamental diagram can be presented from three perspectives: (1) flow-density, (2) speed-flow, and (3) speed-density, and is an important subject of study within traffic flow theory, as it can be used for example to predict the impact of inflow regulation or speed limits to improve the network congestion problem.

A major inefficieny at highways occurs when the so-called critical density is reached. The critical density denotes the point at which we reach the maximum flow for the specified road characteristics. Beyond this point, the average traffic speed can decrease tremendously leading to a large decrease of flow, this phenomenon is denoted as the capacity drop. Many studies have been devoted to this specific subject, ranging from adjustments to the shape of the diagram to extensions that include more regimes [72], or even to capture the entire network into the diagram [29, 30]. An extensive survey regarding this subject can be found in [16].



(a) Bruce Greenshields collecting data in 1933

(b) Linear relation between flow and density

Figure 1.1. The first attempts to model traffic flow by Greenshields [57].

A distinction between traffic flow models can be made based on the level of detail. In increasing order of magnitude, we have *microscopic*, *mesoscopic* and *macroscopic* scale traffic flow models. A comprehensive historical overview of these methods is [139].

1. *Microscopic* traffic flow captures the *individual* dynamics of drivers and their interaction with other individuals or the infrastructure. Well-known microscopic models are car-following models [107], and cellular automata models [102, 27]. In general, these models are represented by a set of ordinary differential equations that take into account the acceleration and deceleration dynamics of the individual vehicles.

2. *Mesoscopic* traffic flow models can be classified into three categories: time-headway distribution models, cluster models and gas-kinetic models. These models aim to capture the behaviour of drivers without explicitly distinguishing their time-space behaviour [65].

3. *Macroscopic* traffic flow models are based on system variables in which individual dynamics are captured in an aggregated manner. A common approach is to use methods from fluid dynamics to model the behaviour of particles in a 'fluid'. Often, these models are represented in terms of partial differential equations. The study of macroscopic traffic flow modelling started with the seminal Lighthill-Whitham-Richards (LWR) model, also referred to as the kinematic wave model. This work was developed by Lighthill and Whitham [88, 89], and Richardson [120], independently. They approximate traffic dynamics by a continuous flow of vehicles in terms of vehicle density $\rho(x, t)$ as a function of space x and time t. Many extensions of this model

have been proposed to better capture phenomena such as capacity drop [59] and hysteresis [97]. Payne [115] developed the so-called velocity equation, based on higher order models, to capture the response time of individuals to the surroundings.

It has been shown that, more often than not, the current methods in the area of traffic flow do not accurately capture the actual dynamics. In various cases, these methods lead to highly inaccurate results [90]. This raises the need for the inclusion of stochasticity in traffic models.

1.3 User behaviour models

Another major field of study within transportation research is user behaviour modelling. This area mainly concerns transportation issues and challenges which involve the social and spatial dimensions. In particular, it is the study of the factors that influence activity and travel choices of people and businesses. These studies are used to answer questions related to the transportation infrastructure, mobility choices, social sustainability, among others.

Within this area, the impact of travellers' departure time and route decisions and the effectiveness of information provision remains an important topic of research. This interest is mostly driven by the expectations that the provision of travel information may help reduce congestion. However, modelling the decisions of travellers in a correct manner remains a complex task. Often these models assume that travellers only decide selfishly which route to take, which is not entirely true, moreover, the uncertainty in the network state and user behaviour complicates the analysis of the optimal modelling approach and optimisation strategy even more. In this dissertation, we study the impact of these uncertainty aspects by extending existing models in Chapters 5 and 6. In Chapter 7 we developed a framework to analyse the impact of a central traffic coordinator with limited control in a stochastic environment. A short introduction of the affiliated research areas is given in Section 1.3.1.

1.3.1 Equilibrium models

In the application area of transportation, network equilibrium strategies are commonly modelled to predict the outcome of traffic scenario's subject to congestion. These equilibrium concept include properties of game theory. Therefore, we first give a brief overview of the main game theoretical methodologies within the context of travel choice. From there, we proceed to economic congestion models and traffic assignment methods which make use of game theoretic equilibrium concepts.

Game theory

The field of game theory [101] is a branch within applied mathematics. It is concerned with the analysis of strategies in competitive situations where the outcome of an individual's choice of action depends critically on the actions of other individuals. The theories are based on formal rules and consequences that individuals share. A common assumption in these type of models is that individuals decide rationally and the main goal is to obtain an equilibrium. Various types of game theoretic models can be distinguished, we explain the most common ones applicable to behaviour in road traffic.

Game theoretical models can be split into *cooperative* and *non-cooperative* games. Cooperative means that players negotiate with each other to create a joint strategy. In non-cooperative games, competition among players is modelled, and each individual tries to maximise its own profit according to a specified utility function. A common strategy is to find a Nash equilibrium, implying that no player wants to deviate from his strategy, as it is the best strategy given the strategy of the other players. This concept has been used to model road traffic phenomena, a famous insight, known as the *Braess paradox* shows, the inefficiency of selfish behaviour of individuals [19].

The idea of traffic equilibrium is related to the Nash equilibrium in game theory, however, in transportation networks many players are involved. In this case, obtaining the Nash equilibrium becomes a difficult task. Wardrop [142] developed two different notions of equilibrium, of which the first one is closely related to the Nash equilibrium. The difference lies in the fact that in the Nash equilibrium individual players are considered, while in the Wardrop equilibrium these individuals are modelled as a fluid.

Wardrop's first principle of equilibrium is often referred to as the User Optimum or User Equilibrium (UE), meaning that each user individually optimises his utility for his own benefit. Wardrop's second principle of equilibrium is referred to as the System Optimum (SO), meaning that each individual chooses his route in order to ensure that the overall journey time of everyone is minimised. In general, non-cooperative UE models often lead to inefficient strategies, which are also observed in practice.

Economic congestion models

From an economical perspective, departure time decisions can be expressed in terms of a disutility function. The disutility is usually modelled in terms of a trade-off between travel time and schedule delay (early or late arrival). At the peak, a traveller faces longer travel times, while at the shoulders of the peak hour the traveller faces a schedule delay due to early or late arrival. In such a model each traveller decides when to depart depending on the choices of others. The aim is to find the non-cooperative Wardrop UE strategy. In the literature, two types of modelling approaches can be distinghuished, the Vickrey bottleneck model [138] and the Henderson model [63]. The main differences between the two is that the Vickrey model uses a queueing approach, while Henderson uses the form of flow congestion. More specifically, Henderson assumes that the travel time depends on the number of travellers that depart at a certain time instant, while Vickrey includes the downstream traffic conditions due to earlier departures.

In this dissertation, we build on the concept of the Vickrey model. A fast amount of literature has been developed for both models, which we will not cover in detail. However, it is worth mentioning that Vickrey's model was independently formulated by Hendrickson and Kocur [64], and Fargier [66] with slightly other configurations. Later on, Arnott et al. adapted Vickrey's model [5], which is nowadays the most common version for which extensions of this model are developed. These models lead to valuable insights and understanding of many features of congestion such as the effects of peak-shifting, pricing strategies and capacity expansion. A drawback is that this model is not directly applicable in practice due to its simplifying assumptions.

Traffic assignment

The aim of traffic assignment methods is to determine the impact of possible future scenario's by reproducing the pattern of vehicular movements based on inserted volumes of traffic flow [24]. This allows for evaluation of the current status of the network and to forecast travellers' behaviour under hypothetical scenarios. The effectiveness of specific traffic management strategies or the impact of future traffic conditions can be predicted. For these models the UE concepts are commonly used to determine the decisions of travellers. In this dissertation we use a simplified version of this approach where we analyse the impact of partial control scenario's for only a single stretch of road.

1.4 Network analysis

To assign the social and spatial dimension within transportation modelling, a network-oriented modelling approach is needed. The research concerned with network analysis cannot be described in a few sentences. However, we restrict ourselves to describe the main directions and focus on the methodologies that underlie, or are applicable to, the research in this dissertation.

Network analysis in transportation systems is mainly concerned with the spatial and temporal nature of movements across the infrastructure and the infrastructural topology. Describing the network in terms of nodes and their linkage to each other, measures such as *accessibility* and *connectivity* can be extracted. Including the flow of movements across the infrastructure can be used to analyse the network performance. However, this is not an easy task as this detailed information is often not available.

A major area of research is concerned with the estimation of path flows, route choice decisions, and mode choice. A model incorporating the decisions and estimations into a framework is known as *discrete choice modelling* [11]. Currently, discrete choice models include an elaborate specification of dynamics and other elements. However, social influences are in general not taken into account in such models, which was first mentioned in [39]. In this case, models from social network analysis come at hand. In recent studies, this aspect is considered to be very important [146]. An overview of the current research is given in [98].

Below, we discuss two methods from social network modelling that are of interest within this scope.

1.4.1 Centrality measures

The step towards analysis of a social network of individuals easily aggregates itself to transportation networks. The social network is represented as a graph, where nodes represent people and edges represent the relationships between people. For a road network, a crossing is represented by nodes and the stretches connecting the crossings as edges. A key question in social network analysis is how to determine the most important or central nodes in a network. Although the question is simple, the answer is not.

There are many definitions of what the 'most important' node in a network is, depending on the definition of importance. As mentioned by Freeman [51]: "There is certainly no unanimity on exactly what centrality is or on its conceptual foundations, and there is little agreement on the proper procedure for its measurement' '. Although this statement is still valid, there are a variety of measures that capture specific centrality aspects of a network. The most common measures are *degree*, *betweenness*, *closeness* and *eigenvector centrality* [82]. A survey on centrality measures can be found in [31].

Computing the centrality measures leads to the identification of important locations in the network. A more general direction of network analysis is to separate a graph by identification of groups, or communities.

1.4.2 Community detection

Another popular approach in social network analysis is the discovery of communities within graphs where social interactions are specified. The aim of community detection is to divide the graph into components based on the topological information of the graph only [43]. These communities consist of groups of nodes that have a stronger connection to each other than to members of another community. Community detection has been used to discover geographical areas by means of telephone data [13, 118]. In this research, the geographical area is partitioned into small regions. These regions are translated to a graph where the regions are represented by nodes, and the intensity of phone calls by edges. Community detection algorithms give insight into the geographical connection and separation by grouping the regions into communities.

1.5 Stochastic models

One of the main sources of congestion in road networks is variability in demand. Congestion has a large impact on the network, causing a reduction in the road capacity. Moreover, it has been shown that small perturbations in demand can have a major influence on the network performance. Empirical studies demonstrate that the distribution of travel times is spread over a wide range and has a long tail to the right [91]. In [90], case studies show that the impact of variability in demand and capacity is most significant in cases where the demand is close to capacity.

Although road traffic is highly volatile and for a large part uncertain, traffic modelling and optimisation is typically based on first order performance metrics such as average travel times. A large number of aspects can be distinguished that influence the volatility in traffic flow. For example, day-to-day variability in traffic demand and capacity, uncertainty in route choice and departure time, variations in driving behaviour and the occurrence of incidents, etcetera. In this dissertation, we mainly focus on variability with respect to demand and capacity in traffic flow and the impact of uncertainty in departure time choice. The variability in the arrival and service process has turned out to be essential to the performance of the system [95]. Stochastic modelling is a powerful means to capture the various sources of variability.

In this section, we will give an overview of the main models and techniques that have been used throughout this dissertation with respect to stochastic modelling. Although our main focus is on road traffic congestion modelling, the mathematical techniques used can often be applied in a broader context.

1.5.1 Stochastic processes

A stochastic process is a sequence of random variables, typically representing the state of a system that is changing over time. We can distinguish two types of stochastic processes, continuous-time random processes that may change at any point in time, and discrete-time processes that change at discrete time steps. In mathematical modelling *Markov processes* are commonly used stochastic processes. A discrete-time stochastic process $\{X_1, X_2, \dots,\}$, on a discrete state space $\mathcal{X}$, is a

Markov process if it satisfies the Markov property:

$$\mathbb{P}\left(X_{i+1} = x_{i+1} | X_i = x_i, \ldots, X_1 = x_1\right) = \mathbb{P}(X_{i+1} = x_{i+1} | X_i = x_i)\,,$$

meaning that the probability distribution of the next state of the system, i.e. at time $i + 1$, is only dependent on the history through the current state. The advantage of this assumption is that it makes the system mathematically more tractable and in some cases closed-form solutions can be obtained.

A Markov process that is often used to model random arrivals into a system is a Poisson process. The Poisson process is particularly appropriate when arrivals occur from a large population of individuals. Throughout the dissertation we use the parameter λ to denote the arrival rate, in other words (the average number of arrivals per time unit). The number arrivals Y over a fixed interval of length t has a Poisson distribution:

$$\mathbb{P}(Y = n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \tag{1.3}$$

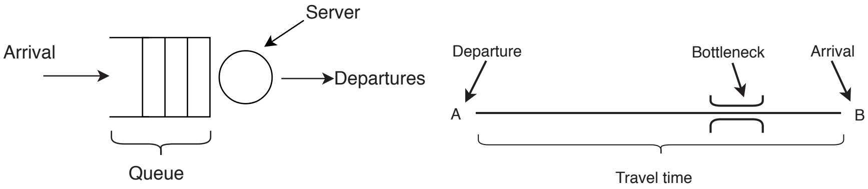
with $n \in \mathbb{N}_0$.

For a general overview of stochastic processes, we refer the reader to [123].

1.5.2 Queueing models

In many applications the variability in the arrival and service process has turned out to be essential to the behaviour of the system. Such systems alternate between periods in which arriving customers are waiting to receive service, and periods where all servers are idle as they are waiting for new customers. To improve, for example, the revenue in such systems, a trade-off between utilisation and queue length has to be made. Results from queueing theory show that in heavily loaded systems, a small increase in arrival rate may quickly saturate such a system, compared to lightly loaded systems. This occurs due to the variability in arrival and service process, and is nicely illustrated by the so-called *waiting-time paradox* [60].

As the variability in driver decision and network capacity are hard to be captured completely, queueing models lend themselves to the subject of traffic flow by modelling the arrival and service rate by probability functions. In this section, we provide an overview of the most common queueing models and the relevant extensions for traffic models. Figure 1.2 shows a representation of a queueing system and a bottleneck in road traffic. A notable difference is the reversed definition of arrival and departure between the two. In the traffic literature, it is common to focus on the origin and destination of a trip, denoting a departure as the start of the trip during which a traveller passes a bottleneck and the arrival when he arrives at the destination. Queueing theorists denote by an arrival the time the traveller arrives at the bottleneck, and a departure the moment a traveller leaves the bottleneck. This can lead to confusion when not well specified. In this dissertation we restrict ourselves to the queueing definition of arrival and departure.



(a) Queueing model

(b) Bottleneck

Figure 1.2. Representation of a queueing model and a road traffic bottleneck.

Characteristics of queueing models

A queueing model is in general characterised by the following four components:

1. **Arrival process**: Specification of the dynamics that determine the occurrence of the arrivals of customers to the system. In many situations, the arrival process is modelled by a *Poisson process*.

2. **Service mechanism**: A quantification of the required service and the service discipline to handle a customer. The required service time is represented by a realisation of a non-negative random variable. Prominent examples of the service discipline are: First-Come-First-Serve, Last-Come-First-Serve and Processor-Sharing. Typically, when the service discipline is not specified, it is assumed to be First-Come-First-Serve.

3. **Service capacity**: Specification of the number of servers available to process customers.

4. **Buffer capacity**: Capacity of the system to queue customers for which no server is available upon arrival.

Kendall introduced a notation to define the most common queueing models in terms of the above-defined characteristics [71]. He used the coding $A/B/c/d$. The letter A specifies the inter-arrival time distribution, B the service time distribution, and c and d represent the number of servers available and the system capacity, respectively. In case that d is not specified it usually means that there is infinite space for waiting.

The most basic model is the $M/M/1$ model, for which both the arrival and service time distribution are assumed to be Markovian (Poisson arrivals and exponential service duration), and there is one server to process individual jobs and the capacity is omitted as it is assumed to be infinite. Other examples are $M/M/\infty$, $M/G/1$, $M/D/1$ and $G/G/1$, where G represents a general distribution, and D stands for deterministic. The number of variations on this simple queueing model is enormous. For example, when we have more servers they could serve at different speeds, the service discipline can be specified regarding the order of service, i.e., First In First Out (FIFO), Last In First Out (LIFO), or Processor Sharing (PS) where each customer receives the same fraction of service. Another example is that arrivals can occur in batches instead of individual customers, or that there are multiple types of customers requiring different amount of service.

For details on the above queueing models and more extensions of this basic notation we refer to Kleinrock [74].

Phase-type distributions

The Markov property is a standard approach in the study of queueing systems, as this property simplifies the modelling tremendously. In transportation systems, the Poisson arrival process is, in general, a valid assumption. However, approximations that rely on the assumption of exponential service durations have been found to overestimate the fluctuations of systems' capacity for real-world applications [148]. For example in road traffic systems, where the time to pass a stretch of road is denoted as the service duration, tends to be less volatile. To adjust for this, the service time can be modelled by using a phase-type distribution. A random variable having a phase-type distribution is composed of multiple phases of exponentially distributed length. In this manner, any service time distribution can be arbitrarily closely approximated by a phase-type distribution.

Stationary performance measures

In general, queueing models are analysed by assuming that the system is in steady state. To compute the stationary distribution of a queue, the system is assumed to be stable, i.e., the offered load remains within the system capacity. The two most popular stationary performance measures are the mean waiting/sojourn time of a customer and the mean number of customers in the system. These measures are typically derived by using two common properties in queueing theory:

1. **Little's law:** States that there is a relation between the mean waiting time of a customer and the average number of customers that enter the system per unit of time [93].
2. **PASTA:** Poisson Arrival See Time Averages states that the expected number of customers in the system seen at arrival moments, equals the time-average number of customers in the systems. This only holds for systems where the arrival process is Poisson [150].

Transient methods

Although the assumption of stationarity is a general modelling approach, the system dynamics in road traffic are time-dependent and behave in a highly non-stationary manner. Moreover, during the peak hours, it often happens that the offered load temporarily exceeds the capacity of the system, resulting in an overload situation. For such systems, it is important to include time in the performance analysis. In the framework of Markov chains, this is captured by the time-varying Poisson process of Equation (1.3) by including time.

In Figure 1.4 an illustration of the queue length development for stationary, non-stationary and a deterministic queueing model is given. This figure is inspired on the lecture notes of [125] and represents the resulting mean queue length for an arrival intensity during a specific time frame. It shows that deterministic models underestimate the queue length at traffic junctions, and that stationary stochastic models are not able to capture a temporal traffic load larger than one. Transient method are able to capture queue length formation during the entire range of traffic loads. We refer to [125] for an explanation of the transient computation for the queue length distribution. Moreover, in research, transient queuing models have been used to model the queue length at intersections [1, 21, 73].

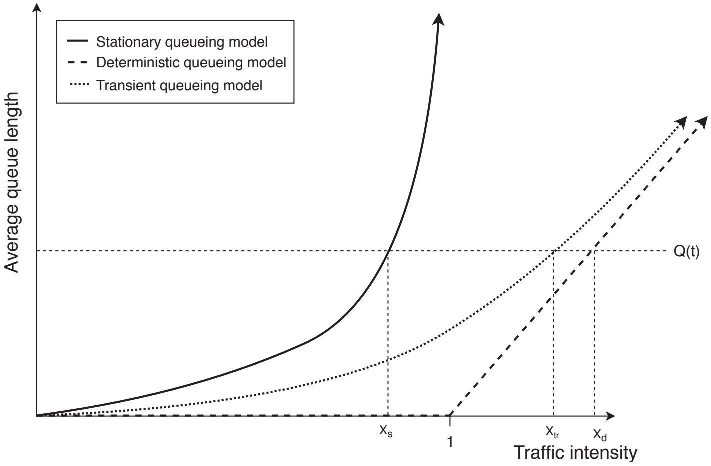


Figure 1.3. Queueing process

Figure 1.4. Queueing behaviour for deterministic, stationary and non-stationary queueing models over a fixed time interval t.

Tandem queues

A natural extension of the basic queueing model to a larger system is a tandem queue. As the name suggests, this model considers a system with two or more queues in series or tandem. The departures of the first queue serve as an input for the second. For tandem queues in which the arrival process is Poisson and the service durations are exponentially distributed, a special property holds. These tandem queues have a product-form solution and are called Jackson networks [68]. The product form remains valid for larger networks as long as the inter-arrival and service times are exponentially distributed, customers are served in a first-come-first-serve manner and after completing their service they move to either another queue or leave the system according to a Markovian or deterministic routing rule. As a final requirement, each queue should have a load smaller than one. An overview of systems for which the product-form solution remains true is [36].

When the product-form solution does not hold, this increases the computational effort tremendously. Computational insights can then be obtained by simulation or approximations. In this dissertation, such approximations are developed to obtain insights into the impact of simplifications of the optimal strategy (Chapter 3), and to allow for extensions of the system in size and realism (Chapter 4).

Application of queueing theory in transportation models

There is a stream of literature that uses models from queueing theory to model traffic flow including its stochastic aspects. In these models, the focus does not lie on the flow of traffic, but on the delay that traffic encounters. A review on the usage of queueing models for highway traffic has been given by van Woensel and Vandaele [147], who validate these models based on empirical data and simulation [149].

Queueing theory has been applied to analyse the behaviour of vehicular traffic flow on a road segment with finite capacity by Jain and Smith [69]. Later on, Osario and Bierlaire [112] presented an analytic queueing network model which preserves the finite capacity of the queues and use structural parameters to capture the between-queue correlation dynamics. Phenomena such as the capacity drop observed at highways are modelled by means of threshold queueing models by Baer and Boucherie [8].

Other well-known work that incorporates queueing theory to road traffic models started by the pioneering work of Webster [144], who approximates the fixed cycle mean queue length at a signalised traffic light by an $M/M/1$ queue. As of today, this approximation is still widely used. Extensions that include the overflow were developed by Newell, Miller and McNeil and compared in [67]. A detailed description of the fixed cycle traffic light queue length distribution is performed by van Leeuwaarden [86]. However, these methods assume stationarity, which is often not reached. Non-stationary queueing models were developed by [1, 21, 73], as mentioned earlier.

1.5.3 Optimisation methods

Optimisation of transportation systems is a popular field of study. The impact on our everyday life and the gain that can be accomplished through optimisation with respect to congestion, air quality, safety, noise and liveability motivates the development of optimisation methods. A definition of optimisation is given in [117]: 'Optimisation is the act of obtaining the best result under given circumstances'. Many optimisation problems result in a large and complex solution space, in these cases, finding the optimal solution becomes infeasible. Therefore, other methods are developed to find a satisfactory solution to the problem. Examples of such methods are asymptotic analysis and heuristics, for which we provide a brief introduction in this section.

Dynamic programming

Dynamic programming is an optimisation technique that finds an optimal solution when the solution space consists of a series of subsequent decisions. The equation to describe the relationship between the subsequent decisions is called the Bellman equation [10]. The solution approach is done in a recursive manner such that the computational complexity is significantly reduced compared to a full evaluation of the complete history. This method has been used in many application areas [12]. The simplest types of dynamic programming problems are deterministic. Examples include Dijkstra's algorithm to find the shortest path [37].

In the context of stochastic processes, a Markov Decision Process (MDP) is an extension of a Markov chain in which actions and rewards are added. In each state, a decision from the action space leads to a reward. Transitions to the next state are determined by means of a probability function. By using dynamic programming we want to obtain the strategy that maximises the specified reward function over all states. This strategy is determined by the best decision for each state in the system taking into account the future impact of this decision. In Chapter 2 such a decision problem is specified. We refer to Puterman [116] for a thorough overview on MDPs.

Asymptotic fluid analysis

Asymptotic analysis is a method to describe the limiting behaviour of a process. As analytical expressions are in most application scenarios not possible, and numerical evaluation become computationally prohibitive, asymptotic analysis is a powerful means to provide valuable insights into the performance characteristics of the system.

A common technique within asymptotic analysis is *fluid scaling*. This means that the stochastic process is rescaled by speeding up the system and decreasing the effect of transitions. The scaling vector is taken to infinity, and it can often be shown that the stochastic process then satisfies the law of large numbers, also referred to as the fluid limit [76]. This approach results in a system where minor fluctuations are eliminated to reveal the long-term evolution of the process. For some applications, this fluid limit might not give sufficient information, and one may be interested in the deviations from this limit. These deviations can be studied by means of diffusion limits [128].

Heuristic optimisation

Heuristic methods can be employed when traditional methods fail to provide a solution within reasonable time. These methods are designed to be fast, but in general fail to find the optimal solution. The word heuristic originates from Greek and can be translated by the verb ‘to find’. As finding a better solution is sufficient in most applications, heuristics are often applied. We can separate heuristic methods into two classes: local search methods and greedy algorithms.

Local search methods start with a candidate solution. By applying local changes, improvements in the current solution can be found in its the neighbourhood. In general, when no local improvements can be obtained, the search algorithm is stuck in a local optimum. In case that the solution space of the problem is convex, local search can provide the optimal solution. In Chapter 7, a local search approach is applied to find a good solution. In many cases, the solution converges to the optimum, however, no guarantee is given.

The approach of a greedy algorithm is to pick the best solution given the current circumstances. The criteria for the best solution of the current circumstances has to be defined. Iteratively, this solution is expanded until the steps leading to a solution have been obtained. In Chapter 4, one of the approximation methods uses a greedy approach to speed up computation to approximate the optimal strategy. In Chapter 8, a greedy algorithm has been used to obtain a ‘good’ partition.

1.6 Research objectives

The *overall goal* of the research in this dissertation is to gain understanding in *the impact of uncertainty on the effectiveness of control mechanisms for road traffic congestion.* The effectiveness of traffic management depends on the interaction between travellers and the settings of roadside systems, amongst others. However, the inclusion of uncertainty in modelling large-scale networks often leads to computational intractability. Therefore, it is crucial to partition the road network into a hierarchical structure of manageable subnetworks to keep a scalable solution. In this context, the overall research question that we aim to answer is:

How to design effective control mechanisms to reduce or prevent congestion on road networks in the omnipresence of uncertainty?

More specifically, we address the following subquestions:

1. How to take into account the uncertainty of traffic demand and road capacity in the deployment of roadside systems?
2. How to take into account the uncertainty in the behaviour and interaction of travellers?
3. How to partition the network to determine the optimal control points to manage traffic?

Accordingly, the thesis is subdivided in the following three parts:

I: Controlling traffic flow by means of roadside systems.

II: Modelling the interaction between users.

III: Partitioning road networks into control areas.

1.7 Overview of the dissertation

This dissertation presents research on the stochastic effects in the area of road traffic. More specifically, the main research focus is on the uncertainty in demand and capacity. In the subsequent chapters, we approach this uncertainty from different perspectives, leading to a division of this thesis into three parts. These parts consist of (local infrastructure) actuator control, user behaviour, and lastly, network analysis.

In part I, the main focus is on modelling and control from the infrastructural perspective by means of local actuators of roadside systems. In the corresponding chapters, we study control strategies that avoid accumulation of traffic at strategic points in a network. In Chapter 2, we introduce two versions of a two-node tandem queue for which we show the interplay between queues when stochasticity of the arrival process and capacity is incorporated. Optimisation of this model can become complex when more realistic aspects are taken into account. Therefore, we introduce two approximation approaches in the subsequent chapters. In Chapter 3, an approximation technique to obtain a fixed threshold level is developed. In Chapter 4, a fluid approximation is proposed to approximate the optimal strategy dynamically. The performance of both approaches is discussed in Chapter 4.

In part II, we turn our attention to rational behaviour of individual users. In practice, travellers can strategically choose their departure times and the routes they take. Congestion occurs when more users simultaneously access the infrastructure than can be sustained by that infrastructure. Understanding the interaction between individual travellers is an important aspect in accurate modelling and effective control of congestion phenomena. The starting point for Chapters 5 and 6 is the seminal bottleneck model of Vickrey [138]. The main goal in that setting is to find compatible departure times of travellers, such that all travellers suffer the same discomfort, which is expressed in a cost function that accounts for three cost components: the cost of being too early at the destination, the cost of arriving too late and the cost of the actual total travelling time; the latter component is determined by the delay due to traffic congestion.

The standard Vickrey model is extended in Chapter 5 with stochastic (uncertain) arrival times and travelling speeds, using a Poisson arrival process with time-fluctuating rate and exponential travel times, aiming at an (approximated) equilibrium in which - again - travellers all experience (approximately) the same cost. In that model, the strategic behaviour of users is captured in the aggregated intensity function of the Poisson arrival process.

In Chapter 6, we use a more detailed model for the rational behaviour of travellers: each can strategically choose a preferred time to join the bottleneck, but the actual time at which the bottleneck is reached is subject to a random shift in time, which captures uncertainty with respect to departure and travelling times prior to joining the bottleneck. We show that the existence of a strategic equilibrium in this setting is questionable, and that, if it exists, it can not be a pure Nash equilibrium, nor can it be a mixed equilibrium with a continuous density.

The strategic scheduling model in Chapter 7 is similar to that of Chapter 5. Similar to the previous two chapters, the goal is to dynamically spread arrivals, but now travel times are optimised in a joint effort between travellers and a central coordinator. The central coordination allows for effective synchronisation of travellers' preferences.

In part III, we analyse the travel behaviour from a network viewpoint. In Chapter 8, a network partitioning algorithm is applied to aggregate travel patterns into high-level partitions of the network. These partitions are composed of historical travel movements in the city of Amsterdam. This study is performed for different time periods, revealing changes over time with respect to high-level network compositions. The results give insights with respect to connectivity and spatial travel patterns, thereby supporting policy makers in their decisions for future infrastructural adjustments.

Part I

Actuator Control

Chapter 2

Modelling Two-stage Systems with Sequential Processing

In this chapter, we introduce a generic model for traffic flow control applications. The stochastic nature of traffic flow in both capacity and demand leads to complex system dynamics. This makes it hard to determine effective control mechanisms to reduce or prevent the impact of congestion. Understanding and quantifying the interplay between queues incorporating the stochasticity of the arrival process and capacity is a starting point for stochastic traffic flow control strategies. In this study, we focus on a control strategy that avoids accumulation of traffic at strategic points in the network.

We introduce two versions of a *Markovian tandem model* for which the service rate of the first queue can be controlled. In the first model, the control of the service rate at the first queue is limited to being turned *on* or *off*. In the second model, the system contains a batch-processing server where the number of jobs to be transferred can be specified at all times. For both models, the objective is to keep the mean number of jobs in the second queue as low as possible, without compromising the total system delay. The balance between these objectives is governed by a linear cost function of the queue lengths. This model can be formulated as a *Markov Decision Process*[1].

[1] This chapter is partly based on [**S1**] and [**S2**].

2.1 Introduction

In this chapter, we investigate a dynamic flow control problem arising in various applications. As a motivating example, consider road traffic control, where trucks enter a crowded metropolitan area to supply goods in the city centre. More often than not, such a scenario leads to clustering of transportation traffic near distribution facilities in the city. Our specific aim here would be to develop a control method that reduces long waiting lines of trucks at distribution centres located in or near cities. As a solution we investigate the effectiveness of a buffer location (e.g., a parking facility for trucks) near a distribution centre to reduce the number of waiting trucks in busy areas, thereby giving more space to other traffic and reducing emissions.

Specifically, we are interested in the effectiveness of such a buffer location when fluctuations in arrival and servicing the trucks play an important role. Indeed, the buffer location will temporarily 'store' trucks and prevent overly crowded areas near the distribution centre. On the downside, the introduction of the buffer location introduces an additional hop in the route for the trucks, creating potential inefficiency. When poorly operated, trucks may be waiting at the buffer location, while the service location at the distribution centre may have cleared all the local backlog.

The problem setting described here illustrates a generic challenge in transportation logistics, manufacturing and production management. For example, one may think of an asphalting machine that must be supplied with liquid asphalt at the correct pace, avoiding too long storage of the perishable material, but also maintaining sufficient supply to avoid an expensive shutdown of the machine due to lack of material. In a production assembly line, one can also imagine the necessity to balance the local inventory of assembly parts with the available space. Similarly, in road congestion management the traffic density near bottleneck junctions must be kept low enough to avoid traffic deadlock, but on the other hand in the upstream direction, the traffic flow should be sufficient to prevent unnecessary delay. We discuss and describe such control problems and design a generic strategy with practical applicability.

To gain an understanding of the impact of uncertainty on the performance of these systems, and to determine the characteristics of an optimal strategy, we simplify the above-sketched situations as a controllable *two-stage tandem queue.* Referring to our initial motivation of the distribution centre, the first stage represents the buffer location where trucks reside and has infinite capacity. The second stage represents the distribution centre at which we want to reduce the number of trucks. We seek an optimal trade-off between the reduction in the number of vehicles at the second stage on the one hand and the additional delay at the first stage on the other hand. We first concentrate on the setting in which the server at the first stage can be controlled by an on-off switch. The 'optimal' operation point is determined by the minimisation of a cost function. This function accounts for waiting time costs in the buffering stage as well as costs for waiting at the distribution centre. Arrivals to this system are modelled by a Poisson process and 'service times' in both queues are exponentially distributed, which facilitates a formulation as a Markov Decision Problem (MDP).

The controllable tandem queue model has been studied extensively in the literature, and it has been shown that it is optimal to serve either at full speed or not to serve at all [122]. This type of strategy is referred to as 'bang-bang strategies'. Under certain cost assumptions, it has been shown that the structure of the optimal service rate gives a switching strategy dividing the state space into two regions: one where the service rate is at its maximum and one where service is paused. Such structural properties of optimality can be proven by showing the convexity of the value function in the Bellman optimality equation, see for example [122] and [75]. Similar convexity properties also hold for networks of queues [143]. Adopting the optimal control, several strategic questions can be answered as well, such as the desired distance of the buffer location from the distribution centre in order to regulate trucks optimally. In our model, this distance is captured by the service rate at the first stage.

Real-time evaluation of the theoretical optimal strategy under changing conditions can become difficult to realise. Moreover, when such strategies are to be analysed for a sequence of bottlenecks for which the second queue of the one system serves as the input for the next. Full characterisation of the theoretically optimal strategy becomes complex when the tandem model is extended to include more realistic factors observed in practice. Examples of these factors include, relaxation of the exponential service distribution, the order of service, and extensions to a network of bottlenecks. Numerical computation of the optimal strategy becomes then numerically overly time-consuming, or even infeasible. Several papers have looked at approximation techniques for this model. The fluid approximation developed in [7] works well when the second station works at a higher rate than the first station. However, for the applications of our interest, the bottleneck station is at the second station, and the question is how to optimally control the outflow of the first station.

In practice, service at the first station may not be limited to one job at a time. Indeed, in our primary example, several trucks may jointly leave the parking facility if the waiting line at the distribution centre is very short. Similarly, in manufacturing and production planning, several items may be produced or delivered at the same time. We, therefore, proceed to study an extension of the tandem queue with controllable service rate, allowing for batch service in the first station. A service batch corresponds, for example, to platoons of trucks jointly driving from the buffer location to the distribution centre. In this setting, it is reasonable to maintain the service rate independent of the number of jobs that are jointly processed. We study the impact of batch service on the first server and determine the structure of the optimal policy. It turns out that the optimal queue level at the second queue is fully determined by the aggregate number of jobs in the two queues (i.e., the sum of the two individual queues). If the optimal levels can be determined, the optimal batch size is then easily computed for all states in the state space.

Several papers have investigated control of similar tandem models, of which we discuss the most relevant. In [137] an inventory control system has been analysed for various control policies in which both the first and the second station can be controlled. However, they assume that the first station represents an inventory level that can be negative, which fundamentally changes the analysis. It is worth noting that with an appropriate translation, their special order-to-stock policy is mathematically equivalent to our single-service model with a fixed threshold at the second queue. In [152], a fuzzy control mechanism is used that computes the decision at each state based on expected reward versus holding costs. This approach is of similar spirit as our fluid-based approximations for which we also use expected costs to approximate the threshold value.

Batch service models with control for single queues have been studied for example in [35, 34]. The optimal batch size is determined by a trade-off between costs of a service initiation, and the waiting time costs of jobs in the system. In the present chapter, we do not consider costs for service initiation, but costs are related to lost capacity. Capacity is lost when the second server becomes idle, while the first queue is not empty. Alternatively, the model in [83] charges costs for abandonment due to impatient customers in a batch service queue. In our model, the adverse effect of being delayed in the first queue is indirectly penalised by the fact that the second queue may idle unnecessarily.

In this chapter, we outline the two versions of the controllable tandem model and their structural properties. Such a model can be seen as a first step towards optimisation of driving times under uncertainty. A next step would be to investigate steps such as sequential optimisation, where the outflow of a single instant of the tandem system is determined by the policy of the upstream tandem system, thereby connecting the impact of each policy in a network setting.

In the remainder of this chapter, we introduce the notation for the on/off server in Section 2.2, denoted as the single service model, and introduce the structural properties leading from this model formulation. In Section 2.3 we introduce the extension to the batch service model, followed by its structural properties. This chapter is a prelude to Chapters 3 and 4 in which a fixed threshold approximation and a dynamic threshold approximation are developed. These approximations are developed to, (1) gain insights into the necessity of an optimal strategy, compared to a near-optimal strategy for specific parameter choices, and (2) allow for extensions to larger or more complex systems that otherwise are computationally demanding or even become infeasible.

2.2 Single service model

2.2.1 Model description

Our model consists of two queues in tandem. As alluded to before, the second queue represents the actual service facility (e.g., distribution centre, production plant or assembly line), whereas the first queue serves as a temporary buffer to alleviate congestion in the second queue. For analysis purposes, we assume that jobs arrive at the first queue according to a Poisson process with rate λ and jobs *can be* processed at rate μ_1.

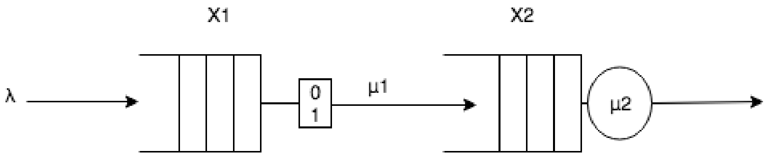


Figure 2.1. *Graphical representation of the tandem queue with an on-off controlling mechanism.*

After service in the first queue, jobs proceed to the second queue, for which the service rate is denoted by μ_2. For stability, we assume that $\lambda < \mu_1$ and $\lambda < \mu_2$.

To control the number of jobs at the second station we introduce a binary decision at the first station, depicted in Figure 2.1. The control mechanism may be interpreted as an on/off switch at the first station with two states: 0 or 1. State 0 represents a service rate of 0, i.e., all jobs waiting at the first station will be blocked for service. State 1 represents the situation where each job at station 1 is served at rate μ_1 and continues to stage 2.

To formulate this as an optimisation problem we introduce constant waiting costs c_{wait}, which are incurred *per job* and *per unit of time.* Jobs queueing at the second station encounter additional costs indicated by c_{loc} which, in our introductory example, represents the costs of residing in the distribution area per unit of time. Thus, the total cost at the first station is $c_1 = c_{wait}$ per job per unit of time, and at the second station it is $c_2 = c_{wait} + c_{loc}$ per unit of time. Naturally, it follows that $0 < c_1 < c_2$. Due to larger costs at station 2, it is more advantageous to hold customers in queue 1 rather than in queue 2. However, one should avoid an empty station 2 when station 1 still has a backlog. We seek an efficient trade-off between these two effects.

We formulate the problem as an MDP. The system will be observed at epochs of arrivals and service completions, i.e., in discrete time. We use uniformisation to discretise the Markov chain as described by Lippman [92]. Our discrete-time MDP consists of the quadruple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}\}$. $\mathcal{S}$ represents the state space of the system, and is defined as $i = (x_1, x_2) \in \mathbb{N}^2$, where x_k is the number of jobs at stations $k = 1$ and $k = 2$, respectively. $\mathcal{A}$ represents the action space, i.e., the set of actions that a decision maker can take. In this problem, $\mathcal{A} = \{0, 1\}$ represents either a *blocked* or an *unblocked* first server, respectively. Action 0 blocks service at station 1, i.e., no jobs can move from station 1 to station 2. For action 1 jobs are served at the first station at rate μ_1 and then move from station 1 to station 2. $\mathcal{P}$ contains the transition probabilities from state i to state j for action $a \in \mathcal{A}$; these can be written as $p^a(i,j)$. Finally, $\mathcal{C}$ denotes the cost function and will be written as $c^a(i)$ which is the expected cost per unit of time for each state $i = (x_1, x_2) \in \mathcal{S}$ and action $a \in \{0, 1\}$.

An optimal strategy satisfies Bellman's equation [9, 116]:

$$V^*(i) + g^* = \min_{a \in \mathcal{A}} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i,j) V^*(j) \Big\} \text{ for } i \in \mathcal{S}, \tag{2.1}$$

where g^* and $V^*(i)$ give the optimal average reward and relative value function. The decision rule can be determined by:

$$f(i) = \arg\min_{a \in \mathcal{A}} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i,j) V^*(j) \Big\} \text{ for } i \in \mathcal{S}, \tag{2.2}$$

where $V^*(j)$ satisfies $V^*(i) + g^* = c^f(i) + \sum_{j \in \mathcal{S}} p^f(i,j) V^*(j)$. Note the slight abuse in notation in writing $c^f(i)$ and $p^f(i,j)$ instead of $c^{f(i)}(i)$ and $p^{f(i)}(i,j)$ as we should have according to our earlier notation. Our goal is to minimise the long-term average cost and determine an optimal decision for each state.

To determine the optimal strategy in our tandem queue we use Equation (2.2), where $c^a(i)$ for $i = (x_1, x_2)$ is given by $c_1 x_1 + c_2 x_2$. Recall that the cost c_1 consists only of the waiting cost per job at station 1 and c_2 is a combination of the waiting costs and additional costs for station 2, and that we take $0 < c_1 < c_2$.

The transition probabilities $p^a(i,j)$ are determined by the transition rates in each state, applying uniformization as described by Lippman [92]. For action $a = 1$ (service in queue 1), we have for $x_1 \geq 0$ and $x_2 \geq 0$: $p^1((x_1, x_2), (x_1+1, x_2)) = \lambda/(\lambda+\mu_1+\mu_2)$, $p^1((x_1+1, x_2), (x_1, x_2+1)) = \mu_1/(\lambda+\mu_1+\mu_2)$, and $p^1((x_1, x_2+1), (x_1, x_2)) = \mu_2/(\lambda+\mu_1+\mu_2)$. On the boundary we have 'dummy transitions' leading to $p^1((0, x_2+1), (0, x_2+1)) = \mu_1/(\lambda+\mu_1+\mu_2)$, $p^1((x_1+1, 0), (x_1+1, 0)) = \mu_2/(\lambda+\mu_1+\mu_2)$, and $p^1((0,0),(0,0)) = (\mu_1+\mu_2)/(\lambda+\mu_1+\mu_2)$.

Similarly, when $a = 0$ (no service in queue 1), we have for $x_1 \geq 0$ and

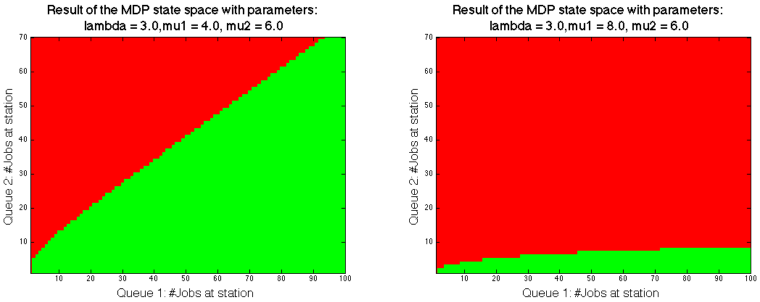


Figure 2.2. *Illustration of the optimal actions for all states. Red indicates blocking (jobs are not served in queue 1) and green indicates that jobs at server 1 are served at rate μ_1.*

$x_2 \geq 0$: $p^0((x_1, x_2), (x_1 + 1, x_2)) = \lambda/(\lambda + \mu_1 + \mu_2)$, and $p^0((x_1, x_2 + 1), (x_1, x_2)) = \mu_2/(\lambda + \mu_1 + \mu_2)$. Now there can be no service in queue 1, giving $p^0((x_1, x_2 + 1), (x_1, x_2 + 1)) = \mu_1/(\lambda + \mu_1 + \mu_2)$. Finally, the remaining transitions on the boundary are $p^0((x_1, 0), (x_1, 0)) = (\mu_1 + \mu_2)/(\lambda + \mu_1 + \mu_2)$.

We use Successive Approximation (SA) to calculate the optimal decision for each state so as to minimize average costs:

$$V_n(i) = \min_{a \in \mathcal{A}} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i,j) V^*_{n-1}(j) \Big\}, \tag{2.3}$$

and

$$f_n(i) = \arg\min_{a \in \mathcal{A}} \Big\{ c^a(i) + \sum_{j \in \mathcal{S}} p^a(i,j) V^*_{n-1}(j) \Big\}. \tag{2.4}$$

2.2.2 Structural properties

We start our discussion of the optimal strategy with a numerical illustration for a particular example. Throughout, we will use $c_1 = 1$ and $c_2 = 3$, meaning that jobs incur waiting costs of 1 per unit of time and, only in queue 2, an additional location cost of 2 units. However, our structural results hold for all values that satisfy $0 < c_1 < c_2$.

First, we illustrate a dichotomy that occurs between the cases $\mu_1 < \mu_2$, i.e., the first server serves jobs at a lower speed than the second server, and the opposite $\mu_1 > \mu_2$. The two graphs in Figure 2.2 show the optimal strategy for the MDP under these two settings. A red colour indicates that it is optimal to block service at the first stage. The green colour implies a system working at maximum service speed at both stages. We observe that in both cases, the optimal action is prescribed by a so-called *switching curve* separating the green area from the red area. The shapes of the switching curves in the two graphs are a bit different. On the left, the curve eventually grows with a constant slope (this will be explained below), whereas the graph on the right flattens for larger values of x_1. This difference appears to be fundamental to the two chosen parameter sets: one where the first server is slower than the second, and the opposite case. We will discuss this in more detail below.

To have a better understanding of the dynamics of the system operating under such a switching curve, we include the drift and trajectory diagrams displayed in Figures 2.3 and 2.4 as explained in [53]. Irrespective of the shape of the switching curve, the drift above the curve is positive in the horizontal direction (due to arrivals at rate λ) and negative in the vertical direction (by departures from the second queue at rate μ_2). Note that because of the stability condition $\lambda < \mu_2$, the horizontal component of the drift is smaller than that in the vertical direction, but that is irrelevant for our discussion here. Below the curve, the horizontal drift changes sign and has magnitude $\mu_1 - \lambda_1$, which is positive due to the stability condition $\lambda < \mu_1$. In the vertical direction, the drift is $\mu_1 - \mu_2$. Here we observe a distinction between the case $\mu_1 < \mu_2$ in Figure 2.3 and the case $\mu_1 > \mu_2$ in Figure 2.4. In the first case ($\mu_1 < \mu_2$), we obtain a negative vertical drift and a corresponding direction toward the horizontal axis. If $\mu_1 > \mu_2$, the vertical drift is positive and the trajectory is directed toward the switching curve from both sides.

We now return to Figure 2.2. The graph on the left suggests a close approximation of the switching curve by a linear function with a positive intercept at the vertical axis. The graph on the right rather suggests an approximation by a horizontal line. The difference in behaviour can be explained by the parameter choice. The linear increasing curve is the effect of a larger service capacity at the second stage, $\mu_1 < \mu_2$. Intuitively, an optimal strategy must aim at avoiding an empty queue 2, when there are jobs in queue 1. The undesirable states are therefore located on the horizontal axis. If $\mu_1 < \mu_2$, the first queue cannot 'catch up' with the second queue, and therefore, it should always provide sufficient inflow for queue 2 even at large system states. To further

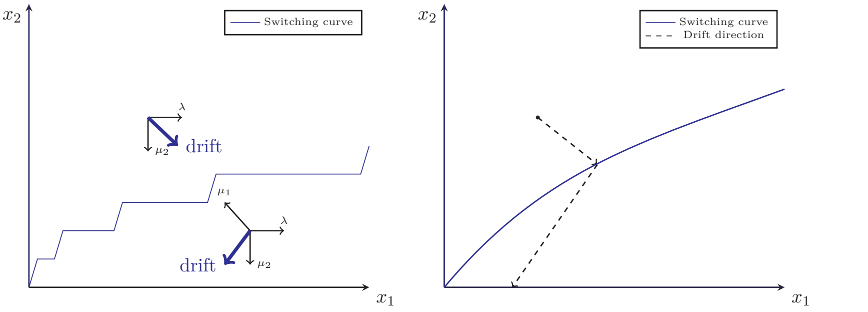


Figure 2.3. *An illustration of the drift above and below the switching curve (left graph) and a typical trajectory (right graph);* $\mu_1 < \mu_2$.

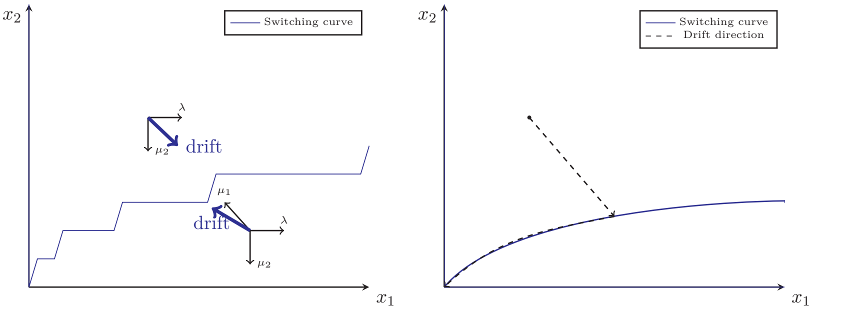


Figure 2.4. *An illustration of the drift above and below the switching curve (left graph) and a typical trajectory (right graph);* $\mu_1 > \mu_2$.

explain this, we refer to Figure 2.3. The typical trajectory leads to the horizontal axis, which is the set of undesirable states. The linearly increasing switching curve avoids that the horizontal axis is hit at a very large level. When $\mu_1 > \mu_2$, the first server *can* catch up with the second server, because it serves at higher speed, thereby decreasing the probability of starvation of the second stage. All trajectories lead to the switching curve and then continue along the switching curve toward the origin. Hence, the size of the second queue can be maintained at a low level, and consequently, the switching curve flattens for larger levels of the first queue. This fundamental difference leads to a likewise fundamentally different analysis of these two cases.

The first case, $\mu_1 < \mu_2$, has been investigated in [7] using a fluid approximation. In this approximation, the random trajectories are replaced by deterministic ones, characterised by their (expected) drifts.

The fluid approximation can be shown to be the exact limit of the stochastic process under an appropriate scaling. The limiting fluid process can be shown to have an optimal linear switching curve, which translates into the optimal action for large system states, but lacks information about the optimal strategy near the origin. As we have observed, at the origin, the optimal switching curve for the stochastic model has a vertical offset. That offset can be approximated using perturbation methods [7], and turns out to give a good representation of the optimal strategy.

Unfortunately, this method does not work when $\mu_1 > \mu_2$, which is the more relevant setting for many of our motivating applications. For example, the 'buffer' location for the distribution centre will likely not be located far from the distribution centre, which corresponds to relatively large values of μ_1. The above fluid approximation applied to $\mu_1 > \mu_2$ gives a switching curve that lies on the horizontal axis which suggests that the first server should never be operated. This is well explained by the sub-linear shape of the switching curve in the right graph of Figure 2.2. On a linear scale, this graph vanishes for large system states.

Therefore, we set out to obtain an approximate analysis for the case $\mu_1 > \mu_2$. Theoretically, it can be seen that the switching curve still increases indefinitely, albeit at a sub-linear pace. The flat shape, however, implies that over large ranges of buffer levels in queue 1, the optimal action switches at a common buffer level of queue 2. This suggests that the optimal switching curve may well be approximated by a horizontal line, i.e., that a fixed threshold-based strategy should be close to optimal. Obtaining the optimal threshold value from the Bellman equations is computationally hard. Therefore, in Chapter 3 we use a matrix-geometric analysis [105] to compute the best threshold value and compare it to the optimal strategy. An alternative method to approximate near-optimal threshold values for the discounted reward MDP was developed in [100, p.439-441]. Unfortunately, when applied to the average reward problem (under the usual limiting argument for discounted reward models [116, Chapter 8.2.2]) the threshold value becomes equal to infinity. For the purpose of this dissertation, our focus lies on an approximation for the average reward case. Alternatively, the authors of [52] approximate the curve using a large-deviations analysis. In that scaling, they obtain an asymptotically optimal switching level.

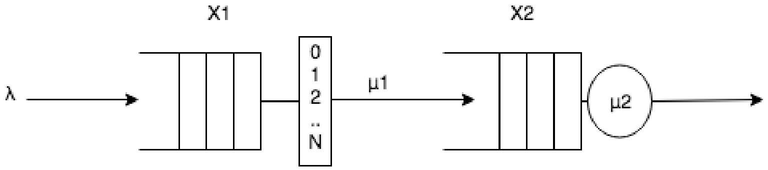


Figure 2.5. *Graphical representation of the tandem queue with batch service in the first queue.*

2.3 Batch transition model

2.3.1 Model description

We now extend the model allowing for batch services in the first queue. To recall that processing multiple jobs at the first server to prevent starvation at the second server is a logical choice for various applications of this model. To clarify in what ways this model extends the previous one, we will describe it while referring to the details of the first model. To gain an understanding of the new model a graphical representation is shown in Figure 2.5. Compared to Figure 2.1 we can see that the first queue is serving N jobs in one single service instead of handling jobs individually.

To allow the first queue to serve in batches we extend the action space from $\{0,1\}$ to $\{0,\dots,x_1\}$ when the number of jobs in the first queue is x_1 (the set of possible actions is thus dependent on the current state). The value of a corresponds to the chosen batch size, which is naturally limited by the number of jobs in the first queue, and the processing rate μ_1 is independent of the batch size. Next, we adapt the transition probabilities described in Section 2.2. For $i=(0,0)$ we have

$$p^a(i,j)=\begin{cases}\frac{\lambda}{\lambda+\mu_1+\mu_2} & \text{if } j=(1,0)\\ \frac{\mu_1+\mu_2}{\lambda+\mu_1+\mu_2} & \text{if } j=(0,0)\end{cases}, \tag{2.5}$$

and for $i=(x_1,x_2)\neq(0,0)$

$$p^a(i,j)=\begin{cases}\frac{\lambda}{\lambda+\mu_1+\mu_2} & \text{if } j=(x_1+1,x_2)\\ \frac{\mu_1}{\lambda+\mu_1+\mu_2} & \text{if } j=(x_1-a,x_2+a) \text{ for } a\in\{0,\dots,x_1\}\\ \frac{\mu_2}{\lambda+\mu_1+\mu_2} & \text{if } j=(x_1,x_2-1) \text{ or } i=j=(x_1,0)\end{cases}. \tag{2.6}$$

For this system, there is always a strategy that is stable as long as $\lambda < \mu_2$, irrespective of the value of $\mu_1 > 0$. This is obvious, since we can always choose to serve all jobs in queue 1 in a single batch, no matter how many there are. Different from the single service model, there will be no clear distinction between the cases $\mu_1 > \mu_2$ and $\mu_1 < \mu_2$, because the first station is always able to 'catch up' with the second station, even for $\mu_1 < \mu_2$. In the batch transition model, we will see that the switching curve typically flattens for larger x_1 values as is illustrated in Figure 2.6. More details on this figure will be given when we investigate the structural properties of the batch service model.

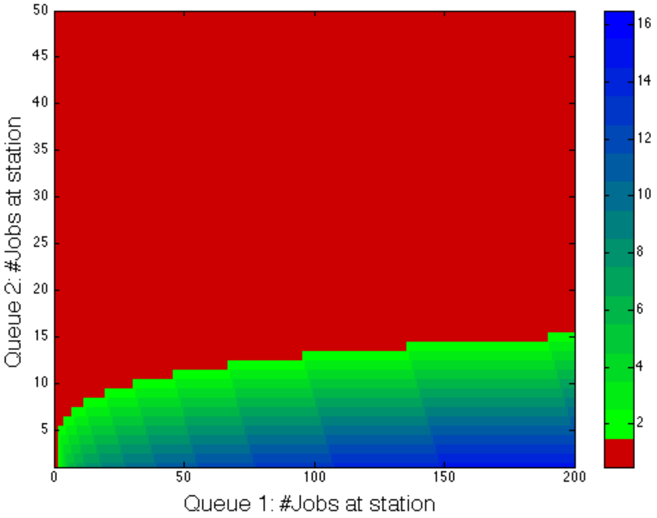


Figure 2.6. *Output of the MDP for the batch service model for parameter set $\lambda = 4$, $\mu_1 = 2$ and $\mu_2 = 6$. The colours in the Figure specify the optimal batch size for each state. In the legend this value is specified, where red is zero and the batch size increases for brighter shades of blue.*

2.3.2 Structural properties of the batch service model

We apply similar numerical calculations to show the main structural properties of the batch service model, and compare these with the single service model. For the single service model, recall that the state space is divided into two regions depending on the optimal decision. For the batch transition model, similar shapes are encountered when grouping

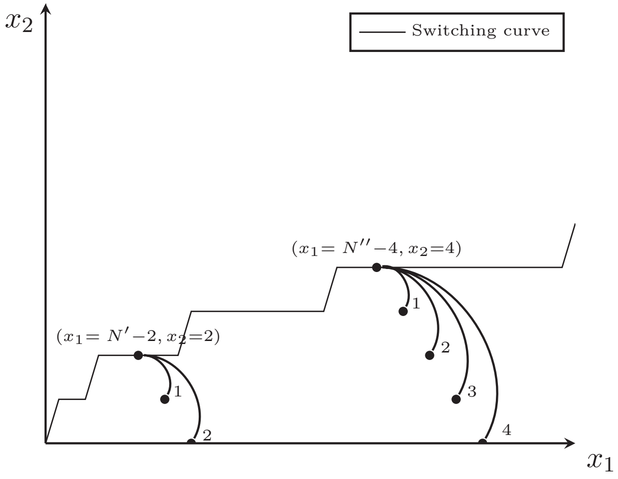


Figure 2.7. *A graphical representation of the optimal batch size for two examples.*

states with the same optimal decision. In Figure 2.6 states coloured in red correspond as before to blocking of service at the first queue. The next layer corresponds to states in which the optimal batch size is one, then we have states with an optimal batch size of two, etcetera. The figure suggests that the optimal trajectories of the process are near the curve dividing red from green coloured states. In this numerically obtained graph, the shape of this curve is again sub-linear. We will indeed show that this is confirmed when investigating a scaled version of the process. Note once more that this shape is not restricted to particular parameter settings, as was the case in the single batch model where the sub-linear shape corresponded to the choice $\mu_1 > \mu_2$. The larger jumps now allow the system to move to the switching curve in one step from any state.

Although this is rather difficult to see from Figure 2.6, we observe that the optimal strategy can again be characterised by the *single* switching curve separating the red states from all others. Given the total number of jobs in the system, say $x_1 + x_2 = N$, the optimal action is to serve a jobs in the first queue such that $(x_1 - a, x_2 + a)$, which also has N jobs in total, is *on* the switching curve. Should this value of a be negative (this happens when (x_1, x_2) is in the red area), then no jobs should be served in the first queue.

A graphical representation of the optimal transitions can be seen in Figure 2.7 for two different values of the total number of jobs in the system: $N = N'$ and $N = N''$. All states on a diagonal $x_1 + x_2 = N$ 'point' to the same destination state at the intersection of this line and the switching curve.

Determining the optimal policy of both the single server and the batch server can become computationally demanding, when these are extended to more complex settings. Examples of such settings are: Adjustments in the cost function, the service distribution is fitted by a phase type distribution, or when the system is extended to a network of tandem queues. The numerical results suggest that the optimal policy shows the potential for approximation methods that either approximate the policy by a threshold level or approximate the structure of the optimal policy. Both options are investigated in Chapters 3 and 4, respectively.

Chapter 3

Tandem Queue with Fixed Threshold Strategies

In this chapter we provide an approximation method for the control models introduced in Chapter 2, to obtain a near-optimal threshold policy. We propose an effective mathematical analysis based on a matrix-geometric solution for calculating the stationary probabilities. This method enables us to compute the relevant stationary measures efficiently and determine an optimal choice for the threshold value. In some of our target applications, it is more realistic to see the first queue as a (controllable) batch-server system. We follow a similar approach as for the first model and specify the computation for the near-optimal threshold policy with batch services.

We find that this method is most appropriate for applications where the system has a low to moderate load and where the policy consists of a fixed threshold strategy[1].

[1] This chapter is based on [**S1**].

3.1 Introduction

In this chapter, we propose an approximation technique for calculating the best threshold value in order to reduce the computational effort of the models described in Chapter 2. This method uses matrix-geometric analysis as described in [105]. Various papers, such as [40], have previously applied this method to speed up computation. The exposition in [84] for a tandem queue similar to ours is particularly relevant to develop our approximation, as it gives an explanation of the blocks which are necessary to capture the details of our model.

In Chapter 2 we introduced two types of tandem queue models. The first is referred to as the *single service model*, which is a well-known model and its structural properties have been studied in [122]. The second type is referred to as the *batch model*. In this chapter, we approximate the batch model in the same manner as the single service model. The extension of the matrix-geometric method for use in the batch model follows along the same lines as [104]. This reference focuses on a system which requires a minimum batch size to initiate service, and additionally, service can be granted up to a predefined maximum batch size. In various other papers optimal batch sizes are determined via a trade-off between startup costs for service and costs per unit time for jobs in the system (see, e.g., [28],[145]). Our model, however, does not have startup costs for batch service. We determine the optimal batch size leading to an optimal threshold level based on properties arising from the MDP formulation.

The remainder of this chapter is structured as follows. In Section 8.3 we set out with a short review of the models explained in Chapter 2. We then turn our attention to determine the best choice for the threshold value using matrix-geometric analysis techniques for both the single service and the batch service mode in Section 3.3. In Section 3.4 we numerically study the appropriateness of the proposed strategies for both models. We conclude this chapter in Section 3.5.

3.2 Model description

In this section, we give a short recap of the two models introduced in Chapter 2, for which we introduce a fixed threshold approach to approximate the optimal policy.

Both models consist of two queues in tandem. Jobs arrive at the first queue according to a Poisson process with rate λ, jobs *can be* processed at the first server at rate μ_1. After this service, jobs proceed to the second queue, for which the job is processed at rate μ_2. In the first model, referred to as the single service model, only one job at a time is served at station 1. For the second model, called the batch model, multiple jobs are served simultaneously without affecting the service duration. For both models, jobs are processed individually at the second station.

Jobs are penalised by inducing costs per unit of time in the system. There are waiting costs at the first station, denoted by $c_1 = c_{wait}$ per job per unit of time, and at the second station we have waiting costs and additional location costs, $c_2 = c_{wait} + c_{loc}$ per unit of time. Naturally, we have $0 < c_1 < c_2$. Due to larger costs at station 2, it is more advantageous to hold customers in queue 1 rather than in queue 2. However, one should avoid an empty station 2 when station 1 still has a backlog. We seek an efficient trade-off between these two effects.

The costs of jobs in the system are minimised, by means of a control mechanism at the first station. This control mechanism encompasses two states: 0 and 1. State 0 represents a service rate of 0, i.e., all jobs waiting at the first station will be blocked for service. State 1 represents the situation where job(s) at station 1 are served at rate μ_1 and continue to station 2. To determine the optimal strategy as to minimise these costs, a formulation in terms of an MDP is introduced. The system will be observed at epochs of arrivals and service completions, i.e., in discrete time.

The discrete-time MDP is formulated by the quadruple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C})$. The first element, $\mathcal{S}$, represents the state space of the system, defined by $i = (x_1, x_2) \in \mathbb{N}^2$, x_k represents the number of jobs at stations $k = 1$ and $k = 2$, respectively. The second element, $\mathcal{A}$, represents the action space, i.e., the set of actions that a decision maker can take. In this problem, $\mathcal{A} = \{0, 1\}$ denotes the action space for the control mechanism. Subsequently, the third element $\mathcal{P}$ contains the transition probabilities from state i to state j for action $a \in \mathcal{A}$; these are written as $p^a(i, j)$. Finally, $\mathcal{C}$ denotes the cost function and will be written as $c^a(i)$ which is the expected cost per unit of time for each state $i = (x_1, x_2) \in \mathcal{S}$ and action $a \in \mathcal{A}$. The action space of the single server model is denoted by $\{0, 1\}$, whereas the action space of the batch model is $\{0, \ldots, x_1\}$. We refer to Chapter 2 for more details on the computation of the optimal strategy and the structural properties of the policy.

3.3 Approximation method

This section describes the approximation methods for the controllable tandem queues for single and batch services, subsequently.

3.3.1 Matrix-geometric method with single service

We have argued that for the case $\mu_1 > \mu_2$ the optimal switching curve can perhaps be well approximated by a horizontal line. In order to compare the effectiveness of such fixed-threshold strategies, we set out to determine the relevant performance measures as functions of the threshold parameter K and then pick the best value of K. In this section we show that the resulting model falls into the class of Quasi-Birth-Death (QBD) processes that allow for a matrix-geometric solution. To cast our model into the framework of [105], we partition the state space into levels and phases, resulting in the generic structure of the generator matrix displayed in Equation (3.1). In our model, each level will correspond to a fixed number of jobs in the first queue, and the phase within a level represents the number of jobs in the second queue. Thus, the generator matrix can be written in the block form of Equation (3.1) below. Transitions between blocks correspond to a change in level (queue 1) and transitions within a block represent a change in phase (queue 2). The number of levels is therefore unbounded and the size of the block matrices (corresponding to the number of phases) is $K + 1$, where K is the fixed threshold level.

Formally, the state space can be described by $\mathcal{S} = \{(x_1, x_2) : x_1 \in \mathbb{Z}^+, 0 \leq x_2 \leq K + 1\}$. The level index x_1 denotes the number of jobs at station 1 and x_2, the phase index, represents the number of jobs at station 2. The maximum number of jobs at station 2 is now bounded by the threshold K.

The generator matrix Q_{single} for this system is given by:

$$Q_{single} = \begin{bmatrix} B_0 & \Lambda & 0 & 0 & 0 & 0 & \cdots \\ M & D & \Lambda & 0 & 0 & 0 & \cdots \\ 0 & M & D & \Lambda & 0 & 0 & \cdots \\ 0 & 0 & M & D & \Lambda & 0 & \cdots \\ 0 & 0 & 0 & M & D & \Lambda & \cdots \\ 0 & 0 & 0 & 0 & M & D & \cdots \\ 0 & 0 & 0 & 0 & 0 & M & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{3.1}$$

In this representation, all blocks are square matrices of order $K+1$, and $M+D+\Lambda$ is a generator of a $K+1$ dimensional Markov process that follows the transitions of the second queue, *conditioned on a non-empty first queue*. The stability condition is given by Neuts' mean drift criterion [105]. We define π to be the equilibrium distribution of a Markov process with generator $M+D+\Lambda$:

$$\underline{\pi}(M+D+\Lambda) = \underline{0}, \text{ where } \underline{\pi}\mathbf{e} = 1, \tag{3.2}$$

where $\mathbf{e}$ is a $K+1$ dimensional vector with all entries equal to 1. The process with generator Q_{single} is stable if and only if $\underline{\pi}M\mathbf{e} > \underline{\pi}\Lambda\mathbf{e}$, i.e., the drift to higher levels should be strictly less than the drift to lower levels to guarantee the stability of the system. For a fixed threshold level K the blocks are defined as follows:

$$B_0 = \begin{bmatrix} -\lambda & \cdots & \cdots & \cdots & 0 \\ \mu_2 & -a_1 & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \mu_2 & -a_1 & \vdots \\ 0 & \cdots & \cdots & \mu_2 & -a_1 \end{bmatrix}, \ \Lambda = \begin{bmatrix} \lambda & \cdots & \cdots & \cdots & 0 \\ \vdots & \lambda & \cdots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \lambda & \vdots \\ 0 & \cdots & \cdots & \cdots & \lambda \end{bmatrix},$$

with $a_1 = \lambda + \mu_2$, and

$$D = \begin{bmatrix} -a_2 & \cdots & \cdots & \cdots & 0 \\ \mu_2 & -a_2 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \mu_2 & -a_2 & \cdots \\ 0 & \cdots & \cdots & \mu_2 & -a_1 \end{bmatrix}, \; M = \begin{bmatrix} 0 & \mu_1 & \cdots & \cdots & 0 \\ \vdots & \cdots & \mu_1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \mu_1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix},$$

with $a_2 = \lambda + \mu_1 + \mu_2$.

The repetitive block structure implies a matrix-geometric form for the stationary probabilities corresponding to Q_{single}. Defining the $K+1$ dimensional vectors $\underline{\pi}_i = (\pi_{i0}, \ldots, \pi_{iK})$, where $\pi_{x_1 x_2}$ is the stationary probability of having x_1 jobs in the first and x_2 jobs in the second queue. The balance equations are defined as

$$\underline{\pi}_{i-1}\Lambda + \underline{\pi}_i D + \underline{\pi}_{i+1} M = \underline{0} \text{ for } i \geq 1, \tag{3.3}$$

and we can write

$$\underline{\pi}_i = \underline{\pi}_{i-1} R \rightarrow \underline{\pi}_i = \underline{\pi}_0 R^i, \tag{3.4}$$

where the matrix R is the minimal non-negative solution to the following quadratic matrix equation:

$$R^2 M + RD + \Lambda = R. \tag{3.5}$$

Via iteration, we may determine R (there are alternative and more efficient routines, see [85]). Once we have determined a solution for R, we can include the boundary conditions. It then remains to compute $\underline{\pi}_0$ via the remaining boundary equation:

$$\underline{\pi}_0 B_0 + \underline{\pi}_1 \Lambda = 0. \tag{3.6}$$

For a unique solution we impose the normalization condition that the probabilities sum to 1. This gives

$$\underline{\pi}_0 \mathbf{e} + \sum_{i=1}^{\infty} \underline{\pi}_0 R^i \mathbf{e} = 1, \text{ or equivalently, } \underline{\pi}_0 (I - R)^{-1} \mathbf{e} = 1. \tag{3.7}$$

In order to compute the cost function, we determine the average queue length for both queues by

$$\begin{aligned}\mathbb{E}[X_1] &= \underline{\pi}_0 R(I-R)^{-2}\mathbf{e}\\ \mathbb{E}[X_2] &= \underline{\pi}_0 (I-R)^{-1} J\end{aligned}\;,$$

where $X_1, X_2 \in \mathbb{N}$ are random variables representing the queue length distribution of station 1 and 2, respectively, and J is the column vector $(0, 1, \ldots, K-1)^T$.

To determine the optimal threshold, we minimise costs over all possible values of K by

$$\min_{K\in\mathbb{N}} \{\underline{\pi}_0 (I-R)^{-1}(c_1 R(I-R)^{-1}\mathbf{e} + c_2 J)\}. \tag{3.8}$$

Now we can compute the best threshold level with respect to the costs and compare it to the MDP policy. From now on, we will refer to this policy as the ‘optimal threshold’ policy, not implying that this policy is the overall optimal, but rather that it is optimal among the threshold policies. Determining the optimal threshold is much less computationally demanding than finding the optimal strategy using the MDP approach.

3.3.2 Matrix-geometric method with batch service

Similar to the case $\mu_1 > \mu_2$ in the single service model, we wish to approximate the (sub-linear) switching strategy with a horizontal line, thereby implementing a threshold-based strategy with, say, threshold value K. This model falls into the class of GI/M/1 type Markov chains that admit a matrix-geometric solution for the stationary distribution. Note that for a fixed threshold value the condition $\lambda < \mu_2$ is not sufficient for stability. For sure, the system cannot be stable if $\lambda \geq K\mu_1$, because $K\mu_1$ is the maximum rate at which jobs can be pushed from the first station. The additional condition $\lambda < K\mu_1$ is necessary, but certainly not sufficient either. The precise stability condition can be shown to be

$$\lambda < \mu_1 \left(K \left(\frac{\mu_2}{\mu_1+\mu_2} \right)^K + \sum_{k=1}^{K-1} k \frac{\mu_1}{\mu_1+\mu_2} \left(\frac{\mu_2}{\mu_1+\mu_2} \right)^k \right). \tag{3.9}$$

This can be obtained by interpreting the right-hand side of this inequality as the exact departure rate from the first station if that station were saturated (i.e., starting with infinitely many jobs in station 1). We do not make this precise here, as this equation can be obtained from Neuts' mean drift condition, which is done below.

To define the batch transition model in matrix-geometric form extra blocks must be added into the generator matrix, that allow for the larger transition jumps. Recall from the structural properties of the batch model discussed in Chapter 2 that the batch size can be derived from the switching curve, effectively redistributing the total number of jobs over the two queues (with the obvious limitation that no jobs can be moved from the second to the first queue).

The generator matrix Q_{batch} now has the following structure:

$$Q_{batch} = \begin{bmatrix} B_0 & \Lambda & 0 & 0 & 0 & \cdots & \cdots \\ B_1 & D & \Lambda & 0 & 0 & \cdots & \cdots \\ B_2 & M_1 & D & \Lambda & 0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \cdots \\ B_{K-1} & M_{K-2} & M_{K-3} & \cdots & \cdots & \ddots & \cdots \\ B_K & M_{K-1} & M_{K-2} & \cdots & \cdots & \ddots & \cdots \\ 0 & M_K & M_{K-1} & \ddots & \ddots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \end{bmatrix}. \tag{3.10}$$

The 0^{th} level of the process represents the boundary states, comparable to the single service model. The matrices B_0, Λ and D remain equal to the ones in Equations (3.3) and (3.6). The block matrices below the diagonal must be adapted to account for the batch services. The matrices B_k, for $k = 1, 2, \ldots, K$, correspond to transitions for which the first queue is emptied. This is only possible when there are 1 up to K jobs in the first queue, and the second queue has sufficient space left to accommodate the batch size

$$B_1 = M = \begin{bmatrix} 0 & \mu_1 & 0 & \cdots & 0 \\ 0 & 0 & \mu_1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 0 & 0 & \mu_1 & \cdots & 0 \\ 0 & 0 & 0 & \mu_1 & \vdots \\ \vdots & \ddots & \ddots & 0 & \mu_1 \\ \vdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 & \cdots & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} = B_K.$$

The transitions corresponding to batch services that do not lead to an empty first station are grouped in the matrices M_k, for $k = 1, 2, \ldots, K$

$$M_K = B_K,$$

$$M_{K-1} = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mu_1 \\ 0 & \cdots & \cdots & 0 \end{bmatrix} = M_1.$$

From Neuts' mean drift criterion [105] we can obtain the stability criterion in (3.9). Similar to the single service model we now define $\underline{\pi}$ to be the equilibrium distribution of a Markov process with generator $\Lambda + D + \sum_{k=1}^{K} M_k$:

$$\underline{\pi}(\Lambda + D + \sum_{k=1}^{K} M_k) = \underline{0}, \text{ where } \underline{\pi}\mathbf{e} = \underline{1}. \tag{3.11}$$

The process with generator Q_{batch} is stable if and only if the drift condition $\underline{\pi} \sum_{k=1}^{K} M_k \mathbf{e} > \underline{\pi} \Lambda \mathbf{e}$ is satisfied.

Again, following [105], the stationary distribution has a matrix-geometric structure $\underline{\pi}_i = \underline{\pi}_0 R^i$, for $i = 1, 2, \ldots$, where the matrix R is the minimal non-negative solution of

$$\Lambda + RD + \sum_{k=1}^{K} R^{k+1} \cdot M_k = 0.$$

The boundary equations now read

$$\underline{\pi}_0 \sum_{k=0}^{K} R^k \cdot B_k = \underline{0},$$

and the normalization condition is

$$\underline{\pi}_0 \sum_{k=0}^{\infty} R^k \cdot \mathbf{e} = \underline{\pi}_0 \cdot (I - R)^{-1} \cdot \mathbf{e} = 1.$$

Again, by computing the stationary distributions for varying values of the threshold K, we may determine the best value of the threshold in terms of the average cost as we did for the single service model using (3.8). Finally, we can compare this result with the optimal MDP policy.

3.4 Simulation experiments

In this section, we illustrate the effectiveness of the threshold policies, obtained using the matrix-geometric representation, with the optimal strategies from the MDP formulation. For our comparison, we will compute both classes of strategies, although for the threshold strategies the reported results can also be directly obtained after computing the stationary distribution.

In Figure 3.1 the costs and the average queue lengths are plotted for varying service rate μ_1 at station 1. We observe that the performance of the best threshold policy is almost identical to that of the optimal MDP policy. The right-hand graph also shows that the two policies are very close to each other in terms of the average queue lengths. The discontinuities in the curves corresponding to the threshold policies correspond to parameter choices where the optimal threshold value shifts by one. As can be expected, the discontinuities for the MDP policies are much less pronounced, as the optimal switching curve may shift only for a small number of states.

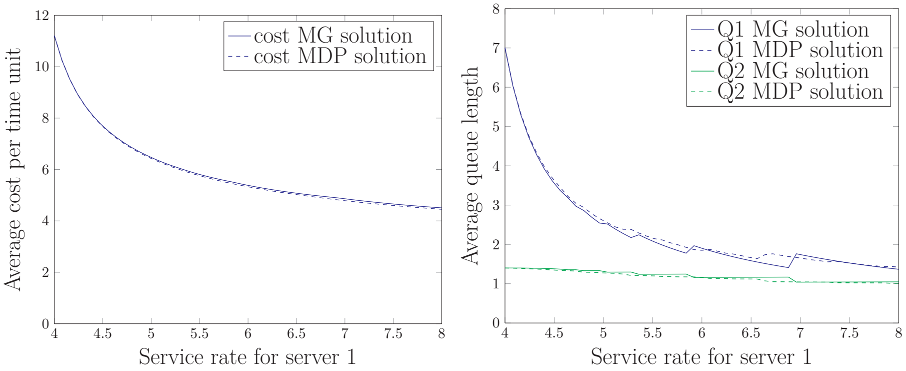


Figure 3.1. *Comparison of the optimal MDP policy model and the threshold-based MG approximation for the single service model for varying service rate μ_1. The parameters are $\lambda = 3, \mu_1 \in [4, 8]$ and $\mu_2 = 6$.*

For the batch service model, the simulation results of the MDP and the threshold strategies are reported in Figure 3.2. We may now take μ_1 to be smaller than λ without compromising the stability of the optimal policy as long as condition (3.9) is satisfied.

We again observe a remarkable fit in terms of costs, for almost all values of the service rate at the first station. As could be expected, the costs are lower compared to the single service model. As for the queue lengths, we observe that the batch service mode allows to keep the first station at lower levels, but the queue lengths at the second station remain at roughly the same level. For now, we defer further comparisons between the models with and without batch services and focus on comparisons between the MDP and the threshold strategies. Due to the jumps in service mode, the costs of the threshold policies are much less smooth than in Figure 3.1 and the optimal mean queue lengths oscillate more for larger values of μ_1. Indeed, changing the threshold value by one has a much larger impact on the resulting policy (that aims to bring the queue length back to the horizontal switching curve in a single service run). The strong fluctuations in queue length make it all the more surprising that the costs of the best threshold policy remain close to those of the optimal MDP strategy.

We have now compared the rightfulness of the approximating threshold strategies. Next, we compare the gain of having batch service in the first station. In Figure 3.3 the best threshold values are determined, again for

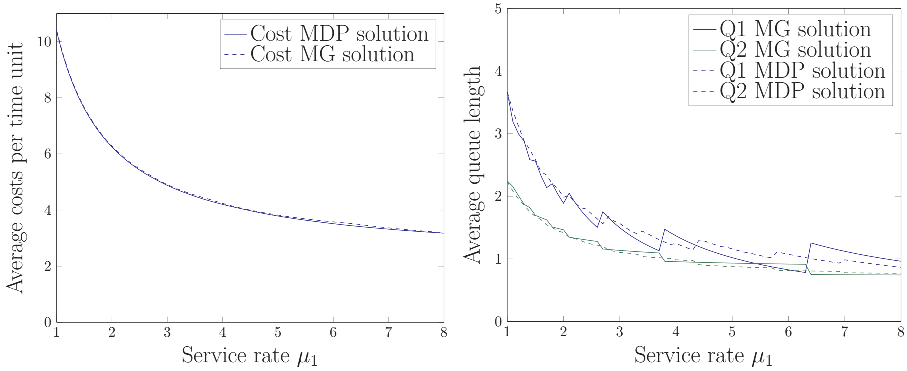


Figure 3.2. *Comparison of the optimal MDP strategy and the threshold-based approximation for the batch service model for varying service rate μ_1. The parameters are $\lambda = 3, \mu_1 \in [1, 8]$ and $\mu_2 = 6$.*

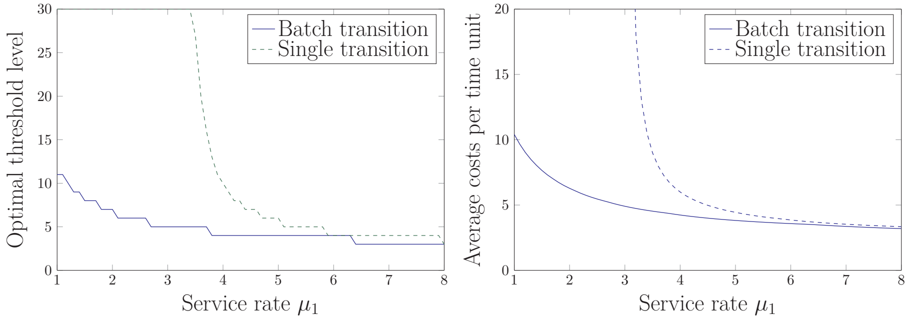


Figure 3.3. *Comparison of optimal threshold strategies for the single service and batch service model for varying service rate μ_1. The parameters are $\lambda = 3, \mu_1 \in [1, 8]$ and $\mu_2 = 6$.*

increasing service rate μ_1 at station 1. The single service model is not stable for $\lambda \geq \mu_1$. We observe that the threshold strategies only perform badly in the single service model when the system approaches instability. For a large range of values with $\mu_1 < \mu_2$, the single-service threshold strategy performs almost as well as the batch-service threshold strategy, although in that case the optimal switching curve for the single-service model has a rather steep (linear) ascent. It is quite surprising that the costs are comparable for the two models, as long as μ_1 does not approach the stability limit (i.e., remains $> \lambda$). The optimal threshold levels do, naturally, differ: that of the single service model is considerably larger than in the batch service model (as could be expected).

K	Single service	Batch transition	Single service	Batch transition
	$\lambda = 4, \mu_1 = 5, \mu_2 = 6$		$\lambda = 4, \mu_1 = 7, \mu_2 = 6$	
3	-	18.39	13.04	7.34
4	50.17	7.16	7.20	6.00
5	14.01	6.87	6.75	6.19
6	11.23	7.06	6.78	6.47
7	10.42	7.29	6.90	6.70
8	10.12	7.47	7.00	6.87
9	10.00	7.61	7.10	6.99
10	9.96	7.71	7.17	7.07
11	9.95	7.78	7.22	7.12
12	9.96	7.83	7.25	7.16

Table 3.1. *Comparison of the costs of single and batch services for various threshold levels K.*

Table 3.1 shows the costs for the two models for various threshold levels. The results show that the batch transition model performs better for all threshold levels, also non-optimal levels, but the difference is not very pronounced. The main advantage of the batch-service model is that the costs are not that sensitive to the exact value of the threshold. For the single-service model, the costs are much more sensitive and small errors in the threshold value may lead to considerable loss of efficiency.

3.5 Conclusion

We have investigated the optimal control of the number of jobs in an expensive service station, by regulating the flow from a preceding buffer station. We started by determining the optimal control policy using an MDP formulation. The optimal strategy can, in general, be characterised by a switching curve. The shape of this curve is determined by whether or not the first station has a larger service rate than the second. If so, the optimal switching curve is rather flat, otherwise, it increases approximately linearly. When the optimal switching curve is rather flat, it can well be approximated by a horizontal one, which corresponds to a fixed threshold strategy. Besides their practical relevance due to the simplicity of implementation, threshold-based strategies have the advantage that they allow a much more detailed analysis. By casting a threshold based control into the framework of Markov models with a matrix-geometric stationary distribution, we can efficiently compute the best threshold level. For this ‘optimal’ threshold level, we indeed verified that it performs very closely to the optimal MDP strategy under medium loaded systems. However, when the load of the system increases, the performance gap between the MDP strategy and the approximation increases. Under heavily loaded systems the structure of the optimal policy becomes more important. We address this issue in the next chapter.

For some of our motivating examples, the ‘feeding’ process from the buffer station need not necessarily be done by individual jobs only. It is quite natural to allow multiple jobs to be served in a single service run from the first station. For this model, we again formulated and studied the corresponding MDP and established that the optimal switching curve always has a flat shape, irrespective of the speed of the servers. Again, threshold-based strategies were shown to be much more efficiently solvable and have close to optimal performance.

Surprisingly, the best single-service threshold and the best batch-service threshold policies were found to give comparable performance, unless the single-service threshold policies were near their stability limit (the arrival rate being near the service rate of the first station). In the latter case, batch-service threshold strategies profit from their larger stability region. Due to the ability to serve an arbitrary large number of jobs at once, the length of average service time at the first station can become larger without affecting the performance. This advantage could be used for instance to place the hub for trucks at a cheap location far from the costly distribution centre with a minor risk of starvation of the distribution centre. Moreover, the insensitivity assumption of the service time for the size of a batch is of negligible influence when these trucks consists of only a small percentage of total road capacity.

Our results also translate into practical design rules. First of all, the simplicity of threshold-based rules makes them much easier to implement in practical scenarios. Note that the optimal switching curve policy requires to operate a different threshold level depending on the load in the first station.

A further insight is that for single service mode at the first station, it is important that the service rate in that station is large enough (preferably larger than, or at least comparable to that at the second station). When applied to the distribution centre setting of Chapter 2, this implies that the parking facility should not be too far from the distribution centre. In fact, the travel time between the two should be smaller than, or comparable to, the unloading time at the distribution centre. If multiple jobs can be simultaneously transferred between the two stations, the distance is not a major issue. In that case, performance is rather insensitive to the service speed in the first station (unless that speed is very low).

Chapter 4

Tandem Queue with Dynamic Threshold Strategies

In this chapter, we continue our investigation of the controllable tandem queue systems of Chapter 2. To overcome the computational burden of the matrix-geometric approximation of Chapter 3 for large loads, we develop new approximations that are especially suitable when the system is under heavy load. We use a fluid analysis approach for which the first queue may contain a large number of jobs, while the 'critical' second queue remains of moderate size. The randomness in the second queue determines the fluid dynamics at the first queue. This fluid-based approach results in two heuristic strategies that provide excellent approximations for a broad range of parameter values, while the computation time is quite insensitive to the system load. Numerical results demonstrate the accuracy of the approximation over a broad range of parameter values[1].

[1] This chapter is based on [**S2**].

4.1 Introduction

We again investigate the controllable two-stage tandem queue as explained in Chapter 2, where the first stage represents a storage buffer in which jobs can be kept before being transferred to the second stage. The second stage represents the service bottleneck for which we want to maintain a small number of waiting jobs. It is assumed that the buffer at the first stage is large enough so that it is reasonable to model it with infinite storage capacity. Our main motivation for this model comes from road traffic control, where one can avoid accumulation of traffic by reducing the upstream traffic flow.

We seek an optimal trade-off between a reduction in the number of jobs at the second stage on the one hand, and the additional delay caused by keeping jobs in the first stage on the other hand. The optimal point of operation is determined by minimisation of a cost function that accounts for waiting time in the buffering stage as well as waiting at the critical stage. Arrivals to this system are modelled by a Poisson process, and service times at both queues are exponentially distributed, which facilitates a formulation as an MDP. Solving the MDP to optimality is often computationally prohibitively demanding. Our main objective in this chapter is to develop two heuristic approaches that closely approximate the optimal threshold strategy. In Chapter 3, the approximation is based on the optimal threshold level, irrespective of the current state of the system. In this chapter, we capture the full structure of the threshold policy. Our heuristics will be based on the analysis of a related controlled fluid model and provides intuition for the optimal decision structure.

The proposed model is rather well understood for the single-service case, in which the first server either serves a single job or idles. This setting has been considered in [7, 75, 122, 143] and is the basis for our analysis of the batch service model; these references are discussed in more detail in Chapter 2.

In this chapter, most of our attention will be targeted at the batch service model. The approximations we propose have natural counterparts for the single-service tandem model as well. To avoid repeating discussions, we will not derive these in detail, but on several occasions, we will briefly refer to the similarities and differences between the two models, and we will also use the single-service model to illustrate the applicability of our heuristics for non-exponential service times.

The remainder of this chapter is organised as follows. In Section 4.2 we give a short recap of the model, introduce the fluid formulation, and give the formulation of the approximation methods. We illustrate the accuracy of these approximations in Section 4.3, first for exponential service durations, followed by a generalisation to phase type service distributions. Finally, Section 4.4 contains conclusions and ideas for further investigation.

4.2 Model description

In Chapter 2 we introduced two versions of a tandem queueing model for which we found the optimal strategy by solving the MDP (using successive approximation). We will reformulate these queueing models as a fluid control problem so as to approximate the optimal strategy.

4.2.1 Model

For a description of the two models that we consider in this chapter, we refer the reader to Chapter 2 (complete description), or to Chapter 3 (short description). In short, we consider a tandem queue model as introduced by [122], where the rate of the server at the first station can be controlled. The first model considers the version described in [122], whereas the second model is an extension that allows for controllable batch processing. We apply fluid scaling to the first queue, while preserving the queueing behaviour at the second queue. This approach is motivated by the earlier observation that the optimal switching curve is rather flat, see Figure 2.4 in Chapter 2 above.

The scaling that we use is different from the standard fluid scaling as first proposed in Kurtz [76]. Since the second queue is not scaled, it maintains its stochastic nature. This randomness is of a different nature than that described in [49] for a model with two queues, where the trajectories of the fluid-scaled components are random. Our scaling is also different from those in the batch-service model of [18]. Their first scaling is the standard one of [76] and in the second the batch sizes are scaled, so that the limiting fluid model has jumps.

Our scaling is closest to that described by Robert [121, Chapter 9.6].

In that work, however, the unscaled components have stationary distributions that do not depend on the position of the scaled components. In our case, the conditional distribution of queue 2 (the unscaled component) depends on the position of the fluid-scaled size of queue 1. In this chapter, we do not formally prove the convergence of the scaled stochastic process to the proposed fluid model, as was done in [121]. Instead, we propose the approximation by investigating local dynamics and illustrate the appropriateness through numerical experiments.

4.2.2 Fluid approximation

In this section, we explain the framework to obtain the fluid approximations.

Let us briefly recall the fluid limits in Avram [7] for the *single-service* controlled tandem model. It turns out that for the case $\mu_1 < \mu_2$ the optimal strategy in the fluid model is determined by a linearly increasing switching line, but that for $\mu_1 > \mu_2$, the switching line lies on the horizontal axis. This can be understood from the flat, unscaled, switching curve: in the fluid scaling, it is indistinguishable from the x-axis. We, therefore, need a different scaling if $\mu_1 > \mu_2$, and the same is true for the batch service model: For the first queue we can apply the usual fluid scaling, but the second queue should remain unscaled.

Formally, the fluid limit for the batch service model is obtained as the limit of a sequence of processes

$$\left\{\left(X_1^{(n)}(t), X_2^{(n)}(t)\right), \, t \geq 0\right\}_{n\geq 1},$$

indexed by n, which we take to be an integer, as $n \to \infty$. The sequence is determined by the queue length processes of the first and second queue, $X_1(t)$ and $X_2(t)$, respectively. Motivated by the observation from [7] discussed above and justified by numerical experiments that show that the optimal control policy indeed employs an asymptotically flat switching curve, we assume that there is a fixed constant K that uniformly bounds the switching curve from above. Our later approximations of the optimal policy are consistent with this assumption. Note that as a consequence, $X_2(t) < K$ for all t. In the next construction of the fluid limit, we follow [121, Chapter 9.6] and define

$$\begin{aligned} X_1^{(n)}(t) &= \frac{1}{n} X_1(nt), \\ X_2^{(n)}(t) &= X_2(nt), \end{aligned}$$

with initial condition $X_1(0) = n$. Thus, the initial condition for the first component is different for each process in the sequence. We will see shortly that the initial condition for X_2 is irrelevant. Note that for the first queue we scale both space and time, while for the second queue, which is uniformly bounded by the fixed constant K, we only scale time. Assuming that it exists, the fluid limit for the first queue is now defined, for $t \geq 0$, as

$$x_1(t) = \lim_{n \to \infty} X_1^{(n)}(t).$$

Note that for any fixed t, the random sequence $X_2^{(n)}(t)$ will converge weakly as $n \to \infty$, with the limiting distribution depending on (the value of the switching curve at) $x_1(t)$. Indeed, in the limit $n \to \infty$, $X_2^{(n)}(t)$ instantly reaches the stationary distribution [121, Chapter 9.6] for each fixed t. In turn, the direction of $x_1(t)$ will depend on the distribution

$$\lim_{n \to \infty} \mathbb{P}\left(X_2^{(n)}(t) \leq x | X_1^{(n)}(t) \right),$$

and in particular on its expectation which we denote by

$$x_2(t) = \lim_{n \to \infty} \mathbb{E}\left[X_2^{(n)}(t) | X_1^{(n)}(t) \right].$$

Note that for all n, $\mathbb{E}[X_2^{(n)}(t)|X_1^{(n)}(t)]$ is random due to the randomness of $X_1^{(n)}(t)$, but for very large values of n, it will be close to deterministic.

Recall that in the stochastic model, the optimal strategy is dictated by a switching curve $K(x)$ that gives the threshold value on the second queue for given $X_1(t)$. If a batch moves from the first to the second queue at time t, the process moves to the state $(y, K(y))$ on the switching curve, with y such that $y + K(y) = X_1(t-) + X_2(t-)$. The size of the batch is $X_1(t-) - y$.

Let us now specify the local dynamics of the fluid limit. Since $K(\cdot)$ is assumed to be bounded, the size of the jump does not scale with n. Therefore, the fluid limit for the first component will not show these jumps. (See also the discussion in [18], where two different scalings are distinguished, one of which has a fluid limit with jumps and the

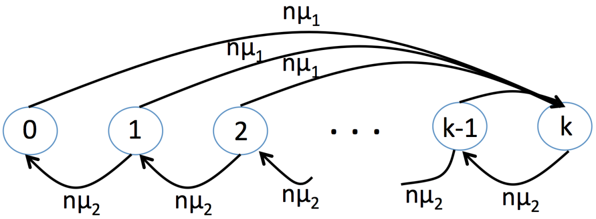


Figure 4.1. *Graphical representation of* $X_2^{(n)}(t)$.

other does not.) In the limit $n \to \infty$, the second component reaches stationarity instantly for any value of the first component. In addition, note that in the n-th system $\left(X_1^{(n)}(t), X_2^{(n)}(t)\right)$ the rate with which batches move from the first to the second queue is $n\mu_1$. In the limit the process $x_1(t)$ will decrease continuously, the speed of movement being determined by the conditional distribution of the second component.

Let us now focus on $X_2^{(n)}(t)$, the size of the second queue in the n-th system, for a given level of $X_1^{(n)}(t)$ with constant threshold value k. For large n, $X_2^{(n)}(t)$ becomes a rapidly moving random variable with the stationary distribution of a batch-arrival queue in which the batches always lift the queue to the level k, as depicted in Figure 4.1. The stationary distribution (given threshold value k) is, therefore,

$$\pi_i^{(k)} = \pi_0^{(k)} \frac{\mu_1}{\mu_2}\Big(\frac{\mu_1+\mu_2}{\mu_2}\Big)^{i-1}, \text{ for } i = 1, 2, \ldots, k, \tag{4.1}$$

and

$$\pi_0^{(k)} = \frac{1}{1 + \sum_{i=1}^k \frac{\mu_1}{\mu_2}(\frac{\mu_1+\mu_2}{\mu_2})^{i-1}}.$$

We will use the shorthand notation $\mathbb{E}\left[X_2|k\right]$ for the expectation of this distribution. The mean batch size is therefore $b(k) = k - \mathbb{E}\left[X_2|k\right]$.

This determines the dynamics of the first component in the fluid limit for a given limiting switching curve $k(x_1)$:

$$x_1'(t) = \lambda - b(k(x(t)))\mu_1,$$

as long as $x_1(t) > 0$, for a given arbitrary initial value $x_1(0)$. In our discussion above we took $X_1^{(n)}(0) = n$, to ensure that it is integer, which corresponds to $x_1(0) = 1$. The arguments remain valid for other positive values of $x_1(0)$ (for example by rounding the initial value of $nx_1(0)$ to an integer). In the next section, we will exploit this description to determine a switching curve $k(\cdot)$ that approximates the optimal switching curve.

4.2.3 Fluid-based approximations of the optimal policy

We approximate the optimal strategy using two different approaches, both based on the fluid description in the previous section. The fluid model is used to approximate the trajectory of the stochastic process $\{X_1(t), t \geq 0\}$ by a smooth path. We emphasize that we *do not* formally work with the fluid limit, but instead directly use it to replace the stochastic process. The first method employs a fixed threshold strategy and the second approximation determines a dynamic threshold based on a greedy heuristic.

Method 1
In our first approach, we ignore the fact that we can adjust the threshold level over time. For any initial value $X_1(0) = x$, we approximate the threshold level $k = k(x)$ for $k \in \mathbb{N}$ that minimises the (approximated) cost until the first component is empty. We will denote the time at which this happens by $T = T(x)$. Replacing the stochastic path of $X_1(t)$ by the trajectory of the fluid model and replacing $X_2(t)$ by its conditional expectation, we obtain:

$$\min_{k \in \mathbb{N}} \Big\{ c_1 (xT(x) + \frac{1}{2}(\lambda - b(k)\mu_1)T(x)^2) + c_2 T(x) \mathbb{E}\left[X_2|k\right] \Big\}. \quad (4.2)$$

To compute the threshold value k that minimises overall costs, we determine the time to empty the system:

$$x + (\lambda - b(k)\mu_1)T(x) = 0, \text{ and hence, } T(x) = \frac{x}{b(k)\mu_1 - \lambda}. \quad (4.3)$$

Equation (4.2) can thus be rewritten as:

$$\min_{k \in \mathbb{N}} \left\{ c_1 x \frac{1}{2} T(x) + c_2 \mathbb{E}\left[X_2|k\right] T(x) \right\}. \quad (4.4)$$

From the stationary distribution in (4.1) we can numerically determine the value of k that minimizes the approximated costs, say k^*. Our first approximation thus replaces the optimal switching curve by a fixed threshold strategy based on the value of k^*.

Method 2

The second method is based on a comparison of costs due to idleness in the second queue (implying loss of capacity if jobs from the first queue could have been moved earlier) and *additional* storage costs at the second queue (when jobs could have been transferred later from the first queue). These storage costs are therefore proportional to the number of jobs at the second queue.

Loss of capacity

Capacity loss is computed in the following manner. We again assume that the number of jobs in the first queue is large and that, at all times, queue 2 is in the equilibrium corresponding to the current threshold (say k, which is determined by queue 1). The maximum customer drain rate from the system per unit of time equals $\mu_2 - \lambda$. However, the effective outflow rate is lower than μ_2, since it is interrupted when the second queue is empty. The fraction of time that the second queue is empty, that is $\pi_0^{(k)} = \mathbb{P}\left(X_2 = 0|k\right)$ in (4.1) is determined by the value of the threshold, i.e., k. The actual outflow from the system is $\mu_2\left(1 - \pi_0^{(k)}\right)$. Dividing the actual drain rate by the maximum drain rate gives the effective capacity per unit of time. The lost capacity can then be obtained as

$$1 - \frac{\mu_2\left(1 - \pi_0^{(k)}\right) - \lambda}{\mu_2 - \lambda} = \frac{\pi_0^{(k)}}{1 - \lambda/\mu_2}. \tag{4.5}$$

Since all jobs in the first queue will be delayed by this inefficiency, we obtain the total costs for capacity loss by multiplying (4.5) with holding cost $c_1 X_1(t)$.

Storage at queue 2

The second component is intuitively easy. The average number of jobs waiting in the second queue is determined by the buffer level k. Each job faces an additional cost of $c_2 - c_1$ per time unit while being at queue 2, so total storage costs at the second queue are computed as $(c_2 - c_1)\mathbb{E}\left[X_2|k\right]$.

We combine the above into the following optimization problem:

$$\min_{k=k(x)\in\mathbb{N}} \left\{ c_1 x_1 \left(\frac{\pi_0^{(k)}}{1-\lambda/\mu_2} \right) + (c_2 - c_1)\mathbb{E}\left[X_2|k\right] \right\}. \tag{4.6}$$

4.3 Experimental results

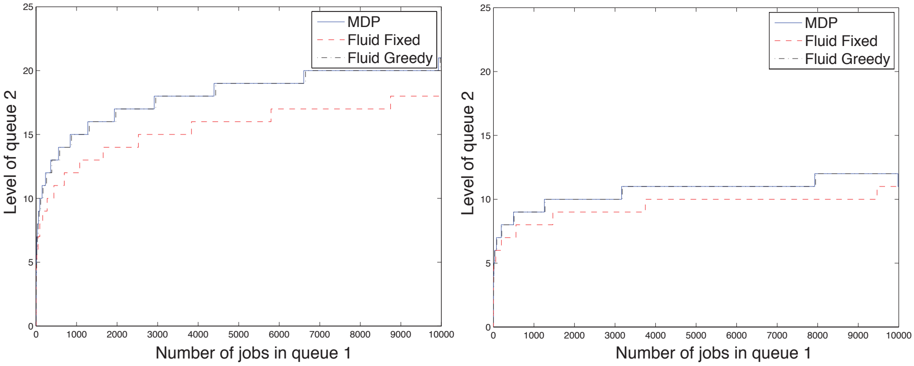
In this section, we assess the accuracy of the fluid approximation methods proposed in Section 4.2. We compute the optimal switching strategies for several parameter choices by using the MDP solution and compare them to the proposed fluid approximation heuristics in terms of average costs and computation time. We applied experiments for both exponential service times, as well as more general service time distributions by using of phase type approximations. The results of these experiments are discussed below subsequently.

4.3.1 Results for exponential service times

In Figure 4.2 we compare the asymptotics of the MDP solution and the two fluid approximations for two distinct parameter sets and for very large system states ($X_1(0) = 10^4$). From both figures, we observe that the greedy heuristic approximates the MDP threshold level very accurately, especially for a large number of jobs in the system. The fixed strategy consistently underestimates the switching curve, but does capture its shape quite well.

In Figure 4.3 we zoom in to lower levels $n = 100$, giving a more detailed picture. We observe that close to the origin the ‘fixed’ strategy overestimates the MDP curve for both parameter sets, while the ‘greedy’ approach gives an underestimation. For smaller service rate at the first queue, the ‘greedy’ heuristic is a worse approximation.

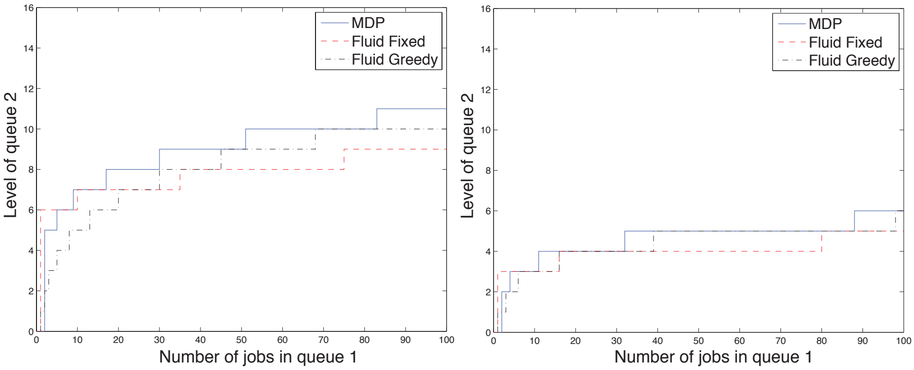
To gain more understanding of the accuracy of the approximations, we compare the average costs of the two fluid approximations with the optimal MDP solution. As a reference, we also compute the average cost for a fixed value threshold policy by using the matrix-geometric method which we have analysed in Chapter 3. The chosen parameter values are those reported above in Table 4.1. Figure 4.4 shows the relative difference in average costs of the two fluid approximations and the fixed threshold method of Chapter 3 with respect to the MDP solution. From



(a) $\lambda = 0.5,\ \mu_1 = 0.5,\ \mu_2 = 1.0$

(b) $\lambda = 0.9,\ \mu_1 = 1.5,\ \mu_2 = 1.0$

Figure 4.2. Comparison of the MDP results and the fluid heuristics for very large n.



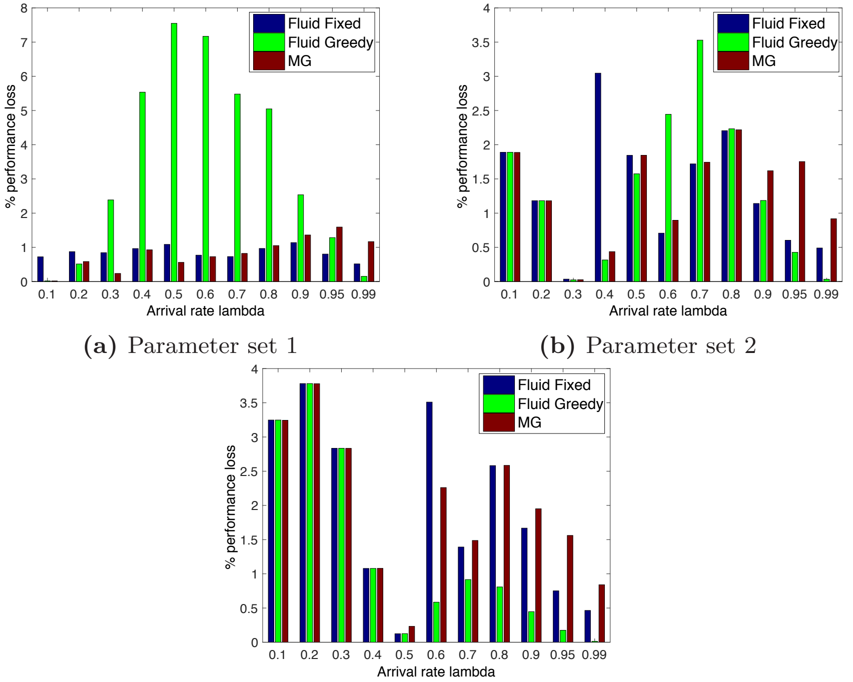
(a) $\lambda = 0.7,\ \mu_1 = 0.5,\ \mu_2 = 1.0$

(b) $\lambda = 0.7,\ \mu_1 = 1.5,\ \mu_2 = 1.0$

Figure 4.3. Comparison of the MDP results and the fluid heuristics for $n = 100$.

these experiments, we see that the increase in average cost is relatively small. The 'greedy' heuristic shows the largest relative deviation of 8% on parameter set 1. In all others, the differences relative to the optimum are not more than a few percent.

Figure 4.4a shows that on parameter set 1 the 'greedy' approximation is much less accurate than the other two approximations. We already observed in the detailed graphs of Figure 4.3 that a low service rate at the first queue causes a larger gap with the MDP threshold curve, particularly for small system states. This is reflected in the cost performance. For a



(a) Parameter set 1

(b) Parameter set 2

(c) Parameter set 3

Figure 4.4. Comparison of the MDP result with both fluid approximations and the fixed threshold approach of Chapter 3 for various parameter choices.

large system load, the typical number of jobs in the system is larger, which reduces the impact of this underestimation of the 'greedy' approach.

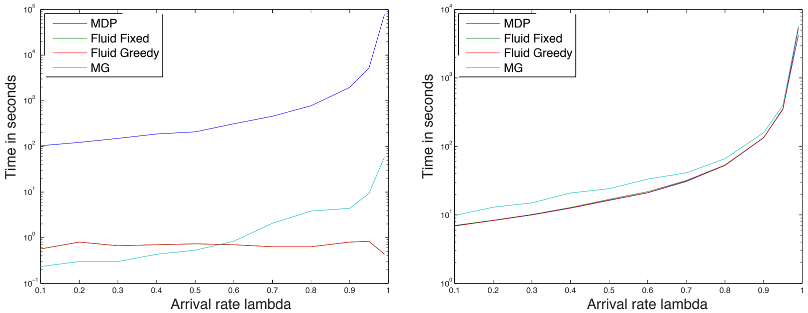
For all three parameter sets in Figure 4.4 we observe that the greedy fluid approximation gives better results for heavy loads ($\frac{\lambda}{\mu_2} \to 1$) than the other two approximations. The fixed fluid approximation appears to be the best all-round approximation.

To show the efficiency of the methods in terms of computation time, we average the computation times of the three parameter sets in Table 4.1 for increasing load. We separately investigate the time needed to compute the policy and the time it takes to compute the average costs of a given policy. The results are illustrated in Figure 4.5.

The more relevant issue is the time needed to determine a good policy. Especially for heavily loaded systems, finding the optimal policy is computationally extremely demanding for the MDP method. Figure 4.5a

Set	λ	μ_1	μ_2	c_1	c_2
1	[0.1, 0.2, . . . , 0.9, 0.95]	0.5	1.0	1	3
2	[0.1, 0.2, . . . , 0.9, 0.95]	1.0	1.0	1	3
3	[0.1, 0.2, . . . , 0.9, 0.95]	1.5	1.0	1	3

Table 4.1. Parameter sets used for the numerical experiments with the batch server.



(a) Time to compute policy **(b)** Time to compute average costs

Figure 4.5. Time in seconds to compute the policy for increasing load of the system averaged over the service rate $\mu_1 \in \{0.5, 1.0, 1.5\}$ at the first queue.

shows that the computation time of the two fluid approximations is only mildly sensitive to the parameter choice, while the other methods quickly become slower for a higher load. Note that the computation time for both fluid models is comparable, which explains the absence of the 'fluid fixed' line in the figure. Even for a small load, the fluid approximation is significantly faster than the MDP solution. We observe that the computation time depends on the load of the system and increases for a higher load.

Although of less relevance, we also compared the time needed to compute the average cost of a given strategy using an iterative approximation. (Note that for the MDP and the matrix-geometric approximation, the average cost is jointly determined with the policy itself. For the fluid approximations, these two phases are carried out separately.) It should be no surprise that for that metric all methods are essentially equivalent. It is quite likely that this computation time can be improved for all policies by using a more sophisticated computation scheme than direct iteration. Our goal here was to show that the differences are small.

4.3.2 Results for general service times

We continue our investigation of the fluid approximations and study their applicability under less restrictive assumptions on the service time distribution in the second queue, which we now take to be of phase-type. We concentrate on the single-service controlled tandem queue. The fluid approximations for the batch service model can also be used with phase-type services in the second queue, but solving the MDP for comparison becomes too demanding.

To apply the fluid approximations of Section 4.2 for the controllable tandem queue with two single server queues, we only need minor modifications: The second queue is now approximated with the usual $M/M/1/k$ queue instead of a batch-arrival queue, and we use its truncated geometric distribution as the conditional distribution for $X_2|k$. Since transfers are now all for single jobs, in the fluid formulation for the first queue we have a more limited control rule $b(\cdot) \in \{0, 1\}$.

Specifically, we will use the Erlang (with low variability) and the hyper-exponential (high variability) distributions for service durations in the second queue, and take the stationary distribution of the corresponding $M/PH/1/k$ queue as the conditional distribution for $X_2|k$. We keep the processing rates at the first and second queue (μ_1 and μ_2) fixed for all experiments, while adjusting the squared coefficient of variation. For an Erlang service distribution with m phases the squared coefficient of variation is given by

$$v^2 = \frac{1}{m}.$$

We parameterise the hyper-exponential distribution with two phases as follows:

$$F(x) = 1 - p_1 \mathrm{e}^{-\nu_1 x} - p_2 \mathrm{e}^{-\nu_2 x},$$

with $0 \leq p_1 = 1 - p_2 \leq 1$ and $\nu_1 > 0$, $\nu_2 > 0$. We use the method of ‘balanced means’ of Tijms [136] to determine these parameters for given mean $1/\mu_2$ and squared coefficient of variation v^2:

$$\nu_1 = 2p_1\mu_2 \qquad \text{and} \qquad \nu_2 = 2p_2\mu_2,$$

where

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{v^2 - 1}{v^2 + 1}}\right) \qquad \text{and} \qquad p_2 = 1 - p_1.$$

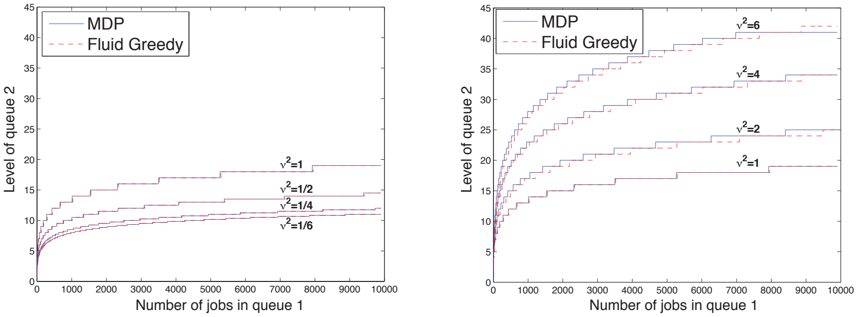
As we will see, using the heuristic rules from the model with exponential services is straightforward, but the MDP solution suffers enormously in terms of computability, which demonstrates the need for approximations.

We extend our earlier experiments with the parameter sets presented in Table 4.2. We specify the service distribution at the second queue in the column ‘Type’. To allow comparison between the different systems, we keep the average service duration at the second queue ($1/\mu_2$) fixed for all experiments and vary the coefficient of variation. In all our numerical experiments the computations were performed by adequately truncating the state space, depending on the specific parameter values.

Set	Type	v^2
4	Exponential	1
5	Erlang-2	1/2
6	Erlang-4	1/4
7	Erlang-6	1/6
8	Hyper-2	2
9	Hyper-2	4
10	Hyper-2	6

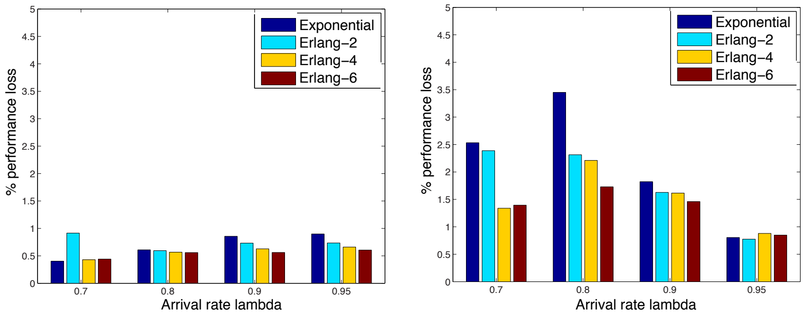
Table 4.2. Parameter sets for the single-service model with phase-type services in queue 2; λ takes values in $\{0.7, 0.8, 0.9, 0.95\}$ and throughout we use $\mu_1 = 1.5, \mu_2 = 1.0, c_1 = 1$ and $c_2 = 3$.

The results of this set of experiments are illustrated in Figure 4.6a for Erlang service times of server 2, and in Figure 4.6b for hyper-exponential services. We also show the corresponding graphs for exponential service durations (parameter set 4) as a reference. Clearly, the optimal switching curve obtained with MDP and the switching curve of the fluid approximation are again very close to each other. As might be expected, the switching curve is lower for less variable distributions (Erlang with many phases), because the departures from queue 2 can be predicted more accurately and thus there is less need to maintain a large buffer in queue 2. Similarly, for the hyper-exponential service durations with



(a) Erlang service in queue 2 **(b)** Hyper-exponential service in queue 2

Figure 4.6. Comparison of the MDP results and fluid approximations for various v^2 in service variability at the second queue with load $\rho = 0.7$.

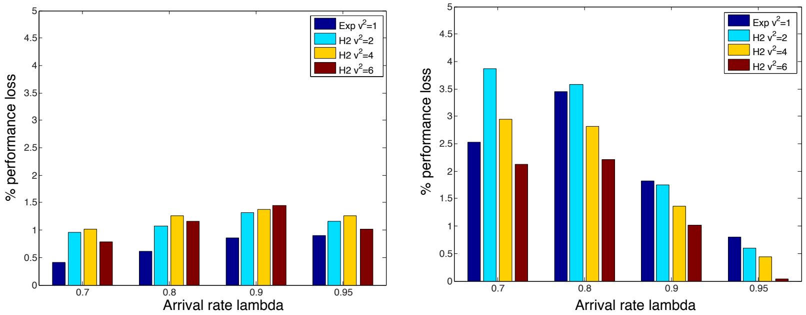


(a) Fluid fixed **(b)** Fluid greedy

Figure 4.7. Performance results of the fluid approximations for parameter sets 4 to 7.

increasing variance a more conservative strategy (larger threshold) is needed.

As before, we also investigate the accuracy in terms of achieving close to minimum cost. In Figures 4.7 and 4.8 we observe that, as before, we obtain a better approximation in terms of cost for more highly loaded systems. This is natural, since the fluid approximation is tailor-made for states far from the origin. For two-phase hyper-exponential services at the second queue, we observe a better performance with the 'greedy fluid' approach for larger coefficients of variation, while the 'fixed fluid' approach does the opposite. This can be explained by the fact that larger states are more easily reached with more variability in the service times, which was better approximated by the greedy approach. For the



(a) Fluid fixed

(b) Fluid greedy

Figure 4.8. Performance results of the fluid approximations for parameter sets 8-10.

lower variation of the Erlang service distributions, we see that both the ‘fluid fixed’ and ‘greedy’ approach give better performance when the coefficient of variation decreases. This suggests, unsurprisingly, that the fluid approximations are well suited for systems that have little variation in the service times.

4.4 Conclusion

We have investigated the structure of the optimal strategy to control the first service stage of a tandem queueing system with batch services. In the Markovian setting, we formulated an MDP to determine the optimal strategy in terms of *when to serve at the first stage* and *how large the batch size should be.* To gain more understanding of the shape of the optimal MDP policy, we developed approximations and computationally efficient heuristics that are very close to the MDP strategy, especially for high loads.

For the design of our heuristics, we noted that the optimal MDP strategy is characterised by a switching curve that is rather flat. In order to formulate a meaningful approximating fluid model, we applied different scalings to the two queues, see the approach in [121, Chapter 9.6]. To the best of our knowledge, this has not been applied to stochastic control problems before.

We have developed two different heuristics based on the fluid model approximation. The ‘fixed fluid’ heuristic underestimates the optimal

MDP strategy in states with a large number of jobs in the system, while the ‘greedy fluid’ approach follows the optimal MDP switching curve quite closely. The average costs of the ‘fixed fluid’ approach remain within a few percent of the average costs of the MDP solution for a wide range of parameters. The ‘greedy’ approach becomes more accurate for a higher load.

Encouraged by the simplicity and the accuracy of the two approximations for the batch tandem system, we investigated the applicability for non-exponential service durations in the second queue. For the batch service model, the MDP formulation quickly becomes numerically intractable, leaving us with no bench mark to test our approximations. For this reason, we illustrated the potential of the approach for more general service times by only allowing single services at the first station. We again obtain an approximation function that closely follows the optimal MDP policy. As before, the accuracy of the ‘greedy fluid’ approximation improves for increasing load and the ‘fixed fluid’ approach performs well for a wide range of parameters.

The proposed fluid approach is computationally very fast. Solutions are available within a second (evaluated on a Macbook Pro dated from 2013, with processor 2.4 GHz Intel Core i5 and 8GB internal memory). This suggests that the approach is worth exploring for larger queueing networks with non-exponential service times.

Appendix

We present an extended overview of the numerical results shown in the figures of Section 4.3. Table 4.3 shows an overview for a range of parameters of the costs for each method using the batch model. Table 4.4 shows this for the generalised service rate examples.

Service Type	Parameters					$\mathbb{E}[Cost]$			
	λ	μ_1	μ_2	c_1	c_2	MDP	Fluid Fixed	Fluid Greedy	MG
Batch	0.10	0.50	1.00	1	3	0.57	0.57	0.57	0.57
	0.10	1.00	1.00	1	3	0.44	0.44	0.44	0.44
	0.10	1.50	1.00	1	3	0.39	0.40	0.40	0.40
	0.20	0.50	1.00	1	3	1.29	1.30	1.29	1.29
	0.20	1.00	1.00	1	3	0.97	0.98	0.98	0.98
	0.20	1.50	1.00	1	3	0.86	0.89	0.89	0.89
	0.30	0.50	1.00	1	3	2.20	2.22	2.25	2.21
	0.30	1.00	1.00	1	3	1.63	1.63	1.63	1.63
	0.30	1.50	1.00	1	3	1.42	1.46	1.46	1.46
	0.40	0.50	1.00	1	3	3.36	3.39	3.55	3.39
	0.40	1.00	1.00	1	3	2.45	2.52	2.45	2.46
	0.40	1.50	1.00	1	3	2.14	2.17	2.17	2.17
	0.50	0.50	1.00	1	3	4.88	4.93	5.25	4.91
	0.50	1.00	1.00	1	3	3.51	3.58	3.57	3.58
	0.50	1.50	1.00	1	3	3.07	3.07	3.07	3.08
	0.60	0.50	1.00	1	3	6.93	6.99	7.43	6.98
	0.60	1.00	1.00	1	3	4.97	5.01	5.10	5.02
	0.60	1.50	1.00	1	3	4.31	4.47	4.34	4.41
	0.70	0.50	1.00	1	3	9.91	9.99	10.46	9.99
	0.70	1.00	1.00	1	3	7.12	7.25	7.38	7.25
	0.70	1.50	1.00	1	3	6.18	6.27	6.24	6.27
	0.80	0.50	1.00	1	3	14.80	14.94	15.55	14.96
	0.80	1.00	1.00	1	3	10.77	11.00	11.01	11.00
	0.80	1.50	1.00	1	3	9.40	9.65	9.48	9.65
	0.90	0.50	1.00	1	3	25.74	26.03	26.39	26.09
	0.90	1.00	1.00	1	3	19.41	19.63	19.64	19.73
	0.90	1.50	1.00	1	3	17.24	17.53	17.31	17.57
	0.95	0.50	1.00	1	3	42.11	42.45	42.65	42.78
	0.95	1.00	1.00	1	3	33.32	33.52	33.46	33.90
	0.95	1.50	1.00	1	3	30.26	30.49	30.32	30.73
	0.99	0.50	1.00	1	3	137.98	138.69	138.19	139.59
	0.99	1.00	1.00	1	3	122.86	123.46	122.90	123.98
	0.99	1.50	1.00	1	3	117.56	118.11	117.58	118.55

Table 4.3. Average costs for the experiments of the batch tandem queue.

Service type	v^2	Parameters					$\mathbb{E}[Cost]$		
		λ	μ_1	μ_2	c_1	c_2	MDP	Fluid Fixed	Fluid Greedy
Exponential	1	0.70	1.50	1.00	1	3	6.58	6.61	6.75
		0.80	1.50	1.00	1	3	10.32	10.39	10.68
		0.90	1.50	1.00	1	3	19.67	19.84	20.03
		0.95	1.50	1.00	1	3	34.97	35.29	35.25
Erlang-2	$\frac{1}{2}$	0.70	1.50	1.00	1	3	5.70	5.75	5.84
		0.80	1.50	1.00	1	3	8.66	8.71	8.86
		0.90	1.50	1.00	1	3	15.85	15.97	16.11
		0.95	1.50	1.00	1	3	27.40	27.60	27.61
Erlang-4	$\frac{1}{4}$	0.70	1.50	1.00	1	3	5.26	5.28	5.33
		0.80	1.50	1.00	1	3	7.82	7.87	8.00
		0.90	1.50	1.00	1	3	13.93	14.02	14.15
		0.95	1.50	1.00	1	3	23.59	23.75	23.80
Erlang-6	$\frac{1}{6}$	0.70	1.50	1.00	1	3	5.11	5.13	5.18
		0.80	1.50	1.00	1	3	7.54	7.58	7.67
		0.90	1.50	1.00	1	3	13.29	13.36	13.48
		0.95	1.50	1.00	1	3	22.32	22.46	22.51
Hyper-2	2	0.70	1.50	1.00	1	3	8.05	8.13	8.36
		0.80	1.50	1.00	1	3	13.15	13.29	13.62
		0.90	1.50	1.00	1	3	26.37	26.72	26.83
		0.95	1.50	1.00	1	3	48.56	49.12	48.85
Hyper-2	4	0.70	1.50	1.00	1	3	10.77	10.88	11.09
		0.80	1.50	1.00	1	3	18.46	18.69	18.98
		0.90	1.50	1.00	1	3	39.14	39.68	39.68
		0.95	1.50	1.00	1	3	74.78	75.73	75.12
Hyper-2	6	0.70	1.50	1.00	1	3	13.36	13.47	13.65
		0.80	1.50	1.00	1	3	23.55	23.82	24.07
		0.90	1.50	1.00	1	3	51.51	52.26	52.03
		0.95	1.50	1.00	1	3	100.03	101.37	100.40

Table 4.4. Average costs for the experiments of the generalised tandem queue for specified service type at the second queue.

Part II

User Behaviour

Chapter 5

Modelling User Interaction at a Stochastic Traffic Bottleneck

In this chapter, we focus on the user response during peak hour periods at bottleneck locations subject to uncertainty. Our analysis is based on a popular approach to model congesting and user response. This model, known as the Vickrey bottleneck model [138], captures travellers' response to congestion in a very simplified manner, thereby enabling equilibrium results to be computed and evaluated. Extensions are easily applied, allowing comparisons across different situations. This model ignores the fact that the demand and capacity at a bottleneck are subject to uncertainty. While this fluid approach may be correct when the number of travellers is large, it fails to yield accurate predictions for a small number of travellers.

Motivated by this, we propose a stochastic version of the bottleneck model, that can also handle a smaller number of travellers. We discuss the error made by the fluid approximation, and show that the Nash equilibrium of the original model, results in highly varying costs when applied in the more realistic setting with stochasticity. We then discuss an algorithm to numerically approximate the equilibrium arrival rate for the stochastic bottleneck model, and propose a closed-form estimation for this equilibrium. This contribution lays the groundwork for future studies into the effect of stochasticity in these bottleneck models[1].

[1] This chapter is based on [**S3**].

5.1 Introduction

Bottlenecks are a common phenomenon in road traffic, and arise when the rate of traffic arriving into a stretch of road temporarily exceeds its capacity. The resulting congestion causes economic damages and discomfort to the travellers. Bottlenecks have been extensively studied in the research literature, starting with the seminal work of Vickrey [138], inspired by a morning commute. Traffic is modelled as a fluid, and travellers experience a penalty for waiting at the bottleneck, as well as for arriving at their office earlier or later than intended. Because the morning commute is a recurring and predictable phenomenon, travellers can learn the behaviour of others, and eventually adjust their departure time from home to minimise costs. This strategic behaviour is modelled in [138, 5] by assuming that traffic arrives according to a Nash equilibrium (NE), meaning that no traveller can shift its arrival into the bottleneck without increasing its costs [103].

This bottleneck model and its variants have been studied extensively in econometrics and transportation literature, and it remains a popular starting point for many recent studies, see [5, 129] for overviews. Extensions include demand elasticity [4], which studies the impact of capacity expansion at the bottleneck. The impacts of heterogeneity among travellers is studied in [6], where there are multiple classes of travellers with different cost parameters and target times for departing the bottleneck. In [106] the conditions for equilibrium existence assuming a heterogeneous distribution are explored. More recent studies consider spatial effects [78], endogenous trip timing effects with respect to group arrival times [48], and the relationship between parking facilities and congestion [133]. The NE can be computed in closed form for a range of these model variants.

In practice, road traffic is not a fluid, but instead consists of *individual* travellers, each of which may have some uncertainty surrounding its arrival time at the bottleneck and its driving speed. The fluid assumption used in the bottleneck literature is accurate when both (1) the number of travellers at the bottleneck, and (2) the bottleneck capacity are large, but is inadequate for ‘smaller’ bottlenecks. To study the effects of variability and the fact that a bottleneck consists of discrete travellers, we modify the traditional deterministic bottleneck model [5] by considering the traffic waiting at the bottleneck as a stochastic process.

While the resulting stochastic model is less tractable than the deterministic bottleneck model, it allows for more detail and accuracy. We first show that the NE computed for the deterministic model does not provide equal costs in the stochastic setting, unless the number of travellers is large. We then compute a similar equilibrium concept for the stochastic model, and discuss how it differs from the NE. Using these results we propose a closed-form approximation for the stochastic equilibrium, and show that it performs well.

The work presented here fits in a larger trend towards modelling uncertainty in the bottleneck literature. Most variants of the bottleneck model that include stochasticity do so exogenously, for instance, by including some random additional travel time due to an incident [110, 33]. In our study, we investigate the impact of endogenous effects, where the uncertainty of the arrival behaviour is included in the model.

The impact of uncertainty over time due to endogeneity is studied in [44, 45, 47]. In particular, in [45] the daily demand and capacity are assumed to be random variables with a known distribution. The authors show that under quite general assumptions the variance of the delay is increasing in its expectation. This phenomenon has been observed empirically in [23], where the authors demonstrate that travel time variance is strongly correlated with the queue length. A paper by [153] adds these effects by increasing the variance of the error term depending on the queue length. However, in our model, these effects are implicitly included, which confirms the accuracy of our modelling approach.

Beyond the transportation science literature, this chapter is closely related to those on the boundary between queueing and game theory. In [55] the authors consider a queueing system with a finite number of customers that must arrive before some time T, where each customer tries to minimise its waiting time by strategically determining its arrival time. This model was extended in [61], where the arrival rate is modelled as a non-homogeneous Poisson process and early arrivals are served at random when the facility opens. This is also related to the so-called concert queueing model [70], where customers aim to arrive at some time T, but incur costs for both waiting and tardiness. Various extensions and generalisations have been studied in [70, 62]. Other related models from game theory are the airport boarding game [134] and the meeting game [58]. This chapter is most closely related to [127], where the authors study a similar model to ours, but consider a different equilibrium concept.

The remainder of this chapter is structured as follows. First, we discuss the traditional deterministic bottleneck model and the Nash equilibrium in Section 5.2. In Section 5.3, we introduce the stochastic bottleneck model, and show how to numerically approximate the equilibrium arrival rate. We use this to propose a closed-form approximation of the stochastic equilibrium. We conclude in Section 5.4, and outline future research directions.

5.2 Deterministic bottleneck model

In this section, we provide some background on the deterministic bottleneck model introduced in [5], and describe the Nash equilibrium that ensures that all traffic experiences the same costs.

5.2.1 Model outline

We consider a single bottleneck, with fluid arriving at time t with rate $\lambda(t)$. The fluid represents identical travellers, and the bottleneck can serve a fixed capacity s of traffic per time unit. Each traveller wants to exit the bottleneck at time t^*, and incurs a penalty for the *waiting time* in the queue and for departing from the queue *earlier* or *later* than the desired time t^*. This penalty is captured by a linear cost function, with cost coefficients α (waiting), β (early arrival), and γ (late arrival).

Let t_q denote the time of the first arrival. Then the cumulative inflow of traffic at the bottleneck up to time t can be written as $a(t) = \int_{t_q}^{t} \lambda(u)\mathrm{d}u$, and the cumulative outflow as $d(t) = s\max\{0, t - t_q\}$ (assuming that the bottleneck only empties once). The sojourn time of a traveller arriving at time t can be computed as $w(t) = a(t) - d(t)$. The cost incurred by an arrival at time t can then be written as

$$c(t, \lambda) = \alpha w(t) + \beta(t^* - t - w(t))^+ + \gamma(t + w(t) - t^*)^+, \qquad (5.1)$$

with $(a)^+ = \max\{a, 0\}$. Here, the $t + w(t) - t^*$ denotes the difference between the departure time of a traveller $t+w(t)$ and its desired departure time t^*. Observe that c depends on λ through the sojourn time w.

5.2.2 Nash equilibrium

In this section, we summarise the equilibrium conditions of the standard bottleneck model as defined by Arnott [5]. Given a total amount of fluid N, we want to find an inflow curve $\lambda_f(t)$ such that no traveller can decrease its costs by altering its arrival time at the bottleneck. It has been shown (see, e.g., [132]) that such a *Nash equilibrium* exists, is unique for $\alpha > \beta$, and is given by

$$\lambda_f(t) = \begin{cases} r_1(t - t_q) & t \in [t_q, t_n) \\ r_1(t_n - t_q) + r_2(t - t_n) & t \in [t_n, t_{q'}] \end{cases}, \tag{5.2}$$

where

$$r_1 = s + \frac{\beta s}{\alpha - \beta}, \quad r_2 = s - \frac{\gamma s}{\alpha + \gamma}, \tag{5.3}$$

and

$$t_q = t^* - \frac{\eta N/s}{1+\eta}, \quad t_{q'} = t^* + \frac{N/s}{1+\eta}, \quad t_n = t^* - \frac{\delta N/s}{\alpha}, \tag{5.4}$$

with $\eta = \frac{\gamma}{\beta}$ and $\delta = \frac{\beta\gamma}{\beta+\gamma}$. This arrival curve gives all travellers equal costs

$$c_f = \delta \frac{N}{s}. \tag{5.5}$$

For $\alpha > \beta$ the inflow rate presented in Equation (5.2) generates a single busy period, i.e., $w(t) > 0$ for all $t \in (t_q, t_{q'})$ [132]. In this equilibrium, the first and last fluid will only incur costs for early/late arrivals, and experience no delay. The fluid leaving exactly at the preferred time t^* encounters costs consisting only of delay.

An example of the NE is illustrated in Figure 5.1. It shows the cumulative inflow $a(t)$ (blue), the cumulative outflow $d(t)$ (black), and the waiting time $w(t)$ (red).

5.3 Stochastic bottleneck model

In practice, road traffic is not a perfect fluid, but consists of individual travellers, which each having some uncertainty surrounding their arrival

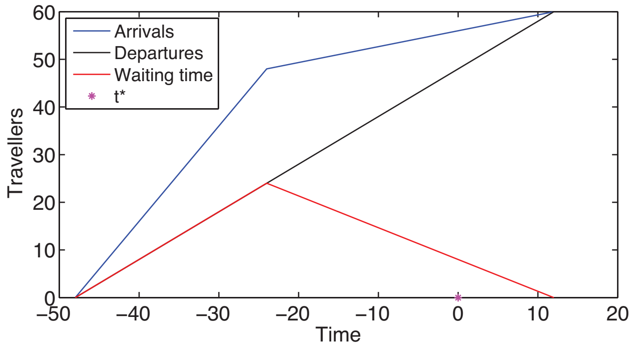


Figure 5.1. Equilibrium inflow, outflow and waiting time for the deterministic bottleneck model with $T = N/s = 60$, $\alpha = 1$, $\beta = 0.5$, $\gamma = 2$ and $t^* = 0$.

time at the bottleneck and their speed. We capture this uncertainty by assuming discrete travellers that arrive at the queue according to a time-dependent Poisson process with rate $\lambda(t)$. The arrival rate function is such that the expected total number of travellers in N, i.e., $\int \lambda(t)dt = N$. The bottleneck can serve only a single traveller at a time, which takes an independent and identically distributed exponential time with rate $\mu = s$. Such assumptions are not uncommon in this setting (see, [21, 62], for example). Note that our model is equivalent to an $M_t/M/1$ queue.

Similar to the deterministic model, each traveller prefers to exit the bottleneck at time t^*, and incurs a linear penalty α for waiting, β for arriving early, and γ for tardiness. Let us denote by $W(t)$ the random variable that represents the sojourn time of a traveller arriving at time t, for $t \geq 0$, which depends on the past arrival rate through the travellers in the queue upon arrival. The cost function for the stochastic model is identical to that of the deterministic model (5.1), with the sojourn time replaced by its stochastic counterpart:

$$C(t, \lambda) = \alpha W(t) + \beta(t^* - t - W(t))^+ + \gamma(t + W(t) - t^*)^+. \quad (5.6)$$

Note that $C(t, \lambda)$ is also a random variable, since it depends on $W(t)$.

To compare our model to that of the deterministic case for the equilibrium we need to find the so-called *symmetric Nash equilibrium*, where we choose the time-dependent arrival rate function of the Poisson process in such a way that no traveller can improve its costs unilaterally [127]. Therefore, we consider the problem of finding an arrival rate such that the expected costs $\mathbb{E}[C(t,\lambda)]$ over time is constant when $\lambda > 0$, and larger otherwise. This gives

$$\begin{aligned}\mathbb{E}[C(t,\lambda)] = \alpha\mathbb{E}[W(t)] + \beta\mathbb{E}[(t^* - t - W(t))^+] \\ + \gamma\mathbb{E}[(t + W(t) - t^*)^+]. \end{aligned} \tag{5.7}$$

In contrast to the deterministic model, we cannot obtain this equilibrium in closed form. Instead, we describe how to compute it numerically in the next sections.

5.3.1 Expected costs computation

To determine whether we can find an equilibrium, we first compute the expected costs over time for a given arrival rate function $\lambda(t)$. The expected cost of a traveller depends on its sojourn time, which is determined by the queue length upon arrival. Below we describe how to compute the transient queue-length distribution for a given arrival rate $\lambda(t)$, and use this to compute the sojourn time distribution and the expected costs. We assume that there exist some $t_0 < t_1$ such that $\lambda(t) = 0$ outside of $[t_0, t_1]$.

We consider a continuous-time Markov chain representing the number of travellers waiting at the bottleneck. At each state, an arrival or departure can take place, except for state 0 in which there is no one waiting. The time-dependent transition matrix $Q(t)$ is given by

$$Q(t) = \begin{bmatrix} -\lambda(t) & \lambda(t) & 0 & \cdots \\ \mu & -(\lambda(t)+\mu) & \lambda(t) & \cdots \\ 0 & \mu & -(\lambda(t)+\mu) & \lambda(t) \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

We denote by $\bar{\pi}(t) = \big(\pi_0(t), \pi_1(t), \dots\big)$ the distribution of the number of travellers waiting at time t. To compute this we use uniformisation, where we embed on time instances according to a Poisson process with rate equal to

$$\nu = \sup_{t\geq 0} \lambda(t) + \mu, \tag{5.8}$$

assuming that this supremum exists. Let $\Delta > 0$ and observe that

$\frac{(\nu\Delta)^n}{n!}e^{-\nu\Delta}$ denotes the probability that n transitions occur in an interval of length Δ. By conditioning on this, we may write

$$\bar{\pi}(t+\Delta) = \bar{\pi}(t)\sum_{n=0}^{\infty}\frac{(\nu\Delta)^n}{n!}e^{-\nu\Delta}P(t)^n, \tag{5.9}$$

where $P(t)$ denotes the transition probability matrix of the embedded Markov chain given by

$$P(t) = I + \frac{1}{\nu}Q(t). \tag{5.10}$$

We can then approximate $\bar{\pi}(t)$ by discretising time into small intervals of length Δ and iterating according to Equation (5.9), starting from $\bar{\pi}(t_0) = (1, 0, \dots)$.

Having outlined a numerical procedure to obtain the queue-length distribution, we can use this to determine the expected costs over time, by first computing the sojourn time distribution at each time instant t for an arriving traveller.

Let $f(\tau, t)$ denote the density function of the sojourn time of a traveller arriving at time t. By conditioning on the number of travellers seen upon arrivals this can be written as

$$f(\tau;t) = \sum_{n=0}^{\infty}\pi_n(t)g_{n+1}(\tau), \tag{5.11}$$

where $g_n(\tau)$ denotes the sojourn time density function given n travellers seen upon arrival, which follows an Erlang-$(n+1, \mu)$ distribution:

$$g_n(\tau) = \frac{\mu(\mu\tau)^n e^{-\tau\mu}}{n!}. \tag{5.12}$$

To obtain the unconditional sojourn time distribution we substitute (5.9) and (5.12) into (5.11).

We are now in position to compute the expected cost incurred by a traveller arriving at time t. To this end, we evaluate the expected costs (5.7) by conditioning on the sojourn time of the traveller arriving at time t:

$$\begin{aligned}\mathbb{E}[C(t,\lambda)] = \alpha & \int_{\tau=0}^{\infty} \tau f(\tau;t)\mathrm{d}\tau \\ & + \beta \int_{\tau=0}^{(t^*-t)} (t^* - t - \tau) f(\tau;t)\mathrm{d}\tau \\ & + \gamma \int_{\tau=t^*-t}^{\infty} (t + \tau - t^*) f(\tau;t)\mathrm{d}\tau.\end{aligned} \tag{5.13}$$

To compute the integrals we partition the sojourn time into small intervals with length Δ to obtain the following approximation:

$$\begin{aligned}\mathbb{E}C[(t,\lambda)] \approx \alpha\Delta & \sum_{k=0}^{\infty} k\Delta f(k\Delta;t) \\ & + \beta\Delta \sum_{k=0}^{\lfloor (t^*-t)/\Delta \rfloor} (t^* - t - k\Delta) f(k\Delta;t) \\ & + \gamma\Delta \sum_{k=\lceil (t^*-t)/\Delta \rceil}^{\infty} (t + k\Delta - t^*) f(k\Delta;t).\end{aligned} \tag{5.14}$$

To illustrate this procedure, we compute the expected costs in the stochastic model for the arrival rate λ_f given by the equilibrium function of Equation (5.2), which is the NE for the standard Vickrey model. In Figure 5.2 we plot these costs $\mathbb{E}[C(t,\lambda_f)]$ over time, for each parameter set defined in Table 5.1. For each parameter set, we vary the value of N and s such that N/s remains constant. We see from Figure 5.2 that $\mathbb{E}[C(t,\lambda_f)]$ varies significantly between travellers, in particular putting travellers arriving towards the end of the busy period at a disadvantage. This demonstrates that the NE, found by the standard bottleneck model is not an accurate equilibrium concept in a realistic setting with stochasticity. As expected, the error is the largest when N is small and disappears as N grows large. For the remainder of the chapter, we will refer to the NE of the standard bottleneck model as the Vickrey equilibrium.

In Figure 5.3 we plot the decomposition of the costs $\mathbb{E}[C(t,\lambda_f)]$ into its three components: *waiting*, *early arrival* and *tardiness*. This figure suggests that the large increase in expected costs just before the peak moment is due to the combination of costs for late and early arrival,

	$T = \frac{N}{s}$	t^*	α	β	γ
Set 1	60	0	1	0.5	2
Set 2	60	0	1	0.5	0.5
Set 3	60	0	1	0.75	0.5

Table 5.1. Parameter sets for numerical experiments.

whereas only early or late costs are encountered in the deterministic model. Moreover, at the end of the bottleneck period in the stochastic model, the queue may not disappear at time $t_{q'}$, giving travellers additional costs $\alpha + \gamma$ for every unit of time spent waiting.

5.3.2 Stochastic equilibrium

In the previous section, we demonstrated that the NE of the deterministic bottleneck model, further referred to as the *Vickrey equilibrium*, fails to provide equal costs for all travellers in a stochastic setting. Therefore, we present a numerical scheme to numerically approximate the equilibrium arrival rate for the stochastic model. That is, we aim to find equilibrium costs c_s and a time-dependent arrival function λ such that $\mathbb{E}[C(t, \lambda)] = c_s$ for all t with $\lambda(t) > 0$.

Our numerical scheme consists of two phases. First, we describe a procedure to obtain an arrival rate λ that satisfies

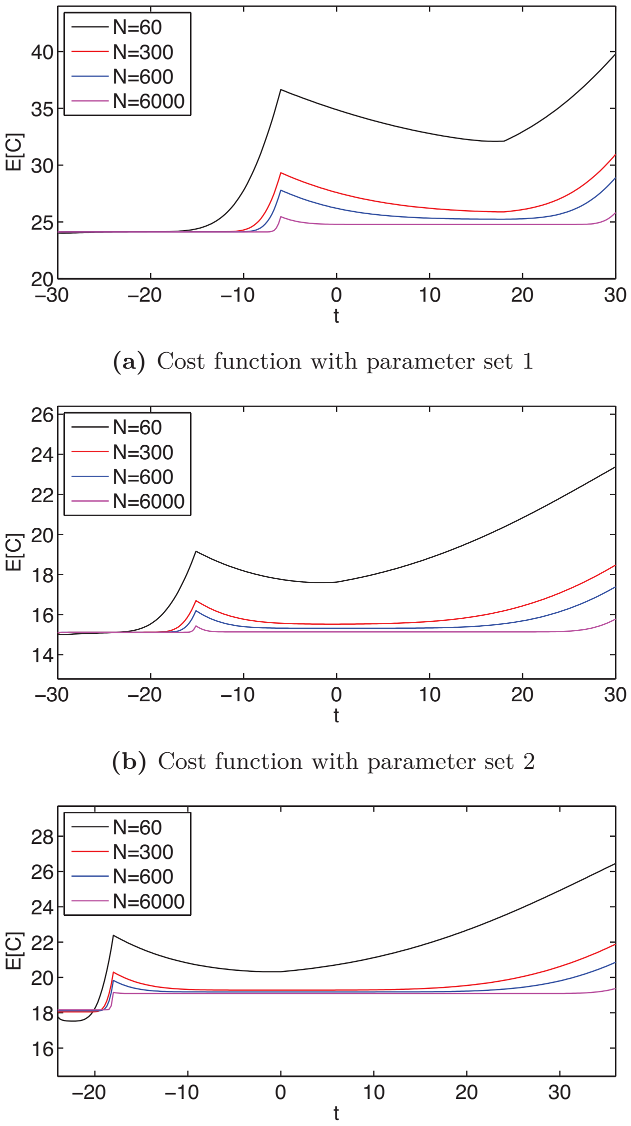
$$\mathbb{E}[C(t, \lambda)] = c \tag{5.15}$$

for any $c > 0$. We then scale the arrival rate and the costs to ensure that in expectation N travellers arrive during the bottleneck period. We use the Vickrey equilibrium costs c_f from Equation (5.5) as a starting point.

Given target costs c, we can extract the start of the stochastic bottleneck period t_0 using the observation that the first traveller to arrive likely incurs no costs for being late, or waiting in the queue. Instead, the traveller is penalised for being early. We solve

$$\beta(t^* - t_0 - \mathbb{E}[W(t_0)]) + \alpha\mathbb{E}[W(t_0)] = c, \tag{5.16}$$

where $\mathbb{E}[W(t_0)] = 1/s$ denotes the service time duration of the traveller.



(a) Cost function with parameter set 1

(b) Cost function with parameter set 2

(c) Cost function with parameter set 3

Figure 5.2. Expected costs $\mathbb{E}[C(t, \lambda_f)]$ for the stochastic model with the arrival rate of the Vickrey equilibrium for different cost parameters function and total number of travellers N.

Solving this we obtain

$$t_0 = t^* - \frac{\beta - \alpha + sc}{s\beta}. \tag{5.17}$$

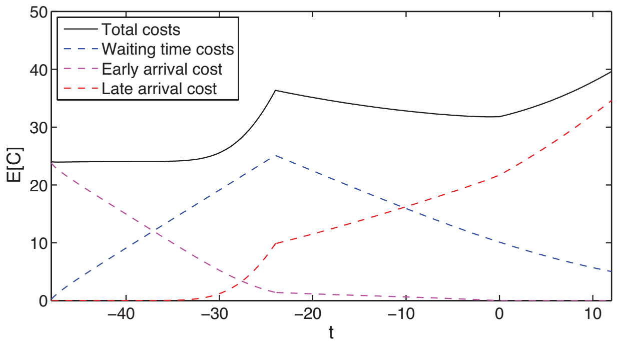


Figure 5.3. Decomposition of $\mathbb{E}C(t, \lambda_f)$, with parameter set 1 and $N = 60$.

We discretise time into small intervals of length Δ in order to find the time-dependent arrival rate that satisfies Equation (5.15). We do so iteratively, exploiting the observation that $\mathbb{E}[C(t, \lambda)]$ only depends on $\lambda(u)$ for $t_0 \leq u \leq t$, due to fact that travellers are served in order of arrival. We let $t \geq t_0$ and assume that λ is such that $\mathbb{E}[C(u, \lambda)] = c$ for all $t_0 \leq u \leq t$. We use this to determine the correct arrival rate for time $t + \Delta$.

In particular, we initialise $\lambda(t + \Delta) = \lambda(t)$, and then adjust the arrival rate by small increments x until we obtain $\mathbb{E}[C(u + \Delta, \lambda)] = c$ within some small error bound ϵ. The direction of the increments can be obtained from the observation that the cost function is increasing for larger arrival rate $\lambda(t)$. In case of an early arrival, the costs change by $\frac{x\Delta}{s}(\alpha - \beta)$, where $\alpha > \beta$. In case of a late arrival, the costs change by $\frac{x\Delta}{s}(\alpha + \gamma)$ which is positive as well. We continue this procedure until we first hit a time t_1 such that $\lambda(t_1) = 0$.

The procedure described above yields an arrival rate λ such that $\mathbb{E}[C(t, \lambda)] = c$ for all $t \in [t_0, t_1]$, but may not in expectation result in the arrival of N travellers:

$$\Delta \sum_{t=t_0}^{t_1} \lambda(t) = N. \tag{5.18}$$

To leverage this procedure to obtain the equilibrium arrival rate for N travellers we modify the target costs c, or equivalently, the bottleneck starting time t_0.

Based on the starting point t'_0 and number of travellers N' obtained from an iteration of the algorithm described above, we determine the new starting point by adding the expected service time of the difference in arrivals $(N - N')/s$:

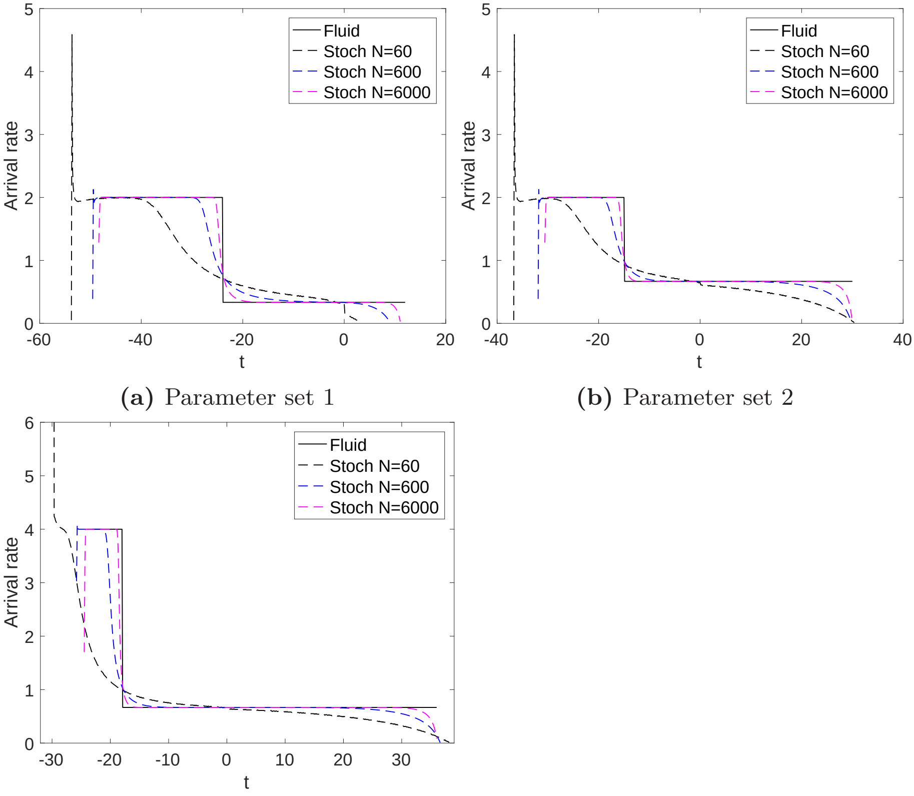
$$t_0 = t'_0 - (N - N')/s. \tag{5.19}$$

The corresponding equilibrium costs can be computed by Equation (5.17). We adjust the starting point until $|N - N'| < \epsilon$, for ϵ small. The entire numerical procedure is summarized in pseudo code in Algorithm 1 below.

Algorithm 1 : Procedure to obtain stochastic equilibrium

```
Inputs:
    $N, s, t^*, \alpha, \beta, \gamma, \epsilon, x$
Initialize:
    $r_1, r_2, t_q, t_{q'}, t_n, c_f$ from (5.3) and (5.4)
    $t_0 \leftarrow t_q$
    $c_s \leftarrow c_f$
    $\lambda(t_0) \leftarrow r_1$
    $N' = 0$
while $|N' - N| > \epsilon$ do
    $t \leftarrow t_0$
    while $\lambda(t) > 0$ do
        $t \leftarrow t + \Delta$
        $\lambda(t) \leftarrow \lambda(t - \Delta)$
        while $|\mathbb{E}[C(t, \lambda(t) - c_s)]| > \epsilon)$ do
            if $\mathbb{E}[C(t, \lambda(t) - c_s)] > 0$ then
                $\lambda(t) \leftarrow \lambda(t) - x$
            else $\mathbb{E}[C(t, \lambda(t) - c_s)] < 0$
                $\lambda(t) \leftarrow \lambda(t) + x$
            end if
            obtain $\mathbb{E}C[(t, \lambda)]$ from (5.3.1)
        end while
    end while
    $N' \leftarrow \Delta \sum_{t=t_0}^{t_1} \lambda(t)$
    $t_0 \leftarrow t_0 - (N - N')/s$
    $c_s \leftarrow \beta(t^* - t_0 + \frac{1}{s}) + \frac{\alpha}{s}$
end while
$t_1 \leftarrow t$
```

Using Algorithm 1, we can approximate the stochastic equilibrium arrival rate. We plot this in Figure 5.4 for parameter set 1 from Table 5.1 and for various values of N and s, keeping N/s constant. The corresponding Vickrey equilibrium is shown for comparison. From Figure 5.4a we observe that instead of a sudden transition between the high and low arrival rate, the stochastic equilibrium shows a gradual decrease. The smaller the total number of travellers N, the smoother this gradual decrease becomes. For each of the parameter sets in Figure 5.4 we



(a) Parameter set 1

(b) Parameter set 2

(c) Parameter set 3

Figure 5.4. Comparison of the equilibrium arrival rate for the deterministic model and the stochastic model for increasing N, for the parameter sets of Table 5.1.

observe an arrival rate peak at the start of the bottleneck period for small values of N. Although we are not certain, this peak could be due to the reduced waiting time uncertainty as there are no arrivals beforehand, making this point in time slightly more attractive compared to time instant later. This effect fades when the number of travellers during the bottleneck period becomes large.

In Table 5.2 the start and end time relative to the corresponding Vickrey equilibrium is shown, as well as the duration of the stochastic equilibrium. Depending on the cost parameters, the stochastic bottleneck period can be larger or smaller than the standard bottleneck period, which is 60 for all three parameter sets. The last column shows the relative increase in

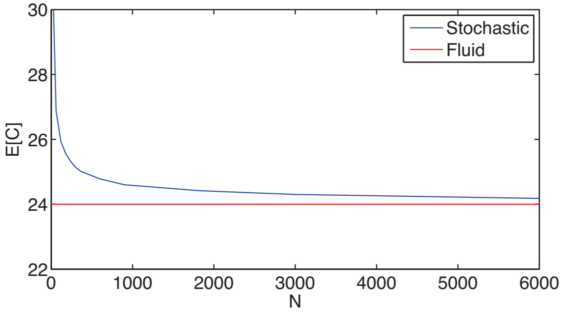


Figure 5.5. Expected cost for increasing the scaled number of travellers N for parameter set 1.

costs for the stochastic equilibrium $c_s(N)$ in comparison to the Vickrey equilibrium c_f. We visualised this in Figure 5.5 for parameter set 1 to show the impact for small values of N, we see that $c_s(N) \to c_f$ for $N \to \infty$, as expected. We see empirically that the computation time for the stochastic equilibrium grows linearly in N, so there is a trade-off between the computational effort and the benefits from the stochastic equilibrium when N becomes large.

We can use the stochastic equilibrium to investigate the uncertainty over time by plotting the mean waiting time against its standard deviation, see Figure 5.6. We observe that both the mean and standard deviation of the waiting time increase until peak congestion is reached, after which the waiting time decreases but the standard deviation keeps growing. Eventually, the standard deviation also decreases as the bottleneck disappears. This suggests that uncertainty at the end of the bottleneck period has a larger impact than at the beginning. Similar results were shown by Fosgerau [45] known as the *counter-clockwise looping phenomenon* also observed in empirical data [23].

Another interesting observation is that the duration of the bottleneck period deviates when we alter costs for waiting, while keeping the costs for early and late arrival fixed. This effect is visualised in Figure 5.7. As mentioned in Arnott [5], the costs for the Vickrey equilibrium do not depend on the costs for waiting α. Since the last and first traveller only experience schedule delay and no waiting time delay, the start and end of the rush hour is independent of α. This can also be seen in Equation (5.5)

	N	$[t_0, t_1] - t_q$	$t_1 - t_0$	$\frac{\mathbb{E}[C]}{c_f}$
Set 1	60	[-5.76 , 51.6]	57.4	12%
	300	[-2.16 , 56.2]	58.3	5%
	600	[-1.44 , 57.4]	58.8	3%
	3000	[-0.48 , 58.8]	59.3	1%
	6000	[-0.24 , 59.3]	59.5	1%
Set 2	60	[-6.72 , 60.7]	67.4	22%
	300	[-2.88 , 60.2]	63.1	10%
	600	[-1.92 , 60.0]	61.9	6%
	3000	[-0.72 , 60.0]	60.7	2%
	6000	[-0.48 , 60.0]	60.5	2%
Set 3	60	[-5.76 , 62.4]	68.2	24%
	300	[-2.40, 61.0]	63.4	10%
	600	[-1.68 , 60.7]	62.4	7%
	3000	[-0.72 , 60.5]	61.2	3%
	6000	[-0.48 , 60.2]	60.7	2%

Table 5.2. Comparing the stochastic and Vickrey equilibrium.

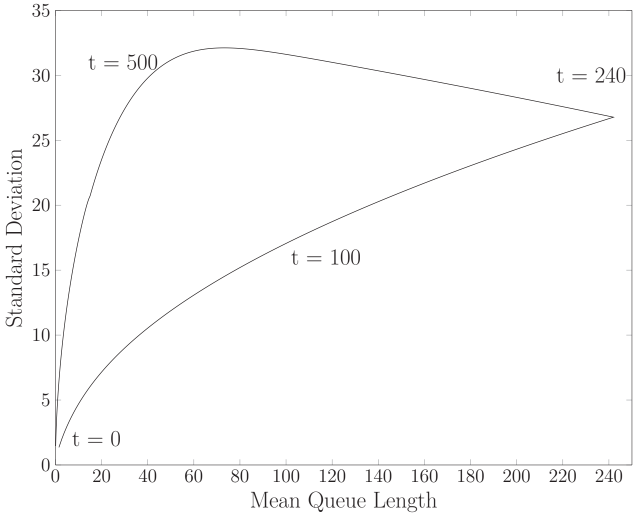


Figure 5.6. Mean waiting time against its standard deviation over time.

which shows that t_q,$t_{q'}$ and the total costs of travel time are independent

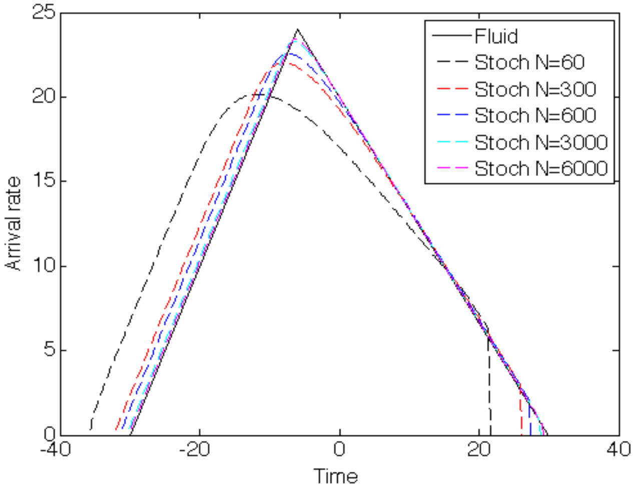


Figure 5.7. Waiting time for parameter set 1 and various scaled values of N.

of α. In the deterministic bottleneck model, the equilibrium costs at the start and the end of the bottleneck period are only comprised of costs for early and late arrival. However, in the stochastic bottleneck model, the queue does not necessarily disappear at the end of the bottleneck period. Moreover, a high costs relative to the early/late costs causes a smaller queue size. The uncertainty in waiting time causes travellers to arrive earlier when the queue is small, causing an earlier start of the bottleneck period, thereby showing the impact of variations in waiting time costs.

5.3.3 Closed-form expression for the equilibrium

The results of Section 5.3.2 show the impact of uncertainty over the bottleneck period. Our numerical procedure for computing the stochastic equilibrium provide useful insights into its behaviour, but it lacks the qualitative insights of analytic expressions. In this section, we derive a closed-form expression for the stochastic equilibrium.

In Figure 5.4 we observed that the arrival rate of the stochastic equilibrium shows a gradual decrease between the rate at the beginning and at the end of the bottleneck period compared to the instantaneous drop observed in the deterministic model. We propose to approximate this gradual decrease by a Sigmoid function. These functions are used in a wide range of fields, for instance in machine learning, biology, and economics [77].

We use a special case of the *Sigmoid function* known as the *generalised logistic function*, which was originally developed as a function to model animal growth [119]. In particular, we choose the following functional form:

$$f(t) = A + \frac{K - A}{(1 + \nu e^{-B(t-M)})^{1/\nu}}, \quad t \in [t_0, t_1]. \tag{5.20}$$

where A and K are the lower and upper asymptotes respectively, B is the growth rate, $\nu > 0$ represents the symmetry parameter and M defines the point of inflection. We are interested in the period $[t_0, t_1]$, which indicates the start and end of the bottleneck period.

To choose the correct parameter values, we draw inspiration from the numerical approximation of the stochastic equilibrium. Figure 5.4 shows that the values of the lower and upper asymptotes of the stochastic equilibrium correspond to the lower and upper rate of the Vickrey equilibrium, respectively, and we choose

$$K = r_1 \text{ and } A = r_2. \tag{5.21}$$

Furthermore, we observe that the inflection point roughly coincides with time t_n of the Vickrey equilibrium, shifted by the difference in starting points of the Vickrey equilibrium and the stochastic equilibrium $t_q - t_0$. This is because the waiting time also starts to decrease at this point. Therefore, we set

$$M = t_n - (t_q - t_0). \tag{5.22}$$

The symmetry parameter ν can be related to the fraction of time the Vickrey equilibrium prescribes the rate $(t_n - t_q)/(t_{q'} - t_q)$ multiplied by the difference in processing rate of the Vickrey equilibrium. This gives us

$$\nu = \frac{(r_1 - r_2)}{s} \frac{t_{q'} - t_q}{t_n - t_q}. \tag{5.23}$$

In contrast to the other parameters, the growth parameter B cannot be readily estimated by relating it to the Vickrey equilibrium, and instead we use nonlinear regression. To this end, first observe from Figure 5.4 that the steepest descent of the stochastic equilibrium is at the inflection point M, and its derivative, denoted by k, is equal to

$$\begin{aligned} k := \frac{\mathrm{d}}{\mathrm{d}t} f(t)\Big|_{t=M} &= B(K-A)(1+\nu e^{-B(t-M)})^{\frac{-1-\nu}{\nu}} e^{-B(t-M)}\Big|_{t=M} \\ &= B(K-A)(1+\nu)^{\frac{-1-\nu}{\nu}}. \end{aligned}$$

The values of K, A and ν can be obtained from Equations (5.21) and (5.23), so once we determine k we can compute B as

$$B = \frac{-k(1+\nu)^{\frac{-1-\nu}{\nu}}}{K-A}. \tag{5.24}$$

From numerical results, we can see that k depends on a combination of the cost parameters α, β and γ, the number of travellers N, and the rate of service s. However, this becomes a very complicated function. Therefore, in our regression model, we estimate the growth rate B for only a few parameters for the most general form, which is the standard cost value of set 1 from Table 5.1. We keep the ratio N/s fixed. Then we adjust the values of N and s by taking multiples of 60 for N, ranging from $N \in [60, 3000]$ and $s \in [1, 50]$. For simplicity, we divide N by 60 in our regression function. Additionally, we vary $\alpha \in [\beta, \gamma]$ and use the waiting costs expressed as

$$\alpha_{perc} = \frac{\alpha - \beta}{\gamma - \beta}. \tag{5.25}$$

The resulting values show a linear dependency when a log scaling is applied. To fit our linear regression model, we thus have to solve

$$\log k = a_0 + a_1 log(N/60) + a_2 log(\alpha). \tag{5.26}$$

We use a least squares non-linear regression and obtain the following values for the coefficients $a_0 = -0.0093 \approx 0$, $a_1 = -1.1896$ and $a_2 = 1.4242 \approx \sqrt{2}$ and with relative residual 0.053. Thus, we estimate k by

$$\hat{k} = -\alpha_{perc}^{-1.1896}(N/60)^{\sqrt{2}}. \tag{5.27}$$

In Figure 5.8 we compare the estimated slope $\hat{k}$ with the actual slope k. This slope is computed based on our numerical approximation of the

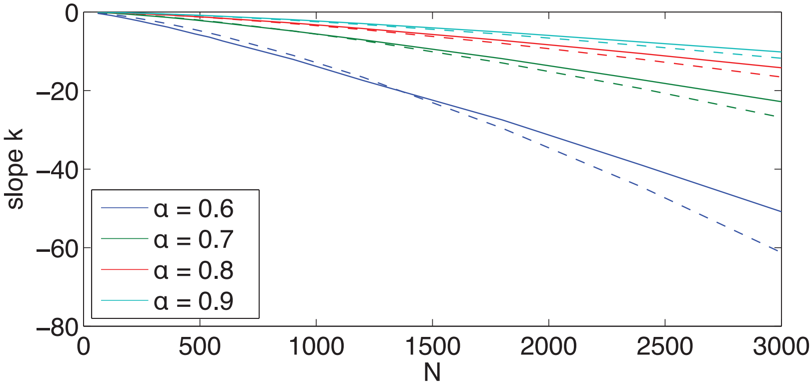


Figure 5.8. Comparison of the estimator $\hat{k}$ (dashed) against the real values k (solid).

stochastic equilibrium. Figure 5.8 shows that $\hat{k}$ is a remarkably accurate estimate for k, in particular for small values of N, which is the most relevant regime. Note that the slope estimator $\hat{k}$ is decreasing in N as expected, since as N grows large, the stochastic model approaches the deterministic model, where the equilibrium has an instantaneous transition from r_1 to r_2.

By substituting $\hat{k}$ from (5.27) into (5.24), along with our estimates for K and A from (5.21) and ν from (5.23), we obtain an approximation for B.

Having determined all parameters for our approximation of (5.20), it remains to find the correct time interval $[t_0, t_1]$ during which arrivals occur in the stochastic equilibrium. To this end, we exploit the fact that the expected number of arrivals during the bottleneck duration must add up to N, and that the expected cost throughout must be equal. For simplicity, we do this assuming that $\nu = 1$, and use this result for general ν. Numerically, we find that this approximation works well.

Define

$$F(t) := \int f(t)\mathrm{d}t\Big|_{\nu=1} = tA + \frac{(K-A)}{B}\log\left(1 + e^{B(t-M)}\right). \quad (5.28)$$

Then the fact that the expected number of arriving travellers must equal

N can be written as

$$F(t_1) - F(t_0) = N. \tag{5.29}$$

Since the expected cost in equilibrium $\mathbb{E}[C(t, \lambda)]$ must be the same throughout the bottleneck duration $t \in [t_0, t_1]$ we have that

$$\mathbb{E}[C(t_0, \lambda)] = \mathbb{E}[C(t_n - t_q + t_0, \lambda)]. \tag{5.30}$$

We can approximate the costs at these two time instances t_0 and $t_n - t_q + t_0$ as follows: Travellers arriving at time t_0 would be the first to enter the system, so its expected sojourn time would be $\frac{1}{s}$ (its own service time), while it would arrive early by an amount of time $\mathbb{E}[t^* - t_0 - X_1]$, where $X_1 \sim \exp(s)$ represents the service time of the traveller. The cost for being late are negligible to the first arrival, so by replacing X_1 by its expectation we can approximate the costs for an arrival at time t_0 as

$$\mathbb{E}[C(t_0, \lambda)] \approx \frac{\alpha}{s} + \beta\left(t^* - t_0 + \frac{1}{s}\right). \tag{5.31}$$

In the deterministic bottleneck model, the travellers arriving at time t_n depart from the bottleneck at exactly time t^*, so they only incur waiting costs. The starting point of the stochastic equilibrium is shifted by $t_0 - t_q$ compared to that of the Vickrey equilibrium, so the costs for travellers arriving at time $t_n + t_0 - t_q$ is dominated by the waiting time, and we approximate

$$\mathbb{E}[C(t_n - t_q + t_0, \lambda)] \approx \alpha W(t_n - t_q + t_0). \tag{5.32}$$

In order to approximate the sojourn time at time $t_n - t_q + t_0$ we use that the expected number of arrivals is equal to $F(t_n - t_q + t_0) - F(t_0)$, while the expected service up to that time is $(t_n - t_q)s$. If we also include the service of the traveller itself, we obtain

$$\mathbb{E}[C(t_n - t_q + t_0, \lambda)] \approx \frac{\alpha}{s}(F(t_n - t_q + t_0) - F(t_0) + 1 - (t_n - t_q)s). \tag{5.33}$$

The start and end times of the bottleneck period of the stochastic equilibrium can be obtained by numerically solving t_0 and t_1 from Equations (5.29) and (5.33).

We plot our approximation in Figure 5.9. One can see that the closed form approximation closely follows the stochastic model, and that it does a better job at equalizing the costs among the travellers compared to the Vickrey equilibrium.

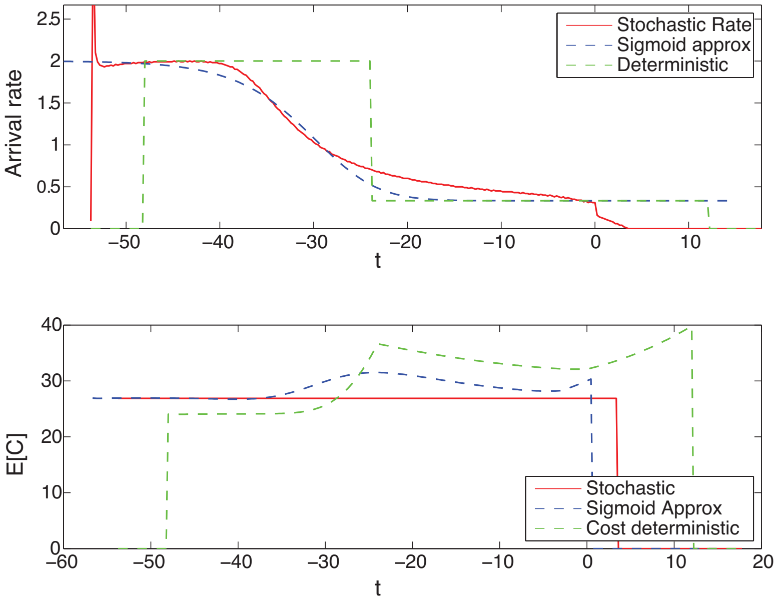


Figure 5.9. Comparing the Vickrey equilibrium with the analytic and numerical approximation of the stochastic equilibrium by using the parameter values of set 1 from Table 5.1 for $N = 60$.

5.4 Conclusion

In this chapter, we presented a general model for predicting the strategic user response to a bottleneck in road traffic. We first reviewed the existing models and results, which rely mostly on deterministic fluid models and equilibria. To allow for more realism we proposed to extend these models by considering travellers as discrete entities, which may be subject to randomness. More specifically, the strategic behaviour of travellers was captured in the Poisson arrival process with time-varying rate. This setting can be motivated when the population of potential travellers is large, each having a small probability of actually travelling at any particular time.

We presented a numerical procedure to compute the equilibrium in this stochastic model, and used these results to find a closed-form approximation for this stochastic equilibrium. Numerically, we showed that this approximation closely follows the equilibrium.

The stochastic bottleneck model gives new insights into the effects of strategic arrival behaviour in response to travel times uncertainty. Our approach can be applied to the many extensions that exist of the standard deterministic bottleneck model giving intuition of the impact of uncertainty in a broad range of transportation models. Examples include heterogeneity among travellers' departure time, interpretation of early and late arrival, and demand elasticity.

In the next chapter, we will consider an alternative extension of this model which captures uncertainty in the choice of travel times of individual travellers.

Chapter 6

Modelling of Arrival Time Uncertainty at a Bottleneck

We investigate the impact of random deviations in planned arrival times on the user equilibrium in an extension of Vickrey's celebrated bottleneck model [138]. In comparison to Chapter 5, where we study the uncertainty in demand and capacity at the bottleneck, the focus of this chapter is on the departure time uncertainty of each individual traveller. The model is motivated by the fact that in real life, users can not exactly plan the time at which they depart from home, or the delay they experience before they join the congestion bottleneck under investigation.

We show that the arrival density advocated by the Nash equilibrium in Vickrey's model is not a user equilibrium in the model with random uncertainty. We then investigate the existence of a user equilibrium for the latter and show that in general such an equilibrium can neither be a pure Nash equilibrium, nor a mixed equilibrium with a continuous density. With numerical examples we illustrate the mechanics that prevent existence of such an user equilibrium. Our results demonstrate that when random distortions influence user decisions, the dynamics of standard bottleneck models are inadequate to describe such more complex situations[1].

[1] This chapter is based on [**S4**].

6.1 Introduction

In the standard bottleneck traffic models, the inclusion of uncertainty in the exact departure moment is not taken into account. More specifically, people that depart from home have an intended time at which they want to depart, but their actual departure time will deviate from day to day. In this chapter we investigate the effects of these deviations on the formation of a single bottleneck. The formation of this bottleneck results from the decision that travellers make based on their valuation of travel time including the uncertainty. Our model is an extension of the Vickrey bottleneck model for which a literature review is given in Section 5.1 of Chapter 5.

Beyond the transportation literature, the response of travellers based on common preferences has been studied for a wide variety of applications that are closely related to the Vickrey model. These models use queuing theory in combination with game theory. The first model which uses a queueing approach was developed by Glazer and Hassin [55]. They consider a game where a population with a Poisson distributed size chooses an arrival time, and where service times at the queue are exponentially distributed. Many extensions have been studied with a broad range of application, such as a concert arrival game of Juneja et al. [70], at which the tardiness was added to the model, causing the order of arrivals to become relevant. Another application is presented in the meeting game of Fosgerau et al. [46], who studies the response of users with uncertainty in their arrival time for a meeting, and lastly, synchronisation under uncertainty by Ostrovsky [113], who studies the optimal strategy of individuals that incur a cost for waiting until the last arrival occurred. In these models the stochastic nature of responses is implicitly included.

In this chapter, we extend the bottleneck model by assuming that arrivals are not perfectly arriving at the planned time instants. Alternatively, we consider a system where people choose a time of arrival, but the actual time of arrival deviates by some predefined probability distribution. This uncertainty results in having a non-convex cost function. In [151], uncertainty on the road to the bottleneck is considered, which is similar to our case. However, they include the travel time to the bottleneck in the cost function, whereas we are interested in the stochasticity effects in the departure time from home primarily, and therefore, do not include this in the delay costs.

The goal of our analysis is to gain insight into the effects of uncertainty in the responses of travellers in equilibrium and the resulting queuing behaviour at the bottleneck. In Section 6.2 we outline the details of the extension of the bottleneck model. We analyse the impact for various scenarios with respect to the cost function and the arrival time uncertainty in Section 6.3. We then continue to investigate whether an equilibrium exists in our model in Section 6.4. We thereby study both a pure and a mixed equilibrium strategy. We conclude this chapter in Section 6.5.

6.2 Model description

In this section, we describe the classical bottleneck bottleneck model including the extension with uncertainty in the individual arrival times of travellers at the bottleneck.

6.2.1 Standard bottleneck model

The standard bottleneck model was presented in Section 5.2. To reduce repetitions, we refer the reader to this section for a detailed explanation and summarise the main formula's in this section.

The classical bottleneck model is a fluid model where a population of N identical travellers passes through a single bottleneck of capacity μ defined as

$$\int_{t_a}^{t_b} a(t)dt = N, \tag{6.1}$$

where $[t_a, t_b]$ denotes the timeframe in which these arrivals occur.

It is assumed that each traveller wants to exit the bottleneck at time t^*, and incurs a penalty for deviations from this preference time and schedule delay. This penalty is captured by a linear cost function with coefficients α, β, γ, for waiting, early and late arrival respectively. The time dependent cost function is represented as follows:

$$c(t) = \alpha w(t) + \beta(t^* - t + w(t))^+ + \gamma(t + w(t) - t^*)^+, \tag{6.2}$$

where $w(t)$ is the waiting time within the bottleneck for an arrival at time t.

Each traveller strategically decides when to arrive at the bottleneck in order to minimise his cost. This results in the following equilibrium arrival rate

$$a(t) = \begin{cases} r_1(t - t_q) & t \in [t_q, t_n) \\ r_1(t_n - t_q) + r_2(t - t_n) & t \in [t_n, t_{q'}] \end{cases}, \tag{6.3}$$

where

$$r_1 = \mu + \frac{\beta\mu}{\alpha - \beta}, \quad r_2 = \mu - \frac{\gamma\mu}{\alpha + \gamma}. \tag{6.4}$$

The costs at the start, peak, and end time of the bottleneck period is computed by

$$t_a = t^* - \frac{\eta N/\mu}{1+\eta}, \quad t_b = t^* + \frac{N/\mu}{1+\eta}, \quad t_n = t^* - \frac{\delta N/\mu}{\alpha},$$

with $\eta = \frac{\gamma}{\beta}$ and $\delta = \frac{\beta\gamma}{\beta+\gamma}$. This arrival curve gives all travellers equal cost of

$$c = \delta\frac{N}{\mu}. \tag{6.5}$$

6.2.2 Bottleneck model with arrival time uncertainty

In reality, travellers do not necessarily arrive at their intended time. We, therefore, extend the above bottleneck model with an uncertainty function that acts as a smoothing kernel over the arrival function $a(t)$. Here, $a(t)$ is assumed to be continuous. The deviation from the intended arrival time of each traveller is modelled by a continuous random variable X, assuming the deviations of different users to be independent. The smoothing kernel $f(u)$ corresponds to the probability density function of X for the arrival deviation u of an arbitrary traveller. The resulting arrival rate function is given by

$$\tilde{a}(t) = \int_{u=-\infty}^{t} f(u)a(t-u)du. \tag{6.6}$$

Ultimately, we are interested in the time-dependent queue length at the bottleneck which can be computed by the difference between the actual arrival rate of (6.6) and the departure rate μ:

$$q(t) = \begin{cases} \tilde{a}(t) - \mu, & Q(t) > 0 \\ (\tilde{a}(t) - \mu)^{+} & Q(t) = 0 \end{cases}. \tag{6.7}$$

The waiting time can then be computed by

$$W(t) = \int_{u=-\infty}^{t} \frac{q(u)}{\mu} du, \tag{6.8}$$

where we use that $q(0) = 0$.

By plugging (6.8) into (6.2) we compute the expected costs of an arrival at time t,

$$\tilde{C}(t) = \alpha W(t) + \beta \left(t^{*} - (t + W(t))\right)^{+} + \gamma \left(t + W(t) - t^{*}\right)^{+}. \tag{6.9}$$

Given a time-dependent arrival rate $a(t)$ we can compute the expected cost for a traveller that has an intended arrival time t by

$$\mathbb{E}[C(t)] = \int_{u=-\infty}^{\infty} \tilde{C}(t+u) f(u) du. \tag{6.10}$$

6.3 Preliminary analysis

To gain insight in the impact of uncertainty in the exact arrival time of a traveller, we analyse the costs over time given that travellers are unaware of this uncertainty aspect. We compute the impact of uncertainty for a number of delay functions. For each, we show the impact for an increasing level of uncertainty.

To obtain the arrival rate over time at which travellers are unaware of each others and their own uncertainty function, we compute the NE arrival rate as computed in the standard bottleneck model [138]. Thus, the actual arrival rate $\tilde{a}(t)$ is computed by the convolution of the NE arrival rate of Equation (6.4), and the arrival uncertainty distribution of X. The actual arrival rate can be obtained by taking the convolution as defined in Equation (6.6). We then plug this rate into (6.9) to obtain the expected costs over time. Finally, the expected costs for a traveller that chooses time t is calculated by (6.10). Numerical evaluation of these formulas requires a discretisation scheme, as these convolutions do not necessarily provide an explicit solution.

We discretise the interval of the bottleneck period into n small segments of length Δ where:

$$n = \left\lfloor \frac{t_{end} - t_{start}}{\Delta} \right\rfloor . \tag{6.11}$$

The probability mass of the k^{th} segment is obtained by

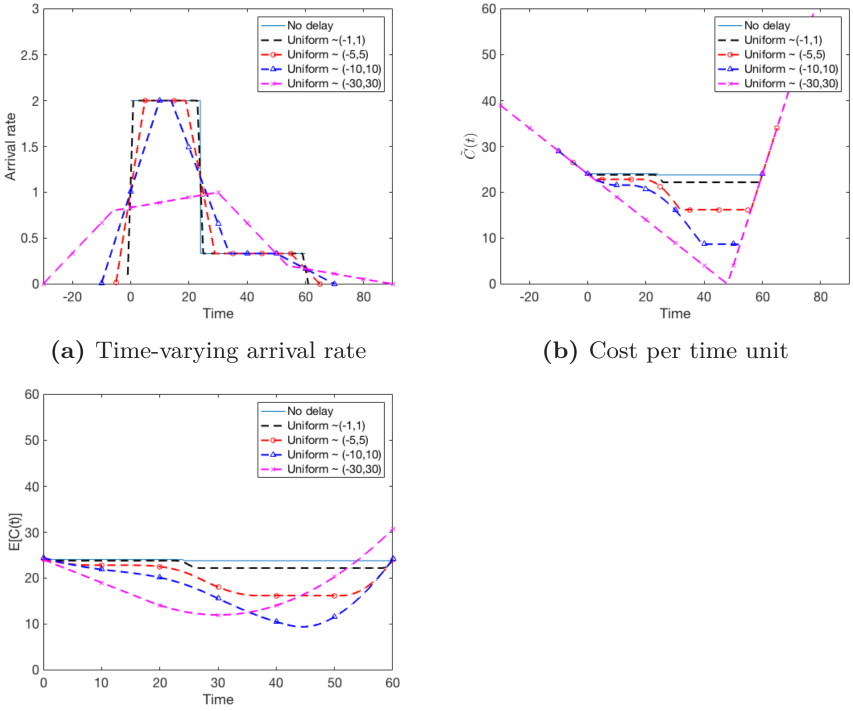
$$p_k = \mathbb{P}[X \leq (k+1)\Delta + t] - \mathbb{P}[X \leq k\Delta + t]. \tag{6.12}$$

First, we take a uniformly distributed arrival uncertainty X. The cost function is taken equal to the standard values where $\beta/\alpha = 0.5$ and $\gamma/\alpha = 2$ [130]($\alpha = 1, \beta = 0.5, \gamma = 2,\ N = 60, s = 1$). In Figure 6.1 the results for $X \sim \text{unif}(\sigma, \tau)$ when $\tau \in \{0, 1, 5, 10, 30\}$ and $\sigma = -\tau$ are visualised. In these examples, the bottleneck period is extended to $t_{start} = t_a + \sigma$ and $t_{end} = t_b + \tau$, since deviations from the intended arrival times will cause users to arrive prior to t_a and later than t_b as well. The results of Figure 6.1c show that the expected cost is below that of the cost without any delay, as long as a queue exists. In the last parameter choice, for which the delay is equal to the period of the bottleneck, there will be no queue at all: travellers will only incur earliness or lateness cost, depending on their arrival time.

In Figure 6.2, the results for $X \sim exp(\mu)$ are shown. This probability density function focuses on the largest arrival volume at the beginning, and the volume quickly shrinks, contrary to the uniform distribution, in which case the volume is equally spread. For both functions the same observations are shown, decreasing in the average cost function for travellers while increasing delay parameter.

In conclusion, travellers will deviate from the standard Vickrey equilibrium arrivals because they can reduce their cost. With the same approach, other delay distributions can be applied as well.

We continue the analysis in the next section, by exploring approaches to obtain an equilibrium accounting for the random delays under the general assumption that everything is known, including the arrival uncertainty distribution.



(a) Time-varying arrival rate

(b) Cost per time unit

(c) Expected cost for a traveller

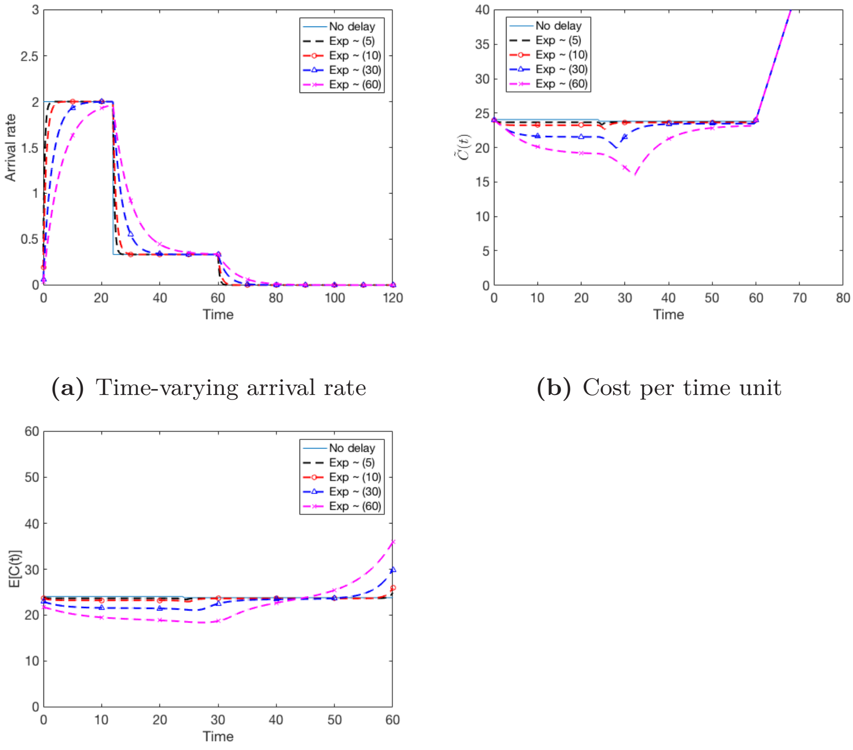
Figure 6.1. Impact of the arrival pattern and costs over time for a uniformly distributed delay.

6.4 Optimal responses

We continue our analysis by investigating whether an equilibrium arrival function exists, and if so, under which conditions. To this end, we explore both the options of a pure equilibrium and a mixed strategy equilibrium. Both cases are investigated assuming a uniform delay function.

6.4.1 Uniform delay function

To explicitly study whether a pure or mixed equilibrium exists, we represent the departure delay by a uniformly distributed random variable



(a) Time-varying arrival rate

(b) Cost per time unit

(c) Expected cost for a traveller

Figure 6.2. Impact of the arrival pattern and costs over time for an exponentially distributed delay.

$X \sim \text{unif}(0, 1)$:

$$f(t) = \begin{cases} 1 & t \in [0, 1] \\ 0 & \text{otherwise} \end{cases}. \tag{6.13}$$

Note that the results for general $X \sim \text{unif}(0, \tau)$ can simply be obtained by re-scaling time. In this section, we outline the simplifications in the model description that are due to the uniform delay assumption.

Plugging this into Equation (6.10) we obtain

$$\mathbb{E}C(t) = \int_{u=0}^{1} \tilde{C}(t+u)du.$$

To investigate whether a mixed strategy or a pure strategy exists for this delay function, we have to show that

$$\frac{d\mathbb{E}C(t)}{dt} = \tilde{C}(t+1) - \tilde{C}(t) = 0, \tag{6.14}$$

which implies that

$$\tilde{C}(t+1) = \tilde{C}(t), \tag{6.15}$$

holds for every $t \in [t_{start}, t_{end}]$.

6.4.2 Pure equilibrium

Next, we analyse whether there are conditions for which a pure equilibrium exists, meaning that the minimum costs is obtained when all travellers intend to arrive at the same time instant. First, we derive the conditions that should hold explicitly. We thereby assume that the delay $f(\cdot)$ is uniformly distributed. Finally, we numerically supplement the explicit derivation to gain more insight in the model and allow for computation with other delay distributions.

Explicit derivation of the pure equilibrium

To determine whether a pure equilibrium exists, we calculate the cost for a tagged traveller arriving at time $s \in \mathbb{R}$, given that the other travellers N all have the same intended arrival time $t \in \mathbb{R}$. We separate between two cases. When $\mu \geq N$, the tagged traveller encounters no waiting time and the moment of arrival of traveller s does not depend on the volume N at time t. The case where $\mu < N$, s does lead to waiting times when the actual arrival time overlaps with the interval of arrival of the volume N. For simplicity we assume $N = 1$.

The case $\mu \geq N$

We determine the cost for a traveller arriving at s by

$$\begin{aligned}
C(s) &= \int_s^{t^*} \beta(t^* - u)du + \int_{t^*}^{s+1} \gamma(u - t^*)du \\
&= \beta \left[t^* u - 0.5u^2\right] \Big|_s^{t^*} + \gamma \left[\frac{u^2}{2} - t^* u\right] \Big|_{t^*}^{s+1} \\
&= \beta \left[t^*(t^* - s) - \frac{1}{2}\left((t^*)^2 - s^2\right)\right] \\
&\quad + \gamma \left[\frac{(s+1)^2}{2} - \frac{(t^*)^2}{2} - t^*(s+1)\right].
\end{aligned}$$

To find the time instant that gives the best response, we compute the solution of $\frac{dC}{ds} = 0$, which gives

$$\frac{dC}{ds} = -\beta t^* + s\beta + \gamma(s + 1 - t^*) = 0, \tag{6.16}$$

resulting in

$$s = t^* - \frac{\gamma}{\beta + \gamma} \tag{6.17}$$

In this case, the best response does not depend on the arrival of the fluid N due to the absence of a queue formation. This is the same solution as the model with no waiting costs by Glazer et al [56].

The case $\mu < N$

For the case $\mu < N$, a queue builds during the arrival interval of the arrival of the volume N. Therefore, we need to consider the time of arrival of this volume, which is given by $\tilde{a}_t(u) = 1 \in [t, t+1]$. Including this in Equation (6.7), we obtain the waiting time by

$$W_t(u) = \begin{cases} (\frac{1}{\mu} - 1)(u - t), & u \in [t, t+1] \\ \frac{1}{\mu} - (u - t), & u \in (t+1, t + \frac{1}{\mu}] \\ 0, & \text{otherwise} \end{cases},$$

where u represents the intended arrival time.

We insert the $W_t(u)$ in Equation (6.9), and compute the cost for a traveller that intends to arrive at time u given that the volume N

intends to arrive at time t by

$$\tilde{C}_t(u) = \alpha W_t(u) + \beta\big(t^* - (t + W_t(u))\big)^+ + \gamma\big(t + W_t(u) - t^*\big)^+.$$

Finally, we compute the expected cost for a traveller that intends to arrive at time s by

$$\mathbb{E}C_t(s) = \int_s^{s+1} \tilde{C}_t(u)du. \tag{6.18}$$

Proposition 6.1. *A pure Nash equilibrium, satisfying Equation* (6.15) *does not exist for* $N > \mu$.

To show that the above proposition is true, we start with the computation of the expected costs for a particle arriving at time s, given that the volume N intends to arrive at time t. Therefore, we split the integral of Equation (6.18) into several cases. We make a division between the case where $s \leq t$ and $s \geq t$, and separate between the point where the earliness costs changes to lateness costs denoted by $x^* = t + \mu(t^* - t)$. This gives

$$\mathbb{E}C_t(s) = \begin{cases} \overbrace{\int_s^t \beta(t^* - u)du}^{(1.1)} \\ + \overbrace{\int_t^{s+1} \alpha W_t(u) + \beta(t^* - u - W_t(u))du}^{(1.2)} & \text{for } s \leq t < s+1 \leq x^* \\ \\ \overbrace{\int_s^t \beta(t^* - u)du}^{(2.1)} + \overbrace{\int_t^{s+1} \alpha W_t(u)du}^{(2.2)} \\ + \overbrace{\int_t^{x^*} \beta(t^* - u - W_t(u))du}^{(2.3)} \\ + \overbrace{\int_{x^*}^{s+1} \gamma(u + W_t(u) - t^*)du}^{(2.4)} & \text{for } s \leq t < x^* < s+1 \leq t+1 \\ \\ \overbrace{\int_s^{t+1} \alpha W_t(u)du}^{(3.1)} + \overbrace{\int_s^{x^*} \beta(t^* - u - W_t(u))du}^{(3.2)} + \\ + \overbrace{\int_{x^*}^{t+1} \gamma(u + W_t(u) - t^*)du}^{(3.3)} \\ + \overbrace{\int_{t+1}^{s+1} \alpha W_t(u) + \gamma(u + W_t(u) - t^*)du}^{(3.4)} & \text{for } t \leq s < x^* < t+1 \leq s+1 \end{cases}. \tag{6.19}$$

Here, we excluded the cases where $s, t > x^*$, and also $s+1, t+1 < t^*$. In these intervals we will not find an optimal choice of s nor of t.

To find the time s for a given arrival of N at time t, we take the derivative of (6.19), which gives

$$\frac{d\mathbb{E}C}{ds} = \begin{cases} \frac{-\alpha(\mu-1)(1+s-t)+\beta(\mu-1-s+2\mu s+t+\mu t)}{\mu} & \text{for } s \leq t < s+1 \leq x^* \\ \frac{-\alpha(\mu-1)(1+s-t)+\beta\mu(s-t^*)+\gamma(1-t^*\mu+s-t+\mu t)}{\mu} & \text{for } s \leq t < x^* < s+1 \leq t+1\,. \\ \frac{-\alpha(\mu-1+s-t)+\beta(s-\mu t^*+(\mu-1)t)+\gamma(1-t^*\mu+\mu t)}{\mu} & \text{for } t \leq s < x^* < t+1 \leq s+1 \end{cases}$$

Next, solving $\frac{d\mathbb{E}C}{ds} = 0$, we obtain

$$s = \begin{cases} \frac{\alpha(\mu-1+t-\mu t)+\beta(1-\mu-t+\mu t)}{\alpha(1-\mu)-\beta(2\mu-1)} & \text{for } s \leq t < s+1 \leq x^* \\ \frac{\alpha(\mu+t-\mu t-1)+\beta\mu t^*+\gamma(\mu t^*+t-\mu t-1)}{\alpha+\gamma+\mu(\beta-\alpha)} & \text{for } s \leq t < x^* < s+1 \leq t+1\,. \\ \frac{\alpha(1-\mu+t)+\beta(\mu t-t-\mu t^*)+\gamma(1-\mu t^*+\mu t)}{\alpha-\beta} & \text{for } t \leq s < x^* < t+1 \leq s+1 \end{cases} \tag{6.20}$$

We distinguish between the cases of (6.20) separately.

Case 1: $s \leq t < s+1 \leq x^*$
Which gives

$$s = \frac{\alpha(\mu-1+t-\mu t)+\beta(1-\mu-t+\mu t)}{\alpha(1-\mu)-\beta(2\mu-1)},$$

then for $s = t = t_e$ we obtain

$$t_e = \frac{\alpha}{\beta}\left(1-\frac{1}{\mu}\right)+\frac{1}{\mu}-1, \tag{6.21}$$

which will give a negative value for any $\mu < 1$ and $\alpha > \beta$.

Case 2: $s \leq t < x^* < s+1 \leq t+1$
Which gives

$$s = \frac{\alpha(\mu+t-\mu t-1)+\beta\mu t^*+\gamma(\mu t^*+t-\mu t-1)}{\alpha+\gamma+\mu(\beta-\alpha)},$$

then, for $s = t = t_e$ we obtain

$$t_e = t^* - \frac{\gamma}{\beta+\gamma}\left(\frac{1}{\mu}\right) - \frac{\alpha}{\beta+\gamma}\left(\frac{1}{\mu}-1\right). \tag{6.22}$$

As t_e has to meet the criteria of $t_e \geq t^* - \frac{1}{\mu}$, we can only find a pure equilibrium when $\mu > \frac{\alpha-\beta}{\alpha}$.

Case 3: $t \leq s < x^* < t+1 \leq s+1$
Which gives

$$s = \frac{\alpha(1-\mu+t) + \beta(\mu t - t - \mu t^*) + \gamma(1 - \mu t^* + \mu t)}{\alpha - \beta},$$

then, for $s = t = t_e$ we obtain

$$t_e = t^* - \frac{\gamma}{\beta+\gamma}\left(\frac{1}{\mu}\right) - \frac{\alpha}{\beta+\gamma}\left(\frac{1}{\mu}-1\right), \tag{6.23}$$

which matches case 2.

The above suggests that a pure equilibrium solution can only exist at the interval of cases 2 and 3. However, we can show that the cost of the tagged traveller arriving at time s for $s = t = t_e$ of Equations (6.22) and (6.23) is not the global minimum, but only a local minimum. The global minimum can be found when $t \leq s \leq x^* \leq t+1 \leq t + \frac{1}{\mu} \leq s+1$. The cost of s for this case is given by

$$\begin{aligned}\mathbb{E}C(s) = &\int_s^{t+1} \alpha W(u)du + \int_s^{x^*} \beta(t^* - u - W(u))du \\ &+ \int_{x^*}^{t+1} \gamma(u + W(u) - t^*)du \\ &+ \int_{t+1}^{t+\frac{1}{\mu}} \alpha W(u) + \gamma(u + W(u) - t^*)du \\ &+ \int_{t+\frac{1}{\mu}}^{s+1} \gamma(u - t^*)du.\end{aligned}$$

To calculate the best response of s, we take the derivation of the above equation and make it equal to zero, which gives

$$\frac{d\mathbb{E}C}{ds} = \frac{-t^*(\beta+\gamma)\mu + s(\beta-\alpha) - \alpha\mu s + \gamma\mu(1+s) + t(\alpha t - \beta)\mu t(\beta - \alpha)}{\mu} = 0,$$

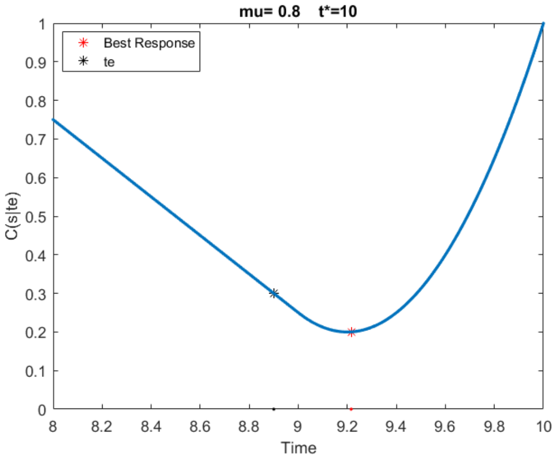


Figure 6.3. Best response time s of Equation (6.25) for a volume N intending to arrive at time t, where $t = t_e$ of Equation (6.23).

$$\tag{6.24}$$

then

$$s = \frac{t^*\mu(\beta+\gamma) - \gamma\mu + t(\alpha-\beta)(\mu-1)}{\alpha(\mu-1)+\beta+\gamma\mu}. \tag{6.25}$$

When we replace t by t_e in (6.22) or (6.23), we obtain a cost that is smaller than cases 2 and 3, as is shown in Figure 6.3. This shows that a pure equilibrium does not exists under these conditions.

Numerical analysis

To gain more insight into the location of the best response time s of a tagged traveller for a given intended arrival time t of volume N, we numerically compute the costs for s over the relevant range.

To numerically approximate Equation (6.18), we discretise both the intended t and the intended s into small steps as explained in Equation (6.11) and (6.12):

$$M = \frac{t_{end} - t_{start}}{\Delta}, \tag{6.26}$$

where Δ is the step size, M gives the number of subintervals and $[t_{start}, t_{end}]$ denotes the interval including the support of the delay function. For a delay function with support length of $[\tilde{v}, \tilde{w}]$, $v = \lfloor \frac{v}{\Delta} \rfloor$ and $w = \lceil \frac{w}{\Delta} \rceil$, we present the discretised version of the cost by

$$\tilde{C}_{i,j} \overset{\substack{t=i\Delta+t_{start},\\ s=j\Delta+t_{start}}}{=} \Delta \tilde{C}_t(s), \tag{6.27}$$

where i and j are the discretised values of s and t, respectively. In Algorithm 2, the computational scheme is described, where p_k denotes the arrival density of the k^{th} interval from the discretised delay function $f(t)$ as given by Equation (6.12).

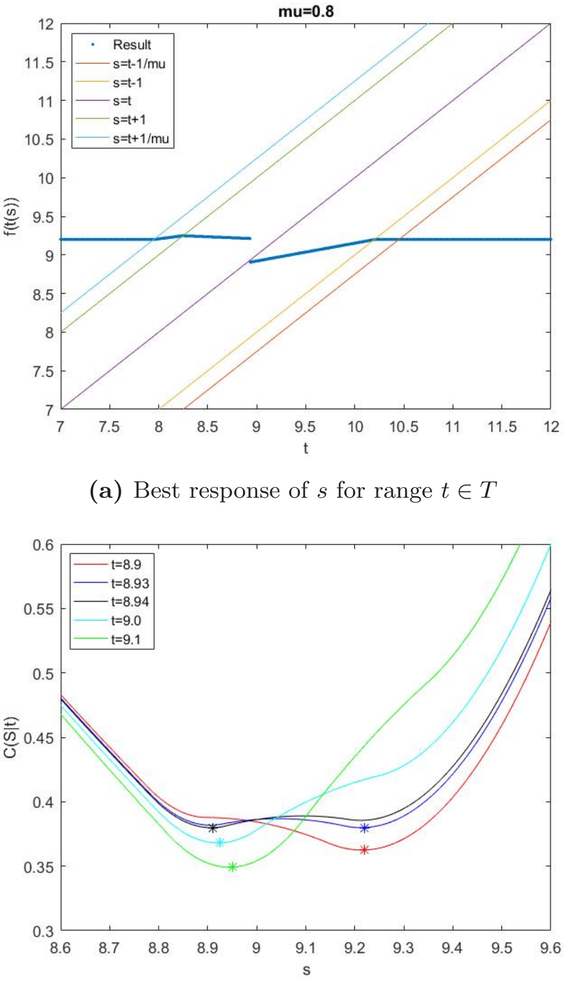
Algorithm 2 : Procedure to obtain best response s given t with delay function $f(t)$

```
Inputs:
    $M, v, w, \tilde{C}_{i,j}, p_k$
for $i = 0, \dots, M$ do
    for $j = 0, \dots, M$ do
        $C_{i,j} = \sum_{k=0}^{j+w-v} p_k \tilde{C}_{i,\max\{0,k+v\}}$
    end for
    $s_i = arg\min_j \{C_{i,j}\}$
end for
```

With the computational scheme of Algorithm 2 we can numerically approximate the result for the best response of s for each t. We use the standard cost function $\alpha = 1$, $\beta = 0.5$, and $\gamma = 2$ to show an example of the time-dependent cost function. The results in Figure 6.4a show the best response s for departure rate $\mu = 0.8$ when t arrives at the time t_e resulting from Equation (6.22) and some small value before and after t_e. In this figure, we see that a jump in the best response s occurs at $t \approx 8.9$. The cost function s for the intended arrival of volume N at time t with minimal costs is shown in Figure 6.4b, where we can see the occurrence of the jump between 8.93 and 8.94. Thus, we observe that for an intended arrival time $t > t_e$ of volume N the best response is within the range of the cases specified in Equation (6.19). As soon as $t = t_e$, s deviates to a point $s > t$ computed by Equation (6.25).

6.4.3 Continuous mixed equilibrium

We continue our analysis by using a mixed arrival strategy and determine whether this leads to an equilibrium. We formulate the conditions that



(a) Best response of s for range $t \in T$

(b) Costs s for specific t

Figure 6.4. Response for a large range of t (left) and in detail at the discontinuity point (right), for $\mu = 0.8$ and $N = 1$.

should hold for a mixed equilibrium for a uniform delay function. We explain the difficulty to obtain a mixed equilibrium for a continuous arrival rate for this example and motivate these by numerical approximation.

Explicit derivation of the continuous mixed equilibrium

To search for a mixed equilibrium strategy, we define an *arrival density profile* $g(t)$ and cumulative density function $G(t)$ for which everyone choses a preferred time t and the probability of arrival happens according to this density function. The delay function $f(t)$ corresponds to the probability density function of someone choosing preferred time t, as defined in Equation (6.6).

We continue the analysis to find out whether a mixed equilibrium exists explicitly, by assuming that the intended arrival time leads to an arrival intensity defined as $X \sim \text{unif}(0, 1)$, as described in Equation (6.13). The arrival rate function of Equation (6.6) simplifies to

$$\tilde{a}(t) = \int_{t_a}^{t} g(t-u)du = G(t-t_a),$$

where t_a denotes the start of the bottleneck period. The time dependent queue length of Equation (6.7) becomes

$$q(t) = \begin{cases} G(t-t_a) - \mu, & Q(t) > 0 \\ (G(t-t_a) - \mu)^+ & Q(t) = 0 \end{cases},$$

where $G(\cdot)$ is the cumulative density function of $g(\cdot)$. This allows us to determine the waiting time by

$$W(t) = \frac{1}{\mu} \int_{t_a}^{t} \Big(G(u-t_a) - \mu \Big) du,$$

where we assume that $q(s) > 0$ for $t_a \leq s \leq t$.

The expected cost for a traveller with intended arrival time t is defined as

$$\mathbb{E}C(t) = \int_{u=t}^{t+1} \alpha W(t) + \beta\big(t^* - (u + W(t))\big)^+ + \gamma\big(u + W(t) - t^*\big)^+.$$

Proposition 6.2. *A continuous mixed equilibrium does not exist.*

To determine whether a mixed equilibrium exists, we have to find $\tilde{a}(t)$, for which the equilibrium condition of Equation (6.15) holds. Suppose, we want to show that this condition is true at $t = t_b$, i.e., the last intended arrival moment. We need to show that $\tilde{C}(t_b) = \tilde{C}(t_b + 1)$. We assume that the delay function is uniformly distributed. This gives

$$\tilde{C}(t_b) = \alpha W(t_b) + \beta(t^* - (t_b + W(t_b))),$$

and hence,

$$\tilde{C}(t_b + 1) = \alpha W(t_b + 1) + \beta(t^* - (t_b + W(t_b + 1))). \tag{6.28}$$

In equilibrium these two equations must be equal. As we assume that $t_b \geq t^*$ we can see that $\tilde{C}(t_b + 1) > \tilde{C}(t_b)$, because $\gamma > \alpha$. This shows that Proposition 6.2 is true, and that a continuous mixed equilibrium does not exists.

Numerical analysis

We search for a numerical procedure to obtain an arrival rate function for which the expected costs per traveller remain constant on most of the support. This approximation visualises the locations in time for which we can not maintain constant costs, as we have proven for one of these in the previous section. We will not only consider the uniform delay function, but show that the same holds for other delay distributions.

Algorithm 3 : Procedure to approximate a mixed equilibrium.

```
Inputs:
    $M, t^*, \beta, C_{target}, v, w, p_j$
Initialize:
    $i_{\text{Loc}} = \frac{t^* - \beta C_{target} - t_{start}}{\Delta}$;
    $r_i = 0$ for $i = 0, \ldots, M$
while $i_{\text{Loc}} \neq \emptyset$ do
    $r_{i_{\text{Loc}}} = r_{i_{\text{Loc}}} + \epsilon$
    for $i = 0, \ldots M$ do
        $C_i = \sum_{j=0}^{i+w-v} p_j \tilde{C}_{j+v}$
    end for
    $i_{\text{Loc}} = arg \min_i \{i : C_i < C_{target}\}$
end while
```

We want to obtain an arrival rate $\tilde{a}(t)$, for which the $\mathbb{E}C(t) \approx c$. To obtain a numerical solution, we discretise the functions of Section 6.4.3 shown in Equation (6.11) and compute the probability distribution of arrival as computed in Equation (6.12). We use the same variables as in Algorithm 2, where the costs at time t is now computed by $C_i \overset{t=i\Delta+t_{start}}{=} \tilde{C}(t)$, C_{target} is equal to Equation (6.5), the arrival rate $\tilde{a}(t)$ is captured in the vector $\bar{r} = (r_1, \ldots, r_M)$, and ϵ is a small value

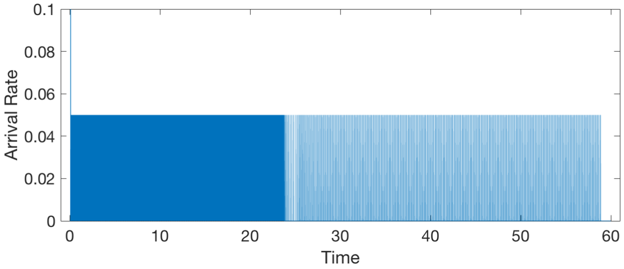
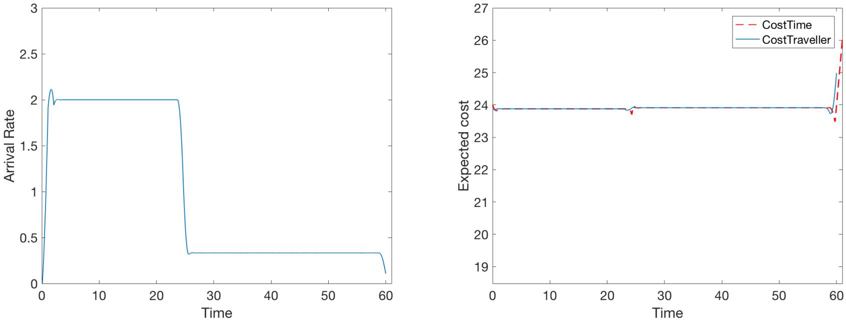


Figure 6.5. Result of Algorithm 3 giving the arrival rate intensities over time.



(a) Time-varying arrival rate

(b) Cost function

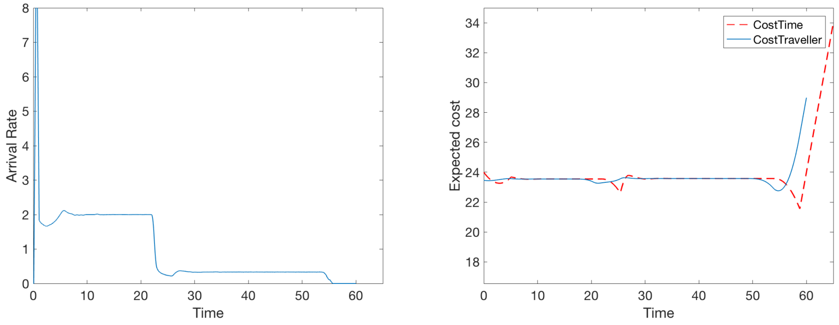
Figure 6.6. Approximated equilibrium function for $X \sim \text{unif}(0, 1)$.

with which we increase the rate at the indicated location.

Algorithm 3 shows the numerical procedure that results in an arrival rate function for which the costs over time remain relatively constant.

In summary, the procedure consists of the following steps. We first set a target costs denoted by C_{target}, which we want to keep constant. We search for the earliest moment of arrival such that this cost constraint is met. At this specific time instant we add a small arrival volume of rate ϵ. Given the updated arrival vector, we compute the new cost function over time. Again, we compute the earliest moment of arrival t such that $\mathbb{E}C(t) \leq C_{target}$. We continue this procedure until this condition can not be met anymore.

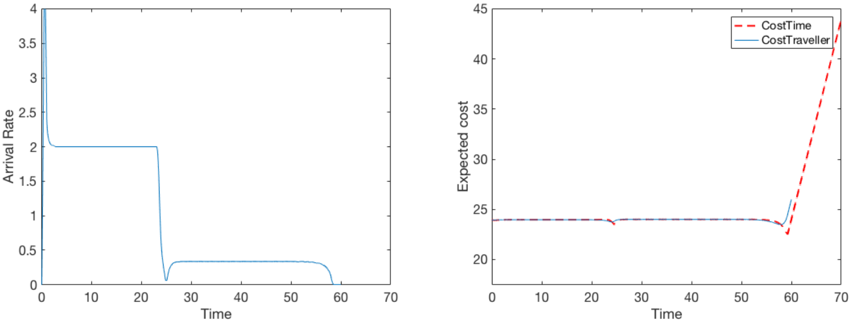
In Figure 6.5, a representation of the outcome of the approximation



(a) Time-varying arrival rate

(b) Cost function

Figure 6.7. Approximated equilibrium function for $X \sim \text{unif}(0, 10)$.



(a) Time-varying arrival rate

(b) Cost function

Figure 6.8. Approximated equilibrium function for $X \sim \exp(1)$.

procedure of Algorithm 3 is visualised. The line density indicates the arrival rate intensity over time. We observe a large density in the beginning, followed by a reduced density at the peak moment x^* ($t = 24$), which increases again shortly after. The arrival rate over a specified period of time is given by the sum of the lines. We apply a moving average filter to obtain the arrival rate function over time.

In Figures 6.6 and 6.7 the results of these rates and the costs over time are visualised for uniform delay function. These figures show that for a larger uncertainty, obtaining a constant cost function becomes more difficult. The results for an exponential density function for the delay is shown in Figure 6.8. We observe that it becomes harder to equalise costs across the bottleneck period with only a small average delay, compared to the uniform delay. In Table 6.1, we observe that the total amount of travellers passing the bottleneck decreases with respect to delay, while fixing the expected cost to the value of Equation (6.5). Conclusions on the impact of arrival time uncertainty with respect to dis-utility can not be made, as the current results are not in equilibrium. However, this does suggest that uncertainty increases the dis-utility of individual travellers.

	Uniform	Exponential
$\tau = 0$	60	60
$\tau = 1$	59.6	59.2
$\tau = 5$	58.7	56.5
$\tau = 10$	57.5	54.1

Table 6.1. Total arrival rate N for fixed costs with varying delay function $f(t)$ and mean delay τ.

6.5 Conclusion

In this chapter, we investigated the impact of uncertainty in arrival time in an extension of Vickrey's bottleneck model. Our model allows a random distortion of the intended arrival times of users at a congestion bottleneck of interest. Such a random distortion models the fact that the actual arrival time of users at a specific congestion point can not be completely controlled by the users. In reality, it is common that the departure times from the points of origin, and the delays incurred before reaching the specific bottleneck can only be estimated up to a certain confidence range.

We have shown that the equilibrium rate of the Vickrey model without distorted arrivals is, in general, not a good approximation for a possible equilibrium in the model with distortions. The equilibrium arrival rate of the standard bottleneck model does give important initial insight into the expected costs over time, in case travellers are unaware of their own and for each other's random delays. The results showed reduced costs for almost the entire bottleneck period, while for highly variable uncertainty distributions, the costs increase at the end of the bottleneck period. This is due to the probability of arrival after the end of the standard bottleneck period.

In fact, we have questioned the existence, in general, of a user equilibrium in the random setting, We showed that, if it exists, it can not be a pure Nash equilibrium, nor can it be a mixed equilibrium with a continuous density for the distortions. To shed light on the nontrivial dynamics in our model and gain intuition regarding the existence of an equilibrium we have numerically investigated the optimal responses of individual users in a range of model settings, including a variety of various delay distributions. Numerically, we showed that an iterative approach for the best responses to previously determined best responses results in a cyclic pattern. In the mixed equilibrium setting with a continuous arrival volume, we observe instabilities at the beginning, the end and the peak of the bottleneck period.

The question whether or not such an equilibrium exists in general remains unanswered in this chapter and is subject of ongoing research. There may exist a more elaborate equilibrium with several arrival volumes, but analysing such patterns proves to be far from trivial.

Chapter 7

Coordinated Scheduling to Enforce Demand Spreading

In this chapter, we study the effectiveness of personal departure advice to enforce peak spreading and alleviate congestion in a setting in which both demand and capacity are stochastic and time-dependent. This advice encompasses a scenario where participants indicate their daily travel schedules by means of restricted time windows of preferred arrival. Based on this, travellers receive a departure advice to meet their personal schedule with an adaptable reliability level. For this study, we split travellers into two groups: (1) *participating* travellers whose departure time interval can be adjusted, and (2) *non-participating* 'background' travellers whose departure times cannot be adjusted. This allows us to assess the impact of the fraction 'adjustable traffic' on the total delay.

Our results give fundamental insight into the optimal scheduling of travellers according to their travel preferences, and show that a significant decrease in average delay can be established when only a small fraction of the total traffic uses a personal departure advice. It can be used to facilitate organisations to improve their accessibility[1].

7.1 Introduction

Active peak spreading can be used as an effective means to reduce congestion. Technological developments create new opportunities to reduce congestion, such as the availability of real-time traffic information

[1] This chapter is based on [**S7**].

and vehicle-infrastructure communication. Combining knowledge of demand and infrastructural capacity creates opportunities to synchronise the departure process to enforce peak-spreading. Motivated by this, we explore the effectiveness of a personal departure time advice for a large group of commuters, explicitly including the personal departure-time preferences of the individual travellers. The goal is to develop a departure advice method that meets the following three requirements:

1. Participating travellers receive a departure advice with a statistical guarantee to arrive at their destination on time.
2. The method spreads traffic to reduce the average congestion level, *both* for scheduled travellers and non-scheduled travellers.
3. The model should be tractable and practically implementable.

To this end, we propose an algorithm to redistribute travellers in time to smooth out traffic and reduce congestion, which is particularly of interest in overcrowded neighbourhoods. This model forms the basis for the development of an application that gives a departure advice for travellers that need to arrive at their destination within a specified time frame.

The concept of peak spreading has been studied extensively. A well-studied model is the bottleneck model by Vickrey [138], of which we extended the basic version in Chapters 5 and 6. This basic model captures the self-organising behaviour of road traffic in a simple manner, which leads to numerous extensions giving insights into traffic behaviour and response.

The effects of information provisioning to enforce peak spreading, combined with the response of travellers is an active field of research. Mahmassani [96] suggests that the effect of information depends on the situation, and that in some cases information provision results in negative effects on the overall system performance. Possible explanations for this negative impact include unsynchronised departure time choices and selfish behaviour of the travellers. The first aspect, unsynchronised departure, occurs when travellers respond in the same manner as real-time travel information when everyone chooses the same alternative. By introducing a central organiser that assigns travellers to a departure time and route choice, this situation could be prevented. The second aspect, selfish behaviour, has recently received a lot of attention. ‘Individuals have a certain social value orientation that determines the extent to which an individual acts selfishly or selflessly' [41]. Moreover, the information acquisition and response of travellers is hard to define. Many modelling approaches have been considered to capture these aspects in a realistic manner (see [25] for a review). These aspects lead us to analyse the effectiveness of a central organiser that synchronises departure time preferences of travellers and provide them with a departure time that meets a predefined statistical guarantee of a timely arrival.

Another topic that has received a lot of attention is the impact of stochasticity on congestion levels, which is neglected in deterministic models. Most often, deterministic models lead to overly optimistic and biased results. By including stochasticity 'more robust and efficient decisions can be made than those made based on deterministic evaluation' [140]. In [141], Waller and Ziliaskopoulos use a *chance-constrained* programming method, to include uncertainty in demand. A uniform distribution of demand is taken, whereby a pre-specified level α defines the reliability constraints. Simulation results show that for some instances, a decrease in long-run average total travel time is experienced because of the more robust solutions. The best reliability level α with respect to the average delay appears to be scenario specific. Furthermore, Waller and Ziliaskopoulos mention that variance in travel time will increase for a larger prediction horizon, which is not included in their paper.

A rich body of literature studies the variability in delay at signalized intersections, starting with the pioneering work of Webster for the fixed-cycle traffic light [144]. Despite its enormous applicability, a main limitation of this model is the assumption that the queue is empty at the beginning of each red period. Hence, this method is overly optimistic, especially in the case of overcrowded intersections. In [1], the author proposes a model that can handle inflow rates that are higher than the number of cars leaving in the green time period. The arrival stream is modelled as a time-inhomogeneous Poisson process, supplemented with the so-called *coordination transformation method*, introduced by Kimber and Hollis [73] to incorporate overload. The downside of this model is that it only works for an empty initial queue, which is often not the case in overloaded systems. Brilon [21] describes a delay formula based on the time-dependent M/M/1 queue by numerical methods. He computes the time-dependent queue length distribution, incorporating initial delay.

To test the applicability and effectiveness of our departure advice model.

We consider a single bottleneck link with two types of traffic streams: (1) *adjustable* traffic, consisting of customers that can be scheduled according to their indicated preferences, and (2) *background* traffic, consisting of customers that cannot be scheduled (see Figure 7.1). We consider a finite time period with known time-dependent arrival rate and a fixed capacity (service rate), in which vehicles queue up at time intervals where the arrival rate exceeds the capacity of the bottleneck. We use the time-dependent queue length computation method to control the congestion level and give a statistical guarantee on arrival time. This control encompasses rescheduling of travellers restricted to their preference interval in order to smooth demand.

To this end, we propose a departure advice algorithm that dynamically schedules travellers on the basis of *predicted* congestion levels and which anticipates the departure time of each traveller based on the departure times given to other travellers. Another important feature of the algorithm is that it includes tail probabilities of the sojourn time *distribution*, which is fundamentally different from the commonly used models that consider mean waiting times only. In other words, we give a statistical guarantee that a traveller arrives at his destination timely as opposed to an advice based only on average travel time. This way, the time-dependent distribution of the deadlines can be determined from a more conservative, risk-adverse approach. Our methodology balances travel demand while incorporating personal preferences. As opposed to the Vickrey model, travellers do not have a single preferred arrival time with a disutility function for delay and late/early arrivals, but instead we use a *flexibility interval*. This method restricts re-scheduling to a time interval in which a user is scheduled. The advantage of such an approach is its simplicity.

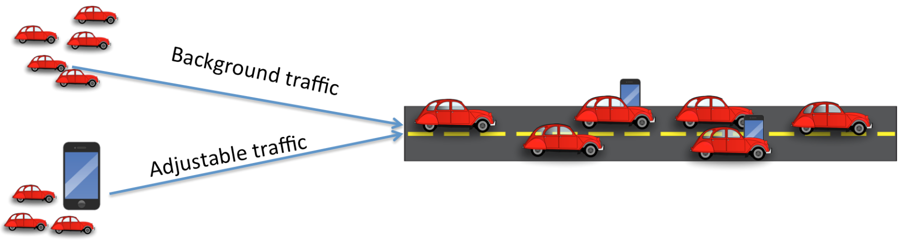


Figure 7.1. Illustration of the model with two traffic classes: (1) adjustable traffic, and (2) background traffic.

The remainder of this paper is organised as follows. In Section 7.2 the modelling approach of the departure advice is described. In Section 7.3 the analysis procedure and the algorithm for the optimal scheduling of customers are described. Section 7.4 discusses the experimental setup to analyse the impact of the departure advice for various scenarios. Section 7.5 gives an outline of the results of these scenarios. Section 7.6 specifies the embedding of the model into a practical application. Finally, Section 7.7 contains conclusions and topics for further research.

7.2 Model description

We study the effectiveness of a personal departure advice system. To this end, we analyse a system where commuters traverse a single road during a finite time span defined by $[0, T]$. This road will further be referred to as *the bottleneck*.

For convenience, we discretise the time span of interest, $[0, T]$, into $n + 1$ time instants T_k, where

$$T_k = \frac{kT}{n} \text{ for } k \in 0, 1, \ldots, n. \tag{7.1}$$

Note that $T_0 = 0$ and $T_n = T$. The time intervals are denoted by

$$I_k = [T_{k-1}, T_k] \text{ for } k = 1, \ldots, n, \tag{7.2}$$

and the length of interval I_k is the same for all k and is given by

$$\Delta t := T_k - T_{k-1} = \frac{T}{n}. \tag{7.3}$$

We assume that the arrivals occur according to a time-dependent Poisson process with piecewise constant rates λ_k for $k = 1, \ldots, n$, and that the service durations are exponentially distributed with mean $1/\mu$ (see Figure 7.2 for an illustration for the case $n = 8$). This $M_t/M/1$ model is referred to as the *original* model. During the timespan of interest $[0, T]$ travellers are served on a first-come-first-served basis. The waiting time of a traveller is defined as the time between his (scheduled) arrival at the bottleneck and his departure. In the original model, travellers do not consider delay at the bottleneck, and we assume that the preferred departure time form the bottleneck is the arrival time in the original

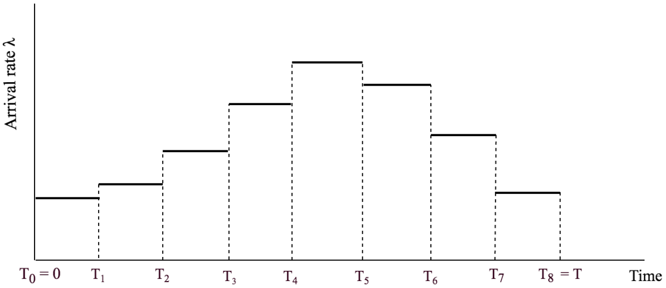


Figure 7.2. Discretisation of the bottleneck time period $[0, T]$ with time-dependent arrival rate λ and $k = 8$.

system. To be specific, for a traveller who arrives at the bottleneck during interval I_k the latest desired departure time is T_k.

The problem is that congestion at the bottleneck causes the travellers to depart later than their preferred departure time. Motivated by this, our goal is to properly (re)schedule travellers based on their desired departure time, providing (statistical) guarantees about their departure time from the bottleneck system. To this end, we assume that a fraction σ $(0 \leq \sigma \leq 1)$ of the customers can be scheduled, and a fraction $1 - \sigma$ cannot be scheduled. This way, the arrival process in the original system is a superposition of two independent processes:

1. a *background* (BG) process of travellers who can *not* be scheduled, which is a Poisson process with rate $\xi_k = \lambda_k(1 - \sigma)$, and

2. a *foreground* (FG) process of travellers who can be scheduled according to their departure-time preferences, which is a Poisson process with rate $R_k = \lambda_k \sigma$ $(k = 1, \ldots, n)$.

We schedule FG travellers according to their preferences such that they get statistical guarantees about their departure from the bottleneck, and at the same time reduce the overall mean waiting time of travellers passing the bottleneck. To be more precise, each FG traveller is characterised by two parameters:

1. The *preferred departure deadline* from the bottleneck T_d (with $d = 1, \ldots, n$), i.e., the latest moment at which the traveller needs to depart from the bottleneck.

2. The *flexibility* $f \in \mathbb{N}$ with respect to the departure moment: the

traveller is willing to be scheduled such that he departs from the system no later than T_d, or earlier than $T_{d-1}, \ldots, T_{d-f}$. Note that this way the traveller is scheduled to depart during one of the intervals $I_{d-f}, \ldots, I_d$.

Throughout, we assume that f is the same for all customers.

The final goal is to assign the FG travellers to a time interval I_k ($k = 1, \ldots, n$) such that congestion is reduced. More specifically, we define the following, we consider optimisation with respect to the following two objectives:

1. Participants Optimum (PO), where we minimise the average sojourn time of the participating travellers (i.e., the FG travellers) only.
2. System Optimum (SO), where we minimise the average sojourn time over *all* travellers (i.e., FG and BG travellers) passing the bottleneck.

These objectives are subject to the (statistical) preference constraints, stating that a participating traveller departs from the bottleneck system before his deadline T_d with probability at least α. The goal of the PO is to create an incentive for travellers to participate, and be subject to scheduling.

7.3 Model analysis

The computation of the optimal arrival schedule proceeds along a number of steps in an iterative manner, as illustrated in Figure 7.3. The complexity of the scheduling algorithm lies in the fact that the relation between the arrival times and departure times depends on the level of congestion, which in turn, depends on the arrival process itself. Therefore, both the sojourn time distribution (see Section 7.3.1 below) and

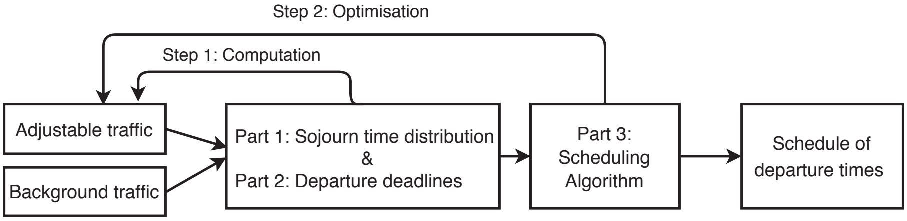


Figure 7.3. Schematic picture of the optimisation process.

the arrival deadlines (Section 7.3.2) have to be recomputed after each schedule adjustment (Section 7.3.3). Therefore, we iterate along the following two successive steps:

Step 1 (Computation of latest arrival deadlines): For each preferred departure deadline T_k, determine the latest arrival time T_i $(i \leq k)$ of a traveller such that his departure time is prior to T_k with probability at least α.

Step 2 (Optimisation with respect to scheduled departure): For each latest departure time T_d $(d = 1, \ldots, n)$, determine the fractions $(p_1^{(d)}, \ldots, p_f^{(d)})$ of travellers to be scheduled for departure in $I_{d-f}, \ldots, I_d$ which minimise the expected sojourn time.

After convergence, the final result is the optimal schedule. In Sections 7.3.1 to 7.3.3 below, we elaborate on the details of the methods involved in the calculation of the optimal schedule visualised in Figure 7.3.

7.3.1 Part 1: Sojourn time distribution

To analyse the delay of the model as defined in Section 7.2, we use the transient results from the time-inhomogeneous $M_t/M/1$ queue. For this queueing model, arrivals are assumed to follow an inhomogeneous Poisson process with parameter λ_k for $k = (1, 2, \ldots, n)$, and departure times are exponentially distributed with rate μ. The behaviour is determined by a continuous-time Markov chain. Hence, the future state (e.g., the number of vehicles in the queue) is only dependent on the present state.

To derive the probability distribution of the queue length over time, we observe the queueing system at time instants T_k, where $k = 0, 1, \ldots, n$. This allows us to consider a discrete-time Markov chain with generator matrix Q_k, where $q_{j,j+1} = \lambda_k$, $q_{j,j-1} = \mu$, $q_{j,j} = -(\lambda_k + \mu)$ for $j \in \mathbb{N}$ and $q_{01} = \lambda_k$, $q_{00} = \mu - \lambda_k$. The states represent the number of travellers waiting at the bottleneck. At each state an arrival or departure can take place, except for state 0 in which there is no one waiting.

To determine the probability to move to an other state, we use uniformisation, where we normalise with respect to the fastest outgoing rate:

$$\gamma \geq \max_j |q_{j,j}|.$$

Moreover, we keep this rate for each time instant k constant. Therefore, we set

$$\lambda_k + \mu + \delta_k = \gamma, \tag{7.4}$$

where δ_k is a dummy transition such that the rate of interval k equals γ for each k. The transition probability matrix is given by:

$$P = I + \frac{1}{\gamma} Q.$$

We denote $\bar{\pi}(k) = (\pi_0(k), \pi_1(k), \dots)$, where $\pi_j(k)$ gives the probability of each state j at time k. We set $\bar{\pi}(0) = (1, 0, \dots)$. Now, we can compute the distribution at time k by:

$$\bar{\pi}(k) = \bar{\pi}(0)e^{Qk} = \bar{\pi}(0)\sum_{n=0}^{\infty} \frac{\gamma k^n}{n!} e^{-\gamma k} P^n, \tag{7.5}$$

where $\frac{\gamma k^n}{n!} e^{-\gamma k}$ gives the probability that n transitions occur in time interval $[0, T_k]$, and where

$$Q_k = \begin{bmatrix} \mu - \lambda_k - \delta_k & \lambda_k & 0 & \cdots \\ \mu & -(\lambda_k + \mu + \delta_k) & \lambda_k & \cdots \\ 0 & \mu & -(\lambda_k + \mu + \delta_k) & \lambda_k \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

In the above case, the queue length distribution is only computed for the first interval, because we initialise the system assuming a queue length of 0: $\bar{\pi}(0) = (1, 0, \dots)$. To recursively compute the queue length distribution from the previous time interval we use:

$$\bar{\pi}(k+1) = \bar{\pi}(0)e^{(k+1)Q_k} = \bar{\pi}(0)e^{kQ_k}e^{Q_k} = \bar{\pi}(k)e^{Q_k}, \tag{7.6}$$

where the interval length between consecutive time intervals k and $k+1$ is of length $T_{k+1} - T_k = \frac{T}{n}$.

The expected queue length at each time instant T_k can be expressed in terms of $\pi_i(k)$ by

$$\mathbb{E}[L_k] = \sum_{i=1}^{\infty} i\pi_i(k), \tag{7.7}$$

where L_k gives the queue length at time T_k.

We want to determine the average sojourn time, denoted by S_k of an arbitrary traveller arriving during interval I_k. By averaging between two consecutive time intervals of Equation (7.7), we compute approximate the average sojourn time of a traveller that departs within his assigned interval I_k:

$$\mathbb{E}[S_k] = \frac{\mathbb{E}[L_{k-1}] + 1 + \mathbb{E}[L_k] + 1}{2\mu}. \tag{7.8}$$

This allows us to compute the arrival interval corresponding to the deadlines of travellers for each interval, which is explained in Section 7.3.2 below.

7.3.2 Part 2: Departure deadline

In this section we use the results of Section 7.3.1 to determine the arrival interval that matches the departure deadline from the bottleneck with a statistical guarantee. To this end, we use the time-dependent queue length distribution, to compute the sojourn time distribution of a traveller. This, in turn, can be used to determine at what time a traveller should arrive at the system, such that he departs before his deadline with probability at least α. Given this latest arrival time, we translate this into a time interval in which the traveller is advised to arrive.

We first need to calculate the probability distribution of L_k, the queue length at time T_k. As follows, this distribution is the same as the distribution of the state of the time-dependent Markov chain:

$$\mathbb{P}(L_k = C) = \pi_C(k) \text{ for } C = 0, 1, \ldots, \tag{7.9}$$

where C represents the number of travellers at the bottleneck queue.

To guarantee that a traveller departs from the bottleneck with a predefined probability, we first compute the probability that we need at least τ time to serve a customer waiting in the queue

$$\mathbb{P}\left(\sum_{c=1}^{C+1} B_c > \tau\right) = \sum_{c=0}^{C}\left(\frac{(\mu\tau)^c}{c!}e^{-\mu\tau}\right), \tag{7.10}$$

where B_c is the service time of the c-th vehicle in the queue. We then substitute Equations (7.6) and (7.10) into

$$\begin{aligned}\mathbb{P}(S_k > \tau) &= \sum_{C=0}^{\infty} \mathbb{P}\left(\sum_{c=1}^{C+1} B_c > \tau\right) \mathbb{P}(L_k = C)\\ &= \sum_{C=0}^{\infty}\sum_{c=0}^{C}\left(\frac{(\mu\tau)^c}{c!}e^{-\mu\tau}\right)\pi_k(C),\end{aligned}$$

where S_k is the sojourn time at time instant T_k.

To obtain the time τ that meets the reliability constraint with level α based on the time-dependent fluctuations in sojourn time, we compute the following equation

$$\tau_\alpha(k) = \min\{\tau : \mathbb{P}(S_k > \tau) < 1 - \alpha\}, \tag{7.11}$$

where $\tau_\alpha(k)$ gives the total amount of time based on the α tail probability of the sojourn time at time T_k. We use this to compute the latest arrival time at the bottleneck such that the traveller departs the bottleneck with probability α. This is given by

$$t^d = \max(t : t + \tau_\alpha(k) \leq T_d), \tag{7.12}$$

where $t^d \in [0, T]$.

In Equation (7.12) we computed a specific time t^d at which the travellers with deadline T_d should arrive at the bottleneck. However, in the scheduling algorithm, travellers are assigned to an arrival interval. We assume that these travellers arrive at the bottleneck uniformly within their assigned interval. Therefore, we need to translate the arrival deadline t^d corresponding to departure deadline T_d to an arrival interval I_{d-l_d}, where $l_d = 0, 1, \ldots, d$. We compute this time interval of arrival by

$$l_d = \left\lceil \frac{t^d - T_d}{T/n} \right\rceil. \tag{7.13}$$

Note that we round the value of l_d to the next integer value. This gives as a result that FG travellers with deadline d depart on average before T_d with probability at least α, i.e., they are ensured to depart with probability α if they arrive in the first half of the interval. This approach ensures that travellers arrive *on average* before their deadline with probability at least α. If we would restrict this to the whole interval, this implies that even for an empty queue, travellers are scheduled an interval before their deadline. This approach would lead to a very conservative arrival advice, and increases the early departure probability substantially.

Travellers which are scheduled to arrive in interval I_k are assumed to arrive homogeneously during that interval. Hence, the resulting arrival process remains Poisson with rates $a_1, \ldots, a_n$, with $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} R_i$. The resulting arrival schedule is given by

$$\lambda'_k = a_k + \xi_k \quad (k = 1, \ldots, n), \tag{7.14}$$

where a_k and ξ_k are the arrival rates of FG and BG travellers in interval I_k, respectively.

7.3.3 Part 3: Scheduling algorithm

To reduce congestion at the bottleneck, we implement a scheduling algorithm that redistributes the FG travellers according to their time preferences and reliability constraints. The objective of the scheduling algorithm is defined from two perspectives: (1) we minimise the average sojourn time of FG travellers throughout referred to as Participants Optimum (PO), and (2) we minimise the average sojourn time of *all travellers*, also referred to by System Optimum (SO).

The SO minimises the average sojourn time over *all travellers*, which is computed by

$$\mathbb{E}S(\bar{\lambda}) = \sum_{k=1}^{n} \lambda_k \frac{\mathbb{E}[L_{k-1}] + 1 + \mathbb{E}[L_k] + 1}{2\mu}, \tag{7.15}$$

where L_k is the queue length at time instant T_k, $\bar{\lambda} = (\lambda_1, \ldots, \lambda_n)$, and λ_k is the average rate of travellers that arrive in interval I_k. Recall that we use the average sojourn time formulation of Equation (7.8).

The PO, which minimises the average sojourn time over FG travellers only is computed by replacing λ_k of Equation (7.15) by the FG rate a_k:

$$\mathbb{E}[S(\bar{a})] = \sum_{k=1}^{n} a_k \frac{\mathbb{E}[L_{k-1}] + 1 + \mathbb{E}[L_k] + 1}{2\mu}, \tag{7.16}$$

where $\bar{a} = (a_1, \dots, a_n)$.

To compute (7.15) and (7.16), we use the arrival rate per interval of the FG travellers $\bar{a}$. A FG traveller with departure deadline T_d is scheduled to arrive at the queue during interval I_{d-l_d}, where $l_d = 0, 1, \dots, d$ denotes the number of intervals that is necessary to depart from the bottleneck before time T_d with probability at least α. By using the computational scheme specified in Section 7.3.2 we can determine the arrival rate of FG travellers denoted by a_k for each interval. For the initial schedule, we translate the deadline rates R_k of FG travellers to a_{k-l_k}, where $l_k \in \{0, \dots, k\}$ for each k. This gives us the arrival rates to compute the average sojourn time.

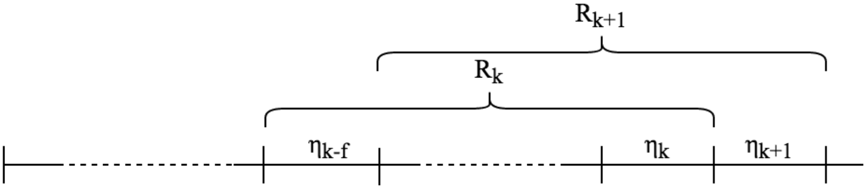


Figure 7.4. Illustration of the intervals in which users with latest arrival interval k and $k + 1$ can be scheduled.

FG travellers indicate their flexibility level $f \in \mathbb{N}$, which means that they can be scheduled to depart from the bottleneck at the f time instants before the final deadline $\{T_{d-f}, \dots, T_d\}$. To capture these new arrival rates, we introduce the vector $\bar{\eta} = (\eta_1, \dots, \eta_n)$, for which $\sum_{k=1}^{n} \eta_k = \sum_{k=1}^{n} R_k$ and $\eta_{k-f} + \cdots + \eta_k \geq R_k$ (see Figure 7.4 for an illustration). We assume that the flexibility time frame f is the same for all FG travellers. This allows us to define the rescheduling problem by the objective defined in (7.15) and the two constraints:

$$\begin{aligned} \underset{\bar{\eta}}{\text{minimize}} \quad & \mathbb{E}[S(\bar{\lambda})] \\ \text{subject to} \quad & \sum_{j=1}^{k} \eta_j \geq \sum_{j=1}^{k} R_k \text{ for } j \in \{f, f+1, \dots, n\} \\ & \sum_{j=1}^{k-f} \eta_j \leq \sum_{j=1}^{k} R_k \text{ for } j \in \{f, f+1, \dots, n\} \\ & \eta_k \geq 0 \text{ for } k = 1, 2, \dots, n, \end{aligned} \tag{7.17}$$

where η_j represents the rate of FG travellers assigned to depart in interval j. The first constraint ensures that enough travellers are scheduled in the intervals before (or at) the latest arrival intervals, i.e., that all users are on time. The second constraint ensures that FG travellers have a flexibility of being scheduled a maximum of f time instants before their latest departure deadline. It is easy to see that these two constraints ensure $\eta_{k-f} + \cdots + \eta_k \geq R_k$, such that the travellers with indicated preference time T_k are scheduled to depart before time instants $T_{k-f}, \dots, T_k$. Our aim is to find the schedule $\bar{\eta}^*$ that minimises $\mathbb{E}S(\cdot)$.

For small and medium-sized problem instances, the optimum can be easily found by standard numerical methods. However, for large model instances, the computation time can become prohibitively large, and local search can be applied to approximate the optimal solution within a reasonable time interval. In general, the solution space is not convex, so that convergence of the local search is not guaranteed, and we stop the computation when a maximum number of iterations is reached. Therefore, as an initial solution, we spread the inflow as much as possible to reduce the chance of a local optimum. More specifically, for given $\bar{\xi} = (\xi_1, \dots, \xi_n)$, our initial solution is the vector $\bar{\eta} = (\eta_1, \dots, \eta_n)$ that minimizes

$$\sum_{i=1}^{n} (\eta_i + \xi_i)^2, \tag{7.18}$$

subject to the constraint in Equation (7.3.3).

7.4 Experimental setup

To assess the effectiveness of the personal departure advice algorithm proposed in Section 7.3, we have performed an extensive numerical experimentation. In this section, we describe the experimental setup and the scenarios considered. In this numerical study, choices have to made with respect to the parameters settings, including the arrival patterns, the capacity of the bottleneck, the size of the discretisation step for the scheduling of travellers, the flexibility level of the travellers, the reliability level for the departure advice and the choice of the objective function.

The effectiveness of the scheduling algorithm will vary depending on the situation of interest. An important factor to be taken into account is the duration and severity of the peak period. The departure advice algorithm applied at a bottleneck with a long peak period with small over-saturation has a different impact than a highly oversaturated road for a short peak period. To analyse the impact of the arrival pattern, we consider four rush hour scenarios, with a total duration of $T = 3$ hours partitioned into 5-minute time intervals denoted by $n = 36$. The service capacity of the bottleneck system is $\mu = 12$ vehicles per minute. The four rush hour scenarios are constructed such that they are representative for different situations, and are referred to as the 'low', 'high', 'peak' and '2 peaks' scenarios:

1. The *low scenario* represents situations where during rush hour there is only a mild overload for a long period,

2. The *high scenario* represents a long and highly oversaturated peak period,

3. The *peak scenario* represents a short but highly oversaturated peak period, and

4. The *2-peaks scenario* represents a peak period with two oversaturated phases.

In line with the observations made in the Highway Capacity Manual [99], we assume that arrival rates are constant over 15-minute time intervals, and given by Table 7.1. Recall that the time granularity for the departure advice is 5 minutes, whereas the arrival rates change at 15-minute intervals. Therefore, for each rush hour type the 12 arrival rates listed

	time	7:00	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45
Type	in min	0-15	15-30	30-45	45-60	60-75	75-90	90-105	105-120	120-135	135-150	150-165	165-180
Low		10.4	11.5	12.3	12.8	13.0	12.8	12.3	11.5	10.4	9.2	7.9	9.2
High		9.3	11.6	13.6	14.9	15.4	14.9	13.6	11.6	9.3	7.0	5.0	7.0
Peak		8.0	8.0	10.0	11.0	36.0	12.3	9.0	8.0	8.0	8.0	7.0	8.0
2-Peaks		8.7	13.6	17.0	8.0	7.0	17.0	18.0	9.0	9.0	9.0	9.0	8.0

Table 7.1. Four scenarios of arrival flows (vehicles/minute) per quarter during rush hour.

in Table 7.1 should be read as $\lambda_{3k} = \lambda_{3k+1} = \lambda_{3k+2}$ for $k = 1, \ldots, 12$. The load of the system is equal for all scenarios; total expected travellers during the time span is $N = 2000$ and the capacity of the bottleneck is given by $\mu * T * 4 * 15 = 2160$ which gives a load of 92.6%.

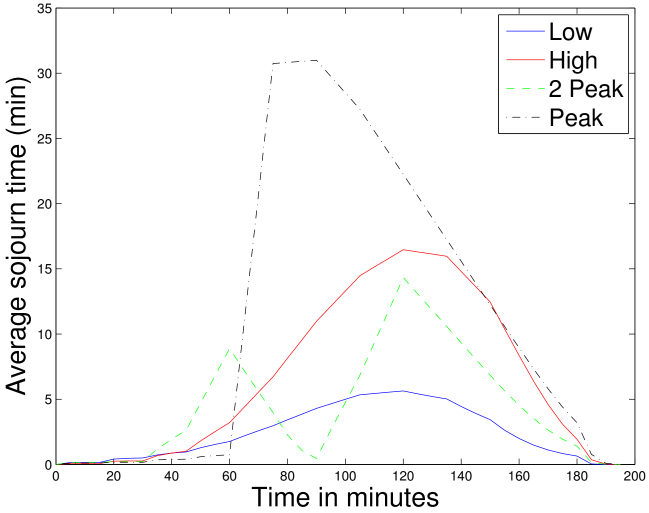


Figure 7.5. Average sojourn time for the arrival scenarios during the bottleneck period.

To illustrate the impact of the arrivals on the queue-length distribution, Figure 7.5 shows how the mean queue length changes over time for each of the four rush-hour scenarios in Table 7.1. The results illustrate the fact that even seemingly mild arrival patterns may have a strong impact on the queueing behaviour of the system. In the next section, we elaborate on the influence of the different parameter settings on the efficiency of the scheduling algorithm.

7.5 Results

In this section, we analyse the impact of the system parameters on the efficiency of the scheduling algorithm. In each of the following subsections we adjust some of the parameters to answer the following questions:

1. What is the impact of the choice of the reliability level, and which one should we use for our experiments?

2. What is the impact of the participation rate, and till what extent is the departure advice a valuable tool?

3. What is the impact of the flexibility time frame, can we determine the minimum level of flexibility required to participate?

4. Which optimisation strategy should be chosen, system optimum, or participants optimum?

We analyse the above four questions in the following three subsections by numerical experiments using the arrival scenarios of Table 7.1.

7.5.1 Reliability parameter

We would like to give a high guarantee to depart the bottleneck before the deadline, while keeping the probability of early arrival as low as possible. In this section, we determine an appropriate level α based on the results of the probability of an early arrival (before T_{d-1}) and late arrival (after T_d) for varying levels of α.

We measure the reliability of the departure advice by two performance indicators: (1) $\mathbb{P}(Early)$, the probability that a traveller departs the bottleneck before time T_{d-1}, and (2) $\mathbb{P}(Late)$, the probability that a traveller departs later than time T_d. In Section 7.3.2 we explained the computation to find the arrival interval I_{d-l_d} based on the α tail probabilities of the sojourn time. This guarantees that travellers arriving in I_{d-l_d} depart, on average, before T_d with probability at least α for each interval. In the same manner, we can compute $\mathbb{P}(Early)$ and $\mathbb{P}(Late)$

for the chosen reliability level α.

In our experiments, we vary α and compute for each of the four scenarios specified in Table 7.1, the corresponding expected earliness, $\mathbb{P}(Early)$, and expected lateness, $\mathbb{P}(Late)$. In Table 7.2 the results of these experiments for α ranging from 10% to 95% are shown. We observe that the earliness probability increases rapidly for high levels of α, whereas the probability of late arrival increases slowly for larger values of α. The results indicate that the lateness probability gives similar results for each scenario, while the earliness probability shows a larger variation.

The reliability level spans the largest interval for $\alpha = 50\%$. Although this gives the largest probability to depart within the specified interval, the probability of late arrival is rather high. Past research indicates that a traveller's experience of late arrival is a lot more unfavourable compared to an early arrival [131]. Taking these aspects into account we set α to 70% for our experiments. This ensures that for each scenario, on average, 9 out of 10 travellers arrive before their deadline, while the discomfort due to earliness remains within acceptable bounds.

	Scenario Low		Scenario High		Scenario Peak		Scenario 2 Peaks	
α	$\mathbb{P}(Early)$	$1-\mathbb{P}(Late)$	$\mathbb{P}(Early)$	$1-\mathbb{P}(Late)$	$\mathbb{P}(Early)$	$1-\mathbb{P}(Late)$	$\mathbb{P}(Early)$	$1-\mathbb{P}(Late)$
10%	0.00	0.69	0.02	0.57	0.01	0.67	0.04	0.63
20%	0.00	0.69	0.04	0.64	0.02	0.74	0.06	0.67
30%	0.03	0.75	0.11	0.76	0.05	0.79	0.09	0.72
40%	0.08	0.82	0.15	0.81	0.05	0.79	0.12	0.76
50%	0.11	0.85	0.20	0.85	0.08	0.83	0.14	0.79
60%	0.14	0.88	0.22	0.85	0.13	0.87	0.30	0.87
70%	0.20	0.90	0.30	0.90	0.26	0.91	0.32	0.89
80%	0.27	0.92	0.42	0.94	0.36	0.94	0.49	0.94
90%	0.47	0.95	0.51	0.96	0.53	0.96	0.56	0.97
95%	0.61	0.97	0.62	0.97	0.61	0.97	0.64	0.98

Table 7.2. Reliability level of departure advice based on the α tail probabilities of the sojourn time.

To summarise, we observe that the $\mathbb{P}(Late)$ of the departure advice for specific level α is similar for each scenario. A larger difference $\mathbb{P}(Early)$ can be observed for the different scenarios. For our examples, we choose a reliability level of $\alpha = 70\%$, to reduce the inconvenience of arriving late, while keeping the $\mathbb{P}(Early)$ reasonable.

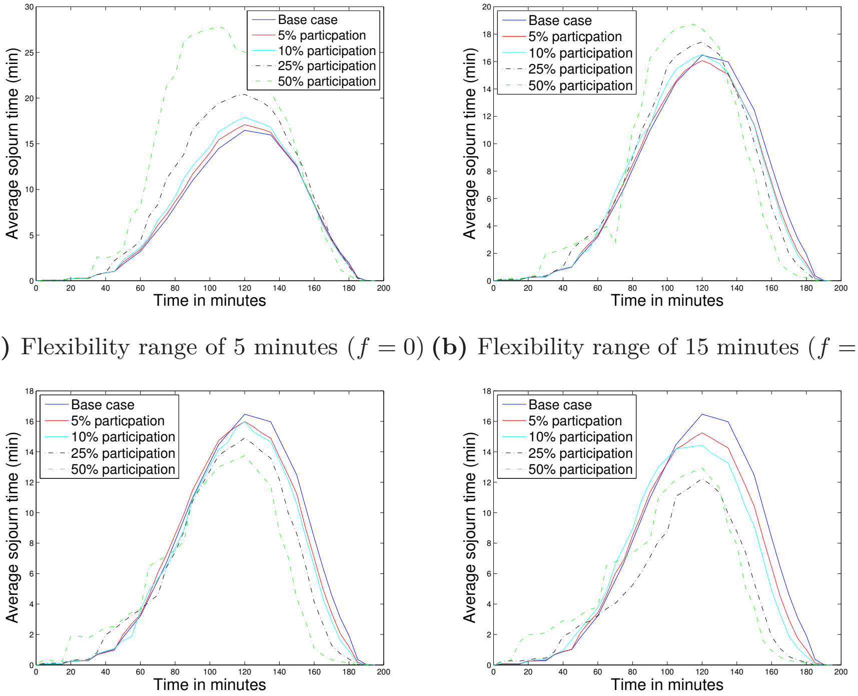
7.5.2 Participation rate and flexibility

To give an answer to the second and the third question, we investigate the effectiveness of the rescheduling method with respect to the participation rate σ and the flexibility time frame f of the travellers. For our experiments, we assume that all the participating travellers take the advice that was computed by the scheduling algorithm and that for a bottleneck scenario all participants have the same flexibility f. We focus on the results of the PO. An overview of the experiments can be found in the Appendix in Tables 7.3 to 7.6.

From the results listed in Tables 7.3 to 7.6 we observe a decrease in average sojourn time for most parameter combinations, compared to the original value. Although we would expect a positive result for all parameter sets, we see that the scenarios with flexibility $f = 0$ have a negative impact on the sojourn time for higher participation rate. In particular, for more severe rush hours like Scenario 'High' and 'Peak', the average sojourn time rapidly increases for larger participation rates. This is caused by the reliability aspect, which schedules travellers to an earlier interval to meet their deadline, causing an earlier build up of the queue at the bottleneck. To compare the impact of the rescheduling without the influence of the reliability constraint, the results for each fraction σ and flexibility f can be compared with the same σ and flexibility $f = 0$.

The impact of the flexibility level on the effectiveness should be considered in combination with the participation rate. In Figure 7.6, the average sojourn time of scenario 'High' is visualised over time for increasing level of flexibility. We observe an increase in average sojourn time for high participation rate and small flexibility time frame. When the flexibility level is 20 minutes, a positive result is obtained for each participation rate. In Figure 7.7 the results for $f = 3$ for scenario 'Peak' and '2 Peak' are visualised. We observe that the 20-minute flexibility does not provide a good solution for scenario 'Peak' when the participation rate is large. The results of the scheduling algorithm clearly depend on the arrival scenario. Therefore, the level of flexibility required for travellers to participate should be determined depending on the expected arrival scenario.

For a 20-minute flexibility time frame, i.e., $f = 3$, we observe a large reduction in average sojourn time. To show the average improvement, we visualise the average of the four arrival scenarios in Figure 7.9. Each bar



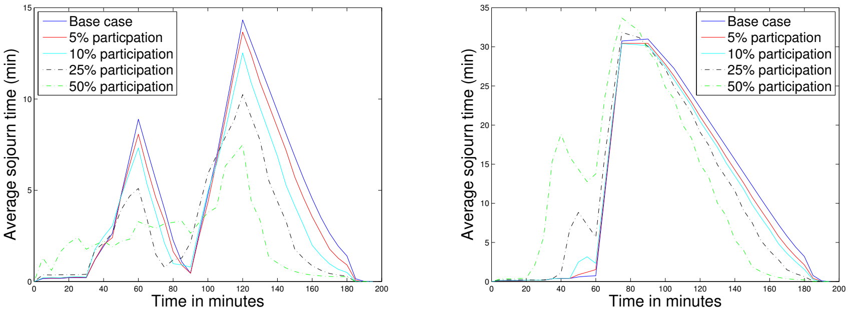
(a) Flexibility range of 5 minutes ($f = 0$) **(b)** Flexibility range of 15 minutes ($f = 2$)

(c) Flexibility range of 20 minutes ($f = 3$) **(d)** Flexibility range of 30 minutes ($f = 5$)

Figure 7.6. Arrival scenario High: average sojourn time for various participation rates with flexibility of 5, 15, 20 and 30 minutes.

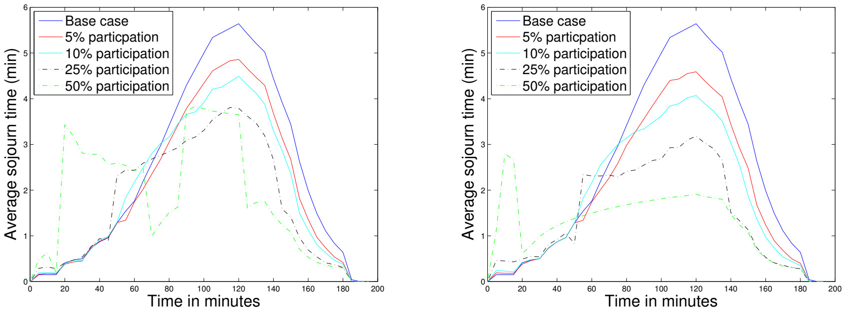
represents the percentage of average sojourn time of BG and FG travellers compared to the base case and $f = 0$ scenario, respectively. Note that the average sojourn time for participating travellers is smaller compared to the average sojourn time for BG travellers in each experiment. This is caused by the fact that in most cases the travellers are assigned to less congested intervals within their flexibility frame.

The scheduling algorithm may cause a complete shift of the peak period for large participation rates combined with a large flexibility f. This is undesirable when travellers who only encountered mild delays have to queue. An example of such a peak shift is shown in Figure 7.8. These peak shifts can be explained by the fact that each deadline corresponds to exactly one arrival interval. Therefore, one interval possibly corresponds to multiple deadlines, and vice versa one interval may correspond to no deadline at all. In some cases, this could lead to a sub-optimal solution,



(a) Scenario 2 Peak ($f = 3$) **(b)** Scenario Peak ($f = 3$)

Figure 7.7. Arrival scenario 'Peak' and '2 Peak': average sojourn time for various participation rates with flexibility of 20 minutes.



(a) Flexibility range of 15 minutes ($f = 2$) **(b)** Flexibility range of 20 minutes ($f = 3$)

Figure 7.8. Arrival scenario Low: average sojourn time for various participation rates with flexibility of 15 and 20 minutes.

see Table 7.5 for $\sigma = 50\%$ and $f \geq 4$. We elaborate that this issue only appears for large participation rate in combination with small flexibility. Including a spread over multiple intervals for each deadline can solve this problem, however, this also decreases the reliability of the departure advice. We did not analyse an extended arrival interval scenario as our study is focused on the impact of a small fraction of participants, and the effect of a larger participation fraction within the same model.

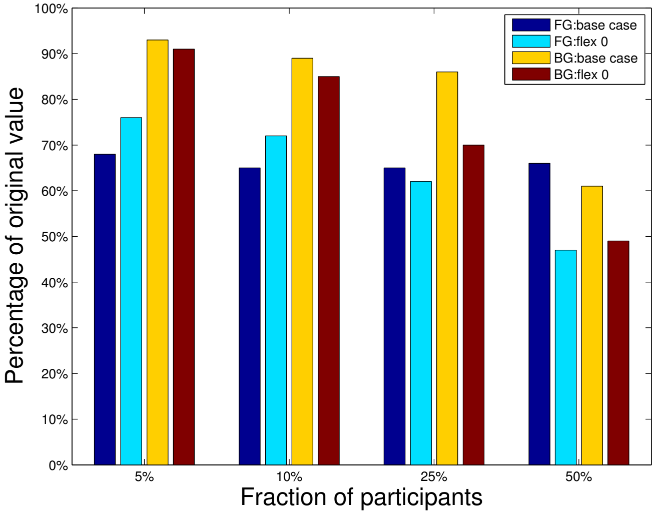


Figure 7.9. Percentage of average sojourn time compared with the base case and $f = 0$ case for BG and FG travellers.

7.5.3 Scheduling objective

So far, we only studied the results for the PO objective, for which we minimise the average sojourn time of the participating travellers. In this section, we compare the impact of both SO and PO, and analyse whether the PO approach is beneficial. Additionally, we discuss the results of the fast search method that provides an initial guess for both objectives. The initial solution is computationally cheap as it only spreads the deadline rates. The results of the initial solution and the two objectives are shown in Tables 7.3 to 7.6 of the appendix.

For both the SO and PO objective we see in Tables 7.3 to 7.6 that the average sojourn time of scheduled travellers is smaller than the sojourn time of all travellers (scheduled and non-scheduled). In general, we reduce delay by rescheduling travellers to less congested intervals. Therefore, participating travellers experience a lower average sojourn time than non-scheduled travellers. We do observe a significant difference between the two objectives with regard to the assigned time interval throughout the bottleneck period. In Figures 7.10 and 7.11 the cumulative arrival times of the FG travellers are visualised for each of the four arrival scenarios. The flexibility time span of the commuters is specified with the dotted lines, representing the minimal and maximal arrival rate for each time step that satisfies the constraints. The black line shows the optimal departure schedule that minimises congestion. What we observe in Figures 7.10 and 7.11 is that commuters are assigned to their earliest departure moment before the peak, and later on they are assigned to their latest departure time.

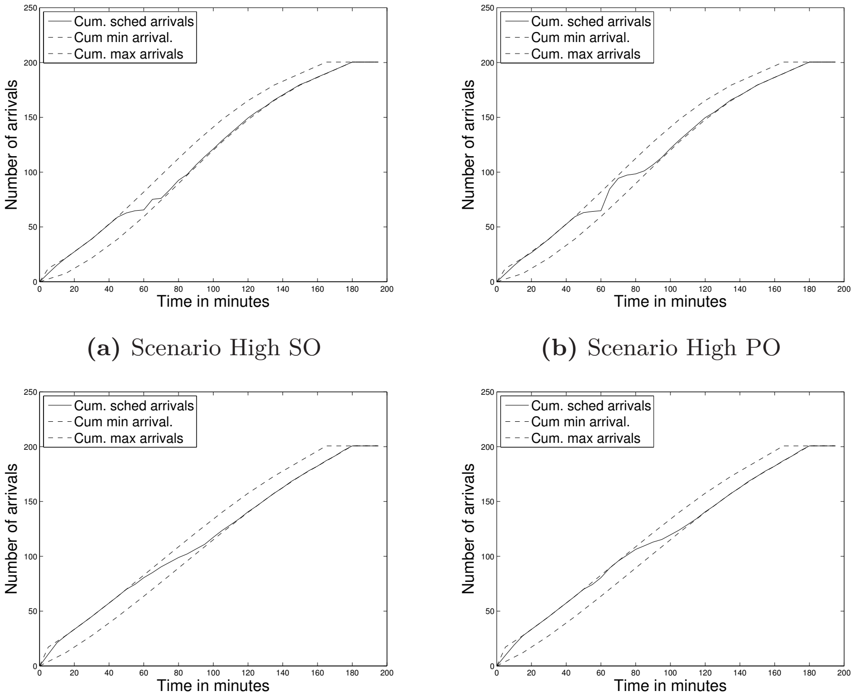
The difference between the objectives PO and SO is shown in Figure 7.10. For the SO objective travellers are scheduled at their earliest departure time until a queue starts to form, while for the PO objective travellers are still scheduled at their earliest deadline. As the sojourn time before the peak rises, the choice of scheduling travellers early reduces the sojourn time of the traveller departing at that moment. However, when the sojourn time increases, scheduling travellers early causes more people to suffer from the traveller arriving early. Therefore, the SO objective starts scheduling travellers to their latest moment when a queue starts to form, while the PO objective waits until the queue is formed.

For the initial solution, we minimise the sum of squares of the departure deadlines. This already shows a substantial reduction in average sojourn time. We use this solution as a starting point for both objectives, SO and PO. This method is computationally very cheap, and significantly reduces the algorithms' convergence time. Moreover, this approach supports real-time adjustments in the schedule.

To summarise, by using our algorithm we see that for both objectives, the FG travellers are in an advantageous position. They are shifted towards less congested time periods, for the PO objectives slightly more than for the SO objective. A risk of the PO objective is the disadvantageous position of BG travellers. This is the case for scenarios with a high fraction of participants combined with a low flexibility level. Within these bounds, our algorithm provides a significant reduction for both BG and FG travellers.

7.6 Implementation

In this section, we shortly describe the software tool in which the scheduling algorithm is embedded. We hereby focus on the embedding and the preconditions that need to be met to effectively use such a tool in practice. The application is meant to support event managers during



(a) Scenario High SO

(b) Scenario High PO

(c) Scenario Low SO

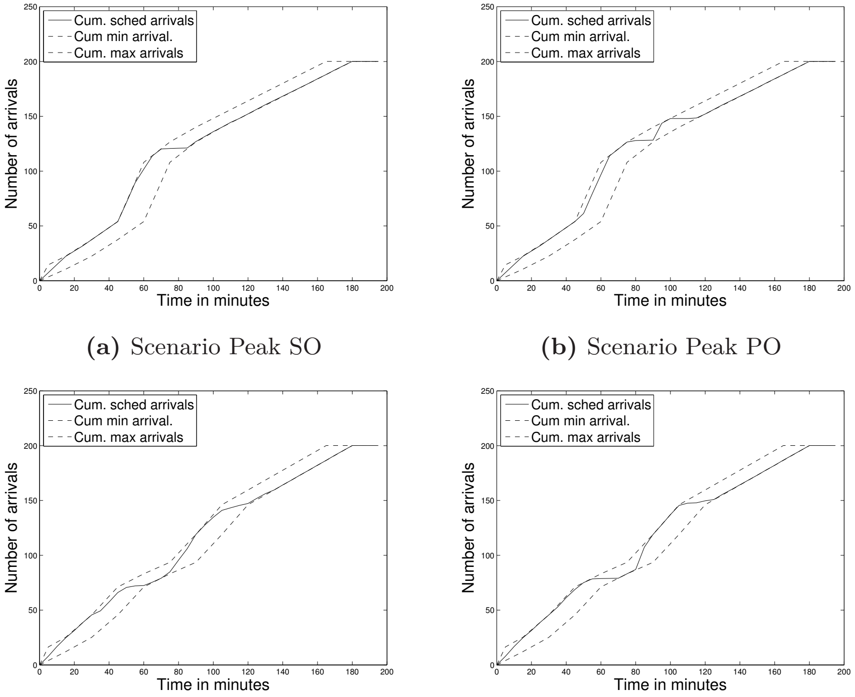
(d) Scenario Low PO

Figure 7.10. Cumulative arrival rate of the departure advice users over time.

an event to reduce the peak flow arrival rate and inform and guide the visitors along their journey to the event. As a result, the application requires more than just the scheduling algorithm.

The software tool that spreads the traffic stream over time at an event location consists of more than just the scheduling algorithm. The application encompasses two components: (1) a web interface application for the traffic manager, and (2) a smartphone application for the end user. A visualisation is shown in Figure 7.12. We describe the two components and its functionalities in more detail.

In the web interface, the traffic manager has to specify the area of interest. This requires the coordinates of the destination(s), the maximum capacity, inflow and outflow constraints. After the initialisation of the destination, the traffic manager has to enter the details for a specific event that he wishes to optimise. This requires information on the date



(a) Scenario Peak SO

(b) Scenario Peak PO

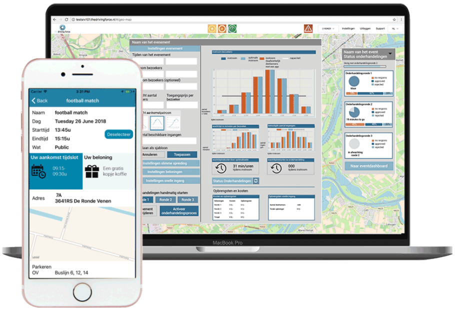
(c) Scenario 2 peak SO

(d) Scenario 2 peak PO

Figure 7.11. Cumulative arrival rate of the departure advice users over time.

and time of the event, the expected number of travellers visiting the event, the granularity of the interval in which travellers are scheduled, and the expected arrival pattern. In order to obtain a feasible schedule, the traffic manager has to specify the maximum number of time slots a traveller can be scheduled from his preferred time, and a target delay to which the congestion should be reduced. The traffic manager can simulate the expected results based on the scenario he initialised and the expected number of participating users. He can adjust the parameters, if needed. For example, when the congestion still exceeds the target delay after optimisation, he can increase the inflow or outflow capacity, or increase the preference bounds. When the details are entered and adjusted to the traffic managers' preference, visitors can sign up for a time slot at the specified event location.

The visitors sign up for a time slot via a smartphone application. They enter their preferred time of arrival and indicate till what extent they



football match

Naam football match

Dag Tuesday 26 June 2018

Starttijd 13:45u

Eindtijd 15:15u

Wat Public

Uw aankomst tijdslot 09:15-09:30u

Uw beloning Een gratis kopje koffie

Adres 7A 3641RS De Ronde Venen

Parkeren

OV Buslijn 6, 12, 14

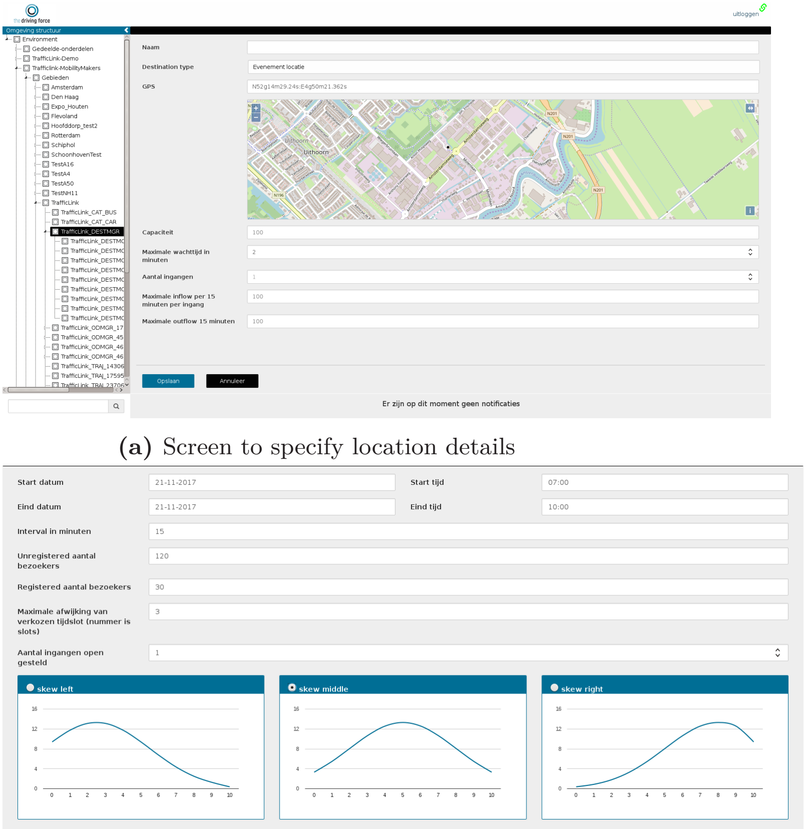
Figure 7.12. Visual of the application tool using the scheduling algorithm.

are flexible with regard to this specific time. Prior to the event, the scheduling algorithm computes the optimal assignment according to the constraints specified by the users and the traffic manager. For practical purposes, each user can specify his own preference bounds instead of a fixed flexibility interval for each participating user as we used in the numerical examples of Section 7.5. Moreover, the traffic manager specifies to what extent these personal bounds are allowed to be exceeded. In case the personal preference bounds of a user are exceeded, this user will receive an acceptance request. This iterative process generates the possibility to interact with the user in order to spread the traffic even more.

Details regarding the departure advice procedure, is visualised in Figure 7.14. A more detailed explanation of these steps is given below:

Step 1: Compute the time-dependent queue length based on the time-dependent arrival rate, and the departure rate. Thereby deriving the moment of departure of the participating travellers, such that they arrive at the destination at their preferred time.

Step 2: Check whether the resulting schedule exceeds the predefined maximum delay time. In case the answer is 'no', we have a feasible solution and continue to the final step (8). In case the answer is 'yes' we continue to (3).



Naam

Destination type: Evenement locatie

GPS: N52g14m29.24s:E4g50m21.362s

Capaciteit: 100

Maximale wachttijd in minuten: 2

Aantal ingangen: 1

Maximale inflow per 15 minuten per ingang: 100

Maximale outflow 15 minuten: 100

Er zijn op dit moment geen notificaties

(a) Screen to specify location details

Start datum: 21-11-2017 Start tijd: 07:00

Eind datum: 21-11-2017 Eind tijd: 10:00

Interval in minuten: 15

Unregistered aantal bezoekers: 120

Registered aantal bezoekers: 30

Maximale afwijking van verkozen tijdslot (nummer is slots): 3

Aantal ingangen open gesteld: 1

○ skew left ● skew middle ○ skew right

(b) Screen to specify optimization details

Figure 7.13. Screenshots of the web application for traffic manager.

Step 3: Minimise the queue length given the constraints of the individual time preference bounds specified per user.

Step 4: Check whether the current schedule exceeds the maximum delay time. In case the answer is ‘no’, we have a feasible solution and continue to (6). In case the answer is ‘yes’ we continue to (5).

Step 5: Increase the range of the preference bounds by one, whereby we switch between left and right increase. In case we exceed the maximum deviation from the preferred time, we stop the process and continue to (6), otherwise we return to (4).

Step 6: Match the optimised arrival schedule to the participating travellers. A list is returned containing the users scheduled outside

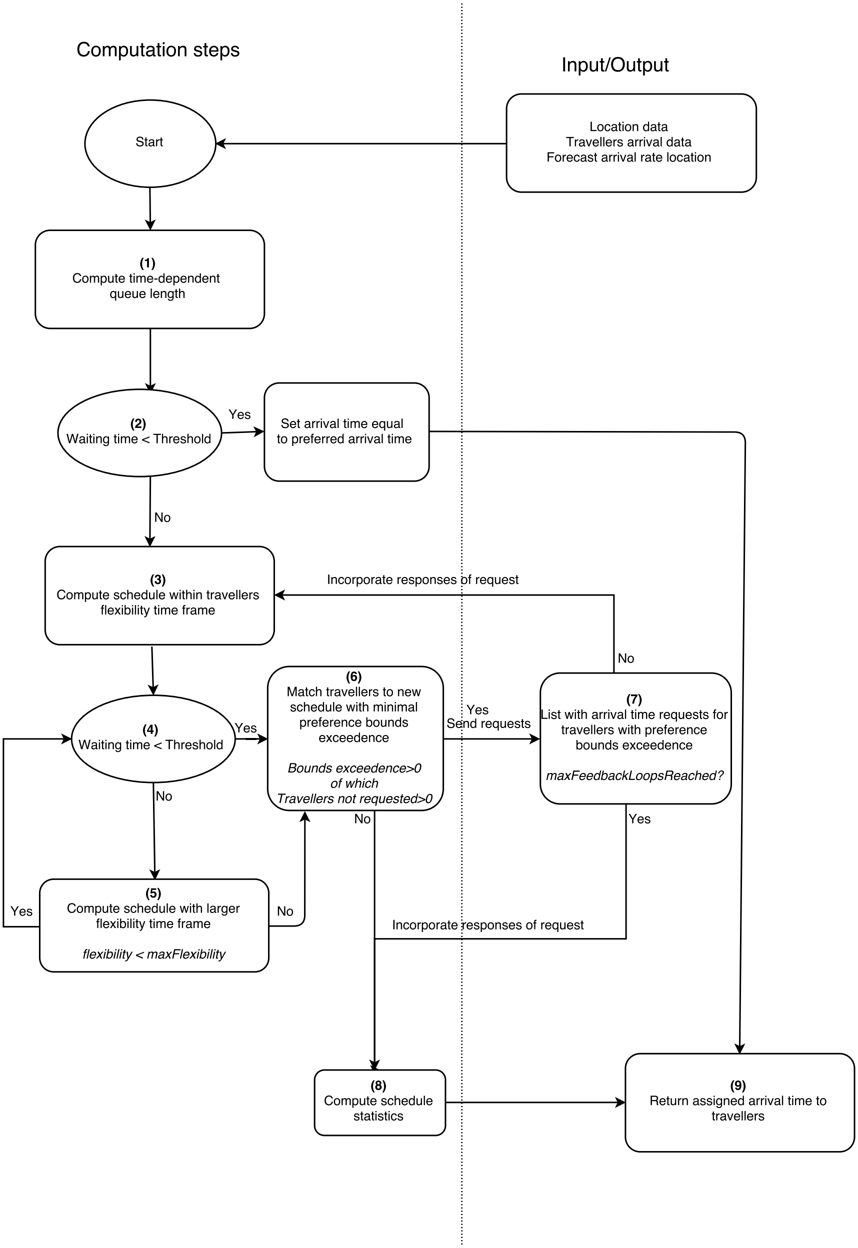

Computation steps
Input/Output
Location data
Travellers arrival data
Forecast arrival rate location
Start
(1)
Compute time-dependent queue length
(2)
Waiting time < Threshold
Yes
Set arrival time equal to preferred arrival time
No
(3)
Compute schedule within travellers flexibility time frame
Incorporate responses of request
No
(4)
Waiting time < Threshold
Yes
(6)
Match travellers to new schedule with minimal preference bounds exceedence
Bounds exceedence>0 of which Travellers not requested>0
Yes
Send requests
(7)
List with arrival time requests for travellers with preference bounds exceedence
maxFeedbackLoopsReached?
No
No
Yes
(5)
Compute schedule with larger flexibility time frame
flexibility < maxFlexibility
Yes
No
Incorporate responses of request
(8)
Compute schedule statistics
(9)
Return assigned arrival time to travellers

Figure 7.14. Procedure of the rescheduling process.

their preference bounds that have not received an adjustment request in a previous iteration. These users receive a request whether they accept the adjustment. In case the maximum number of feedback loops have not been exceeded we proceed to (7), if this sheet is empty we continue with (8).

Step 7: If we did not reach the maximum number of feedback loops. We incorporate the traveller's response and continue to optimise the schedule with this input from (3), otherwise, we go to (8).

Step 8: Given the response, we determine the final schedule, compute the time-dependent queue length and the resulting statistics. We assign each participating traveller to the newly obtained schedule and the resulting schedule is returned as output.

An additional feature in the software application is the reward system. In order to stimulate the participation of visitors, the traffic manager can assign rewards to the visitors. He can specify a reward to travellers based on the adjustments visitors make in their schedule. When a user is asked to arrive outside his preference bounds, the traffic manager can, for example, give this visitor a coupon for a drink or a snack as a compensation. Research on the reward assignment is out of the scope of the current research. However, this component should be explored in further research.

7.7 Conclusion

We studied the effectiveness of a personal travel advice as a measure to reduce congestion. We measured the effectiveness based on three requirements: (1) the departure advice should give a reliable advice that is met with a statistical guarantee, (2) the rescheduling algorithm should give a significant reduction in average delay at the bottleneck, and (3) the method should be easily applicable in practice.

To gain insights into the reliability of a departure advice method, we set up a test environment for which we constructed four types of high load arrival scenarios and modelled the time-dependent arrival and departure process by an $M_t/M/1$ queue. The results of these scenarios show that for each reliability level α, the lateness probability between the scenarios gives similar results. A larger fluctuation is seen for the earliness probability, especially for peaked scenarios, the probability of an early arrival increases rapidly. For the experiments we set $\alpha = 70\%$, this ensures that on average 9 out of 10 travellers arrive before their deadline, while the earliness remains relatively low.

For the second requirement, we studied the impact of this departure advice method on the average reduction in delay. We analysed the four arrival scenarios for various parameter combinations for both the fraction of participating travellers as their flexibility time frame. The fraction of participating travellers was extracted based on percentile arrival rates over time. Most experiments show a significant reduction in delay as a result of rescheduling. Especially, for relatively low participation rate, a steep decrease in average delay is established. However, for larger participation rate, we observe experiments with a negative impact on the delay. To ensure timely arrival, delay can increase substantially, especially when the fraction of participants becomes large and their flexibility time frame is small. It is therefore important to analyse the impact of the scheduling method for the arrival scenario, combined with the fraction of participants and their flexibility level.

To ensure practical applicability of the model, we created a model that uses current technologies that allow for a direct implementation of our method. We did not consider methods such as road pricing, which gives ethical complications. Instead, we focused on reliability and advantageous departure time windows for participants, to incentify travellers. Furthermore, the classical penalty form, where each user encounters extra costs for arriving earlier or later than preferred is relaxed by using a flexibility interval. This interval gives the boundaries of a users' preferred departure time. To create an additional advantage to participate, we computed the schedule that minimises the delay only for the travellers using the departure advice, denoted by the PO. The PO gives an extra incentive to participate, however, it also causes an increase in the sojourn time for BG travellers for some scenarios. Therefore, this idea should be used with caution. In general, for both objectives, the delay of the participants becomes smaller than the delay of the BG travellers.

The results from the above three requirements on the effectiveness of the rescheduling method are promising. The availability of resources such as historical arrival data, real-time sojourn time information, and fast exchange of information creates opportunities to implement this method effectively. A personal travel advice gathers departure information and returns a departure interval for each traveller. Exchange of real-time information from the traffic sensors as well as from the user increases reliability and the ability to react to sudden changes in arrival demand. The algorithms computational efficiency is optimised by means of a local search algorithm that ensures a fast convergence of the scheduling solution to quickly adjust to real-time information. The model has been tested for hypothetical arrival curves where the $M_t/M/1$ model is used to simulate the effects of variability in arrival and gain insights into the impact of a personal travel advice. In order to implement this algorithm for a specific bottleneck, historical arrival patterns should be used to calibrate the model.

Eventually, the effectiveness of the departure advice application depends on the participation of the travellers. The departure curve is partly dependent on the rescheduled travellers, and whenever these travellers do not follow the travel advice, this could influence the expected delay at the bottleneck. Our results reveal that travellers are likely to be rescheduled at the outer intervals of their preference. Whenever a larger group ignores the advice and departs at other time instants, this could result in a negative influence on the delay at the bottleneck. This study only shows the possible impact of a departure advice. Another factor influencing the effectiveness is background traffic. The response of background traffic on the change in delay over time is not included in this study. A field study should be performed to determine the acceptance of travellers for this type of measures.

This model showed that peak-spreading by means of a personal travel advice can be used to alleviate congestion at daily bottlenecks. In particular, for bottlenecks close to the origin of travellers' trip, real-time adjustments and last-minute updates show a substantial reduction in delay. This method could be extended to include multiple route options. Travellers that already departed cannot adjust their departure time, but they can adjust their route. Especially for incidental delays, this extension can inform travellers over the best route alternative according to expected delays.

Appendix

The results of the average sojourn time for each parameter set are displayed in the Tables 7.3 to 7.6. The values represent the average sojourn time encountered upon arrival for: all travellers, participating travellers, and non-participating travellers. This is given for each scenario in three column groups. The first group represents the results after the first search optimisation step. The second and third group give the results of the PO and SO, respectively.

Part. rate	flexibility f	First Search			PO			SO		
		Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled
0%	0	170	-	170	170	-	170	170	-	170
5%	0	172	171	172	172	171	172	172	171	172
	1	167	165	168	151	146	151	151	146	151
	2	163	158	163	146	138	146	146	139	146
	3	158	153	158	139	126	139	140	131	140
	4	140	121	142	138	110	140	133	118	134
	5	136	113	137	133	101	135	128	110	130
10%	0	177	176	177	177	176	177	177	176	177
	1	158	152	158	152	145	153	151	146	151
	2	147	137	148	140	129	141	137	130	138
	3	139	125	140	130	115	132	127	115	128
	4	127	110	129	119	101	121	116	104	118
	5	124	104	126	109	91	111	108	93	110
25%	0	214	214	213	214	214	213	214	214	213
	1	158	154	160	153	151	154	154	152	155
	2	139	131	141	123	117	125	122	116	124
	3	116	108	119	102	96	104	101	95	103
	4	99	91	102	86	80	88	85	79	86
	5	88	86	89	71	67	72	70	66	71
50%	0	269	271	268	269	271	268	269	271	268
	1	146	144	148	145	143	148	145	142	146
	2	107	109	106	130	132	127	104	106	102
	3	91	94	87	86	89	85	79	80	78
	4	91	94	87	86	89	85	79	80	78
	5	91	94	87	86	89	85	79	80	78

Table 7.3. Results of the sojourn time in seconds for scenario 'Low' for parameters σ and f.

Part. rate	flexibility f	First Search			PO			SO		
		Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled
0%	0	489	-	489	489	-	489	489	-	489
5%	0	507	478	509	507	478	509	507	478	509
	1	494	453	496	481	425	484	475	439	477
	2	482	427	485	476	390	481	465	422	467
	3	472	404	475	473	358	479	454	406	457
	4	460	379	464	466	323	473	440	379	443
	5	452	359	457	454	293	463	428	358	431
10%	0	530	503	533	530	503	533	530	503	533
	1	514	476	518	496	452	500	492	455	496
	2	492	449	496	488	418	496	466	431	470
	3	470	433	475	453	403	458	444	406	448
	4	453	413	458	429	386	433	428	386	433
	5	422	337	431	440	288	457	404	326	413
25%	0	633	605	643	633	605	643	633	605	643
	1	578	538	591	551	514	563	550	518	560
	2	523	475	538	493	451	507	500	457	514
	3	452	403	469	420	379	434	445	397	461
	4	386	334	404	370	325	385	393	340	410
	5	341	294	356	314	269	330	331	281	347
50%	0	877	847	907	877	847	907	877	847	907
	1	773	734	812	682	655	709	675	646	704
	2	570	535	605	510	490	530	563	530	596
	3	455	424	485	387	361	413	380	356	403
	4	410	380	439	361	335	386	363	337	389
	5	410	380	439	361	335	386	363	337	389

Table 7.4. Results of the sojourn time in seconds for scenario ‘High’ for parameters σ and f.

Part. rate	flexibility f	First Search			PO			SO		
		Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled
0%	0	329	-	329	329	-	329	329	-	329
5%	0	331	304	332	336	313	337	336	313	337
	1	315	262	318	315	262	318	315	262	318
	2	310	250	313	311	242	314	299	248	302
	3	297	232	300	298	223	301	296	236	299
	4	280	184	285	278	179	283	273	183	278
	5	271	175	276	271	107	280	260	151	265
10%	0	341	307	346	341	307	346	341	307	346
	1	313	260	319	306	241	313	312	263	317
	2	288	221	295	287	191	297	286	221	293
	3	274	197	283	260	156	272	262	179	271
	4	250	175	258	250	175	258	250	175	258
	5	238	169	245	221	116	233	211	135	220
25%	0	388	359	398	388	359	398	388	359	398
	1	331	289	345	320	272	336	324	288	336
	2	268	221	283	255	192	276	259	212	274
	3	224	184	238	206	147	227	191	163	200
	4	170	137	181	159	125	170	162	127	173
	5	149	126	156	134	86	151	131	106	139
50%	0	498	473	523	498	473	523	498	473	523
	1	359	331	388	346	319	374	331	304	357
	2	202	185	218	185	169	202	190	173	206
	3	206	197	215	158	149	167	160	153	166
	4	254	259	251	203	211	194	182	194	169
	5	254	259	251	203	211	194	182	194	169

Table 7.5. Results of the sojourn time in seconds for scenario ‘2-Peak’ for parameters σ and f.

Part. rate	flexibility f	First Search			PO			SO		
		Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled	Total	Scheduled	Non-scheduled
0%	0	881	-	881	881	-	881	881	-	881
5%	0	898	659	911	898	659	911	898	659	911
	1	887	630	901	877	563	893	872	582	887
	2	868	590	882	857	505	876	852	536	868
	3	844	547	859	834	448	854	825	498	842
	4	823	503	840	810	383	832	801	444	820
	5	803	464	821	794	370	817	784	422	803
10%	0	926	659	956	926	659	956	926	659	956
	1	896	616	928	881	575	916	878	604	908
	2	873	622	901	850	556	882	839	555	871
	3	827	563	856	816	466	855	807	512	839
	4	793	521	823	780	406	821	764	436	800
	5	758	465	791	730	378	769	735	440	767
25%	0	1049	826	1123	1049	826	1123	1049	826	1123
	1	985	745	1066	980	721	1067	968	735	1046
	2	897	661	976	936	661	1028	896	641	982
	3	832	626	900	898	625	988	831	618	902
	4	746	535	816	835	529	937	728	518	799
	5	656	458	722	744	447	843	655	458	721
50%	0	1337	1185	1489	1334	1181	1486	1334	1181	1486
	1	1165	1006	1324	1187	1004	1370	1135	980	1290
	2	971	806	1137	1017	872	1163	1221	1064	1377
	3	896	769	1022	910	760	1060	845	694	996
	4	772	667	877	813	705	921	780	675	885
	5	749	662	835	775	663	888	739	647	831

Table 7.6. Results of the sojourn time in seconds for scenario ‘Peak’ for parameters σ and f.

Part

III

Network Modelling

Chapter 8 Network Partitioning on Origin-Destination Traces

In this chapter we use an unsupervised learning algorithm to identify clusters of travel patterns in a network based on historical travel data. This is in contrast to the previous chapters, where we mainly used modelling and optimisation techniques to gain insight into or to improve specific infrastructural strategies. This chapter's starting point is the historical data, due to the volume and the level of detail that is captured within this set, direct interpretation becomes difficult. The techniques applied in this chapter are to transform the data in a comprehensible manner to reveal the structure and patterns to gain insight that can be used in practice.

By means of clustering, we examine the structure of an empirical data set consisting of time-dependent origin-destination pairs in terms of connectedness. We show that we can distinguish spatially connected regions when we use a performance metric called *modularity* and the trip directionality is incorporated. From this we proceed to analyse variations in the partitions that arise due to the non-optimal greedy optimisation method. We use a method known as *ensemble learning* to combine these variations by means of the overlap in community partitions. Ultimately, the combined partition leads to a more consistent result when evaluated again, compared to the individual partitions. Analysis of the partitions can give insights with respect to connectivity and spatial travel patterns, thereby supporting policy makers in their decisions for future infrastructural adjustments[1].

[1] This chapter is based on [**S5**] and [**S6**].

8.1 Introduction

In a densely populated, compact city as Amsterdam it is of great importance to understand the travel patterns of people, as congestion in the city centre is a main concern. With the rise of ubiquitous sensor data, detailed information with respect to mobility is available. Not only can we analyse the infrastructure performance more accurately, it also opens up new opportunities to for estimation, integration and validation of existing models.

For this study, we had access to origin-destination (OD) intensities for the metro region of Amsterdam. These ODs represent neighbourhoods within Amsterdam, and municipalities for the metro-region of Amsterdam. The OD intensities are based on electronic trace data collected from smartphone data by Google. These traces are aggregated at neighbourhood and municipal level by their volume of trips on an hourly basis for a six month period of time.

The aim of this research is to analyse whether travel patterns in Amsterdam can be aggregated into high-level patterns to detect flow trends in both space and time. In the literature, this is called community detection, where the high-level patterns are identified as communities. The results of such an approach can be exploited to analyse major flow patterns between areas based on the obtained communities. Moreover, the obtained communities can be used to support practitioners with strategic decisions. For example to identify or justify the expansion of public transport between specific areas.

We apply clustering to identify communities based on historical travel data. Clustering or graph partitioning is based on nodes that share common properties or behave in a similar manner. In this context, community detection is used to group nodes based on the edge properties only. We thereby want to identify the typical traffic behaviour in Amsterdam from both a temporal and spatial point of view.

In the literature, a wide range of community detections algorithms exist, as well as the number of metrics to evaluate the partition quality of the detection algorithms. A fairly complete review of this topic is given in [43]. By far, the most popular metric to determine the performance of the resulting clusters is called *modularity*, introduced by [108]. Modularity is a metric to measure the strength of a network partitioned into communities based on the intra-inter community edge weight, i.e., the more weight captured within each community compared to the weight between communities, the stronger the connection and the larger the modularity value. The problem of finding the partitioning of a graph with the maximum modularity value is known to be NP-complete [20]. Various heuristics exist to optimise the modularity value. An overview of these methods can be found in [43, Chapter 6].

In a recent study, spatial clusters based on telephone calls have been examined by Blondel et al. [14], who developed an efficient heuristic procedure to find a partition of the network that maximises the modularity known as the Louvain algorithm. In a similar study, this algorithm has been applied to telephone data in Great Britain by Ratti et al. [118]. In both of these papers, the resulting communities of the algorithm are spatially connected, while no spatial characteristics are considered in the algorithm. Both these datasets consist of a large number of connections between the nodes of the network. This algorithm is of interest to us, as the geographical component and densely connectedness both apply to our dataset. The analysis of movement patterns by means of clustering results in aggregated information on the structure of the city, potentially creating a new type of regional analysis for infrastructure developments and planning [118].

Another feature that is included in our dataset is directionality of the trips. In the original Louvain algorithm [15], analysis including directionality is not applied. However, the method is easily extendable to allow for directionality, as is explained in [38]. We will show that the Louvain method produces very good results to determine clusters based on origin-destination pairs in the city of Amsterdam when directionality is included. Moreover, we will show that different time slices of the data (i.e., weekday, months, etcetera), give variations in the obtained communities. However, the comparison is not straightforward as the Louvain method generates variations between each run for the same time slice as well.

The efficiency of the Louvain heuristic with minimal computational effort allows for a more elaborative analysis on the variations between partitions of a network. To this end, we use a technique known as *ensemble learning* to obtain a more consistent partition, i.e., less variation between partitions resulting from the same algorithmic procedure. In [50], they explain this procedure applied to graph partitioning. A more consistent partition of the community structure for a specific time slice allows for a better comparison of partitions of other time slices.

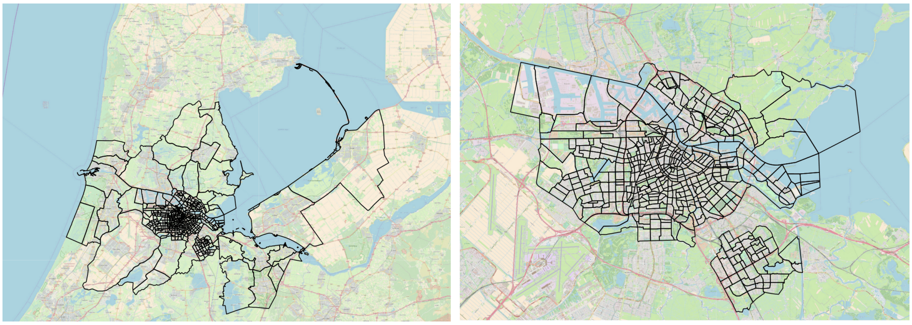
The remainder of this chapter is organised as follows. In Section 8.2 we give a detailed description of the data and specify the filter and preprocessing steps used for the model, which we introduce in Section 8.3. The preliminary results are then given in Section 8.4. From there, we proceed to characterise the obtained communities in terms of connectivity strength in Section 8.5 and consistency in Section 8.6. We conclude in Section 8.7.

8.2 Data analysis

Before we move to the step of clustering, we first give a more detailed description of the data. By applying some preliminary aggregation steps, we obtain initial insights into the value of the data. Some of these confirm assumptions, such as daily fluctuations in travel density, while they also reveal some deficiencies of the data. Moreover, these preliminary results illustrate the motivation to direct the research of this chapter to the use of clustering methods.

8.2.1 Data specification

The travel data used for our analysis is based on travel movements registered by Google on Android phones for the Amsterdam Metro region. This data spans a period of six months that starts 1 April 2016 until 30 September 2016. These trips are aggregated at neighbourhood level for Amsterdam and the surrounding is aggregated at municipal level, both are grouped hourly. These neighbourhoods of Amsterdam and surrounding municipalities are based on the division made by Statistics Netherlands found in [22], who split the area into 512 small pieces as visualised in Figure 8.1a, and in more detail in Figure 8.1b. This division results in more than 300 million data points, consisting of weights from each origin to each destination on an hourly basis. Due to privacy issues, the real intensity has not been disclosed, the intensity is given by a weight which represents a relative value. More specifically, all intensities have been divided by the largest hourly intensity over these 6 months, resulting in weight values between 0 and 1.



(a) The greater metropolitan area

(b) The municipality

Figure 8.1. Neighbourhood and district level division of the greater metropolitan area of Amsterdam.

In Table 8.1 a summary of the weights observed in the data set is given based on the frequency. We observe that the total number of hours that contains weights larger than 0 is close to 30%. As the data consists of all destinations for each origin for every hour, we observe fully connected graphs during most peak hour periods. A large number of the weights consist of small values, an overview is presented in Table 8.1.

Weight	% occurrence	% Total weight
0	71.62%	0%
0.000365764	17.67%	36.42%
0.000731529	6.66%	27.44%
0.001097293	2.49%	15.38%
0.001463058	0.93%	7.68%
> 0.001463058	0.63%	13.08%

Table 8.1. Frequency values for each weight in percentage of occurrence and total density.

8.2.2 Filtering and preprocessing

For the clustering procedure, we restrict ourselves to the travel characteristics within Amsterdam. In this section, we analyse the behaviour of people travelling within the city, and the travelling behaviour from and to the city from the Metro region (defined in Figure 8.1a), to grasp the main traffic characteristics and identify deviating patterns.

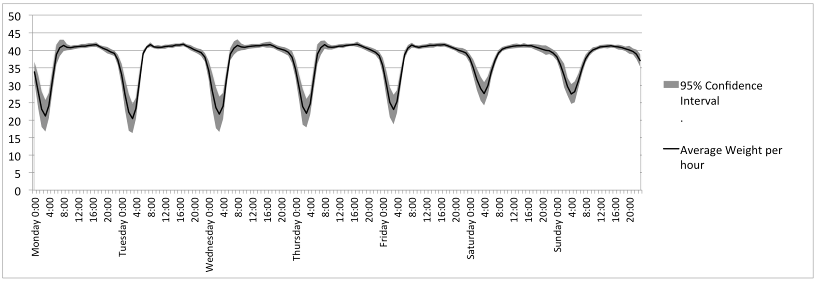
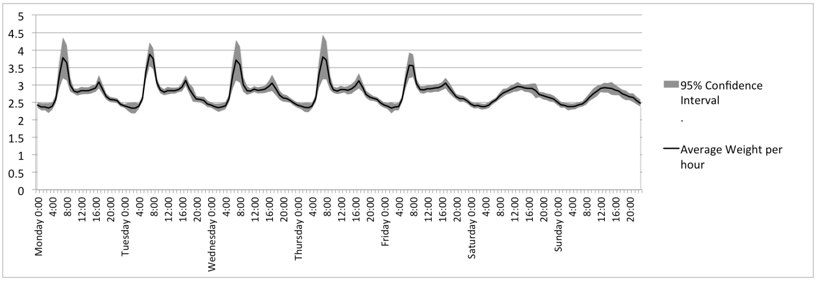


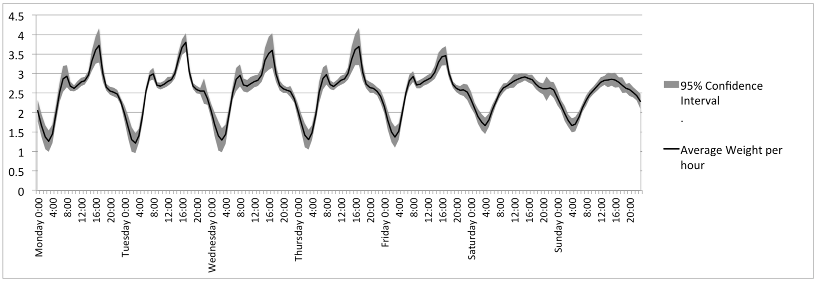
Figure 8.2. Weekly pattern of weights per hour with a 95% confidence interval.

In Figure 8.2 the weekly pattern of trips within Amsterdam is visualised. As can be seen, the rush hour is not so clearly present, and the number of trips in the weekend is nearly as large as during the weekdays. Of course, this data contains not only car travel movements, but also walking and cycling which could explain the intensity of trips throughout the day. The rush hour of trips between Amsterdam and the Metro region area visible in Figure 8.3. In the morning a clear migration from the greater region of Amsterdam is observed to the city of Amsterdam, and in the evening vice versa. In Figure 8.4, the spatial spread of these trips is visualised. The dark red areas in Figure 8.4b all contain large business districts, which could be expected. However, we observe that Figure 8.4a and 8.4b do not show a similar pattern. In Figure 8.4a trips are homogeneously spread over Amsterdam, whereas in Figure 8.4b larger variations in weight between neighbourhoods is observed. A more detailed analysis on this aspect will be discussed below.

In Figure 8.5 of the total trip weight for each neighbourhood as an origin and as a destination for trips within Amsterdam is visualised. We again observe a similar pattern as in Figure 8.3. The destination figure shows a homogeneously spread pattern, while the origin figure shows more variation between the areas. This suggests that certain parts of Amsterdam have more inflow than outflow over a large period of time, which does not make sense considering that these trip intensity are the sum of a half year period. In Figure 8.5c the total inflow and outflow per neighbourhood are visualised. It shows that certain parts of Amsterdam have larger inflow than outflow, except for the first 30 values which belong to the metro region areas. These observations suggests that a transformation has been applied to censor the data.



(a) Weekly pattern for Amsterdam as a destination



(b) Weekly pattern for Amsterdam as an origin

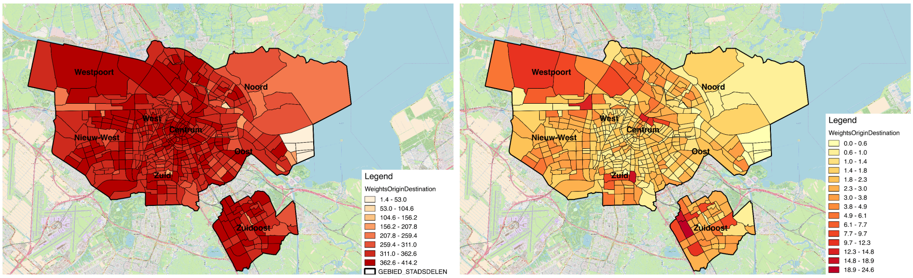
Figure 8.3. Time intensity pattern to and from Amsterdam.

In order to restore the disbalance of the in- and outflow, we rescale the rows of the OD matrix such that the row and column sums become equal. This is done by solving a system of equations where the OD weights are used as Markov chain weights [111]. The resulting stationary probability vector provides the scaling of rows such that the OD matrix disbalance is restored. We use the origin weights as a reference and 'repair' the destination weights. In recent work by Tesselkin [135], this scaling method has been used to reconstruct the OD matrix from traffic flow observations on road segments.

In short, the computation consists of the following steps. We denote the OD matrix by an n by n matrix W, where n denotes the total number of neighbourhoods, and $W_{i,j}$ denotes the intensity of trips from neighbourhood i to neighbourhood j, for $i, j = 1, \ldots, n$. To restore the balance we simply have to solve the linear set of equations

$$W\bar{x} = W^T\bar{e}, \tag{8.1}$$

where $\bar{x}$ is the scaling vector and $\bar{e}$ is a vector of ones. The solution



(a) Inflow intensity pattern

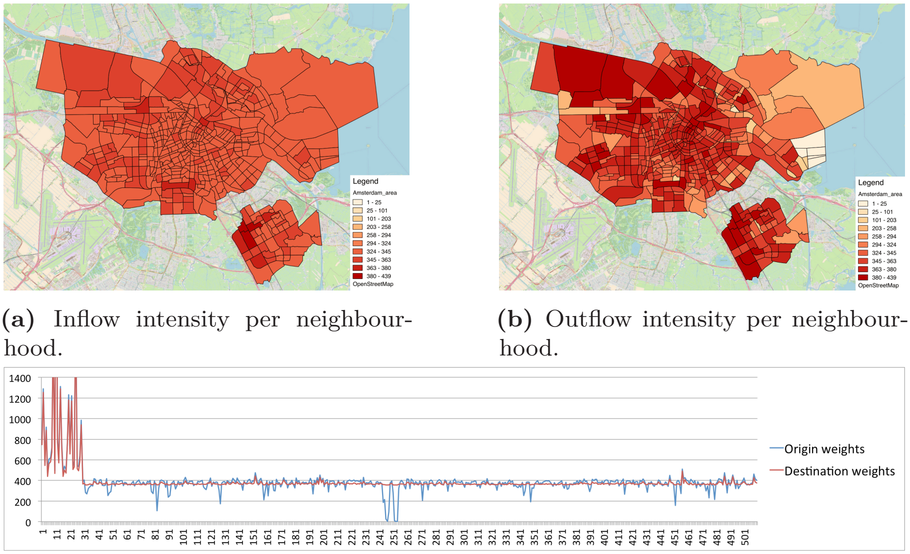
(b) Outflow intensity pattern

Figure 8.4. Spatial intensity pattern of Amsterdam of trips from and to the surrounding Metro region of Amsterdam spanning the six month period.

of $\bar{x}$ is then obtained by $\bar{x} = W^{-1}W^T\bar{e}$, where W^{-1} is the pseudo-inverse of W. The resulting scaling values are visualised in Figure 8.6a. It can be seen that a few areas have a scaling vector close to zero, which is due to the small total outflow compared to the inflow of the specific neighbourhoods. These neighbourhoods are visualised in yellow in Figure 8.6b. We consider these areas as outliers. For analysis purposes, these can be removed from the data, or the scaling factor can be used. In this chapter, we do not adjust or remove neighbourhoods to keep the analysis as clean as possible. Instead, we use the outlier analysis to explain behaviour caused by these deviations.

To give an indication of the travel characteristics per neighbourhood, we visualise the inflow intensity of two neighbourhoods in Figure 8.7. We choose the inflow pattern, as the outflow shows a homogeneous pattern as observed in Figure 8.5a. From both Figures 8.7a and 8.7b, it is observed that travel intensities are larger around the area specified. This suggests that spatially connected communities might arise when neighbourhoods are clustered based on trip intensities.

In this section, we analysed the travel patterns captured in the historical OD data from a spatial and temporal perspective. In general, we can conclude that these patterns match our expectations of travel intensity behaviour, except for the observed disbalance between in- and outflow. We investigated the extensiveness of this disbalance by equalising the in- and outflow. This led to the conclusion that, although we suspect an unexplained transformation on the data, we are confident that this transformation has minor impact on our results. Therefore, in the remainder of this chapter, we perform our analysis on the original data

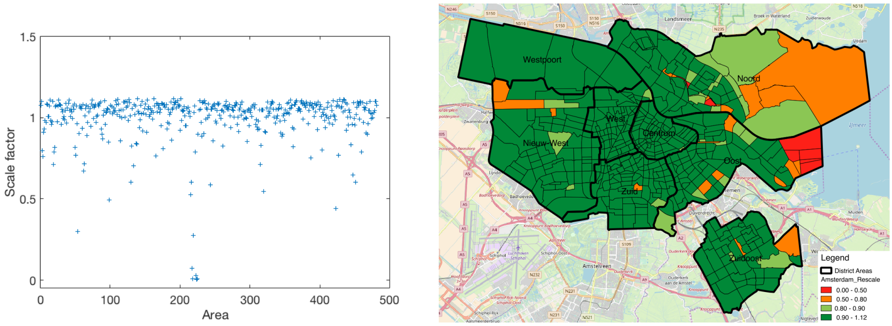


(a) Inflow intensity per neighbourhood.

(b) Outflow intensity per neighbourhood.

(c) Total inflow and outflow intensity per neighbourhood of trips within Amsterdam.

Figure 8.5. Visualisations of the travel intensities within Amsterdam at each neighbourhood spanning the six month period.

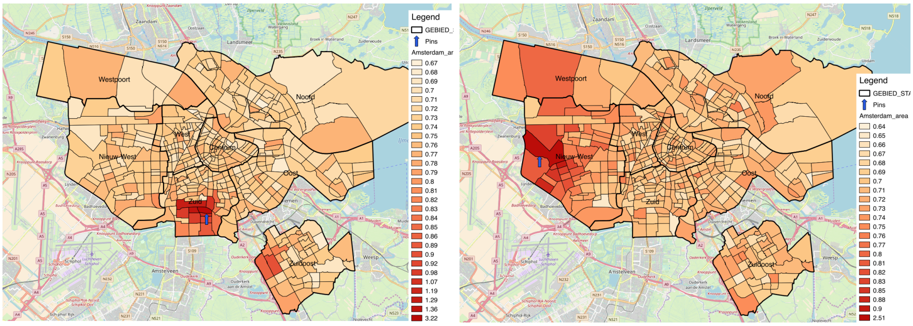


(a) Scaling values of numbered neighbourhoods

(b) Geographical visualisation of scaling values

Figure 8.6. Rescaling values of each neighbourhood, visualising the imbalance of the data between inflow and outflow, where a value of 1 represents no imbalance.

set.



(a) Neighbourhood in 'Zuid' pinpointed in blue

(b) Neighbourhood in 'Nieuw-West' pinpointed in blue

Figure 8.7. Visualisation of travel flow from a single destination.

8.3 Model description

Recall that the goal of our analysis is to discover whether spatially connected communities can be found based on travel intensities only. Moreover, we want to analyse whether the obtained communities represent the districts as defined by the municipality of Amsterdam, which allows for policy evaluation.

First, we explain the transformation of the OD trip matrix to a connected graph. We then introduce the metric, denoted as *modularity*, to evaluate the quality of a network partitioned into communities. We give an explanation of several models that heuristically optimise this modularity metric. Finally, we show the results of one of these heuristic methods of which we obtain interesting results for various subsets of the data.

8.3.1 Network description

We represent the OD trip matrix W in terms of a directed weighted graph $G(\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V} = \{1, \ldots, n\}$ represents a neighbourhood and each edge $(i, j) \in \mathcal{E} \subset \mathcal{V}^2$ represents an OD pair. Each edge has a weight that corresponds to the travel intensity across the respective OD pair, denoted by $w_{i,j} \geq 0$, where $i, j \in \mathcal{V}$. We partition the graph into C communities, where, for each node i mapping index function $c_i = k$, for $k = 1, \ldots, C$ to its corresponding community. We define $\mathcal{V}_k := \{i \in \mathcal{V} : c_i = k\}$ as the set of nodes that belong to community k. Moreover, we define $\mathcal{C}_i := \{k \in \mathcal{V} : c_i = c_k\}$ as the set of nodes that belong to the same community as node i. Note that $\mathcal{C}_i = \mathcal{V}_{c_i}$. The graph is initialised by either assigning each node to a unique community ($C = n$, $c_i = i$), or by assigning all nodes to one community ($C = 1$, $c_i = 1$).

8.3.2 Modularity metric

Modularity is a well-known metric to determine the quality of a graph partitioned into communities. It is a measure of strength of the partition of the network into communities and is defined by a scalar value $Q \in [-1, 1]$. In the literature, the modularity value is often computed for undirected graphs. Therefore, we first present the undirected version before we explain the directed one. The modularity value Q for an undirected graph is defined by

$$Q = \frac{1}{2m} \sum_{k=1}^{C} \sum_{i,j \in \mathcal{V}_k} \left[w_{i,j} - \frac{w_i w_j}{2m} \right], \tag{8.2}$$

where $m = \frac{1}{2} \sum_{i,j \in \mathcal{V}} w_{i,j}$ is the total weight in the graph, and $w_i = \sum_{j \in \mathcal{V}} w_{i,j}$ defines the total edge weight attached to node i. This formula measures the density of edges inside communities to edges outside communities, the value $w_{i,j} - \frac{w_i w_j}{2m}$ defines the differences between the actual weight between nodes i and j and the average node degree weight of i and j. Maximising the modularity value theoretically, results in the best possible grouping of nodes of according to the inter and intra cluster trips for a given network. However, going through all possible iterations of the nodes into groups is impractical so heuristic algorithms are used.

The modularity metric of Equation (8.2) can easily be extended to include directionality as was shown by Leicht and Newman [87]. They show that the total weight connected to these two edges, should be split into the total in-degree weight of one edge and the total out-degree weight of the other edge. Moreover, we specify the total weight by $m_d = \sum_{i,j \in \mathcal{V}} w_{i,j}$ instead of $2m$ as we now count each edge weight only once. This results in the following equation

$$Q_d = \frac{1}{m_d} \sum_{k=1}^{C} \sum_{i,j \in \mathcal{V}_k} \left[w_{i,j} - \frac{w_i^{in} w_j^{out}}{m_d} \right], \tag{8.3}$$

where $w_i^{in} := \sum_{j\in\mathcal{V}} w_{j,i}$, and $w_i^{out} := \sum_{j\in\mathcal{V}} w_{i,j}$.

8.3.3 Heuristic clustering technique

Clustering based on optimisation of the modularity value is a popular approach [43]. Many heuristic techniques exist for modularity optimisation. A comparative study has been conducted by in [79]. Most of these heuristics are only implemented for undirected graphs, while our data consists of directed OD pairs.

For this research, we will not dive into all the clustering heuristics and their performances. Instead, we only focus on a method well-known for its computational efficiency, developed in [14], throughout referred to as the *Louvain method.* This method was first used to detect communities in geographical regions by means of telephone data. The result which captures our interest is the spatially connected clusters that were found, although no spatial characteristics were included in the algorithm. Moreover, this algorithm has shown to outperform many other heuristic methods for benchmark graphs. It has been ranked as second-best heuristic algorithm [79]. The infomap algorithm by Rosvall and Bergstrom [124], which is based on compression, has been ranked as first. Later on, we shortly mention its performance on our dataset.

We now briefly explain the partitioning procedure of the Louvain algorithm; a more detailed description is given in [14]. This algorithm can be classified as a greedy hierarchical approach for modularity optimisation and is known for its computational efficiency. The algorithm consists of a two-step procedure which is iterated until the modularity value is no longer improved. The first step is the ‘greedy’ assignment of nodes to communities, and the second step contains the hierarchical component, where the obtained communities are combined.

Initialisation:
The graph is initialised by a partition into singletons, meaning that each node represents a community.

Step 1:
A loop initiates that runs through all the nodes in a random order. For each node $i \in \mathcal{V}$ the neighbouring nodes are identified, i.e., $w_{i,j} > 0$. For each neighbouring node j, the modularity gain is computed when node

i is added to the community c_j of neighbouring node j. The node i is then added to the neighbouring node j's community that creates the largest positive increase in modularity, computed by (8.3.3). The first loop is re-initiated until the modularity gain is no longer improved.

Step 2:
All nodes that belong to the same community are combined into one node representing the community. This means that the total weight to an external node is combined from all the nodes within the community, and the total weight of nodes within the community is summed, representing the total weight from the community to itself.

Stopping criterium:
Repeat steps 1 and 2 until the final communities between the current and previous iteration are equal.

To speed up the above computation, we focus on the change in modularity when node i is moved to the community of node j, rather than recomputing the modularity by (8.2). For given modularity Q the new modularity becomes $Q' = Q + \Delta_Q(i,j)$, where $\Delta_Q(i,j)$ is defined by

$$\begin{aligned}\Delta_Q(i,j) = &-\frac{1}{2m}\left[\sum_{k\in\mathcal{C}_i\setminus i}\left(w_{i,k}+w_{k,i}-\frac{w_i w_k}{m}\right)+\left(w_{i,i}-\frac{w_i^2}{2m}\right)\right] \\ &+\frac{1}{2m}\left[\sum_{k\in\mathcal{C}_j}\left(w_{i,k}+w_{k,i}-\frac{w_i w_k}{m}\right)+\left(w_{i,i}-\frac{w_i^2}{2m}\right)\right] \\ &=\frac{1}{2m}\left[\sum_{k\in\mathcal{C}_j} w_{i,k}+w_{k,i}-\frac{w_i w_k}{m}\right] \\ &-\frac{1}{2m}\left[\sum_{k\in\mathcal{C}_i\setminus i} w_{i,k}+w_{k,i}-\frac{w_i w_k}{m}\right].\end{aligned} \tag{8.4}$$

Similarly, we can compute the *change in modularity* of Equation (8.3.3) for the directed case by

$$\begin{aligned}\Delta_{Q_d}(i,j) = \frac{1}{m_d}\left[\sum_{k\in\mathcal{C}_j} w_{i,k} + w_{k,i} - \frac{w_i^{in}w_k^{out} + w_k^{in}w_i^{out}}{m_d}\right] \\ - \frac{1}{m_d}\left[\sum_{k\in\mathcal{C}_i\setminus i} w_{i,k} + w_{k,i} - \frac{w_i^{in}w_k^{out} + w_k^{in}w_i^{out}}{m_d}\right].\end{aligned} \tag{8.5}$$

8.3.4 Evaluation technique

In this section we explain the evaluation metric that we use to give an indication of the partition quality of the OD network, and to make a comparison of the obtained communities between various time slices. We explain how we can use the evaluation metric, as the ‘true’ partition of the network is not known.

In the literature, many evaluation techniques are proposed to determine the quality of the obtained network partitions. Almeida et al. [2] describe various metrics that exist to determine the quality. However, no straightforward method exists to evaluate the quality of a partition when the ‘true’ partition is unknown. Some of the evaluation techniques can however be used to compare results and give an indication of their quality.

The most common quality metric is the Normalised Mutual Information (NMI) [26]. We use this metric to compare our partition realisations based on their similarity. This metric is in the range of [0,1] and equals 1 if two partition realisations are identical. This value computes the *mutual information* $I(\cdot,\cdot)$ between the two partitions and normalises it based on the entropy value $H(\cdot)$ of each realisation. The *entropy* is a value of the uncertainty present in a realisation. The mutual information gives a reduction in uncertainty by using the information of the first partition to estimate the second partition. In other words, it computes to what extent the realisations overlap. The NMI is defined as

$$\mathrm{NMI}(P_i,P_j) = \frac{2I(P_i,P_j)}{\sqrt{H(P_i)\cdot H(P_j)}} = \frac{2\left(H(P_i) - H(P_i|P_j)\right)}{\sqrt{H(P_i)\cdot H(P_j)}}, \tag{8.6}$$

where P_i and P_j denote the clustering labels, and $H(P_i) = -\sum_{l=1}^{C}(p_l \log p_l)$ the entropy value, where C defines the number of clusters in P_i, and p_l

the fraction of nodes belonging to cluster l.

The value is normalised such that it corrects for differences in the total number of clusters obtained between realisations. This metric has been used to compute the quality of various clustering algorithms [80, 81].

The NMI compares two realisations, whereas we have a group of realisations and want to determine the overall similarity between these partitions. To obtain the mutual information over a group of partitions, we can compute the so-called *average*-NMI, as defined by Ana and Jain [3]

$$\textit{average}\text{-NMI}(\mathcal{P}) = \sum_{i \neq j} \text{NMI}(P_i, P_j) / \binom{r}{2}, \tag{8.7}$$

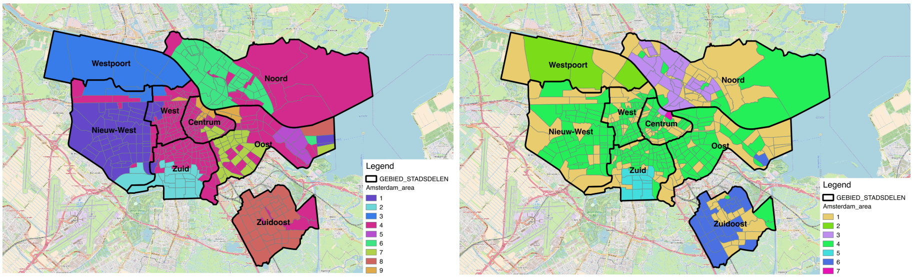
where r denotes the number of cluster realisations, $\mathcal{P} = \{P_1, \ldots, P_r\}$ the group of partitions, and P_i, P_j the individual cluster realisations, for $i, j = 1, \ldots, r$ and $\neq j$.

8.4 Preliminary cluster results

In this section, we show the resulting communities of the OD data in Amsterdam by using the Louvain algorithm. We partition the dataset based on time slices and discuss the observed differences in communities by using the evaluation metrics described in Section 8.3.4.

Various clustering heuristics are compared in [79] such as Fast Greedy, Walktrap, infomap and OSLOM. In contrast to the positive results on the benchmark sets used in [79], these methods proved to be unsuccessful when applied to the OD data of Amsterdam. These methods either failed to converge or returned near to zero modularity values. Near to zero modularity is an indication that the corresponding clusters do not represent any cohesion. In Figure 8.8b the clusters resulting from the undirected implementation of the Louvain algorithm are visualised, a close to zero modularity value is obtained. Visually we observe that these clusters show a certain degree of spatial connectedness, although the low modularity value indicates that a high spatial connectedness in the network exists.

Although our data set does not consist of millions of nodes and edges, we do have a large number of edges to nodes ratio. The dataset consists



(a) Louvain directed

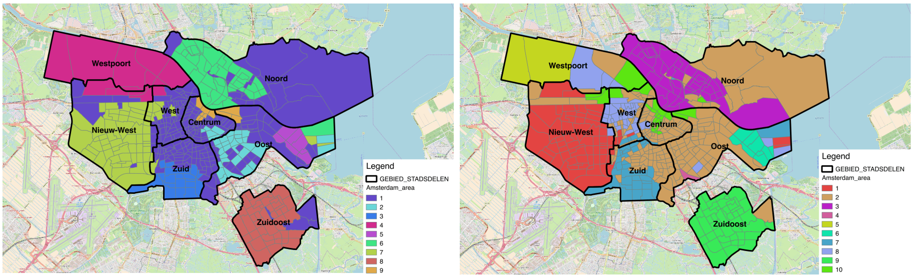
(b) Louvain undirected

Figure 8.8. Clustering with respect to destination for each district.

of an almost fully connected graph, which is probably the reason that most clustering methods do not find good communities. Moreover, the directionality of the connections in the data was not included in most of these heuristics. Therefore, we continued the analysis by using an implementation of the directed Louvain method developed in [126]. An output of this method is visualised in Figure 8.8a. As can be seen, the clusters that result from the Louvain method including directionality appear spatially close, although no spatial aspects are taken into account. Moreover, some of the communities have a close resemblance with the districts of Amsterdam. For example, the ‘Zuid-Oost’ district, which is more isolated from the rest of Amsterdam, is nearly covered by a single community. Nevertheless, the modularity value of the resulting clusters is 0.01, although larger than the undirected output, it is still rather small.

The geographical visualisation of the directed Louvain method show connected clusters and are grouped at locations that we would expect. Therefore, we explore continue to explore the results for this methods for subsets of the data. We divided the data based on the trips during the week and the weekend and applied the Louvain clustering algorithm. The results are shown in Figure 8.9. In both figures, similar clusters appear. However, there are some clear differences. The main differences are the clusters in district ‘Oost’ and ‘Westpoort’ that appear only for the week data, and the cluster in the ‘Amsterdam West’ district that pops up in the weekend data. The clusters at the outskirts of Amsterdam appear to be the most prominent.

Table 8.2 shows the average similarities between the clustering realisa-



(a) Week **(b)** Weekend

Figure 8.9. Clustering with respect to destination for each district using the Louvain method.

Period	Total Period	April	May	June	July	August	September
Total	0.94	0.82	0.83	0.82	0.85	0.79	0.78
Week	0.94	0.84	0.88	0.91	0.83	0.85	0.87
Weekend	0.85	0.43	0.44	0.48	0.48	0.49	0.46

Table 8.2. Comparison of the similarity between cluster realisations for different subsets of the data by using the similarity metric NMI.

tions over the same dataset by using the *average*-NMI value of equation 8.7. We divided the data based on the 'Total' trips, trips during the 'Week' and 'Weekend' and the similarity results over the 'Total period' and on a monthly basis. The *average*-NMI values show that most subsets show consistent results between runs. However, the weekend data shows an overall smaller *average*-NMI value, especially when the monthly division is used. The consistency of the weekend data for each run is smaller compared to the week and total data sets. These variations can be caused by the smaller number of days covered, as well as less regular travel patterns in the weekend.

To analyse whether large differences and similarities between months are present, we again use the *average*-NMI value of Equation 8.7 to compare the resulting partitions. The results are shown in Table 8.3. The *average*-NMI value of each month with itself is shown as well. The largest NMI value for each subset is with itself, denoted by the values on the diagonal. The last row compares the total data set with each month. We do not observe extreme differences between the months in this comparison.

Period	April	May	June	July	August	September	Total Period
April	0.85	-	-	-	-	-	-
May	0.75	0.86	-	-	-	-	-
June	0.74	0.76	0.91	-	-	-	-
July	0.77	0.76	0.75	0.87	-	-	-
August	0.73	0.73	0.72	0.77	0.86	-	-
September	0.73	0.73	0.72	0.75	0.72	0.87	-
Total period	0.82	0.83	0.82	0.84	0.79	0.78	0.93

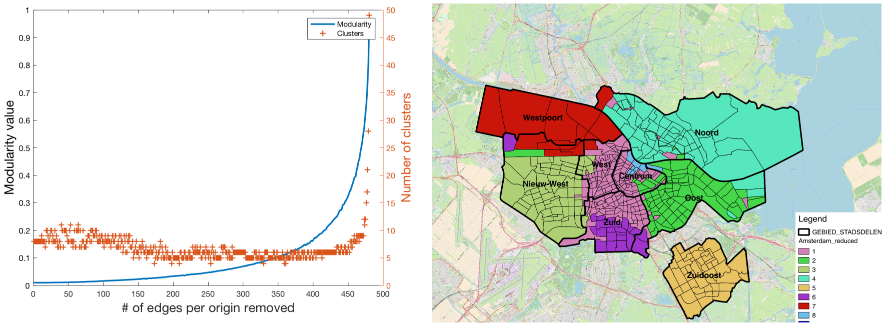
Table 8.3. Comparison of the similarity between cluster realisations for monthly subsets of the data by using the similarity metric NMI.

The small modularity value appears for each of the subsets of the data. The small modularity combined with a fully connected network is not a surprising result. The fully connected graph indicates a well-connected network with a lot of interaction throughout the whole area of Amsterdam. Nevertheless, we do observe spatially connected communities that are closely related to district boundaries and the subsets are relatively similar to each other. We therefore continue our analysis to determine the strength of the communities.

8.5 Robustness of communities

In the previous sections, we observed the spatially connected partitions of OD data in Amsterdam. To gain more insight in the generated communities, we analyse the strength of the communities relative to each other in terms of connectivity.

To analyse what fraction of all edges contributes to the detection of these district boundary clusters we propose a simple method for analysis. We remove the smallest x weights edges from each neighbourhood, where $x \in \{1, \ldots, n\}$ and n denotes the number of nodes of the network. In Figure 8.11a the results show that the modularity value increases when the number of smallest weights x removed increases, as would be expected. More interestingly, the number of clusters found remains relatively constant until almost all values are removed. In Figure 8.11c the clusters found when 10% of the smallest weights were removed are visualised. It can be seen that this partition represents the regional boundaries even more closely than the clusters of the complete set. This



(a) Modularity and cluster results

(b) Results based on removal of 10%

Figure 8.10. Cluster analysis for increasing number of removed edges per neighbourhood.

suggests that although the trips within Amsterdam are well spread, trips within regional boundaries have higher weights in almost every district. Only part of the ‘Centrum’, ‘West’ and ‘Zuid’ region remain connected as one cluster.

In addition to the question which edges contribute to the spatially connected clusters, another question that arises is the connectedness of the communities with respect to each other. Which are the most prominent communities in the dataset, and which communities are less prominent. Although there is no specific metric available in the literature to evaluate such property of connectedness [43, Chapter XIV], in [54] the authors evaluate the connectedness by adding random noise to the edge weights.

We applied the same methodology as in [54]. We add random weights to the edges with a predefined variance. Thus, for each edge in the network, we draw a random variable X, where $X \sim N(0, \sigma^2)$, add these values to the OD matrix, run the Louvain algorithm and visualise the obtained clusters. We gradually increase the value of σ and evaluate the resulting clusters between each increment until no coherent structure can be found. This gives an indication of the connectedness of each community relative to the other communities in a visual manner.

In Figures 8.11, a realisation for an increasing variability in random noise is shown. It should be kept in mind that these graphs only show the result of a single realisation and are only indicative of the impact of noise. We observe that the cluster ‘Zuidoost’ and ‘Noord’ remain visible although a large noise value is added to the edges. The cluster ‘Centrum‘ is the first to disappear, and dissolves in the ‘Oost’ cluster. This is a first step towards analysis of connectedness of clusters with respect to each other. In [54] a more thorough analysis of the consistency of the individual nodes is applied. For the current analysis we have so far only used visually indicative results. In Section 8.6.2 we continue introduce a method that reduces the variations between each realisation of the same subset to obtain more consistent results at each run, allowing for comparison between subsets.

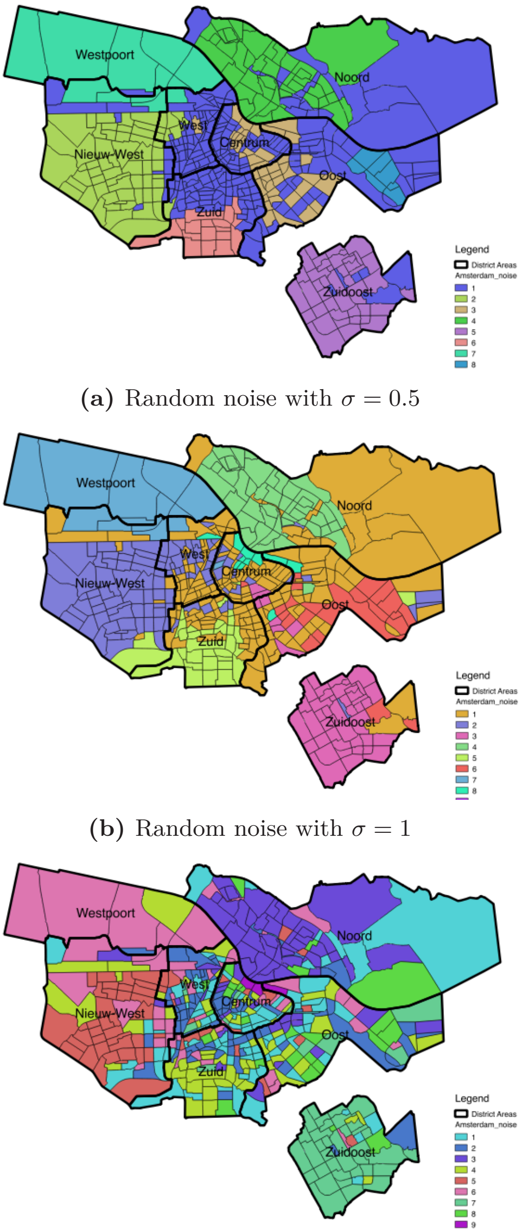
8.6 Consistency of communities

So far, we analysed the community structure of the communities resulting from the Louvain clustering heuristic method applied on the OD data set. We observed variations between realisations of the same dataset and between subsets of the data. To compare the communities of the subsets relative to each other, we need a procedure that gives more or less consistent communities when applied on the same subset. We use a procedure called *consensus clustering* to obtain this consistency. Consensus clustering is an ensemble learning method that combines multiple realisations to create a more consistent final result. An example applied to graph clustering is explained in [114].

8.6.1 Consensus clustering procedure

To determine the consistency of each community, we analyse which neighbourhoods characterise the community. In this section, we explain the procedure to obtain such a characterisation for the current data set.

The Louvain method aims to maximise the modularity value in a greedy manner. The greedy approach makes it computationally efficient, and makes it applicable for clustering on large data sets. However, due to a randomisation in the approach each realisation can deviate from a previously obtained realisation. The algorithm evaluates nodes based on their modularity gain when clustered. Due to randomisation in the order of which these nodes are evaluated, deviations in initial clusters occur. As the algorithm progresses, these initial clusters can result in a node ending up in another cluster than for other initial clusters. Moreover,



(a) Random noise with $\sigma = 0.5$

(b) Random noise with $\sigma = 1$

(c) Random noise with $\sigma = 2$

Figure 8.11. Clustering results under random noise addition.

some communities might not appear due to initial clusterings of nodes which in other realisations belong to different communities. We want to exploit these variations to find the neighbourhoods that can be defined as the 'core' of the community, as well as the neighbourhoods that are on the boundary between communities.

A method known as cluster ensemble learning can be used to obtain the core cluster result, ensemble-based learning is a procedure that combines the results of a certain number of weak learners to obtain a final more robust result. In [50], an ensemble-learning procedure is explained which they call evidence accumulation clustering. We use this procedure to obtain our final partition. The evidence accumulation method is composed of three steps. We will explain each step and specify the implementation that we choose to generate our results.

Step 1 (Generating an ensemble): A cluster ensemble is generated consisting of m clustering partitions, denoted by

$$\begin{aligned}\mathcal{P} &= (P_1, \ldots, P_m)\\ P_1 &= \left(c_1^{(1)}, c_2^{(1)}, \ldots, c_n^{(1)}\right)\\ &\vdots\\ P_m &= \left(c_1^{(m)}, c_2^{(m)}, \ldots, c_n^{(m)}\right),\end{aligned}$$

where $c_i^{(j)} \in \{1, \ldots, C^{(j)}\}$ gives the cluster $k \in \{1, \ldots, C^{(j)}\}$ that node i of partition j is assigned to. These partitions can be obtained by either using different representations of the data, the choice of algorithms, or the algorithmic parameters. The randomised order of the node evaluations in the Louvain algorithm causes variations between each realisation in our dataset, which make it an appropriate method to apply the algorithmic parameter approach. The randomisation of the nodes is then the parameter adjustment.

Step 2 (Determine the similarity): The second step is to combine the cluster realisations by combining 'evidence' in the so-called *co-association matrix* A, with entries $A = (a_{i,j})$, with

$$a_{i,j} = \frac{n_{i,j}}{m}, \tag{8.8}$$

where $n_{i,j} = \sum_{k=1}^{m} \mathbf{1}_{\left\{c_i^{(k)}=c_j^{(k)}\right\}}$ represents the number of times that nodes i and j belong to the same cluster among the m partitions.

Step 3 (Obtain the final partition): The final step in the evidence-based clustering method is to obtain the final cluster partition from the generated similarity matrix A. Any clustering can be applied over this matrix to generate this partition. A hierarchical clustering algorithm is used to combine the nodes and generate the resulting dendrogram [32]. This last step can become complicated when the number of nodes is large. However, in [50] Fred and Jain propose to group similar nodes together before generating the dendrogram. We use this approach to group the nodes that belong to the same partition in all iterations before generating the final dendrogram. We obtain our matrix A by applying the following three steps:

(a) We combine the nodes which are in the co-association matrix A with the value 1, meaning that they are grouped in the same cluster for all realisations. This gives a reduced form matrix $A' \subset A$.

(b) The subset A' is then used to obtain the dendrogram using the Complete-Link method [32]. The Complete-Link method computes the dendrogram based on the furthest neighbour method. This method is known to generate clusters that are well separated and compact and is one of the most commonly used methods for hierarchical clustering. As it computes the furthest neighbour, we have to determine the dissimilarity between nodes. This means that we use $A'' = 1 - A'$.

(c) Having obtained the dendrogram, we then need to determine the cutoff value to disentangle the dendrogram into separate clusters. We determine the cutoff value that leads to the identification of k clusters, where we set k equal to the closest integer value of the mean number of clusters from the cluster ensemble input $\mathcal{P}$.

In the next section, we show the results obtained from the above procedure.

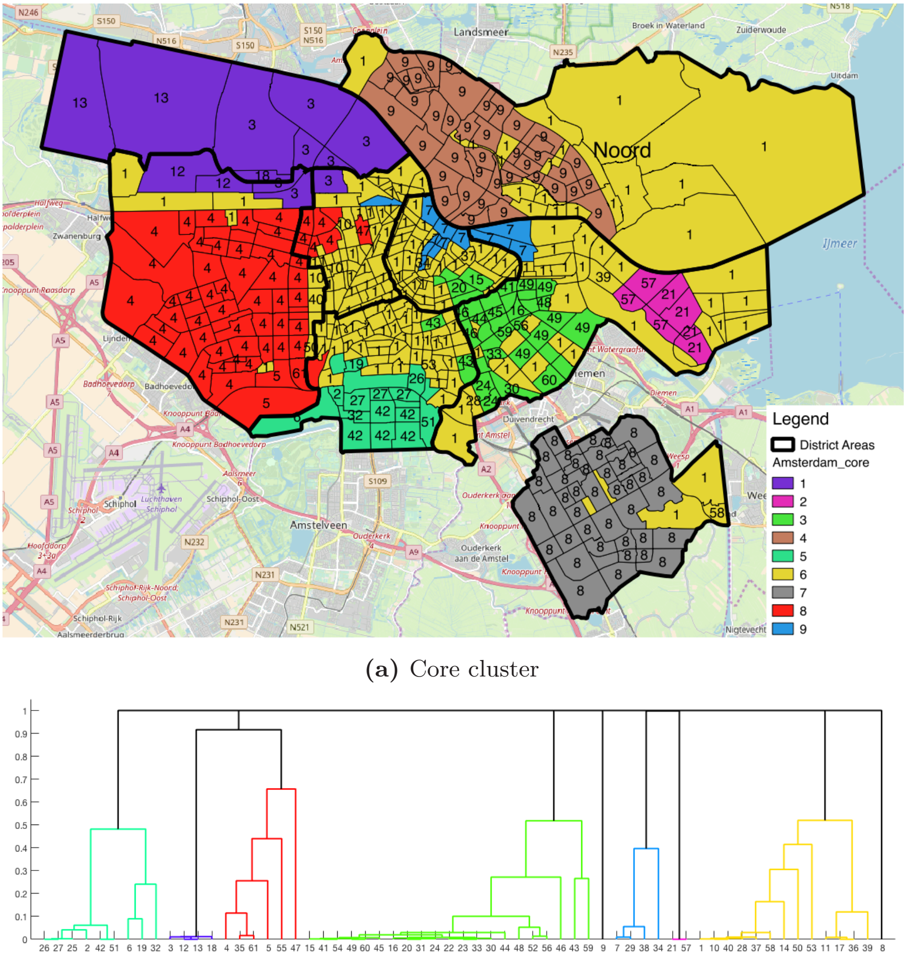
8.6.2 Experimental results

We apply the evidence-based learning algorithm to show the consistency and variation between communities and neighbourhoods. We use this approach to compare the monthly subsets in a more robust manner as well.

We applied the ensemble-learning method initially over the whole data set. We computed the results based on $N = 1000$ realisations of the Louvain algorithm and computed the co-association matrix. We grouped the nodes of the co-association matrix of Equation (8.8) when they belong to the same community over all realisations. This results in a subset of the co-association matrix of size 6, of which 34 values are due to individual nodes. For the current analysis, the size of the dendrogram is still manageable. However, when many individual nodes occur, the reduction of the co-association matrix can also be performed based on a high similarity value.

The resulting core clusters of the total OD-data are visualised in Figure 8.12. Figure 8.14a shows the spatial representation of the final core cluster assignments by individual colours. The value in each neighbourhood represents the initial partition of 61 values in the co-association matrix. The dendrogram of Figure 8.13b shows the dissimilarity between the node groups that are grouped together. For example, the pink group with node groups 21 and 57 have a dissimilarity value near to zero. This means that in only a few realisations of the algorithm they were not assigned to the same cluster. On the opposite side, we can observe that node group 47, denoted by red, does not have a large similarity value compared to the other node groups in this cluster. It is interesting to observe that we have two core clusters consisting of only a single group of nodes. These two node groups were identified in all N realisations, meaning that the neighbourhoods in these groups were consistently grouped together.

We applied the same analysis for the weekdays and weekend subsets of the OD dataset. The resulting partitions and dendrograms are visualised in Figure 8.14. Especially for the weekend subset we observe more diversity between the clustering result. The dendrogram of Figure 8.14b shows that in particular the centre cluster shows large fluctuations over the partitions. No major differences are observed between the week and weekend partitions, suggesting that travel behaviour shows similar



(a) Core cluster

(b) Core cluster dendrogram

Figure 8.12. Core clusters for the complete data set.

groups for the week and weekend days.

We continue to use the cluster ensemble technique to obtain more robust results for the monthly subsets. In Tables 8.2 and 8.3 the NMI-values of the cluster partitions of the same dataset were relatively low. Making it hard to draw conclusions when compared to each other. We use cluster-ensemble method and run this method several times to compute the *average*-NMI values over the subsets. The results are shown in Tables 8.4 to 8.6. It shows that the self-similarity is increased for the

Period	April	May	June	July	August	September	Total period
April	0.97	-	-	-	-	-	-
May	0.79	0.98	-	-	-	-	-
June	0.76	0.79	0.96	-	-	-	-
July	0.81	0.83	0.76	1.00	-	-	-
August	0.76	0.78	0.72	0.81	0.95	-	-
September	0.74	0.74	0.70	0.74	0.71	0.90	-
Total period	0.85	0.88	0.82	0.90	0.81	0.77	0.98

Table 8.4. Average-NMI values of the consensus clustering result based on monthly data.

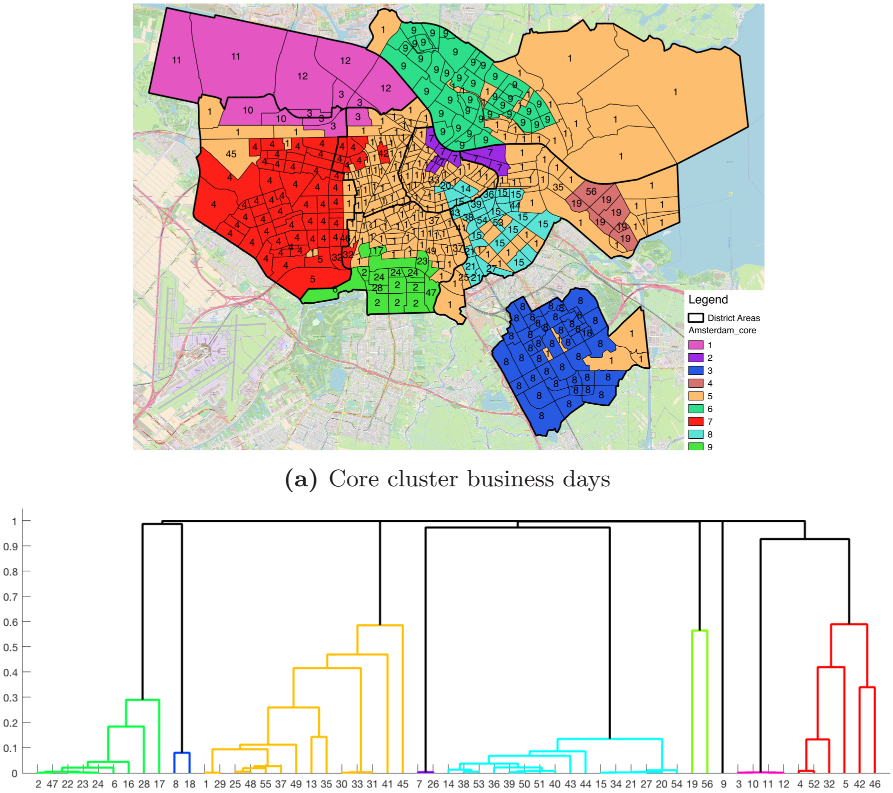
Period	April	May	June	July	August	September	Total period
April	0.97	-	-	-	-	-	-
May	0.76	0.97	-	-	-	-	-
June	0.77	0.79	0.99	-	-	-	-
July	0.75	0.75	0.71	0.93	-	-	-
August	0.74	0.77	0.72	0.76	0.95	-	-
September	0.75	0.74	0.72	0.71	0.72	0.95	-
Total period	0.70	0.73	0.70	0.72	0.73	0.64	0.98

Table 8.5. Average-NMI values of the consensus clustering result based on monthly data during business days.

total and weekly dataset, obtaining values close to 1. This allows for a better comparison between the several months as the monthly subset gives more self-consistent results.

The *average*-NMI values in Tables 8.4 to 8.6 show that in particular September gives a lower NMI value compared to other months. To determine whether specific results deviate, visual representations of the maps should be compared. However, we first analysed the number of clusters that were formed for each month. The average number of clusters for each monthly subset are shown in Table 8.8 for the weekdays, weekend and total set. Interestingly, the September month shows fewer clusters compared to the other months, possibly explaining the lower similarity value. June and August result in slightly more clusters compared to the other months. We expect that the main differences between the resulting core clusters are caused by the number of partitions.

An easy analysis of the partition differences observed in Table 8.8 of August and September is a geographical visualisation of the core result



(a) Core cluster business days

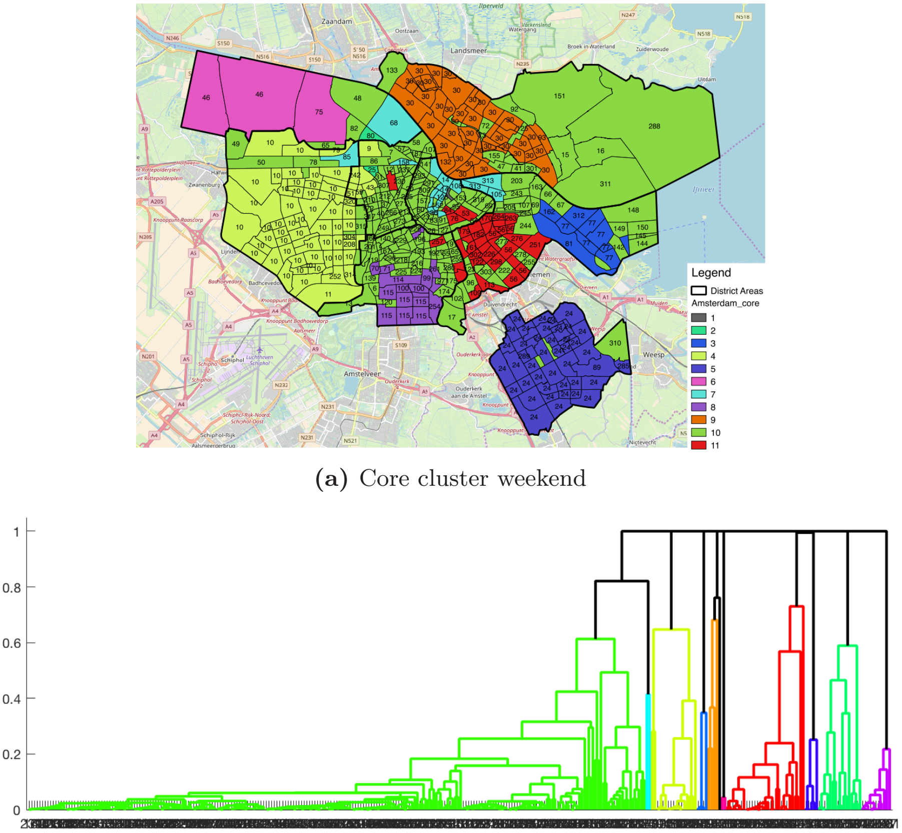
(b) Core cluster dendrogram business days

Figure 8.13. Core clusters for the week data set.

Period	April	May	June	July	August	September	Total period
April	0.64	-	-	-	-	-	-
May	0.38	0.66	-	-	-	-	-
June	0.38	0.40	0.66	-	-	-	-
July	0.40	0.45	0.44	0.69	-	-	-
August	0.38	0.40	0.41	0.42	0.69	-	-
September	0.38	0.39	0.39	0.42	0.39	0.65	-
Total period	0.49	0.54	0.55	0.57	0.52	0.53	0.99

Table 8.6. Average-NMI values of the consensus clustering result based on monthly data during the weekend.

obtained from the ensemble method. In Figure 8.15 both months are shown. We observe that in Figure 8.15b, representing September, the



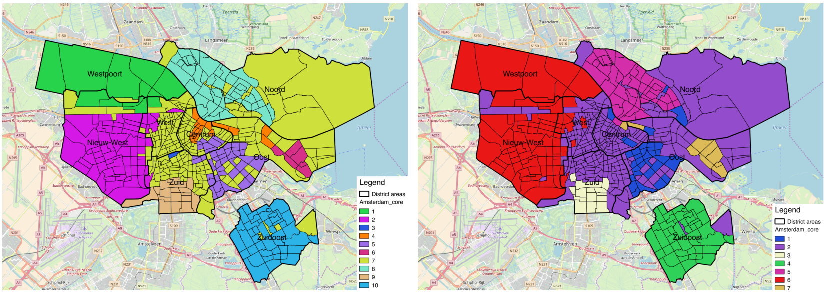
(a) Core cluster weekend

(b) Core cluster dendrogram weekend

Figure 8.14. Core clusters for the weekend data set.

Period	All	Week	Weekend
April	8	9	8
May	9	9	9
June	9	10	8
July	9	9	8
August	10	10	9
September	7	8	7
Total	9	9	9

Table 8.7. Number of clusters in the final core cluster result.



(a) Core cluster of August **(b)** Core cluster of September

Figure 8.15. Core clusters for the complete data set.

Period	All	Week	Weekend
Total	0.51	0.70	0.51
April	0.63	0.67	0.95
May	0.69	0.57	0.96
June	0.62	0.57	0.92
July	0.48	0.69	0.93
August	0.66	0.67	0.94
September	0.57	0.61	0.92

Table 8.8. Cutoff values, representing the highest dissimilarity between cluster branches that were combined to obtain the core cluster results.

centre cluster disappeared. A possible explanation for the absence of this cluster is the end of the tourist season, generating fewer trips in the city centre. As for August, we observe a very prominent cluster at the border of district ‘West’ and ‘Zuid’, consisting of only one neighbourhood. This neighbourhood consists of the largest city park in Amsterdam, which is a famous hotspot during warm weather.

Finally, we can interpret the cutoff values that are used to determine the cluster results. The cutoff value is determined by the dissimilarity value for which we obtain a specific number of clusters, which is equal to the average number of clusters in the ensemble of partitions $\mathcal{P}$. The higher the cutoff value, the higher dissimilarity value to combine the correct number of partitions. As expected, the cutoff value is larger for the monthly weekend data, again confirming our observations that the weekend trip data shows less consistent clusters.

8.7 Conclusion

In this chapter we analysed travel behaviour in Amsterdam based on Origin-Destination travel intensity data. We analysed both the spatial variation as well as the time-dependent variation of the intensity of trips. We then applied a specific clustering method which uses the spatial travel intensity to discover strongly connected regions. This clustering technique heuristically optimises a metric known as modularity. We analysed the consistency between the heuristic optimisation method and used ensemble learning to increase the consistency in the obtained clustered regions. This allowed us to discover deviations in the spatial travel patterns for specific time slices.

The weekly pattern and spatial plots confirm expected behaviour, such as the morning and evening commute. We observed that the trips taken from the metro region of Amsterdam are largely commuting trips. There are three areas that show a high density of trips coming from the Metro region, each of them contains large business districts. However, from this analysis, we also discovered a gap between the total inflow and outflow. As there is no logical explanation for this behaviour, we assume that this occurs due to some transformation to censor the data. In order to properly analyse the data we restored this imbalance, and obtained scaling values for each neighbourhood. This revealed a couple of outliers in the data. Especially in the East of Amsterdam a few neighbourhoods which mostly consist of water showed a large difference between the total inflow and outflow value. With these outliers in mind we continued our analysis.

We were able to identify clusters when the directionality is taken into account. These clusters happen to be very similar to the regional districts defined in Amsterdam. Especially at the outskirts of Amsterdam we can clearly identify clusters. The city centre is represented by one large cluster, together with parts of the east of Amsterdam. When the method is separated into monthly periods, and a division between the weekend and weekday trips is introduced, the results suggest that we observe slightly different clusters in the weekend compared to the week data. Although this is difficult to conclude, given the inconsistency between different instances of the same data partition.

We analysed the results when part of the data is removed. This revealed that a lot of small weight edges can be removed without losing the spatially obtained clusters. We can conclude from the above analysis that trips in Amsterdam are quite homogeneously spread over the city. However, we do observe clustering, although not very prominently. Finally, we used a cluster-ensemble technique to obtain more consistent results, allowing for a better comparison. The results from the cluster ensemble method show minimal deviations between the obtained clusters over the time-dependent subsets, regarding week and weekend. The monthly subsets revealed some differences in the number of partitions obtained in each month. Nevertheless, no big differences in clusters were found between these subsets, suggesting that the partitioning is quite robust over the entire period.

The current cluster results show that the city of Amsterdam is highly connected in terms of travel behaviour, however, by means of clustering on the travel intensity we were able to identify spatially connected regions. These regions correspond to what would be expected. This confirms on one hand that people are inclined to travel within specific regional boundaries. On the other hand it shows that such clustering techniques is a powerful means to detect structure in travel patterns without prior domain knowledge.

For future studies it is interesting to extend the above analysis to discover the main factors that contribute to the appearance of these clusters, i.e., which type of trips contribute to the formation of these regions. An options would be to develop a method for dynamic time-window clustering to see whether we can detect periods in which clusters appear more strongly. Moreover, a segmentation on for example trip purpose (commutes, leisure) can be done to reveal which factors contribute to the occurrence of strongly connected clusters and which do not.

Summary

In this dissertation, we develop models and control techniques for road traffic congestion in which the main focus lies on incorporating the impact of uncertainty by means of quantitative stochastic methods.

The dynamics in road traffic are subject to uncertainty, which is for a large part due to the stochastic nature of behaviour of individuals. As a result, fluctuations in travel demand and road capacity play a major role and have a main impact on the effectiveness of traffic management solutions. Often, these fluctuations are not incorporated in these traffic management solutions. In this dissertation, we focus on the uncertainty from different perspectives, specifically: actuator control, user behaviour, and lastly, network analysis.

Actuator control
In Chapters 2-4, we study control strategies that avoid accumulation of traffic at strategic points in a network. In Chapter 2, we introduce two versions of a tandem queue for which we analyse the interplay between queues when randomness of the arrival process and capacity is incorporated. This tandem queue consists of two queue in series which arrivals pass sequentially. We show that the structure of the optimal control strategy in our tandem queue model is a switching strategy dividing the state space into two regions: one where the service rate is at its maximum, and one where service is paused. Full characterisation of the theoretically optimal strategy becomes complex when the tandem model is extended to include more realistic factors observed in practice. Examples of these factors include relaxation of the exponential service time distribution, the order of service, and extensions to a network of bottlenecks. Numerical computation of the optimal strategy becomes then numerically overly time-consuming, or even infeasible. Therefore, we introduce two approximation approaches in the subsequent chapters.

When the optimal switching curve is rather flat, it can be well approximated by a horizontal one, which corresponds to a fixed threshold strategy. In Chapter 3, we develop an approximation technique to investigate the effectiveness of such fixed threshold strategies. For the ‘optimal’ threshold level, we verify that it performs very closely to the optimal

MDP strategy under medium loaded systems. However, when the load of the system increases, the performance gap between the MDP strategy and the approximation increases. Under heavily loaded systems the structure of the optimal policy becomes more important. Therefore we develop a dynamic approximation strategy in Chapter 4. We use a fluid-based approach that results in two heuristic strategies which provide excellent approximations for a broad range of parameter values. Moreover, the computation time required to obtain the approximation is quite insensitive to the system load.

The average costs of the ‘fixed fluid’ approach remain within a few percents of the average costs of the MDP solution for a wide range of parameters. However, the ‘fixed fluid’ heuristic consistently underestimates the optimal MDP strategy in the states with a large number of jobs in the system, causing less accurate results for heavily loaded systems. The second heuristic, denoted as the ‘greedy fluid’ approach, follows the optimal MDP switching curve quite closely, especially when the number of jobs in the system becomes large. For higher load systems, the ‘greedy’ approach outperforms the ‘fixed’ heuristic method as well as the fixed threshold method of Chapter 3. Moreover, the proposed fluid approach is computationally very fast. This suggests that the approach is worth exploring for larger queueing networks with non-exponential service times.

User behaviour

In practice, travellers can strategically choose their departure times and the routes they take. Congestion occurs when more users simultaneously access the infrastructure than can be sustained by that infrastructure.

The models in Chapters 5 and 6 are based on a popular and simplistic approach to model congestion and user response, where the main goal is to find compatible departure times of travellers, such that all travellers suffer the same discomfort. This discomfort is expressed in a cost function that accounts for three cost components: the cost of being too early at the destination, the cost of arriving too late and the cost of the actual total travelling time; the latter component is determined by the delay due to traffic congestion. The compatible departure times are found by the Nash equilibrium, in which no traveller can improve its costs by shifting its departure time.

In Chapter 5 this model is extended with stochastic (uncertain) arrival times and travelling speeds by using a Poisson arrival process with time-fluctuating rate and exponential travel times. The strategic behaviour of users is captured in the aggregated intensity function of the Poisson arrival process. We discuss the error made by the fluid approximation, and show that the Nash equilibrium of the original model results in highly varying costs when applied in the more realistic setting with stochasticity. We then develop an algorithm to numerically approximate the equilibrium arrival rate for the stochastic bottleneck model, and propose a closed-form estimation for the approximated equilibrium. Our approach can be applied to many extensions that exist of the standard deterministic bottleneck model giving intuition of the impact of uncertainty in a broad range of transportation models. Examples include heterogeneity among travellers' departure time, interpretation of early and late arrival, and demand elasticity.

In Chapter 6, we use a more detailed model for the rational behaviour of travellers: each can strategically choose a preferred time to join the bottleneck, but the actual time at which the bottleneck is reached is subject to a random shift in time. This captures uncertainty with respect to departure and travelling times prior to joining the bottleneck. We show that the arrival density advocated by the Nash equilibrium in Vickrey's model is not a user equilibrium in the model with random uncertainty. We then investigate the existence of a user equilibrium for the latter and show that, in general, such an equilibrium can neither be a pure Nash equilibrium, nor a mixed equilibrium with a continuous density. With numerical examples, we illustrate the mechanics that prevent existence of such a user equilibrium. Our results demonstrate that when random distortions influence user decisions, the dynamics of standard bottleneck models are inadequate to describe such complex situations.

The strategic scheduling model in Chapter 7 is similar to that of Chapter 5. As in the previous two chapters, the goal is to dynamically spread arrivals, but now travel times are optimised in a joint effort between travellers and a central coordinator. The central coordination allows for effective synchronisation of travellers' preferences. For this study, we split the travellers into two groups: (1) participating travellers whose departure time interval can be adjusted, and (2) non-participating 'background' travellers whose departure times cannot be adjusted. This allows us to assess the impact of the fraction 'adjustable traffic' on the total delay. Our results show that a significant decrease in average delay can be established when only a small fraction of the total traffic uses a personal departure advice.

Network analysis

In Chapter 8, we examine the structure of an empirical data set consisting of time-dependent origin-destination pairs in terms of connectedness. A network partitioning algorithm is applied to aggregate travel patterns into high-level partitions of the network. These partitions are composed of historical travel movements in the city of Amsterdam. We show that we can distinguish spatially connected regions when we use a heuristic method that optimises a performance metric called modularity. We proceed to analyse variations in the partitions that arise due to the non-optimal greedy optimisation method. We use a method known as ensemble learning to combine these variations by means of the overlap in community partitions. Ultimately, the combined partition leads to a more consistent result when evaluated again, compared to the individual partitions.

Samenvatting

In dit proefschrift ontwikkelen we modellen en controletechnieken voor files in het wegverkeer. Hierbij ligt de nadruk op de integratie van de impact van onzekerheid door middel van kwantitatieve stochastische methoden. De dynamiek in het wegverkeer is onderhevig aan onzekerheid, deze is voor een voor een groot deel te wijten aan de stochastische aard van het keuzegedrag van individuen. Een gevolg hiervan is dat schommelingen in het verkeersaanbod en de wegcapaciteit een grote impact hebben op de effectiviteit van verkeersmanagement oplossingen. In veel verkeersmanagement oplossingen worden deze schommelingen vaak niet meegenomen. In dit proefschrift richten we ons op de onzekerheid vanuit verschillende perspectieven, in het bijzonder: aansturing vanuit wegkantsystemen, gebruikersgedrag en netwerk analyse.

Controlestrategie van wegkantsystemen
In de hoofdstukken 2-4 bestuderen we controlestrategieën om accumulatie van verkeer op strategische punten in een netwerk te voorkomen of te reduceren. In hoofdstuk 2 introduceren we twee versies van een tandem wachtrijmodel in andere woorden, we beschouwen een systeem waarbij een gebruiker sequentieel twee wachtrijen doorloopt alvorens het systeem te verlaten. In dit model analyseren we het samenspel tussen deze twee wachtrijen onder invloed van fluctuaties in het verkeersaanbod en de wegcapaciteit. Door middel van een controle mechanisme tussen de eerste en de tweede rij, kunnen we de accumulatie van de wachtrij voorkomen door in te grijpen. We laten zien dat de structuur van de optimale controle strategie in ons tandem wachtrijmodel bestaat uit een schakelstrategie die de toestandsruimte verdeelt in twee regio's: een waar de doorstroom op maximaal gezet wordt, en een waarbij de doorstroom gepauzeerd wordt. Een volledige karakterisering van de optimale strategie wordt al snel complex wanneer we het model uitbreiden met realistische praktijk factoren. Voorbeelden van praktijkfactoren zijn onder meer versoepeling van de exponentiële verdelingsduur van de doorstroomsnelheid, de volgorde van afhandeling en uitbreidingen naar een netwerk van meerdere wachtrijen. Het numeriek berekening van de optimale strategie wordt dan in dit geval zeer tijdrovend of zelfs

onhaalbaar. Daarom introduceren we twee manieren om de optimale strategie te benaderen in de volgende hoofdstukken. Wanneer de optimale schakelstrategie vrij monotoom is, kan deze goed worden benaderd door middel van een horizontale lijn. Een zogenaamde vaste drempelwaardestrategie. In hoofdstuk 3 ontwikkelen we een benadering om de effectiviteit van een dergelijke vaste drempelstrategie te bepalen. Hieruit volgt dat de prestatie van het 'optimale' drempelniveau zeer dicht bij de optimale MDP-strategie ligt voor systemen van middelhoge belasting. Echter, wanneer de belasting van het systeem toeneemt, neemt de prestatiekloof tussen de MDP-strategie en de benadering drastisch toe. Voor zwaar belaste systemen wordt de exacte structuur van de optimale strategie belangrijker. Daarom ontwikkelen we een dynamische benaderingsstrategie in hoofdstuk 4. In dit hoofstuk gebruiken we een vloeistofbenadering. We beschrijven twee heuristische strategieën die uitstekende benaderingen weergeven voor een grote set van parameterwaarden. Bovendien is de benodigde rekentijd voor deze benadering is slechts een fractie van de rekentijd voor de MDP-oplossing en is deze ook vrij ongevoelig voor de systeembelasting. De gemiddelde kosten van de eerste heuristische benadering, de 'vaste vloeistof'-benadering, wijkt slechts met een paar procent af van de MDP-oplossing voor een breed scala van parameters. Echter, de 'vaste vloeistof' heuristiek onderschat consequent de optimale MDP-strategie wanneer het systeem zich in een grotere toestand bevind. Dit resulteert inminder nauwkeurige resultaten voor zwaarbeladen systemen aangezien deze zich vaker in hogere toestanden zullen bevinden. De tweede heuristiek, aangeduid als de 'greedy fluid'-benadering, volgt de optimale MDP-strategie vrij nauwkeurig, met name wanneer het systeem zich in grote toestanden bevind. Ook presteert deze 'greedy fluid'-benadering voor zwaarbeladen system beter dan de drempelmethode van hoofdstuk 3. Bovendien is de vloeistof aanpak computationeel zeer efficient. Dit suggereert dat deze aanpak interessant is om te verkennen voor grotere wachtrijnetwerken met niet-exponentiële servicetijden.

Gebruikersgedrag

In de praktijk baseren reizigers hun vertrektijden en routekeuzes vaak vanuit een strategisch perspectief met betrekking tot reisduur en tijd van aankomst. Files treden op wanneer meerdere gebruikers tegelijkertijd gebruikmaken van de infrastructuur dan de capaciteit toelaat. Deze plekken worden gezien als knelpunten. De modellen in hoofdstuk 5 en

6 zijn gebaseerd op een populaire en simplistische benadering van het ontstaan van knelpunten door strategisch reizigersgedrag. Dit model gaat ervan uit dat de strategische keuzes van reiziger resulteren in een aankomstpatroon waarbij elke reiziger hetzelfde ongemak ervaart. Dit ongemak wordt uitgedrukt door middel van een kostenfunctie bestaande uit drie componenten: kosten voor te vroeg arriveren op de bestemming, kosten voor het te laat arriveren op de bestemming en kosten voor extra reistijd; het laatste component wordt bepaald door de vertraging als gevolg van de onstane file. Er wordt gezocht naar een Nash-evenwicht in de vertrektijden, wat betekent dat geen enkele reiziger zijn kosten kan verbeteren door zijn vertrektijd aan te passen. In hoofdstuk 5 wordt dit model uitgebreid naar een model met stochastiek in de aankomstintensiteit en doorstroomsnelheden. Er wordt hierin gebruik gemaak van de aanname dat aankomsten gemodelleerd kunnen worden volgens een Poisson-aankomstproces met tijd- fluctuerende snelheid en dat reistijden van exponentiële duur zijn. Het strategische gedrag van gebruikers wordt vastgelegd op basis van een geaggregeerde intensiteitsfunctie van het Poisson aankomstproces. We bespreken hierin de fout die gemaakt wordt door de deterministische aanpak, en laten zien dat het Nash-evenwicht van het oorspronkelijke model resulteert in sterk variërende kosten in de stochastische setting. Vervolgens benaderen we numeriek het Nash-evenwicht voor het stochastische model en onwikkelen we een methodiek die een schatting van het aankomstpatroon in een gesloten vorm formule geeft voor het benaderde evenwicht. Bovenstaande aanpak kan worden toegepast op een groot aantal extensies van het originele deterministische model. Dit leidt tot een intuïtie van de impact van onzekerheid in een breed scala aan vervoersmodellen. Voorbeelden van extensies zijn onder meer heterogeniteit in de vertrektijd van reizigers, de interpretatie van vroege en late aankomst, vraagelasticiteit, etcetera. In hoofdstuk 6 gebruiken we een meer gedetailleerd model voor het rationele gedrag van reizigers: in dit model kiest elke reiziger strategisch een voorkeurstijd om te arriveren bij het knelpunt. Echter is het werkelijke tijdstip waarop de reiziger het knelpunt bereikt onderhevig aan een zekere verschuiving in tijd waar de reiziger geen invloed op heeft. Hiermee wordt onzekerheid in de exacte vertrek- en reistijden voorafgaand aan toetreding tot het knelpunt meegenomen. Wij laten zien dat de aankomstintensiteit die leidt tot een Nash-evenwicht in het originele model niet tot een evenwicht leidt in het model met onzekerheid in de exacte aankomsttijd.

Vervolgens onderzoeken we het bestaan van een Nash-evenwicht in onze uitbreiding van het model. We laten zien dat voor dit model in ieder geval geen zuiver Nash-evenwicht bestaat, noch dat er een gemengd evenwicht met een continue aankomstdichtheid bestaat. Met numerieke voorbeelden illustreren we de mechaniek die het bestaan van een dergelijk gebruikersevenwicht voorkomt. Onze resultaten demonstreren dat wanneer willekeurige verstoringen gebruikersbeslissingen beïnvloeden, de dynamiek van standaard knelpuntmodellen ontoereikend zijn om evenwichtsituatie te vinden. Het strategische planningsmodel wat beschrevn is in hoofdstuk 7 is in de modellering vergelijkbaar met dat van hoofdstuk 5. Net als in de vorige twee hoofdstukken, gaan we uit van een tijdsafhankelijk aankomstpatroon onderhevig aan stochastiek. In dit geval kiezen reiziger niet meer strategisch hun eigen vertrekmoment, maar worden reistijden geoptimaliseerd in een gezamenlijke inspanning tussen de reizigers door middel van een centrale coördinator. Deze centrale coördinator zorgt voor een effectieve synchronisatie van de voorkeuren van reizigers. Voor deze studie, splitsen we de reizigers in twee groepen: (1) deelnemende reizigers wiens vertrektijd kan worden aangepast, en (2) niet-deelnemend 'achtergrond'-reizigers waarvan de vertrektijden niet kunnen worden aangepast. Deze stelt ons in staat om de impact van de fractie 'aanpasbaar verkeer' op de totale vertraging in kaart te brengen. Onze resultaten tonen aan dat een aanzienlijke afname van de gemiddelde vertraging kan worden vastgesteld wanneer slechts een kleine fractie van het totale verkeer bestaat uit deelnemende reizigers.

Netwerk analyse

In Hoofdstuk 8 onderzoeken we de structuur en de verbondenheid van een empirische dataset die bestaat uit tijdsafhankelijke oorsprong-bestemmingsparen op basis van historische reisbewegingen in Amsterdam. We laten zien dat we door middel van een netwerkclustering algoritme reispatronen kunnen aggregeren naar een hoger abstractie niveau om de structuur van het netwerk te ontdekken. We laten zien dat we geografische verbonden gebieden kunnen onderscheiden door het toepassen van een heuristisch clusteringsalgoritme welke een bekende prestatiemetriek optimaliseert genaamd modulariteit. Door de heuristische aanpak observeren we variaties in de resulterende clusters. Om tot een constanter eindresultaat te komen maken we gebruik van een ensemble methodiek. Deze methodiek combineert de gevonden variaties door middel van meest voorkomende overlappingen. Dit leidt tot een gecombineerde partitie die de variaties in de resultaten reduceert.

Publications of the author

[S1] D. van Leeuwen and R. Núñez-Queija, 'Near-optimal switching strategies for a tandem queue'. In: *Markov Decision Processes in Practice*, eds. R.J. Boucherie and N.M. van Dijk (2017), pp. 439-459.

[S2] D. van Leeuwen and R. Núñez-Queija, 'Optimal dispatching in a tandem queue'. In: *Queueing Systems* (2017), Vol.87.3-4 pp. 269-291.

[S3] D. van Leeuwen and P. van de Ven, 'Modelling user behaviour at a stochastic road traffic bottleneck'. In: *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools* (2017), pp. 140-147, ACM.

[S4] D. van Leeuwen, S. Ghazanfari, L. Ravner and R. Núñez-Queija, 'Modeling of uncertainty of the arrival time at a bottleneck'. To be submitted.

[S5] D. van Leeuwen, J.W. Bosman and E.R. Dugundji, 'Spatio-temporal clustering of time-dependent origin-destination electronic trace data'. In: *Procedia Computer Science*, 130 (2018), pp. 359-367, Elsevier.

[S6] D. van Leeuwen, J.W. Bosman and E.R. Dugundji, 'Network partitioning on time-dependent origin-destination electronic trace data'. Submitted to *Personal and Ubiquitous Computing.*

[S7] D. van Leeuwen, M. Hoogeboom, R.D. van der Mei and F. Ottenhof. 'Demand spreading in overcrowded neighbourhoods: a personal departure advice'. To be submitted.

References

[1] R. Akçelik. 'Time-dependent expressions for delay, stop rate and queue length at traffic signals'. In: *Australian Road Research Board, Internal report AIR 367-1.* (1980).

[2] H. Almeida, D. Guedes, W. Meira and M.J. Zaki. 'Is there a best quality metric for graph clusters?' In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer. 2011, pp. 44–59.

[3] L.N.F. Ana and A.K. Jain. 'Robust data clustering'. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on.* Vol. 2. IEEE. 2003.

[4] R. Arnott, A. De Palma and R. Lindsey. 'A structural model of peak-period congestion: A traffic bottleneck with elastic demand'. In: *The American Economic Review* (1993), pp. 161–179.

[5] R. Arnott, A. De Palma and R. Lindsey. 'Economics of a bottleneck'. In: *Journal of Urban Economics* 27.1 (1990), pp. 111–130.

[6] R. Arnott, A. de Palma and R. Lindsey. 'Schedule delay and departure time decisions with heterogeneous commuters'. In: *Transportation Research Record* 1197 (1988).

[7] F. Avram. 'Optimal control of fluid limits of queuing networks and stochasticity corrections'. In: *Mathematics of Stochastic Manufacturing Systems: AMS-SIAM Summer Seminar in Applied Mathematics, June 17-22, 1996, Williamsburg, Virginia.* Vol. 33. American Mathematical Soc. 1997.

[8] N. Baer, R.J. Boucherie and J. van Ommeren. 'The PH/PH/1 multi-threshold queue'. In: *International Conference on Analytical and Stochastic Modeling Techniques and Applications.* Springer. 2014, pp. 95–109.

[9] R. Bellman. *Dynamic Programming.* Courier Corporation, 2013.

[10] R. Bellman. 'The theory of dynamic programming'. In: *Bulletin of the American Mathematical Society* 60.6 (1954), pp. 503–515.

[11] M.E. Ben-Akiva and S.R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand.* Vol. 9. MIT press, 1985.

[12] D.P. Bertsekas. *Dynamic Programming and Optimal Control.* Vol. 1.2. Athena Scientific Belmont, MA, 1995.

[13] V. Blondel, G. Krings and I. Thomas. 'Regions and borders of mobile telephony in Belgium and in the Brussels metropolitan zone'. In: *Brussels Studies. The e-journal for academic research on Brussels* (2010).

[14] V.D. Blondel, J. Guillaume, R. Lambiotte and E. Lefebvre. 'Fast unfolding of communities in large networks'. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008).

[15] V.D. Blondel, J.L. Guillaume, R. Lambiotte and E. Lefebvre. 'The Louvain method for community detection in large networks'. In: *Journal of Statistical Mechanics: Theory and Experiment* 10 (2011).

[16] Transportation Research Board. *Foundations of Traffic Flow Theory: The Fundamental Diagram.* Vol. E-C149. Transportation Research Circular, 2011.

[17] J. Bongaarts. 'United Nations, department of economic and social affairs, population division.' In: *Population and Development Review* 40.2 (2014), pp. 380–380.

[18] L. Bortolussi and M. Tribastone. 'Fluid limits of queueing networks with batches'. In: *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering.* ACM. 2012, pp. 45–56.

[19] D. Braess. 'Über ein Paradoxon aus der Verkehrsplanung'. In: *Unternehmensforschung* 12.1 (1968), pp. 258–268.

[20] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski and D. Wagner. *On modularity-np-completeness and beyond.* Univ., Fak. für Informatik, Bibliothek, 2006.

[21] W. Brilon. 'Time dependent delay at unsignalized intersections'. In: *Proceedings of the 17th International Symposium on Transportation and Traffic Theory (ISTTT17), London, Elsevier.* 2007, pp. 555–582.

[22] *CBS wijk- en buurtkaart.* 2017. URL: `https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/geografische%20data/wijk-en-buurtkaart-2017`.

[23] C. Chen, A. Skabardonis and P. Varaiya. 'Travel-time reliability as a measure of service'. In: *Transportation Research Record: Journal of the Transportation Research Board* 1855 (2003), pp. 74–79.

[24] Y. Chiu, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller and J. Hicks. 'Dynamic traffic assignment: A primer'. In: *Transportation Research E-Circular* E-C153 (2011).

[25] C.G. Chorus, E.J.E. Molin and B. Van Wee. 'Use and effects of Advanced Traveller Information Services (ATIS): a review of the literature'. In: *Transport Reviews* 26.2 (2006), pp. 127–149.

[26] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & sons, 2012.

[27] M. Cremer and J. Ludwig. 'A fast simulation model for traffic flow on the basis of Boolean operations'. In: *Mathematics and Computers in Simulation* 28.4 (1986), pp. 297–303.

[28] G.L. Curry and R.M. Feldman. 'An M/M/1 queue with a general bulk service rule'. In: *Naval Research Logistics (NRL)* 32.4 (1985), pp. 595–603.

[29] C.F. Daganzo. 'Improving city mobility through gridlock control: an approach and some ideas'. In: *UC Berkeley online repository* (2005).

[30] C.F. Daganzo. 'Urban gridlock: Macroscopic modeling and mitigation approaches'. In: *Transportation Research Part B: Methodological* 41.1 (2007), pp. 49–62.

[31] K. Das, S. Samanta and M. Pal. 'Study on centrality measures in social networks: a survey'. In: *Social Network Analysis and Mining* 8.1 (2018).

[32] W.H.E. Day and H. Edelsbrunner. 'Efficient algorithms for agglomerative hierarchical clustering methods'. In: *Journal of Classification* 1.1 (1984), pp. 7–24.

[33] A. De Palma and M. Fosgerau. 'Random queues and risk averse users'. In: *European Journal of Operational Research* 230.2 (2013), pp. 313–320.

[34] R.K. Deb. 'Optimal control of batch service queues with switching costs'. In: *Advances in Applied Probability* 8.1 (1976), pp. 177–194.

[35] R.K. Deb and R.F. Serfozo. 'Optimal control of batch service queues'. In: *Advances in Applied Probability* 5.2 (1973), pp. 340–361.

[36] N.M. van Dijk. *Queueing Networks and Product Forms: a Systems Approach.* Vol. 4. John Wiley & Son Limited, 1993.

[37] E.W. Dijkstra. 'A note on two problems in connexion with graphs'. In: *Numerische Mathematik* 1.1 (1959), pp. 269–271.

[38] N. Dugué and A. Perez. 'Directed Louvain: maximizing modularity in directed networks'. PhD thesis. Université d'Orléans, 2015.

[39] E. Dugundji and J. Walker. 'Discrete choice with social and spatial network interdependencies: an empirical example using mixed generalized extreme value models with field and panel effects'. In: *Transportation Research Record* 1921.1 (2005), pp. 70–78.

[40] A. El-Rayes, M. Kwiatkowska and G. Norman. 'Solving infinite stochastic process algebra models through matrix-geometric methods'. In: *School of computer science research reports (CSR) - University of Birmingham* (1999).

[41] M. van Essen, T. Thomas, E. van Berkum and C. Chorus. 'From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels'. In: *Transport Reviews* 36.4 (2016), pp. 527–548.

[42] Eurostat. *Urban Europe - Statistics on cities, towns and suburbs.* Luxembourg: Publications office of the European Union, 2016.

[43] S. Fortunato. 'Community detection in graphs'. In: *Physics Reports* 486.3 (2010), pp. 75–174.

[44] M. Fosgerau. *Congestion costs in bottleneck equilibrium with stochastic capacity and demand.* Tech. rep. University Library of Munich, Germany, 2008.

[45] M. Fosgerau. ‘On the relation between the mean and variance of delay in dynamic queues with random capacity and demand’. In: *Journal of Economic Dynamics and Control* 34.4 (2010), pp. 598–603.

[46] M. Fosgerau, L. Engelson and J.P. Franklin. ‘Commuting for meetings’. In: *Journal of Urban Economics* 81 (2014), pp. 104–113.

[47] M. Fosgerau and A. Karlström. ‘The value of reliability’. In: *Transportation Research Part B: Methodological* 44.1 (2010), pp. 38–49.

[48] M. Fosgerau and K.A. Small. ‘Endogenous scheduling preferences and congestion’. In: *International Economic Review* 58.2 (2017), pp. 585–615.

[49] S. Foss and A. Kovalevskii. ‘A stability criterion via fluid limits and its application to a polling system’. In: *Queueing Systems* 32.1-3 (1999), pp. 131–168.

[50] A.L.N. Fred and A.K. Jain. ‘Combining multiple clusterings using evidence accumulation’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.6 (2005), pp. 835–850.

[51] L.C. Freeman. ‘Centrality in social networks conceptual clarification’. In: *Social Networks* 1.3 (1978), pp. 215–239.

[52] A. Gajrat, A. Hordijk and A. Ridder. ‘Large-deviations analysis of the fluid approximation for a controllable tandem queue’. In: *Annals of Applied Probability* (2003), pp. 1423–1448.

[53] A. Geyer-Schulz and M. Ovelgönne. ‘The randomized greedy modularity clustering algorithm and the core groups graph clustering scheme’. In: *German-Japanese Interchange of Data Analysis Results*. Springer, 2014, pp. 17–36.

[54] D. Gfeller, J. Chappelier and P. De Los Rios. ‘Finding instabilities in the community structure of complex networks’. In: *Physical Review E* 72.5 (2005).

[55] A. Glazer and R. Hassin. ‘?/M/1: On the equilibrium distribution of customer arrivals’. In: *European Journal of Operational Research* 13.2 (1983), pp. 146–150.

[56] A. Glazer, R. Hassin and L. Ravner. ‘Equilibrium and efficient clustering of arrival times to a queue’. In: *arXiv preprint arXiv:1701.04776* (2017).

[57] B.D. Greenshields, W. Channing, H. Miller et al. ‘A study of traffic capacity’. In: *Highway Research Board Proceedings*. Vol. 14. National Research Council (USA), Highway Research Board. 1935.

[58] O. Guéant, J. Lasry and P. Lions. ‘Mean field games and applications’. In: *Paris-Princeton Lectures on Mathematical Finance* (2011), pp. 205–266.

[59] F.L. Hall and K. Agyemang-Duah. ‘Freeway capacity drop and the definition of capacity’. In: *Transportation Research Record* 1320 (1991).

[60] K. van Harn and F.W. Steutel. ‘Infinite divisibility and the waiting-time paradox’. In: *Stochastic Models* 11.3 (1995), pp. 527–540.

[61] R. Hassin and Y. Kleiner. ‘Equilibrium and optimal arrival patterns to a server with opening and closing times’. In: *IIE Transactions* 43.3 (2010), pp. 164–175.

[62] M. Haviv. ‘When to arrive at a queue with tardiness costs?’ In: *Performance Evaluation* 70.6 (2013), pp. 387–399.

[63] J.V. Henderson. ‘Road congestion: a reconsideration of pricing theory’. In: *Journal of Urban Economics* 1.3 (1974), pp. 346–365.

[64] Chris Hendrickson and George Kocur. ‘Schedule delay and departure time decisions in a deterministic model’. In: *Transportation science* 15.1 (1981), pp. 62–77.

[65] S.P. Hoogendoorn and P.H.L. Bovy. ‘State-of-the-art of vehicular traffic flow modelling’. In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 215.4 (2001), pp. 283–303.

[66] V Hurdle, E Hauser and Ph Fargier. ‘Effects of the choice of departure time on road traffic congestion. Theoretical approach’. In: *Transportation and traffic theory* 8 (1983), pp. 223–263.

[67] T.P. Hutchinson. ‘Delay at a fixed time traffic signal-II: Numerical comparisons of some theoretical expressions’. In: *Transportation Science* 6.3 (1972), pp. 286–305.

[68] J.R. Jackson. ‘Networks of waiting lines’. In: *Operations Research* 5.4 (1957), pp. 518–521.

[69] R. Jain and J.M. Smith. ‘Modeling vehicular traffic flow using M/G/C/C state dependent queueing models’. In: *Transportation Science* 31.4 (1997), pp. 324–336.

[70] S. Juneja and R. Jain. ‘The concert/cafeteria queueing problem: a game of arrivals’. In: *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools.* ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2009, pp. 1–59.

[71] D.G. Kendall. ‘Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain’. In: *The Annals of Mathematical Statistics* (1953), pp. 338–354.

[72] B. Kerner. ‘Theory of breakdown phenomenon at highway bottlenecks’. In: *Transportation Research Record* 1710 (2000), pp. 136–144.

[73] R.M. Kimber and E.M. Hollis. *Traffic queues and delays at road junctions.* Tech. rep. Transport and Road Research Laboratory, 1979.

[74] L. Kleinrock. *Queueing Systems, volume 2: Computer Applications.* Vol. 66. Wiley New York, 1976.

[75] G. Koole. 'Convexity in tandem queues'. In: *Probability in the Engineering and Informational Sciences* 18.1 (2004), pp. 13–31.

[76] T.G. Kurtz. 'Solutions of ordinary differential equations as limits of pure jump Markov processes'. In: *Journal of Applied Probability* 7.1 (1970), pp. 49–58.

[77] N. Kyurkchiev and S. Markov. 'Sigmoid functions: some approximation and modelling aspects'. In: *Some Moduli in Programming Environment Mathematica* (2015).

[78] R. Lamotte and N. Geroliminis. 'The morning commute in urban areas with heterogeneous trip lengths'. In: *Transportation Research Procedia* 23 (2017), pp. 591–611.

[79] A. Lancichinetti and S. Fortunato. 'Community detection algorithms: a comparative analysis'. In: *Physical Review E* 80.5 (2009).

[80] A. Lancichinetti, S. Fortunato and J. Kertész. 'Detecting the overlapping and hierarchical community structure in complex networks'. In: *New Journal of Physics* 11.3 (2009).

[81] A. Lancichinetti, S. Fortunato and F. Radicchi. 'Benchmark graphs for testing community detection algorithms'. In: *Physical review E* 78.4 (2008).

[82] A. Landherr, B. Friedl and J. Heidemann. 'A critical review of centrality measures in social networks'. In: *Business & Information Systems Engineering* 2.6 (2010), pp. 371–385.

[83] M. Larrañaga, O.J. Boxma, R. Núñez-Queija and M.S. Squillante. 'Efficient content delivery in the presence of impatient jobs'. In: *International Teletraffic Congress (ITC 27), 2015 27th International*. IEEE. 2015, pp. 73–81.

[84] G. Latouche and M.F. Neuts. 'Efficient algorithmic solutions to exponential tandem queues with blocking'. In: *SIAM Journal on Algebraic Discrete Methods* 1.1 (1980), pp. 93–106.

[85] G. Latouche and V. Ramaswami. 'A logarithmic reduction algorithm for quasi-birth-death processes'. In: *Journal of Applied Probability* 30.3 (1993), pp. 650–674.

[86] J.S.H. van Leeuwaarden. 'Delay analysis for the fixed-cycle traffic-light queue'. In: *Transportation Science* 40.2 (2006), pp. 189–199.

[87] E.A. Leicht and Mark E.J. Newman. 'Community structure in directed networks'. In: *Physical Review Letters* 100.11 (2008), p. 118703.

[88] M.J. Lighthill and G.B. Whitham. 'On kinematic waves. I. Flood movement in long rivers'. In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* (1955), pp. 281–316.

[89] M.J. Lighthill and G.B. Whitham. 'On kinematic waves II. A theory of traffic flow on long crowded roads'. In: *Proceedings of the Royal Society*

of London. Series A, Mathematical and Physical Sciences 229 (1955), pp. 317–345.

[90] H. van Lint, O. Miete, H. Taale and S. Hoogendoorn. 'Systematic framework for assessing traffic measures and policies on reliability of traffic operations and travel time'. In: *Transportation Research Record* 2302 (2012), pp. 92–101.

[91] J.W.C. van Lint, H.J. Van Zuylen and H. Tu. 'Travel time unreliability on freeways: Why measures based on variance tell only half the story'. In: *Transportation Research Part A: Policy and Practice* 42.1 (2008), pp. 258–277.

[92] S.A. Lippman. 'Applying a new device in the optimization of exponential queuing systems'. In: *Operations Research* 23.4 (1975), pp. 687–710.

[93] J.D.C. Little. 'A proof for the queuing formula: L= λ W'. in: *Operations Research* 9.3 (1961), pp. 383–387.

[94] M. Luis, L.M. Correia and K. Wünstel. 'Smart Cities Applications and Requirements'. In: *Net! Works European Technology Platform* (2011).

[95] H. Mahmassani, T. Hou and J. Dong. 'Characterizing travel time variability in vehicular traffic networks: deriving a robust relation for reliability analysis'. In: *Transportation Research Record: Journal of the Transportation Research Board* 2315 (2012), pp. 141–152.

[96] H.S. Mahmassani. 'Dynamic network traffic assignment and simulation methodology for advanced system management applications'. In: *Networks and Spatial Economics* 1.3-4 (2001), pp. 267–292.

[97] H.S. Mahmassani and S. Peeta. *Network Performance Under System Optimal and User Equilibrium Dynamic Assignment: Implications for ATIS.* Transportation Research Board, 1993.

[98] M. Maness, C. Cirillo and E.R. Dugundji. 'Generalized behavioral framework for choice models of social influence: Behavioral and data concerns in travel behavior'. In: *Journal of Transport Geography* 46 (2015), pp. 137–150.

[99] Highway Capacity Manual. 'Transportation research board'. In: *National Research Council, Washington, DC* 113 (2000).

[100] S. Meyn. *Control Techniques for Complex Networks.* Cambridge University Press, 2008.

[101] R.B. Myerson. *Game Theory.* Harvard university press, 2013.

[102] K. Nagel and M. Schreckenberg. 'A cellular automaton model for freeway traffic'. In: *Journal de Physique I* 2.12 (1992), pp. 2221–2229.

[103] J.F. Nash et al. 'Equilibrium points in n-person games'. In: *Proceedings of the National Academy of Sciences* 36.1 (1950), pp. 48–49.

[104] M.F. Neuts. 'A general class of bulk queues with Poisson input'. In: *The Annals of Mathematical Statistics* 38.3 (1967), pp. 759–770.

[105] M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach.* Courier Corporation, 1981.

[106] G.F. Newell. 'The morning commute for nonidentical travelers'. In: *Transportation Science* 21.2 (1987), pp. 74–88.

[107] G.Frank. Newell. 'A simplified car-following theory: a lower order model'. In: *Transportation Research Part B: Methodological* 36.3 (2002), pp. 195–205.

[108] M.E.J. Newman and M. Girvan. 'Finding and evaluating community structure in networks'. In: *Physical Review E* 69.2 (2004), p. 026113.

[109] R.B. Noland and J.W. Polak. 'Travel time variability: a review of theoretical and empirical issues'. In: *Transport Reviews* 22.1 (2002), pp. 39–54.

[110] R.B. Noland and K.A. Small. 'Travel-time uncertainty, departure time choice, and the cost of morning commutes'. In: *Transportation Research Record* 1493 (1995), pp. 150–158.

[111] J.R. Norris. *Markov Chains.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[112] C. Osorio and M. Bierlaire. 'An analytic finite capacity queueing network model capturing the propagation of congestion and blocking'. In: *European Journal of Operational Research* 196.3 (2009), pp. 996–1007.

[113] M. Ostrovsky and M. Schwarz. 'Synchronization under uncertainty'. In: *International Journal of Economic Theory* 2.1 (2006), pp. 1–16.

[114] M. Ovelgönne and A. Geyer-Schulz. 'An ensemble learning strategy for graph clustering.' In: *Graph Partitioning and Graph Clustering* 588 (2012).

[115] H.J. Payne. 'Models of Freeway Traffic and Control.' In: *Mathematical Models of Public Systems* (1971).

[116] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, 2014.

[117] S.S. Rao. *Engineering Optimization: Theory and Practice.* John Wiley & Sons, 2009.

[118] C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton and S.H. Strogatz. 'Redrawing the map of Great Britain from a network of human interactions'. In: *Public Library of Sciences* 5.12 (2010).

[119] F.J. Richards. 'A flexible growth function for empirical use'. In: *Journal of Experimental Botany* 10.2 (1959), pp. 290–301.

[120] P.I. Richards. 'Shock waves on the highway'. In: *Operations Research* 4.1 (1956), pp. 42–51.

[121] P. Robert. *Stochastic Networks and Queues.* Vol. 52. Springer Science & Business Media, 2013.

[122] Z. Rosberg, P. Varaiya and J. Walrand. 'Optimal control of service in tandem queues'. In: *IEEE Transactions on Automatic Control* 27.3 (1982), pp. 600–610.

[123] S.M. Ross. *Stochastic Processes.* Wiley. 1996.

[124] M. Rosvall and C.T. Bergstrom. 'Maps of random walks on complex networks reveal community structure'. In: *Proceedings of the National Academy of Sciences* 105.4 (2008), pp. 1118–1123.

[125] N. Rouphail, A. Tarko and J. Li. 'Traffic flow at signalized intersections'. In: *Traffic Flow Monograph* (2008), pp. 1–32.

[126] A. Scherrer. *Matlab Louvain Implementation.* online. 2008.

[127] E. Sherzer and Y. Kerner. 'When to arrive at a queue with earliness, tardiness and waiting costs'. In: *arXiv preprint arXiv:1709.03374* (2017).

[128] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis: Queues, Communication and Computing.* Vol. 5. CRC Press, 1995.

[129] K.A. Small. 'The bottleneck model: An assessment and interpretation'. In: *Economics of Transportation* 4.1 (2015), pp. 110–117.

[130] K.A. Small. 'The scheduling of consumer activities: work trips'. In: *The American Economic Review* 72.3 (1982), pp. 467–479.

[131] K.A. Small, E.T. Verhoef and R. Lindsey. *The Economics of Urban Transportation.* Routledge, 2007.

[132] M.J. Smith. 'The existence of a time-dependent equilibrium distribution of arrivals at a single bottleneck'. In: *Transportation Science* 18.4 (1984), pp. 385–394.

[133] Y. Takayama and M. Kuwahara. *Scheduling preferences, parking competition, and bottleneck congestion: A model of trip timing and parking location choices by heterogeneous commuters.* Tech. rep. University Library of Munich, Germany, 2016.

[134] R. Talak, D. Manjunath and A. Proutiere. 'Strategic Arrivals to Queues Offering Priority Service'. In: *arXiv preprint arXiv:1704.05986* (2017).

[135] A. Tesselkin and V. Khabarov. 'Estimation of Origin-Destination Matrices Based on Markov Chains'. In: *Procedia Engineering* 178 (2017), pp. 107–116.

[136] H.C. Tijms. *Stochastic Modelling and Analysis: a Computational Appraoch.* John Wiley & sons, New York, 1986.

[137] M.H. Veatch and L.M. Wein. 'Optimal control of a two-station tandem production/inventory system'. In: *Operations Research* 42.2 (1994), pp. 337–350.

[138] W.S. Vickrey. 'Congestion theory and transport investment'. In: *The American Economic Review* 59.2 (1969), pp. 251–260.

[139] F. van Wageningen-Kessels, H. van Lint, K. Vuik and S. Hoogendoorn. 'Genealogy of traffic flow models'. In: *EURO Journal on Transportation and Logistics* 4.4 (2015), pp. 445–473.

[140] S. Waller, J. Schofer and A. Ziliaskopoulos. 'Evaluation with traffic assignment under demand uncertainty'. In: *Transportation Research Record* 1771 (2001), pp. 69–74.

[141] S.T. Waller and A.K. Ziliaskopoulos. 'A chance-constrained based stochastic dynamic traffic assignment model: Analysis, formulation and solution algorithms'. In: *Transportation Research Part C: Emerging Technologies* 14.6 (2006), pp. 418–427.

[142] J.G. Wardrop. 'Some theoretical aspects of road traffic research'. In: *Operational Research Quarterly* 4.4 (1953), pp. 72–73.

[143] R.R. Weber and S. Stidham. 'Optimal control of service rates in networks of queues'. In: *Advances in Applied Probability* 19.1 (1987), pp. 202–218.

[144] F.V. Webster. *Traffic Signal Settings*. Tech. rep. 1958.

[145] H.J. Weiss. 'The computation of optimal control limits for a queue with batch services'. In: *Management Science* 25.4 (1979), pp. 320–328.

[146] B. Wichmann, M. Chen and W. Adamowicz. 'Social networks and choice set formation in discrete choice models'. In: *Econometrics* 4.4 (2016).

[147] T. van Woensel and N. Vandaele. 'Modeling traffic flows with queueing models: a review'. In: *Asia-Pacific Journal of Operational Research* 24.04 (2007), pp. 435–461.

[148] T. van Woensel and N. Vandaele. 'Queueing models for uninterrupted traffic flows'. In: *Proceedings of the 13th Mini-EURO Conference Handling Uncertainty in the Analysis of Traffic and Transportation Systems*. 2002, pp. 636–640.

[149] T. van Woensel, B. Wuyts and N. Vandaele. 'Validating state-dependent queueing models for uninterrupted traffic flows using simulation'. In: *4OR* 4.2 (2006), pp. 159–174.

[150] R.W. Wolff. 'Poisson arrivals see time averages'. In: *Operations Research* 30.2 (1982), pp. 223–231.

[151] W. Xin and D. Levinson. 'Stochastic congestion and pricing model with endogenous departure time selection and heterogeneous travelers'. In: *Mathematical Population Studies* 22.1 (2015), pp. 37–52.

[152] R. Zhang and Y.A. Phillis. 'Fuzzy control of arrivals to tandem queues with two stations'. In: *IEEE Transactions on Fuzzy Systems* 7.3 (1999), pp. 361–367.

[153] S. Zhu, G. Jiang and H.K. Lo. 'Capturing value of reliability through road pricing in congested traffic under uncertainty'. In: *Transportation Research Procedia* 23 (2017), pp. 664–678.