Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 52 (2021) 59–66

Transportation Research Procedia

www.elsevier.com/locate/procedia

23rd EURO Working Group on Transportation Meeting, EWGT 2020, 16-18 September 2020, Paphos, Cyprus

Combining Simulation and Optimisation to Design Reliable Transportation Services with Autonomous Fleets

Rico Kötschau[a,*], Kai Martins-Turner[b], Jan Fabian Ehmke[c], Kai Nagel[b]

[a] *Otto von Guericke University Magdeburg, Management Science Group*
[b] *Technische Universität Berlin, Chair of Transport Systems Planning and Transport Telematics*
[c] *University of Vienna, Department of Business Decisions and Analytics*

Abstract

We develop and investigate reliable services for autonomous freight and passenger transport in urban areas. We consider the individual requirements of travelers for reliable arrival times. To ensure these in an environment of congested urban traffic networks, we investigate how much buffer we need while avoiding excessive idle times. Using the traffic simulation MATSim, we derive time-dependent travel times, integrate them into offline buffer planning, and evaluate the resulting tours. We implement and compare the use of standard deviation and an adapted linear programming model to define buffers and to enable more reliable transportation services at affordable costs.

Keywords: reliable transportation service; traffic simulation; MATSim; vehicle routing problem; autonomous vehicles; stochastic travel times; avoid lateness; reliability buffers; travel time buffers

1. Introduction

Fleets of autonomous vehicles offer the possibility to design new transportation services in a more economic way (Gurumurthy et al., 2019). Among the most important potentials are cost savings and a quick response to plan changes. Due to their high level of connectivity and rapid adaptability to new situations, fleets of autonomous vehicles provide new opportunities. For instance, compared to fleets operated by conventional drivers, premium services could be offered that guarantee reliable arrival times, which could be a significant advantage compared to today's taxi fleets that do not offer any service guarantees because they are dispatched manually.

In this context, we want to develop and investigate concepts for more reliable transportation services in urban areas. This includes the evaluation of different service levels regarding the reliability of arrival and departure times. Reliable

* Corresponding author.
E-mail address: rico.koetschau@univie.ac.at

10.1016/j.trpro.2021.01.091

arrival times are a critical factor in any mobility solution (Redmond et al., 2020), e.g. to reach an appointment in time. We are particularly interested in how to design reliable services with autonomous fleets in an urban environment, where it is challenging to offer premium services with a guaranteed arrival time.

The main reason for delays in urban areas is the congestion of transport infrastructure. When planning transportation services too tightly, it is likely that delays will accumulate over time (Ehmke and Campbell, 2014). This could have an even more dramatic effect on autonomous fleets, as they are not bound to any working hours or breaks and can operate nearly 24/7. If departure and arrival times are to be guaranteed, this will require significant buffers leading to a higher possibility of idle times and additional costs (Tierney et al., 2018). We want to show the potential of smart buffers and motivate further studies in this area.

Of particular importance are time-dependent travel times caused by rush hour traffic. In order to investigate their impact on premium services of autonomous fleets, we combine optimisation with the agent-based traffic simulation MATSim. The combination of optimization and simulation is used to derive realistic traffic demand and urban congestion as well as to evaluate buffer concepts for ensuring reliable arrival and departure times.

In the following, first, we will present related literature and introduce our framework of buffer creation and evaluation. Then, the design of the experiments will be described, using the urban area of Berlin, Germany. For this setting, we will present results comparing different approaches at buffer creation and service levels.

2. Related Literature

The problem at hand is related to vehicle routing problems with stochastic travel times; for a literature review, we refer to Rajabi-Bahaabadi et al. (2019) and Vareias et al. (2019). In the following, we will focus on literature that aims at guaranteeing a certain level of service in the environment of predefined time windows. Specifically, we are interested in the trade-off between ensuring punctual arrival times and extensive usage of transport resources.

In the area of attended home deliveries, Ehmke and Campbell (2014) propose to include buffers in order to maintain delivery time windows. They assume that these time windows are "hard", which means that arrival before or after a time window is not feasible. The corresponding buffers are dynamic and based on the standard deviation of travel times between customers. They consider the accumulation of previous delays as well as buffers naturally arising from waiting at hard time windows. In the context of liner shipping, Tierney et al. (2018) investigate the most reliable compliance with hard port time windows, allowing variable travel speeds of vessels. They show that this can save costs at normal speeds, but speeds can be increased to maintain time window adherence if necessary. While these papers address service levels in conjunction with hard time windows, in practice, time windows are not always perceived as strict. Ta et al. (2014) investigate reliable and efficient solutions to routing problems with soft time windows, taking into account penalties for deviations. Rajabi-Bahaabadi et al. (2019) adopt this approach for correlated travel times.

While above papers assume that time windows are chosen by the customer, we focus on the rare approach that the time windows are assigned and communicated by service providers. These are called "self-imposed time windows", a concept introduced by Jabali et al. (2015). Application examples are craftsmen appointments or standard home deliveries. Jabali et al. (2015) propose a hybrid solution method that works in two steps: first, routes are planned heuristically, and second, the expected departure times are determined by a linear programming model. The self-imposed time windows are then derived from the expected arrival times. The presented approach was developed further by Vareias et al. (2019), who integrate the selection of a suitable time window length into the model.

In the scope of this paper, we will apply and extend methods of reliable planning to the management of autonomous fleets and reliable premium services. Self-imposed time windows as provided by Jabali et al. (2015) will be used to investigate the potential of buffers to establish reliable transportation services. Representing a first step, self-imposed time windows eliminate the problem of dealing with buffers arising from hard time windows being determined by customers. This problem seems to have little presence in the literature so far, given the fact that numerous taxi and shuttle service providers promise a high degree of reliability but do not have the tools to guarantee this in practice.

3. Framework

3.1. Overview

To plan and evaluate schedules for our transportation service in a realistic environment, we propose an integrated framework of optimization and simulation. For the evaluation of schedules, an interface to MATSim (Horni et al., 2016) was developed. MATSim is an open source activity-based Multi-Agent Transport Simulation software for large-scale scenarios. Several extensions are available as well as open-access scenarios. Expected travel times can be derived for planning, and schedules can be evaluated with regard to their actual reliability in an environment of realistic traffic demand and urban congestion. Our approach for creating and evaluating a schedule can be divided into the following five steps.

(1) *Request creation.* The first step is about creating possible travel requests that an autonomous vehicle will perform. Representing a passenger transport service, each request comes with two locations specifying origin and destination. A traveler's request executed by our taxi service is called a service trip. When having finished a service trip, the vehicle performs an empty trip to the next service trip. The schedule for each autonomous vehicle is represented by an ordered set of trips T.
(2) *Travel time derivation.* The trips are routed in an empty network identifying the links used in the traffic network. Then, through observation from MATSim traffic simulation, time-dependent travel times are derived for all trips $t \in T$. In particular, we monitor each trip t for a series of points in time P. For each point p, the resulting travel time obtained by MATSim is represented in a matrix M_{tp}. For example, if the travel times were retrieved every three hours for one day, this results in eight values $P = \{0, 3, 6, ..., 21\}$ and the matrix M_{tp} would contain eight travel times for every trip t.
(3) *Buffer modeling.* This step addresses the key component of our paper, namely how to define reasonable buffers for each trip to create reliable premium services. Details are given in Section 3.2. Schedules are created by combining the free flow or average travel times as observed in MATSim with the appropriate amount of buffer. The resulting schedule contains expected arrival and departure times of service trips, differing for each applied buffer variant.
(4) *Self-imposed time windows.* Based on the given schedule, self-imposed time windows are derived around every expected arrival and departure.
(5) *Schedule evaluation.* The last step is the simulation of schedules with MATSim. In case of a simulated arrival before the scheduled arrival time window, the vehicle waits if it is a pick-up location; otherwise, it drops the traveler early and continues to the next traveler immediately. As a result of a simulation run with MATSim, we obtain detailed event logs containing all pick-up and drop-off activities of a vehicle and their exact times of occurrence. Based on these event logs, the following key figures are investigated: earlier than time window, within time window, later than time window, average early, maximum early, average late, maximum late and the average deviation. Earlier and later than as well as within the time window indicate how often the vehicle arrived before, after or within the scheduled time windows, respectively. The average deviation represents the additional tour duration due to the applied buffers compared to a tour without buffers. We analyse potential delays separately for pick-ups at origins and drop-offs at destinations.

3.2. Buffer Creation

For buffer creation, we consider three basic ideas; these ideas and their variants are summarized in Table 1. For all variants, the expected arrival time at each location i is determined by the scheduled travel times d_t, the specific buffer b_t and the service time u_i, with $t = i$ meaning that a trip t is always assigned to the origin location i. The scheduled travel times are based on either free flow (*"min"*) or average (*"avg"*) travel times. The minimum and average travel times are derived from the MATSim observations represented in M_{ip}, separately for each trip i. The buffers are created as follows.

Baseline: $BASE_{min}$, $BASE_{avg}$. As the baseline, we use the minimum and average travel times without additional buffers. $BASE_{min}$ represents the departure and arrival times of our service assuming a free-flow traffic network and is

Table 1: Summary of buffer variants.

variant	input travel time	buffer
$BASE_{min}$	min	–
oLP_{min}	min	original LP by Jabali
aLP_{min}	min	adapted LP of Jabali
$BASE_{avg}$	avg	–
oLP_{avg}	avg	original LP by Jabali
aLP_{avg}	avg	adapted LP of Jabali
SD_{avg}	avg	standard deviation

used to gain a general understanding of the delays occurring during schedule execution. $BASE_{avg}$ represents a more sufficient planning approach, since it uses average travel times.

Standard deviation: SD_{avg}. Our first idea to create reasonable buffer times is inspired by Ehmke and Campbell (2014). In this case, we use the standard deviation of the observed travel times of trips to define buffers. In particular, the buffer b_i is calculated for every trip i, using the set of corresponding travel times in M_{ip}. This buffer reflects a simple approach, as it considers each trip individually.

Original LP model: oLP_{min}, oLP_{avg}. For more sophisticated buffers that consider the impact of buffer creation across a given schedule, we implement the model of Jabali et al. (2015). Their linear programming model is based on delay scenarios Ψ_i representing a set of delays for a specific trip i. Each delay is provided with a particular probability of occurrence. The objective function finds the appropriate balance between a longer tour duration due to incorporated buffers and resulting later departure times as well as a possible accepted delay Δ_{ijk} at each location. In particular, l_{ik} defines the length of the delay for trip i in scenario $k \in \Psi_i$.

$$(1) \qquad s_i + d_i + l_{ik} + \sum_{m=i+1}^{j-1} (u_m + d_m) \leq s_j - u_j + W_j + \Delta_{ijk} \qquad i \in R_r \setminus \{n_r + 1\}; j \in R_r \setminus \{0\}; i < j; k \in \Psi$$

Equation (1) ensures that despite the occurrence of a delay, all subsequent travelers can be reached in an appropriate time by defining the planned departure time s_j, with i being the traveler after whom the delay occurs, and j the subsequent traveler affected by this delay. The time window W_j assists in ensuring the scheduled departure time and the entire length is considered. In our approach, we derive the self-imposed time windows around each departure time. Therefore, within this paper, only the additional half of the given time window is considered in the LP model. Note that u_i is service time and d_i is travel time for each trip i and the corresponding origin location. As a result, we can derive a set of departure times s_i. The required buffer b_i is determined by comparing the received departure times s_i with the underlying travel times d_i as shown in Equation (2).

$$(2) \qquad b_{t=i} = s_{i+1} - u_{i+1} - d_{t=i} - s_i.$$

To create delay scenarios, we use the deviations of d_i and each corresponding observed travel time in M_{ip}. Hereby, the number of observations contained in P defines the number of delay scenarios Ψ_i. Based on minimum and average travel times, we create two variants oLP_{min} and oLP_{avg}. For oLP_{min}, the minimum travel time d_i is deducted from any observed time-dependent value represented in M_{ip} for each i, which results in a specific delay scenario value l_{ik} for each P. For oLP_{avg}, negative values may occur, which are set to zero. In the original version as presented by Jabali et al. (2015), it is assumed that a delay occurs for exactly one trip.

Adapted LP model: aLP_{min}, aLP_{avg}. We also consider two variants of an adapted version. Instead of simply assuming exactly one delay in our schedule, we now assume that a certain delay scenario k can occur at several or even all trips i. To reflect this, we adapted the model such that if a delay l_{ik} occurs, the delays corresponding to the scenario k also occur on all subsequent trips. In the original case, all service times u and travel times d are added up beginning from the point of delay; in the modified case, we consider all service times u, travel times d and the delays of the respective scenario l_k, as presented in Equation (3). The resulting buffers are computed analogously to Equation (2).

$$(3) \qquad s_i + d_i + l_{ik} + \sum_{m=i+1}^{j-1} (u_m + d_m + l_{mk}) \leq s_j - u_j + W_j + \Delta_{ijk} \qquad i \in R_r \setminus \{n_r + 1\}; j \in R_r \setminus \{0\}; i < j; k \in \Psi.$$

4. Experimental Setup

We evaluate the potential of buffers for schedules of a premium passenger taxi service in the urban area of Berlin, Germany. We use time-dependent travel times from the publicly available MATSim Open Berlin Scenario (OBS). The Open Berlin Scenario has been calibrated on traffic counts and mode-specific trip distance distributions. It represents the real-world traffic in the state of Berlin realistically (Ziemke et al., 2019) and contains the complete day plans of travelers represented by agents who reflect all adult inhabitants and their activities living in the Greater Berlin region. The road network is based on OpenStreetMap (OSM) and includes all road categories for Berlin and all main roads for the surrounding State of Brandenburg. All relevant modes are represented, i.e. using a private car as driver, using a private car as passenger, bicycle, walking, and public transport.

Based on the Open Berlin Scenario, time-dependent travel times M_{tp} were derived for each trip t of a schedule T on an hourly basis p, resulting in $P = 24$ values for every t. We chose 2000 randomly selected locations spread over the area of Berlin to represent 1000 travel requests, each with origin and destination. These were randomly divided into 100 tours of equal length, with 10 service trips each. Note that we do not optimize the order of service trips, since requests for premium services arise dynamically in practice and our focus is on the impact of buffers in schedules of autonomous fleets. The service time is assumed to be constant (two minutes) for pick-ups at origins and drop-offs at destinations. Five different service levels are investigated through self-imposed service time windows of different lengths (between 1 and 10 minutes), with tight time windows of 1 minute representing our premium option. Schedules are created for the different buffer variants $BASE_{min}$, oLP_{min}, aLP_{min}, $BASE_{avg}$, oLP_{avg}, aLP_{avg}, and SD_{avg}.

LP-specific parameters (for oLP and aLP) are defined as follows. The probability for occurrence of a delay of a trip is equally distributed over all trips. We assume the probability of a certain delay scenario is also spread equally, so the chance is one divided by the number of 24 scenarios. In addition, we need to set the tardiness penalty, applied to the late arrival at a location, and the overtime penalty. The penalty for tardiness is set to 200 and the overtime penalty to 1. The tour start is set to zero and the tour end is defined as estimated tour duration without any delay, i.e. the sum of all scheduled travel times is added to the sum of all service times.

Altogether, based on the seven buffer and five time window variants for 100 different tours with 24 possible departure times (at every full hour of a day), 84000 schedules are created and simulated with MATSim.

5. Results

In the following, we first present results for an exemplary tour and illustrate the derived key figures. Then, we analyse overall results for our different buffer variants to investigate the trade-off between service quality and costs. Results represent aggregates summarised over 2400 simulations.

For the example tour, Table 2 presents the schedules in terms of expected arrival times for each service trip as well as the extent to which the simulated results deviate from the planned schedule, denoted by sim Δ. In addition, “e”, “w” and “l” are used to indicate arrival earlier, within or later than the expected time window. For the presented schedule in Table 2, we use tight departure and arrival time windows of 2 minutes. The time windows are represented as +/- 1 minute, based on the expected arrival times arising from planning with the three buffer alternatives $BASE_{min}$, oLP_{min} and aLP_{min}.

With $BASE_{min}$, one can clearly see how delays accumulate when planning with free-flow travel times and no additional buffer. Using these free-flow travel times and adding buffers according to oLP_{min}, expected departure and arrival times are much closer to the simulated times. Nevertheless, there are delays of up to about 9 minutes, which decrease towards the end. With aLP_{min}, no delays occur, and the vehicle rather arrives early before departure and arrival time windows.

Table 3 summarizes the key figures for the example tour, including how often the time windows were met and, if the time window was not respected, the maximum and average earliness/lateness relative to the time window. An early arrival at the pick-up location indicates unnecessary idle time caused by the waiting time until the traveler arrives at the vehicle. The last column (“deviation”) shows the additional tour duration caused by delays and buffers relative to $BASE_{min}$. Generally, it can be seen that fewer delays are associated with longer tour durations as well as that serious delays occur when planning without buffers. Considering the simulated deviation, the advantage of oLP_{min} is

Table 2: Arrival times of example tour with 2 minutes time windows.

		$BASE_{min}$			oLP_{min}			aLP_{min}		
		expected [hh:mm:ss]	sim Δ [mm:ss]	tw	expected [hh:mm:ss]	sim Δ [mm:ss]	tw	expected [hh:mm:ss]	sim Δ [mm:ss]	tw
c1	origin	10:00:00	00:00	w	10:00:00	00:00	w	10:00:00	00:00	w
	destination	10:39:28	+ 07:12	l	10:46:00	+ 00:40	w	10:46:08	+ 00:32	w
c2	origin	11:18:39	+ 17:12	l	11:33:34	+ 02:17	l	11:37:02	- 01:11	e
	destination	11:52:23	+ 23:42	l	12:12:57	+ 03:08	l	12:16:45	- 00:30	w
c3	origin	12:01:55	+ 26:04	l	12:25:13	+ 02:46	l	12:29:15	- 01:06	e
	destination	12:15:57	+ 29:14	l	12:42:22	+ 02:49	l	12:47:40	- 02:14	e
c4	origin	12:35:04	+ 30:45	l	13:03:43	+ 02:06	l	13:09:25	- 03:21	e
	destination	12:47:26	+ 31:30	l	13:16:05	+ 02:51	l	13:22:23	- 00:54	w
c5	origin	13:10:26	+ 36:25	l	13:42:23	+ 04:28	l	13:49:43	- 01:03	e
	destination	13:33:06	+ 41:02	l	14:08:50	+ 05:18	l	14:16:46	- 00:47	w
c6	origin	13:48:14	+ 44:24	l	14:27:04	+ 05:34	l	14:36:00	- 01:30	e
	destination	14:14:10	+ 51:37	l	14:58:43	+ 07:04	l	15:09:27	- 00:20	w
c7	origin	14:32:06	+ 53:20	l	15:18:20	+ 07:06	l	15:29:50	- 01:07	e
	destination	14:58:08	+ 59:55	l	15:49:25	+ 08:38	l	16:03:15	- 02:01	e
c8	origin	15:32:15	+ 66:04	l	16:29:38	+ 08:41	l	16:45:14	- 02:58	e
	destination	16:21:41	+ 74:33	l	17:29:21	+ 06:53	l	17:47:20	- 04:01	e
c9	origin	17:00:43	+ 83:55	l	18:18:04	+ 06:34	l	18:37:08	- 05:44	e
	destination	17:26:00	+ 86:50	l	18:46:25	+ 06:25	l	19:07:33	- 02:15	e
c10	origin	17:54:32	+ 89:26	l	19:17:17	+ 06:41	l	19:39:18	- 02:54	e
	destination	18:07:18	+ 90:49	l	19:32:10	+ 05:57	l	19:54:22	- 02:11	e

demonstrated clearly. Here, with the same tour duration, the maximum and average delay is reduced significantly. By applying aLP_{min}, delays could be completely avoided, but this comes with a longer tour duration, which increases by about 14 minutes. The best results would predict the arrival times at every hour of the day without being earlier than the time window.

Table 3: Key figures of the example tour.

arrival at	version	early [%]	within [%]	late [%]	avgEarly [mm:ss]	maxEarly [mm:ss]	avgLate [mm:ss]	maxLate [mm:ss]	deviation [%]
origin	$BASE_{min}$	0	10	90	-	-	+ 48:44	+ 89:26	0.00
	oLP_{min}	0	10	90	-	-	+ 04:08	+ 07:41	0.00
	aLP_{min}	90	10	0	- 01:19	- 04:44	-	-	2.42
destination	$BASE_{min}$	0	0	100	-	-	+ 48:38	+ 90:49	0.00
	oLP_{min}	0	10	90	-	-	+ 04:27	+ 07:38	0.00
	aLP_{min}	50	50	0	- 01:32	- 03:01	-	-	2.42

Overall results are summarized in Table 4. In the following, we will focus on the differences between (1) $BASE$ and oLP, (2) oLP and aLP as well as (3) aLP and SD. These comparisons serve to show the potentials of the different buffer variants. Generally, the average tour duration of $BASE_{min}$ for 2400 considered schedules is about 9 hours for each time window variant. This is about one hour longer than planned according to simulation with MATSim.

Confirming the results of the example tour, the comparison of $BASE_{min}$ and oLP_{min} illustrates the potential of buffers. Planned with 10 minute time windows, 22% improvement can be seen with oLP_{min}. For a similar tour duration, the delays have been reduced and the maximum lateness decreases by about 37 minutes. With rising service levels, due to shorter time windows, oLP_{min} can reduce the number of delays even further, to about 14 percent. As shown in the Equation (3) of Chapter 3 on the right, the time window is added to the departure time at a certain location after a delay has occurred. Therefore, longer time windows reduce the expected departure time, respectively the additional buffer; longer time windows offer a certain flexibility and a higher chance to respect the window. The results show that the original model strongly reacts to longer time windows.

For $BASE_{avg}$, the average travel times already represent some kind of buffer induced from observed travel times, which can reduce delays to 61% (1 minute time window) or even 38% (10 minutes time window). As can be seen, oLP_{avg} shows little impact on 10 minute time windows compared with $BASE_{avg}$. At the one minute service level, however, the results are even better than with oLP_{min}.

Table 4: Results at travelers destination

time window	version	early [%]	within [%]	late [%]	avgEarly [mm:ss]	maxEarly [mm:ss]	avgLate [mm:ss]	maxLate [mm:ss]	deviation [%]
1 min	$BASE_{min}$	0.0	2.0	98.0	- 00:00	- 00:00	+ 33:27	+ 92:39	0.00
	$BASE_{avg}$	31.5	7.0	61.5	- 02:22	- 06:36	+ 09:24	+ 30:08	2.45
	oLP_{min}	69.7	16.5	13.8	- 02:47	- 10:25	+ 01:36	+ 05:36	6.21
	oLP_{avg}	75.3	14.9	9.9	- 02:48	- 10:41	+ 01:20	+ 04:35	6.82
	aLP_{min}	87.5	9.3	3.2	- 03:06	- 11:49	+ 00:59	+ 02:31	8.84
	aLP_{avg}	87.6	9.1	3.3	- 03:06	- 11:51	+ 00:58	+ 02:31	8.89
	SD_{avg}	77.9	11.9	10.2	- 02:35	- 10:07	+ 01:29	+ 04:53	6.57
2 min	$BASE_{min}$	0.0	3.2	96.8	- 00:00	- 00:00	+ 33:21	+ 92:09	0.00
	$BASE_{avg}$	30.0	12.6	57.3	- 02:25	- 06:34	+ 09:27	+ 29:37	2.40
	oLP_{min}	51.2	25.2	23.6	- 02:58	- 09:59	+ 02:42	+ 09:06	4.81
	oLP_{avg}	61.7	25.1	13.2	- 02:51	- 10:22	+ 01:54	+ 06:08	5.65
	aLP_{min}	81.3	16.4	2.2	- 03:04	- 11:42	+ 01:01	+ 02:10	8.11
	aLP_{avg}	81.9	15.9	2.2	- 03:03	- 11:42	+ 00:58	+ 02:07	8.16
	SD_{avg}	75.8	18.5	5.7	- 02:36	- 10:05	+ 01:28	+ 03:57	6.48
3 min	$BASE_{min}$	0.0	4.3	95.7	- 00:00	- 00:00	+ 33:11	+ 91:39	0.00
	$BASE_{avg}$	28.9	17.0	54.1	- 02:26	- 06:33	+ 09:25	+ 29:05	2.35
	oLP_{min}	38.2	26.0	35.8	- 03:08	- 09:34	+ 04:31	+ 14:44	3.73
	oLP_{avg}	50.9	30.9	18.2	- 02:56	- 10:02	+ 02:40	+ 08:17	4.78
	aLP_{min}	76.2	22.0	1.8	- 03:03	- 11:32	+ 00:59	+ 01:59	7.52
	aLP_{avg}	77.0	21.3	1.7	- 03:03	- 11:34	+ 00:58	+ 01:56	7.57
	SD_{avg}	73.5	23.0	3.5	- 02:38	- 10:03	+ 01:24	+ 03:10	6.39
5 min	$BASE_{min}$	0.0	6.7	93.3	- 00:00	- 00:00	+ 33:02	+ 90:39	0.00
	$BASE_{avg}$	26.9	24.7	48.4	- 02:30	- 06:31	+ 09:19	+ 28:04	2.25
	oLP_{min}	23.8	25.1	51.1	- 03:11	- 08:50	+ 09:13	+ 27:08	2.29
	oLP_{avg}	38.3	35.0	26.7	- 03:04	- 09:35	+ 04:11	+ 13:00	3.65
	aLP_{min}	68.3	30.2	1.5	- 03:09	- 11:22	+ 00:53	+ 01:38	6.60
	aLP_{avg}	68.4	30.3	1.3	- 03:03	- 11:16	+ 00:53	+ 01:34	6.71
	SD_{avg}	70.0	28.7	1.3	- 02:41	- 10:01	+ 00:59	+ 01:42	6.20
10 min	$BASE_{min}$	0.0	12.2	87.8	- 00:00	- 00:00	+ 32:28	+ 88:09	0.00
	$BASE_{avg}$	23.2	38.7	38.1	- 02:38	- 06:30	+ 08:49	+ 25:34	2.02
	oLP_{min}	8.8	25.4	65.9	- 02:45	- 07:21	+ 19:03	+ 50:48	0.59
	oLP_{avg}	26.2	41.5	32.2	- 03:10	- 08:50	+ 06:48	+ 19:19	2.42
	aLP_{min}	52.2	46.6	1.2	- 03:37	- 11:32	+ 00:54	+ 01:36	5.25
	aLP_{avg}	52.5	46.4	1.1	- 03:19	- 11:09	+ 00:54	+ 01:33	5.58
	SD_{avg}	61.9	37.9	0.2	- 02:48	- 09:58	+ 00:12	+ 00:16	5.75

Relative to oLP, aLP achieves significantly better results in terms of punctuality. It can be seen that with additional tour duration, delays can be reduced considerably, especially for longer time windows. In addition, the maximum and average lateness is lower with only a slight increase in earliness. This is probably due to the waiting time at the pickup location of the traveler. Remarkable is that aLP compared to oLP shows better results with decreasing service level and therefore does not seem to react too strongly to time windows. Finally, aLP_{min} and aLP_{avg} create similar results, but using minimum travel times seems to be slightly better, especially due to the shorter actual tour durations.

As with *aLP*, the delays in SD_{avg} decrease continually with lower service levels, which is not surprising, since time windows are not considered in the calculation of these buffers. Therefore, they are identical for each time window and the chance of adhering increases with the time window length. SD_{avg} shows excellent results for 10 minute time windows, with only a slightly increased tour duration compared to *aLP*.

6. Conclusion

This work demonstrated the impact of simple and more complex buffers for planning of more reliable taxi services with autonomous fleets. Based on an integrated framework of optimization and simulation, we could show that smart buffers can create a compromise of premium service for travelers and cost of operations. Further adjustments to the model could include the integration of dependencies between delay scenarios, e.g. at rush hour times. In particular, probabilities for transitions between the scenarios should be integrated since it is unlikely that one vehicle will change from a scenario with small delays to one with very long delays without considering possible intermediate scenarios.

In addition, the following further methodological modifications could be beneficial. In this paper, we focused on the evaluation of tours which were carried out with the same schedule at different times of the day. This should be extended to include time-dependent driving times and applied to different days. While the delay scenarios of this paper are based on hourly driving times, a scenario could also represent one day with different driving times depending on the time of day. Also a mix of standard and premium services could be an interesting extension. Finally, given that autonomous fleets operate with environmental-friendly electric engines, energy consumption and the required charging times could also be taken into account.

Acknowledgements

This research was funded by the German Federal Ministry of Transport and Digital Infrastructure.
(FKZ: 16AVF2147)

References

Ehmke, J.F., Campbell, A.M., 2014. Customer acceptance mechanisms for home deliveries in metropolitan areas. European Journal of Operational Research 233, 193 – 207. doi:10.1016/j.ejor.2013.08.028.

Gurumurthy, K.M., Kockelman, K.M., Loeb, B.J., 2019. Chapter four - sharing vehicles and sharing rides in real-time: Opportunities for self-driving fleets, in: Fishman, E. (Ed.), The Sharing Economy and The Relevance for Transport. Academic Press. volume 4 of *Advances in Transport Policy and Planning*, pp. 59 – 85. doi:10.1016/bs.atpp.2019.09.001.

Horni, A., Nagel, K., Axhausen, K.W. (Eds.), 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity, London. doi:10.5334/baw.

Jabali, O., Leus, R., Woensel, T., Kok, T., 2015. Self-imposed time windows in vehicle routing problems. OR Spectr. 37, 331352. doi:10.1007/s00291-013-0348-1.

OBS, . Last accessed on 04 July 2020, https://github.com/matsim-scenarios/matsim-berlin.

OSM, . Last accessed on 09 July 2020, http://www.openstreetmap.org.

Rajabi-Bahaabadi, M., Shariat-Mohaymany, A., Babaei, M., Vigo, D., 2019. Reliable vehicle routing problem in stochastic networks with correlated travel times. Operational Research doi:10.1007/s12351-019-00452-w.

Redmond, M., Campbell, A.M., Ehmke, J.F., 2020. Data-driven planning of reliable itineraries in multi-modal transit networks. Public Transport 12, 171–205. doi:10.1007/s12469-019-00221-0.

Ta, D., Dellaert, N., [van Woensel], T., [de Kok], T., 2014. The time-dependent vehicle routing problem with soft time windows and stochastic travel times. Transportation Research Part C: Emerging Technologies 48, 66 – 83. doi:10.1016/j.trc.2014.08.007.

Tierney, K., Ehmke, J., Campbell, A., Müller, D., 2018. Liner shipping single service design problem with arrival time service levels. Flexible Services and Manufacturing Journal doi:10.1007/s10696-018-9325-y.

Vareias, A.D., Repoussis, P.P., Tarantilis, C.D., 2019. Assessing customer service reliability in route planning with self-imposed time windows and stochastic travel times. Transportation Science 53, 256–281. doi:10.1287/trsc.2017.0748.

Ziemke, D., Kaddoura, I., Nagel, K., 2019. The matsim open berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. Procedia Computer Science 151, 870 – 877. doi:10.1016/j.procs.2019.04.120.