Computers and Operations Research 108 (2019) 1–19

Contents lists available at ScienceDirect

Computers and Operations Research

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

Same-Day delivery with pickup stations and autonomous vehicles

Marlin W. Ulmer*, Sebastian Streng

Technische Universität Braunschweig, Institut für Wirtschaftsinformatik, Mühlenpfordtstraße 23, Braunschweig 38106, Germany

ARTICLE INFO

Article history:
Received 18 July 2018
Revised 27 March 2019
Accepted 28 March 2019
Available online 30 March 2019

Keywords:
Same-Day delivery
Pickup stations
Autonomous vehicles
Dynamic vehicle routing
Dynamic dispatching
Policy function approximation

ABSTRACT

Same-day delivery has become a very important challenge for e-commerce providers. However, effective delivery concepts are still not established. In this paper, we analyze the potential of combining parcel pickup stations and autonomous vehicles for same-day delivery. This combination may allow for consolidation of goods, automated delivery processes, and reduced operation costs. We consider the problem on the operational planning level. A depot, a set of pickup stations, and an autonomous delivery fleet are given. Customers dynamically order goods to a preferred pickup station and expect fast service. The goods need to be delivered from the depot to the preferred pickup station or another pickup station in the close neighborhood. Autonomous vehicles are directly dispatched between the depot and the stations. To decide where and when to dispatch a vehicle and about the corresponding goods to load, we present a policy function approximation (PFA). Our intuitive PFA allows real-time decision making while balancing the tradeoff between fast delivery and consolidation. We conduct an extensive computational study for the pickup station network of the city of Braunschweig. We show how our delivery concept allows fast delivery for up to 100 delivered goods per vehicle and day. We further pair the PFA with a value function approximation to account for heterogeneity in demand. We finally derive important managerial insights for strategical and tactical planning and present an extensive outlook on promising future research directions.

1. Introduction

E-commerce experiences a steady growth worldwide. The advantage of shopping online is that customers can select their goods at home and do not need to drive to a store, to find a parking spot, to search the store for goods, and to wait in long lines. One major disadvantage is the missing "instant gratification." In contrast to store shopping, the customer needs to wait until the goods arrive. This waiting time reduces the probability for customers making orders online (Lowe et al., 2014). As a consequence, same-day delivery (SDD) has become one of the hottest topics in e-commerce. Many major e-commerce companies such as Amazon, Alibaba, or JD already offer SDD in many cities worldwide, often within a few hours after order (Stevenson and Black, 2018). However, companies struggle to develop cost-efficient SDD-delivery concepts (Ram, 2015).

Efficient SDD is challenging for many reasons. As already experienced in regular parcel delivery, the "last-mile" causes the majority of delivery costs (Bernau et al., 2016). This results from the lack of consolidation opportunities because customers need to be served individually at different locations in the city. Furthermore, drivers face many uncertainties in the parking situation, the exact delivery location, and the customer availability. The last-mile cost factors are amplified for SDD because customers order over the course of the day and expect timely delivery. In contrast to conventional last-mile delivery, SDD-vehicles conduct several delivery trips per day. Due to this additional temporal customer spread, the amount of SDD-parcels per delivery trip is even smaller compared to regular delivery. Thus, the costs of delivery are increased even further. As previous studies show, delivery vehicles are usually not able to serve more than 30 customers per day (Klapp et al., 2018a; Ulmer, 2017a; Voccia et al., 2019).

An alternative way of delivery are pickup stations (Savelsbergh and Van Woensel, 2016). Pickup stations are lockers where delivery companies store a larger amount of parcels. Once a parcel is stored, the customer is informed and uses an individual access code to pick up the ordered goods. For regular parcel delivery, pickup stations are already established in many countries in Europe as well as in China. Currently, Amazon is developing an own network of pickup stations (Lunden, 2017). Pickup stations have several advantages for customers (Vakulenko et al., 2018; Yuen et al., 2018): Pickup stations are often relatively close to the customers' home leading to short trips for the customer to made. In comparison to brick and mortar stores, no in-store selection of goods or waiting

* Corresponding author.
E-mail address: m.ulmer@tu-braunschweig.de (M.W. Ulmer).

https://doi.org/10.1016/j.cor.2019.03.017
0305-0548/

in line for checkout are necessary. Furthermore, customers are flexible when to pickup their goods and do not need to wait at home for a delivery to arrive (Verlinde et al., 2018). Pickup stations also have many advantages for the delivery companies. Vehicles deliver larger amounts of parcels to an automated pickup station instead of the time-consuming and often unpredictable service of individual customers. Furthermore, companies are often flexible with respect to the specific pickup station for a customer leaving room for even more consolidation. Thus, pickup stations may be the “Golden Mean” also for same-day delivery.

Another major advantage of pickup stations is that the delivery can be conducted by autonomous vehicles. Many companies invest substantially in both autonomous ground vehicles and autonomous air vehicles (drones) (Condon, 2019; McFarland, 2017; Mitchell, 2019). The main advantage of autonomous vehicles is that they are comparably cost-efficient due to increasing driver shortages and high wages. However, it is still unresolved how autonomous vehicles can be embedded in logistical supply chains, because of their limitations compared to conventional vehicles with drivers. One major issue is that in the foreseeable future, not every street or region in the city may be eligible for autonomous travel, especially, for (not fully) autonomous vehicles of the SAE-levels 3 and 4 (NHTSA, 2017; SAE International, 2016). Instead, autonomous vehicle routing will be limited to standardized operations in selected streets (Beirigo et al., 2018; Scherr et al., 2018; 2019). Furthermore, the “last-meter” delivery process to the customer is still an unresolved issue for both autonomous ground and air vehicles (Kolodny, 2017). In Heutger and Kückelhaus (2016), DHL recently stated that “Machine-to-person handover of parcels and letters may be beyond the current capability of autonomous technologies [...]. Instead, machine-to-parcel station handover is certainly achievable.”

This statement was confirmed in discussions with engineers working on autonomous vehicles indicating that pickup stations could alleviate many of the shortcomings autonomous vehicles have. The loading and delivery environment can be controlled and the processes can be automated (Haas and Friedrich, 2017). Autonomous vehicles of SAE-levels 3 or 4 only traverse on known, direct paths between the warehouse and each pickup station. A direct contact between customers and autonomous vehicles is avoided. There are already several concepts for delivery to parcel stations with autonomous ground vehicles (Heutger and Kückelhaus, 2016; Tipping and Kauschke, 2016). Delivery to parcel stations with drones is already tested. DHL delivers goods by drones to some parcel stations (DHL, 2016) and in Singapore, the SKYWAYS system for delivery to parcel stations by drones is planned to be implemented soon (Chong, 2017). Simultaneously, drones are developed that are able to carry many parcels at once, especially beneficial for consolidated delivery to parcel stations (Murphy, 2017).

In this paper, we analyze the operational level of SDD with pickup stations and autonomous vehicle. We denote the resulting mathematical problem as the *Stochastic Dynamic Dispatching Problem for SDD with Pickup Stations and Autonomous Vehicles* (SDDPSAV). A depot, a set of capacitated pickup stations, and a delivery fleet of capacitated autonomous vehicles are given. Over an order horizon, customers dynamically request delivery to a preferred pickup station and expect fast service. The requests are stochastic and follow a known probability distribution. The ordered goods need to be transported from the depot to the preferred pickup station or another pickup station in the close neighborhood. Due to the discussed routing limitations, only direct dispatches are allowed: the travel from depot to one station and back. Decisions are made about sending a vehicle (or to wait), what to load on a vehicle, and to which station to send the vehicle. Once goods are shipped, the customer is informed about the pickup station and the earliest pickup time. After the delivery is loaded to a pickup station, the customer picks up the order. The time of pickup is stochastic as well. The dispatcher aims at a fast delivery for every customer minimizing the expected average delivery time, defined as the time span between order and earliest possible pickup.

Solving the SDDPSAV is challenging. First, the Markov decision process model is very complex in all dimensions. Second, the time to make a decision is highly limited because dispatching is made in real-time. Thus, we propose a runtime-efficient policy function approximation (PFA). A PFA usually bases on a practical commonsense idea (Powell, 2011). For the SDDPSAV, we assume that there is a tradeoff between early and late dispatches. Early dispatches allow for fast deliveries and an early return of the vehicles to perform a next trip. Late dispatches allow for a better consolidation because potential future orders can be integrated. Our PFA addresses this issue by defining dispatching thresholds. The PFA dispatches a vehicle if the vehicle can at least transport a certain number of parcels to a station. The thresholds are determined by means of sample average approximation.

We present a computational study for the City of Braunschweig, Germany. Braunschweig is a medium-sized city with a population of a quarter-million and traditional European city layout. The existing network consists of 12 pickup stations and the depot. Most of the pickup stations are located at large streets, many on the main ring road of Braunschweig where (supervised) autonomous vehicles already drive successfully since 2010 (Nothdurft et al., 2011). We test different instances with up to 1000 expected orders per day and up to 10 vehicles. We show that with pickup stations, vehicles are generally able to deliver about 100 customer orders per day and orders are available at a pickup station in less than 2 h on average. Our study further reveals that the threshold has a significant impact on the solution quality. With increasing customer demand, suitable thresholds increase as well shifting the focus between fast dispatches and consolidation.

We use our PFA to analyze the impact of customer behavior, speed of the autonomous vehicle, and capacities of vehicles and pickup stations. We show that if customers are flexible in their choice of a pickup station, the delivery times can be reduced significantly. Further, if customers pick up their goods early, overall delivery times decreases as well because of the freed capacity at that station. In contrast, delivery times increase if customers insist of delivery to a specific station and picking up their goods late. We analyze the setup of vehicles' and stations' capacities and derive suitable settings for different demand and customer behavior. We further give guidance how the PFA can be paired with value function approximation to adapt to heterogeneous customer demand. Finally, we derive managerial implications for the strategical and tactical planning levels.

The main contribution of our paper is to connect the three strong developments of same-day delivery, autonomous delivery fleets, and parcel station networks to a new and promising delivery concept. Our concept utilizes the strengths of autonomous vehicles while avoiding many of their weaknesses. Our paper is the first combining same-day delivery with pickup stations. Our paper is also the first combining delivery routing of autonomous vehicles with pickup stations. Our paper is further one of the first papers using autonomous vehicles for dynamic same-day delivery. With the SDDPSAV, we present an unambiguous MDP model of the delivery problem. We develop an intuitive and runtime-efficient dispatching strategy tailored to the individual instance settings. Based on this strategy, we conduct a comprehensive computational study for the city of Braunschweig, systematically analyzing the dimensions of the MDP. We generate important managerial implications about the setup of the pickup stations and fleet, about the impact of customer behavior, and about suitable dispatching strategies based on customer demand. We further illustrate how our intuitive policy can be extended to meet heterogeneity in customer demand. We finally present a comprehensive outlook on promising follow up research.

The paper is organized as follows. In Section 2, we present the related literature. We define the problem in Section 3 and describe our solution methods in Section 4. In Section 5, we present our computational study. The paper ends with a summary and an outlook in Section 6.

2. Literature review

In the following, we present the related literature. To the best of our knowledge, our paper is the first that combines autonomous vehicles with pickup stations and dynamic same-day delivery. It is further the first combining autonomous vehicles and pickup stations as well as pickup-stations and same-day delivery. However, there is work on the individual aspects of the SDDPSAV. In the following, we first present research on delivery routing with autonomous vehicles. We then present work for SDD and work on dynamic dispatching. Finally, we address related literature from other problem domains.

2.1. Delivery routing with autonomous vehicles

The amount of research on delivery routing with autonomous vehicles is constantly growing. Most of the research focuses on drone routing in a static environment. That means that all customers are known in advance and an assignment of customers to drones and/or regular vehicles need to be determined. The most prominent example for this problem class is the “flying sidekick”-TSP (Agatz et al., 2018; Carlsson and Song, 2017; Ha et al., 2018; Murray and Chu, 2015). A vehicle is equipped with a drone. Instead of visiting every customer by the vehicle, the drone can be sent, delivering the goods and returning to the vehicle. An extension of this problem is given by Ham (2018) where drones can also perform pickup of parcels. In Cheng et al. (2018), drones can serve several customers within one trip. Arbanas et al. (2016, 2018) consider other extensions of the flying sidekick-TSP, where several autonomous ground vehicles are coordinated to support drones in their delivery operations.

The SDDPSAV is stochastic and dynamic because customers request while the vehicles are on the road. Work on dynamic routing for autonomous vehicles is limited. Ulmer and Thomas (2018) suggest dynamic delivery with separate fleets of drones and vehicles. To decide whether to serve a customer with a drone or a vehicle, they propose a policy function approximation based on the travel distance to the customer. They also analyze the impact of customers not eligible for drone delivery. They show that their policy performs well as long as the majority of customers can be served by both drones and vehicles. Dayarian et al. (2018) use drones to dynamically resupply conventional delivery vehicles on the road. They present several policy function approximations to analyze different routing and assignment strategies. To the best of our knowledge, there is no other work on dynamic delivery routing with autonomous vehicles. However, there is some work on dynamic routing of autonomous vehicles for other purposes than delivery, for example (Basilico et al., 2016; Bullo et al., 2011). Bullo et al. (2011) dynamically routes robots to conduct services in an area. Basilico et al. (2016) addresses a problem where drones dynamically patrol in a certain area.

2.2. Same-Day delivery

For SDD, vehicles are routed dynamically to deliver goods from a depot to customers. These customers request over the course of the day. For each customer, the goods are picked up at the depot before delivery to the customers takes place.

Early work is presented by Azi et al. (2012). To determine suitable trips, they apply the multiple-scenario approach (MSA) by Bent and Van Hentenryck (2004). The MSA samples a set of scenarios for future customer requests and uses the samples to determine a suitable set of trips. In contrast of the SDDPSAV, Azi et al. (2012) assume that customers can be rejected. Instead of real-time decision making, they apply their method every hour with runtimes of several minutes.

In the last years, work on dynamic routing for same-day delivery has increased. Voccia et al. (2019) use the MSA to analyze waiting at the depot for consolidation. They show that waiting at the depot for future orders is beneficial. They also allow rejection of requests. They decide minute-by-minute and experience significant runtimes per decision point. Ulmer et al. (2018) shift the computational burden to an offline learning phase by means of value function approximation (VFA). For a single vehicle, Ulmer et al. (2018) analyze the value of preemptively returning to the depot to pick up more goods. Ulmer (2017b) uses VFA to price different delivery deadlines based on the current workload of the vehicles.

The SDDPSAV shows similarities to SDD with respect to the stochastic customer orders and the requirement for real-time decision making. However, there are significant differences. Instead of dynamic routing to a set of customers, vehicles are directly dispatched to a limited number of capacitated stations and the dispatcher may be flexible with respect to the delivery location. Furthermore, besides the stochasticity in the customer requests, we experience stochasticity in the pickup process at the stations. Finally, the sizes of our SDDPSAV-instances is significantly larger than for the SDD-literature. For the presented problems, usually 3 to 5 vehicles perform in sum up to 100 deliveries per day. In our computational study, we analyze instances with up to 1000 orders per day.

2.3. Dynamic dispatching

The SDDPSAV also shows similarities to dynamic dispatching problems. For these problems, a horizons of several dispatching epochs (hours, days, etc.) is considered. Goods arrive randomly between epochs and in every epoch, the dispatcher needs to decide which goods to ship. Dynamic dispatching problems are considered by Klapp et al. (2018a), Klapp et al. (2018b), van Heeswijk et al. (2019), and Rivera and Mes (2017).

Klapp et al. (2018a,b) split the time horizon into hours. Vehicles can only be dispatched every hour. To determine suitable trips, they solve a computationally challenging mixed-integer program on a rolling horizon. The work by van Heeswijk et al. (2019) considers the challenge of dynamically dispatching goods from a consolidation center to a set of locations. The consolidation center is supplied by trucks with stochastic customer orders either preannounced or without information. As for the SDDPSAV, orders are assigned to a fleet of vehicles with fixed capacity to serve a set of customer locations. In contrast to the SDDPSAV, dispatching is not made in real-time and orders may be outsourced to a third party. To solve the problem they use value function approximation based on linear programs. Similar to van Heeswijk et al. (2019), and Rivera and Mes (2017) use dynamic dispatching for freight selection. On a daily basis, freight is dispatched either with trucks or barges. The authors propose VFA to minimize the total costs balancing the tradeoff between fast delivery and consolidation. Similarities to the SDDPSAV can be seen in the dispatch to a set of known locations. Furthermore, our PFA addresses the tradeoff between fast dispatches and consolidation observed in Rivera and Mes (2017).

2.4. Other related literature

The SDDPSAV shows similarities to other problem areas. For example, flexibility in the delivery location is also addressed in problems such as the close-enough traveling salesman problem where customers can be served within a radius around their locations, for example, for meter reading (Carrabs et al., 2017). Other research addresses flexibility by roaming delivery locations over the course of the day, for example, at work or at home (Reyes et al., 2017). The question whether to wait or leave is also addressed in dynamic queuing problems, for example, for salespersons or at taxi stands (Zhang et al., 2018). The phenomenon of stochastic customer pickup is related to dynamic inventory routing problems, where goods are consumed over time (Coelho et al., 2014). Other related work is vehicle routing with release dates where goods become subsequently available at the depot (Archetti et al., 2015). A special case is restaurant delivery where orders are delivered from restaurants to customers ordering over the course of they. These problems have uncertainty in customer orders and in the ready times at the restaurant (Ulmer et al., 2017). In contrast, SDDPSAV has uncertainty in the pickup times. Finally, the direct dispatching from a warehouse to a set of stations shows similarity to the first level of two-echelon routing from warehouses to satellites (Zhou et al., 2018) and also part of the general concept of a physical internet (Crainic and Montreuil, 2016).

3. The dynamic dispatching problem for same-day delivery

In this section, we define the SDDPSAV. We first give a problem description. We then give an example for a decision state of the problem. We model the problem as a Markov decision process and embed the example in the MDP-notation.

3.1. Problem description

The notation for the problem parameters is depicted in Table 1. We consider a problem where goods are delivered by a fleet of m vehicles $\mathcal{V} = \{v_1, \dots, v_m\}$ from a depot D to a set of n pickup stations $\mathcal{P} = \{P_1, \dots, P_n\}$ distributed in the city. The vehicles and pickup stations have an order capacity $\kappa^V_{\max}$ and $\kappa^P_{\max}$, respectively. Each order requires one unit of station capacity. Because of the discussed limitations of autonomous vehicles to navigate in the city, we assume that the vehicles can only perform direct trips from depot to a pickup station and back. The travel time between depot D and pickup station P is given by function $d^V(D, P)$. Customers order goods over a discrete time horizon $T^O = (0, 1, \dots, t^O_{\max})$. The time steps can be for example seconds or, as we will use for our computational study, minutes. The orders are unknown before they are issued, however, the dispatcher has knowledge about the request distribution. The order time of a customer C is denoted by τ^C.

Table 1
Problem Notation.

Description	Notation
Problem Parameters	
Vehicles	$\mathcal{V} = \{v_1, \dots, v_m\}$
Depot	D
Pickup Stations	$\mathcal{P} = \{P_1, \dots, P_n\}$
Vehicle Order Capacity	$\kappa^V_{\max}$
Pickup Station Order Capacity	$\kappa^P_{\max}$
Vehicle Travel Time Function	$d^V(\cdot, \cdot)$
Order Horizon	$T^O = (0, 1, \dots, t^O_{\max})$
Delivery Horizon	$T^D = (0, 1, \dots, t^V_{\max})$
Dispatch Setup Time	τ^D
Loading Time at Station	τ^P
Customer Order	C
Time of Customer Order	τ^C
Preferred Pickup Station for C	P^C
Customer Travel Time Function	$d^C(\cdot, \cdot)$
Radius for Acceptable Stations	ρ
Earliest Pickup Time for Customer C	$\tau^{\text{early}}(C)$
Pickup Station for Customer C	$P(C)$
Realized Pickup Time for Customer C	$\tau^{\text{pickup}}(C)$

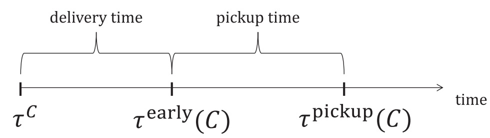

Fig. 1. Example for a Decision State, a Potential Decision, and a Potential new State.

Individual customer locations are not provided. Instead, each customer order C has a preferred pickup station P^C. However, the customer also accepts delivery to adjacent pickup stations. To avoid more severe inconveniences for the customer, we constrain the vicinity of customer travel time around the preferred station by a customer travel time radius ρ. The customer travel time between two pickup stations may differ from the travel time of the vehicles and is given by function $d^C(P_1, P_2)$. Within a sufficiently large dispatching horizon $T^D = (0, 1, \dots, t^V_{\max})$[1], the dispatcher repeatedly sends out vehicles to deliver sets of orders to stations. Vehicles have a dispatch setup time of τ^D minutes and a unloading time at a pickup station of τ^P minutes. A dispatching decisions needs to consider if the target pickup station has sufficient capacity for the corresponding goods and if the target station is within the radius ρ for all the corresponding orders. Whenever a vehicle is scheduled for dispatch, the customers of the corresponding goods are informed about an earliest pickup time $\tau^{\text{early}}(C)$ and the pickup station $P(C)$. Time $\tau^{\text{early}}(C)$ is the point of time the goods are loaded to the pickup station $P(C)$ and are therefore available for pickup. Once a customer is informed, the customer picks up the order at time $\tau^{\text{pickup}}(C)$. The time of pickup is uncertain and unknown beforehand but the stochastic distribution of pickup times is known. The process and the values τ^C, $\tau^{\text{early}}(C)$, and $\tau^{\text{pickup}}(C)$ are depicted in Fig. 1. We denote the time span between τ^C and $\tau^{\text{early}}(C)$ "delivery time" and the time span between $\tau^{\text{early}}(C)$ and $\tau^{\text{pickup}}(C)$ "pickup time".

The dispatcher aims on fast delivery for every customer. The objective is to minimize the expected sum of delivery times, i.e., times between τ^C and $\tau^{\text{early}}(C)$ over all customers. Notably, this number is directly connected to the average delivery time per customer. However, minimizing the sum of delivery times simplifies the mathematical model substantially.

3.2. Example for a decision state

Before we present the mathematical model of the SDDPSAV, we give an example for a decision state, a potential decision, and the transition to the next state. We use this example to introduce the MDP. We will repeatedly return to the example and embed it in the MDP-notation.

The example is depicted in Fig. 2. The figure consists of three portions. The left portion depicts a decision state, the central portion shows a potential decision, and the right portion gives an example for a transition to a new decision state. Each portion shows the network of depot and pickup stations. The depot is depicted as black square, the pickup stations as stacks of rectangles. Each rectangle represents a potential storage room. In the example, we assume that three pickup stations are given, each with a capacity

[1] Because of the stochastic nature of the SDDPSAV, this time horizon needs to be chosen very large to capture extreme instance realizations. However, in our computational study, the latest dispatch is made not later than four hours after the end of the order time horizon.

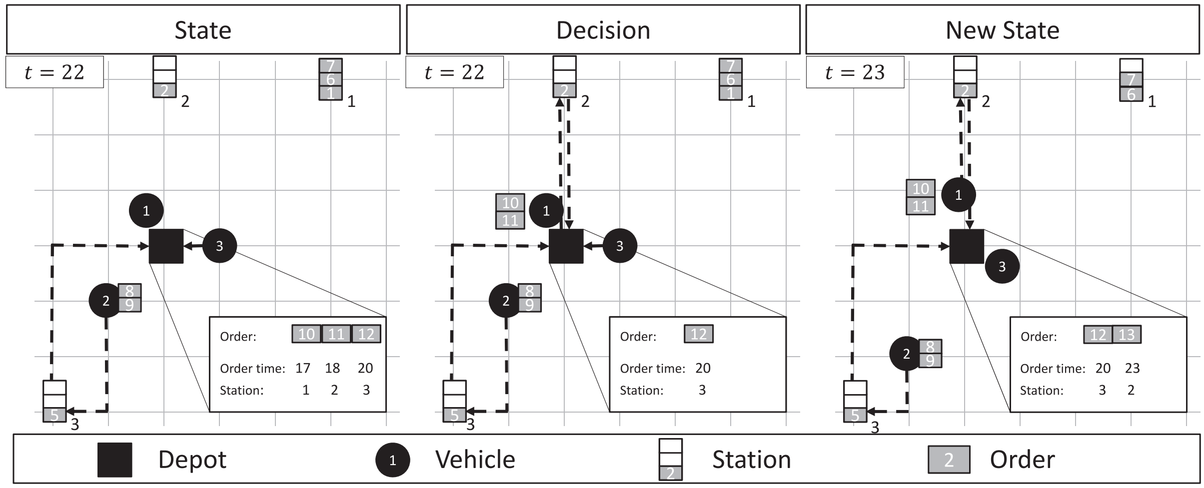

Fig. 2. Example for a Decision State, a Potential Decision, and a Potential new State.

of three. Each portion further shows a street network. For simplicity of the example, we assume a Manhattan-grid. We further assume customers and vehicles travel by the same speed and each segment of the grid requires one time unit of travel for both. The vehicles are represented as black circles. In the example, we consider three vehicles with capacity of 2. Finally, the orders are represented by grey rectangles. Because many orders accumulate at the depot, we depict the information of these orders in large box in the lower right corner. We will now subsequently describe the three portions in detail. For the purpose of presentation, we omit setup and loading times in this example. We further set the acceptable neighborhood radius to $\rho = 3$ time units.

The left portion depicts a decision state. The current point of time is 22. We have three orders at the depot. The Orders 10, 11, 12 occurred at times 17, 18, 20 and the corresponding preferred pickup stations are 1, 2, 3, respectively. Station 1 is currently full with Orders 1, 6, 7. Station 2 contains Order 2 and Station 3 contains Order 5. Order 4 was already delivered and picked up earlier in the day and is therefore not part of the state anymore. Vehicle 1 is currently idling at the depot. Vehicle 2 is on its way to Station 2 carrying Order 8 and 9 and will return to the depot at time 22+3+5=30. Vehicle 3 will return to the depot at time 22+1=23.

For the sake of simplicity, the earliest pickup times for the orders at the stations are not depicted in Fig. 2. However, they are important to estimate when an order may be picked up. For this example, we assume the earliest pickup time was 15 for all Orders 1, 2, 5, 6, 7. The earliest pickup time for Orders 8 and 9 is 22+3=25.

The dispatcher now decides if to dispatch a vehicle and, if so, what to load on the vehicle and where to send it. A potential decision is shown in the center of Fig. 2. This decision loads Orders 10 and 11 to Vehicle 1 and dispatches the vehicle to Station 2. The preferred station for Order 10 is Station 1. However, because the customer travel time between the two stations is 3, this order can also be shipped to Station 2. Furthermore, a shipment of Order 12 is not feasible because Station 3 is currently not able to store another order. The three spots are occupied by Orders 5, 8, and 9. By dispatching Vehicle 1 with Orders 10 and 11, the earliest pickup times for these orders become known and the customers are informed. The time is 22+3=25. Thus, the objective value increases by 15 units: 8 units for Order 10 and 7 units for Order 11.

After determining a decision, the time proceeds to the next step. This step leads to a new state in time 23. A potential new state is shown on the right side of Fig. 2. We observe that all vehicles traveled one segment in the network. Vehicle 3 is back at the depot. We further observe that Order 1 was picked up at Station 1. Finally, a new customer ordered at time 23; Order 13 preferably delivered to Station 2.

Table 2
Notation for the Markov Decision Process.

Description	Notation
Decision State	
State in time t	S_t
Arrival time vector	A_t
Arrival time of vehicle v	a_t^v
Information about orders at depot	$\mathcal{C}_t^D$
Information about orders at stations	$\mathcal{C}_t^{\mathcal{P}}$
Information about orders at station P_i	$\mathcal{C}_t^{P_i}$
Decisions	
Set of decision in state S_t	$\mathcal{X}_t(S_t)$
Decision in S_t	x_t
Set of vehicles dispatched given x_t	$\mathcal{V}_t^x$
Station vehicle v^x is dispatched to	P^{v^x}
Set of orders loaded onto vehicle v^x	$\mathcal{C}_t^{v^x}$
Arrival time of vehicle v given x_t	$a_t^{v,x}$
Information about the remaining orders at the depot given x_t	$\mathcal{C}_t^{D,x}$
Information about the orders at station P given x_t	$\mathcal{C}_t^{P,x}$
Post-decision state given S_t and x_t	S_t^x
Cost function given S_t and x_t	$R(S_t, x_t)$
Transition	
Set of potential realizations	Ω_{t+1}
Specific realization	ω_{t+1}
New orders between t and $t+1$	$\mathcal{C}^{D,\omega_{t+1}}$
Picked up orders at station P between t and $t+1$	$\mathcal{C}^{P,\omega_{t+1}}$

3.3. Markov decision process

In the following, we present the Markov decision process (MDP) model. The notation for the parameters are depicted in Table 2. An MDP models a problem as a sequence of states S_t connected by decisions x_t and transitions based on a stochastic realization ω_{t+1}. In the following, we define the components of the MDP for the SDDPSAV and embed the example in the MDP's notation.

Decision State

A decision state contains all the information available to the decision maker. A decision point occurs in every time step in T^D. State S_t at time t contains three types of information: the status of the vehicles, the orders at the depot, and the orders in the pickup stations. Mathematically, the components are as follows:

1. The information about when the vehicles can be dispatched again. The state contains a vector of arrival times $A_t =$

$(a_t^{v_1}, \ldots, a_t^{v_m})$. If a vehicle v_i is idling at the depot, the arrival time is the current time: $a_t^{v_i} = t$. Else, $a_t^{v_i}$ is the time the vehicle will return.
2. The information about the orders at the depot $\mathcal{C}_t^D$. The information is represented by a set of tuples of order time and preferred station (τ^C, P^C), one tuple for each order C.
3. The information about the orders at each of the n pickup stations $\mathcal{C}_t^{\mathcal{P}} = \left(\mathcal{C}_t^{P_1}, \ldots, \mathcal{C}_t^{P_n}\right)$. The information for each station P_i is again a set containing the information about the orders currently at the station or currently transported to the station: $\mathcal{C}_t^{P_i} = (\tau^{\text{early}}(C_1^{P_i}), \ldots, \tau^{\text{early}}(C_h^{P_i}))$ with h the number of orders currently at or on their way to Station P_i. The relevant information for each order C^{P_i} in P_i is the earliest time it can be picked up $\tau^{\text{early}}(C^{P_i})$ because it indicates the expected pickup time. The time values may be larger than t because it also considers orders currently transported to this station.

In summary, the state in time t is $S_t = (A_t, \mathcal{C}_t^D, \mathcal{C}_t^{\mathcal{P}})$. The initial state is in $t = 0$ with $a_0^v = 0$ for every vehicle. The termination state is in the first $t \geq t_{\max}^O$ with $\mathcal{C}_t^D = \emptyset$ when all orders have left the depot.

In the example given in Section 3.2, we are in state S_{22}. The arrival times of the vehicles are $A_{22} = (22, 30, 23)$. The set of orders at the depot is $\mathcal{C}_{22}^D = ((17, 1), (18, 2), (20, 3))$. The sets for the orders at the pickup stations are the earliest pickup times. Recalling that the earliest pickup time for the orders in the stations was 15 in the example, the set is then

$$\mathcal{C}_{22}^{\mathcal{P}} = \left(\underbrace{(15, 15, 15)}_{\mathcal{C}_{22}^{P_1}}, \underbrace{(15)}_{\mathcal{C}_{22}^{P_2}}, \underbrace{(15, 25, 25)}_{\mathcal{C}_{22}^{P_3}} \right)$$

because Order 8 and 9 will be available for pickup in three time units at time 25.

Decision and Reward Function

Based on a state S_t, the dispatcher selects a decision x_t from the set of decisions $\mathcal{X}_t(S_t)$. A decision consists of dispatching vehicles with a set of loaded orders to a set of stations. A decision is represented by a subset of $0 \leq j \leq m$ vehicles to dispatch $\mathcal{V}_t^x = \{v_1^x, \ldots, v_j^x\}$ and a dispatching decision for each vehicle $v^x \in \mathcal{V}_t^x$ determining the station P^{v^x} and the set of l orders $\mathcal{C}_t^{v^x} = \left((\tau^{C_1^x}, P^{C_1^x}), \ldots, (\tau^{C_l^x}, P^{C_l^x})\right)$ the vehicle delivers to this station. A decision is feasible if the dispatched vehicles are currently at the depot, if orders are only shipped to eligible stations, and if the capacity constraints of vehicles and stations hold. Mathematically, a decision is feasible if the following conditions hold:

1. Vehicles at depot: $a_t^{v^x} = t \; \forall v^x \in \mathcal{V}_t^x$.
2. Eligible stations: $d^C(P^{C^x}, P^{v^x}) \leq \rho \; \forall C^x \in \mathcal{C}_t^{v^x} \; \forall v^x \in \mathcal{V}_t^x$.
3. Vehicle capacity: $|\mathcal{C}_t^{v^x}| \leq \kappa_{\max}^V \; \forall v^x \in \mathcal{V}_t^x$.
4. Station capacity: $|\mathcal{C}_t^P| + \sum_{v^x \in \mathcal{V}_t^x : P^{v^x} = P} |\mathcal{C}_t^{v^x}| \leq \kappa_{\max}^P \; \forall P \in \mathcal{P}$.

A decision can be summarized as $x_t = \left(\mathcal{V}_t^x, (P_t^{v_1^x}, \ldots, P_t^{v_j^x}), (\mathcal{C}_t^{v_1^x}, \ldots, \mathcal{C}_t^{v_j^x})\right)$. A decision changes the status of the vehicles and the sets of orders for depot and stations. Mathematically, a decision x_t changes the arrival time vector A_t to $A_t^x = (a_t^{v_1,x}, \ldots, a_t^{v_m,x})$ as well as the orders at the depot $\mathcal{C}_t^D$ to $\mathcal{C}_t^{D,x}$ and at each station $\mathcal{C}_t^P \in \mathcal{C}_t^{\mathcal{P}}$ to $\mathcal{C}_t^{P,x} \in \mathcal{C}_t^{\mathcal{P},x}$ as follows:

1. Arrival time vector:
   - $a_t^{v^x,x} = t + 2 \times d^V(D, P^{v^x}) + \tau^D + \tau^P \; \forall v^x \in \mathcal{V}_t^x$.
   - $a_t^{v,x} = a_t^v \; \forall v \notin \mathcal{V}_t^x$.
2. Orders at depot: $\mathcal{C}_t^{D,x} = \mathcal{C}_t^{D,x} \setminus \left(\bigcup_{v^x \in \mathcal{V}_t^x} \mathcal{C}_t^{v^x} \right)$.
3. Orders at station: $\mathcal{C}_t^{P,x} = \mathcal{C}_t^P \cup \left(\bigcup_{v^x \in \mathcal{V}_t^x : P^{v^x} = P} \mathcal{C}_t^{v^x} \right) \forall P \in \mathcal{P}$.

The changed (post-decision) state is $S_t^x = (A_t^x, \mathcal{C}_t^{D,x}, \mathcal{C}_t^{\mathcal{P},x})$. Each decision is associated with costs $R(S_t, x_t)$. The function R indicates the increase in delivery time:

$$R(S_t, x_t) = \sum_{v^x \in \mathcal{V}_t^x} \left[\sum_{(\tau^C, P^C) \in \mathcal{C}_t^{v^x}} \left(\underbrace{t + d^V(D, P^{v_x}) + \tau^D + \tau^P}_{\tau^{\text{early}}(C)} - \tau^C \right) \right]. \tag{1}$$

In our example, the decision is to send Vehicle 1 to Station 2 with Orders 10 and 11. The decision is therefore $x_{22} = (\{v_2\}, (P_2), ((17, 1), (18, 2)))$. The costs of the decision are $R(S_{22}, x_{22}) = (25 - 17) + (25 - 18) = 15$. The post-decision state is $S_{22}^x = \left(A_{22}^x, \mathcal{C}_{22}^{D,x}, \mathcal{C}_{22}^{\mathcal{P},x}\right)$ with $A_{22}^x = (28, 30, 23)$, $\mathcal{C}_{22}^{D,x} = ((20, 3))$, and $\mathcal{C}_{22}^{\mathcal{P},x} = ((15, 15, 15), (15, 25, 25), (15, 25, 25))$.

Exogenous Information and Transition Function

The realization of exogenous information ω_{t+1} between time t and $t+1$ from the overall set Ω_{t+1} is twofold for the SDDPSAV. First, new orders occur. Second, customers pick up their goods at the station. Mathematically, the exogenous information is $\omega_{t+1} = \left(\mathcal{C}^{D,\omega_{t+1}}, (\mathcal{C}^{P_1,\omega_{t+1}}, \ldots, \mathcal{C}^{P_n,\omega_{t+1}})\right)$. Set $\mathcal{C}^{D,\omega_{t+1}}$ is the set of new orders made. Set $\mathcal{C}^{P_i,\omega_{t+1}} \subset \mathcal{C}_t^{P_i,x}$ is the set of picked up orders at station P_i. Based on post decision state S_t^x and exogenous information ω_{t+1}, the new state is generated as follows:

1. Arrival time vector:
   - $a_{t+1}^v = t + 1 \; \forall v \in \mathcal{V} : a_t^{v,x} = t$ (Vehicle idles at depot).
   - $a_{t+1}^v = a_t^{v,x} = a_t^v$, else.
2. Orders at the depot: $\mathcal{C}_{t+1}^D = \mathcal{C}_t^{D,x} \cup \mathcal{C}^{D,\omega_{t+1}}$.
3. Orders at pickup station P_i: $\mathcal{C}_{t+1}^{P_i} = \mathcal{C}_t^{P_i,x} \setminus \mathcal{C}^{P_i,\omega_{t+1}}$.

In the example, the realization is that a new order occurs (Order 13) and that Order 1 is picked up at Station 1. Mathematically, the realization is $\omega_{23} = (\{(23, 2)\}, (\{(15)\}, \emptyset, \emptyset))$. The new state is $S_{23} = (A_{23}, \mathcal{C}_{23}^D, \mathcal{C}_{23}^{\mathcal{P}})$ with $A_{22} = (28, 30, 23)$, $\mathcal{C}_{23}^D = (, (20, 3), (23, 2))$ and $\mathcal{C}_{23}^{\mathcal{P}} = ((15, 15), (15, 25, 25), (15, 25, 25))$.

Objective Function

A solution for an MDP is a policy π from the overall set of policies Π. A policy π is a set of decision rules that assigns a decision $X_t^\pi(S_t) \in \mathcal{X}(S_t)$ to every state S_t in the state space. The decision $X_t^\pi(S_t)$ is the decision given state S_t and π in time t. An optimal policy π^* for the SDDPSAV minimizes the expected costs over all decision points beginning from an initial state S_0. Mathematically, π^* is given by:

$$\pi^* = \arg\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{t_{\max}^V} R(X_t^\pi(S_t)) | S_0 \right]. \tag{2}$$

For the SDDPSAV, the costs represent the sum of time between order and earliest pickup over all customer. The optimal policy minimizes the expected overall costs.

4. Dynamic dispatching strategies

In this section, we present our solution method. We give a general motivation and overview in Section 4.1. We then give the algorithmic details of the procedure in Section 4.2.

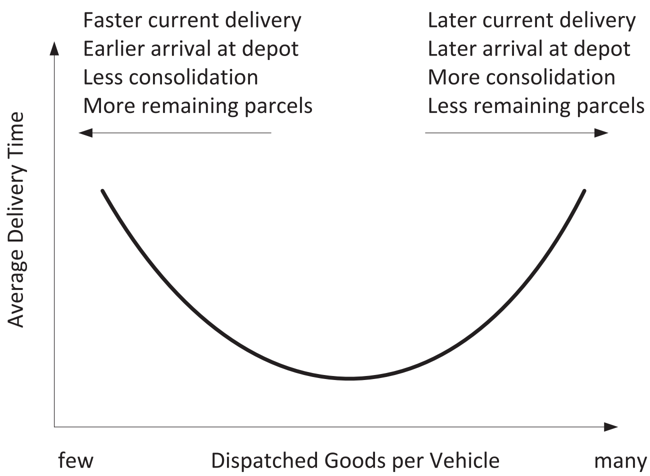

Fig. 3. Tradeoff between Fast and Late Delivery Dispatches.

4.1. Motivation and overview

Finding an optimal policy for the SDDPSAV is challenging because of the well-known "Curses of Dimensionality." The state space, decision space, and space of exogenous information are tremendous. The state comprises information about the orders at the depot and all stations as well as the vehicles. Decisions comprise a subset selection of orders, an assignment of vehicles to these orders, and the set of vehicles to dispatch to a certain point of time. Finally, exogenous information does not only comprise new requests but also the pickup of orders at the pickup stations. Essentially, optimal policies are nearly impossible to obtain. Thus, we draw on a method on approximate dynamic programming, policy function approximation (PFA). PFAs are suitable for the "case that we have a very good idea of how to make a decision, and we can design a function (i.e., a policy) that returns a decision which captures the structure of the problem." (Powell, 2011, p.232). PFAs have two advantages: first, they generally do not require substantial computation in a decision point. As we highlighted in our literature review, other methods such as lookaheads require extensive time for calculations. Such time is not available in real-time decision making for the SDDPSAV. Another important advantage is that decision making with PFAs is often relatively intuitive and therefore easy to communicate to practitioners.

PFAs are highly problem-specific because they utilize the structure of the problem. For the SDDPSAV, we design a PFA that addresses the tradeoff between fast dispatches of a few goods and late dispatches of many goods, often observed in the dynamic dispatching literature (Rivera and Mes, 2017). This tradeoff is illustrated in Fig. 3. The x-axis depicts the number of goods per dispatch.[2] The y-axis depicts the average delivery time over all customers of the entire day.

Fast dispatches of few goods have the advantage that the dispatched goods arrive fast at the station and the corresponding customers do not need to wait very long. Further, the vehicles are available for the next dispatch relatively early. However, fast dispatches reduce the possibility of consolidation and many parcels may remain at the depot once the vehicle returned. Late dispatches have the advantage that usually more orders can be shipped and fewer remain at the depot. However, the customers of the goods currently at the depot need to wait longer and the vehicles arrive later at the depot to start the next trip. As the development of the average delivery time in Fig. 3 suggests and our computational study will confirm, both strategies may therefore lead to inferior solutions and a compromise is needed. Furthermore, the tradeoff is also dependent on the relative resources available for each order. With many resources, fast dispatches are beneficial. With scarce resources, consolidation becomes important (Ulmer et al., 2017). Our PFA addresses this tradeoff by dispatching a vehicle whenever at least a certain number of parcels can be shipped. The minimal number of parcels per shipment is defined by a threshold. A small threshold value leads to fast dispatches of few goods. A large threshold value leads to late dispatches of many goods. In the next section, we will describe the algorithmic procedure to implement the threshold-idea of the PFA.

4.2. Procedure

To implement the PFA, two steps are required. First, for every state, potential dispatching decisions need to be derived. Second, because different instance settings may require different policies, the respective best threshold needs to be determined for each individual instance setting. We will describe the two steps in the following. For the pseudo-code of the algorithm, we refer to Section A.1 of the Appendix.

Potential Dispatching Decisions

First, we describe how dispatching decisions are made by the PFA. In the following, we assume an externally given threshold θ. In every state S_t, a decision x_t needs to be made about the number of vehicles to dispatch, their destination, and the goods loaded to these vehicles. We now describe how we determine this decision in combination with the threshold parameter.

If at least one vehicle for dispatching is available in state S_t, the heuristic iterates through all the stations. For every station P_i, we determine the maximum amount of parcels that can be dispatched. We denote this number $c_t^{P_i}$ for station P_i. This amount depends on the eligible orders for this station at the depot. These orders are orders at the depot for both the particular station as well as for stations in the neighborhood with customer travel time not larger than radius ρ. The set of eligible orders is summarized as $\{C \in \mathcal{C}_t^D : d^C(P^C, P^i) \le \rho\}$. Further, the amount $c_t^{P_i}$ for station P_i depends on the free capacity at the station, $\kappa_{\max}^P - |\mathcal{C}_t^{P_i}|$. In essence, value $c_t^{P_i}$ is defined as

$$c_t^{P_i} = \min\{|\{C \in \mathcal{C}_t^D : d^C(P^C, P^i) \le \rho\}|, \kappa_{\max}^P - |\mathcal{C}_t^{P_i}|\}. \tag{3}$$

The heuristic now selects the station P_t^* with the highest possible amount:

$$P_t^* = \arg\max_{P_i \in \mathcal{P}} \{c_t^{P_i}\}. \tag{4}$$

In case there are several stations with the same amount, the station is chosen with more free capacity available. If several stations have the same free capacity, the station with the smallest travel time is selected. In the (very rare) case that several stations also have the same travel time, the station with the smaller index is selected.

If the amount of parcels shipped to P_t^* is smaller than the threshold ($c_t^{P^*} < \theta$) the heuristic terminates and no vehicle is dispatched in this decision state. If value $c_t^{P^*}$ is equal or larger than the threshold, a vehicle is sent to station P_t^*. In case there are more eligible parcels than free capacity at station P_t^*, the heuristic applies first-in-first-out to avoid postponing orders indefinitely. This procedure is repeated as long as vehicles are available for dispatching and the heuristic has not terminated yet.

There is a special case for states S_t after the order phase, $t \ge t_{\max}^O$. For these states, no future orders will occur, however, there

[2] We additionally experimented with varying the frequency of dispatches. The results are inferior and can be found in the Appendix.

are still stochastic pick ups at the stations. The goal is to ship the remaining orders fast. To this end, the described procedure is applied but the threshold is set to $\theta = 1$. That means that all vehicles are dispatched when available and they can deliver at least one parcel.

Determining the threshold

We now need to determine the threshold for each individual instance setting. Mathematically, we have a set of PFA policies $\bar{\Pi} \subset \Pi$ with $\bar{\Pi} = \{\pi_\theta : \theta = 1, \dots \kappa^V_{\max}\}$. We are looking for the policy $\pi^*_\theta \in \bar{\Pi}$ minimizing the expected costs:

$$\pi^*_\theta = \arg\min_{\pi_\theta \in \bar{\Pi}} \mathbb{E}\left[\sum_{t=0}^{t^V_{\max}} R(X_t^{\pi_\theta}(S_t))|S_0\right]. \quad (5)$$

Because the number of potential policies is the same as the vehicle capacity, the best policy can be found by means of enumeration. However, even for the reduced set of policies, calculating the expected value is impossible due to the dimensionality of the state and information space. Thus, we combine the enumeration of the policies with sample average approximation. For each instance setting, we generate N realizations of the exogenous process $\hat{\omega}^1, \dots, \hat{\omega}^N$ and calculate the average costs over the N realizations:

$$\pi^*_\theta = \arg\min_{\pi_\theta \in \bar{\Pi}} N^{-1} \sum_{i=1}^{N} \left[\sum_{t=0}^{t^V_{\max}} R(X_t^{\pi_\theta}(S_t))|S_0, \hat{\omega}^i\right]. \quad (6)$$

In our computational study, we set $N = 1000$. We then use π^*_θ as the PFA-policy. Because the suitability of thresholds depends on the dimensions of the instance setting, we determine individual thresholds for each instance setting. We denote our policy π^{PFA}. For an example of the tuning results, the interested reader is referred to Section A.2 of the Appendix.

5. Computational study

In this section, we present our computational study. We first describe the instance settings for the real-world network of Braunschweig, Germany and define the benchmark policies. We then compare our PFA with the benchmark policies and analyze the threshold values for different instance settings. We present a sensitivity analysis for the relevant dimensions of the problem with a particular focus on the impact of customer behavior on the delivery times. We analyze how different neighborhood sizes impact the utilization of the stations to derive implications for strategical and tactical planning. We finally pair our PFA with value function approximation to account for heterogeneity in customer demand.

5.1. Instances

We generate instance settings for our main study and, later, additional instance settings for our sensitivity analysis. In the following, we describe the instance settings for our main tests.

We test our policy for the city of Braunschweig, Germany. Braunschweig is a medium-sized city in a traditional European City layout. The city as well as the depot and the pickup stations are depicted in Fig. 4. The depot is set to the DHL delivery base in Braunschweig and the pickup stations represent the “DHL Packstation” scheme in Braunschweig (Persiel, 2017). We observe that the pickup stations are often close to major city streets. These major streets are likely to be the first eligible for autonomous vehicles. As an example, autonomous vehicles of SAE-level 3 have driven on the main ring road of Braunschweig since 2010 (Nothdurft et al., 2011). Stations 2, 3, 5, 7, 8, 9, and 10 are directly located at this main ring road.

The DHL networks is also used for conventional delivery and drop off of parcels. In our main study, we assume that the DHL stations reserve a capacity of 20 orders for same-day delivery. Thus, we set $\kappa^P_{\max} = 20$. In our main experiments, we draw on a fleet of

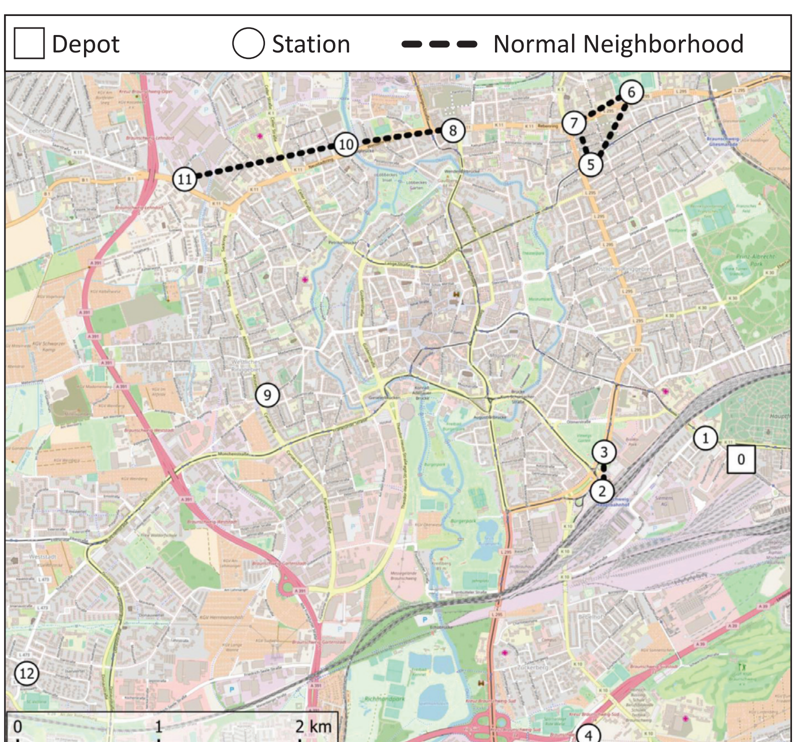

Fig. 4. Pickup Station Network for Braunschweig (©OSM).

six vehicles. The setup and unloading time for the vehicles is set to $\tau^D = \tau^C = 10$ min. Because it is challenging to determine the specifications of the autonomous vehicles with certainty, we vary the parameters. In our main study, we analyze different speeds (In our sensitivity analysis, we later also analyze different capacities). After discussions with researchers conducting experiments on autonomous vehicles in Braunschweig, we assume three different speeds: conservative, normal, and optimistic. We set *conservative* as half the speed assumed for cars by Google Maps. *Normal* reduces the speed by one third, and *optimistic* assumes the original speed. The travel times are derived from Google Maps at time without traffic and are depicted in Table A2 in the Appendix. Notably, the travel times base on the road network of Braunschweig. This network is needed for autonomous ground vehicles but aerial vehicles may be able to choose direct travel ignoring the road network. However, in Germany, drone flight over buildings is prohibited. Thus, drones may also need to follow the road network. For our main tests, we set the capacity of a vehicle to $\kappa^P_{\max} = 10$ but we vary it in our sensitivity analysis.

The order horizon is set from 8am to 4pm and comprises 8 h, $t^O_{\max} = 480$ min. The delivery horizon is set sufficiently large to 20 h, $t^D_{\max} = 1200$ min.[3]

For the customer orders, we assume a Poisson request process over time. We test instances with number of orders 100, 200,..., 1000. We assume an expected pickup time of 60 min. We model the pickup time distribution as Gamma distribution with coefficient of variation of 0.1. A Gamma distribution has a long tail. That means that many orders are picked up around 60 min after the customer is informed but a few orders may be picked up a significant amount of time later.

We analyze three neighborhoods. We call the neighborhoods *none, normal*, and *large*. We assume the original Google Maps travel times for customer travel times. Thus, all neighborhoods' travel time radii are not impacted by the vehicle speed. The *none* neighborhood requires delivery to the preferred pickup station, the radius is $\rho = 0$ min of original Google Maps travel time. The *normal* neighborhood allows a radius of $\rho = 3$ min. We finally test *large* neighborhoods with a travel times radius of $\rho = 5$ min. An example for the impact of the neighborhoods is depicted in Fig. 4. The dashed lines connect pairs of stations that are eligible for delivery given a normal neighborhood size. For example, deliveries for Station 5 can also be delivered to Station 6 and 7. This phenomenon is regularly observed by one of the authors where goods are not delivered to the preferred Station 5 but to Stations 6 or 7. Furthermore, deliveries for Station 10 may be delivered to Station 8 or 11. However, deliveries for Station 8 are not eligible to be delivered to Station 11.

The combination of 10 numbers of orders, 3 different speeds, and 3 different neighborhoods leads to overall 90 different instance settings. For each of the instance settings, we run 1000 evaluation runs. Besides the main tests, we additionally test selected settings with different pickup times, capacities, and number of vehicles in our sensitivity analysis.

5.2. Benchmark heuristics

To analyze the tradeoff between fast delivery due to early dispatches and consolidation due to late dispatches, we compare our policy to two benchmark policies. The first policy π^{fast} dispatches a vehicle whenever orders can be shipped. It can be seen as π^{PFA} with threshold of $\theta = 1$. The second policy π^{late} aims on consolidation. It therefore only sends full vehicles. It can be seen as π^{PFA} with threshold of $\theta = \kappa^V_{\max}$. Both policies draw on the same procedure in the determination of goods to load and shipping station as π^{PFA}. They also follow the same procedure as π^{PFA} after the order phase, for $t \geq t^O_{\max}$.

[3] Another major advantage of autonomous vehicles is that they do not have any working hour restrictions. In our experiments, all orders are usually delivered before 8pm except in a few extreme instance settings with few resources and high demand such as 1000 orders, no neighborhood, and conservative vehicle speed.

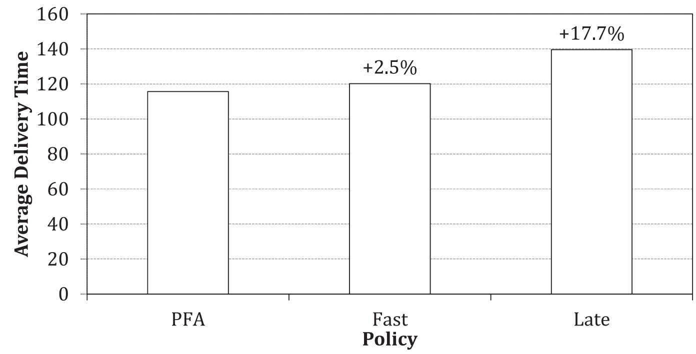

Fig. 5. Average Delivery Times and Average Improvement of PFA.

5.3. Average delivery times

In the following, we compare the average delivery times for our different policies. To this end, we calculate the average delivery time over all 1000 realizations for every instance setting for policies π^{PFA}, π^{fast}, and π^{late}. The individual results are listed in the Appendix. We then calculate the average over these values. The results are depicted in Fig. 5. The x-axis shows the policy. The y-axis shows the average results. We observe that π^{PFA} performs best with average delivery times of about 115 min. Policy π^{fast} performs relatively well with average delivery times of slightly more than 2 h while π^{late} performs substantially worse with delivery times of about 140 min. Additionally, we calculated the average improvement of π^{PFA} over a policy π as

$$\frac{\pi - \pi^{\text{PFA}}}{\pi}.$$

The individual improvements are listed in the Appendix. The average improvement of π^{PFA} compared to π^{fast} and π^{late} are depicted as percentages at the bars. We observe that π^{PFA} outperforms π^{late} significantly by 17.7% while the improvement compared to π^{fast} is less distinct with 2.5%. However, as the detailed results in the Appendix show, π^{PFA} outperforms π^{fast} by up to 8.4% for individual instance settings, especially with higher workload.[4]

In the following, we analyze the average delivery time in more detail. We also show how the threshold is impacted by the instance dimensions. We first analyze how average delivery time and threshold are impacted by the workload; the expected number of orders. To this end, Fig. 6 depicts two types of results: the average delivery time with increasing workload and the threshold development with increasing workload.

The x-axis shows the workload as the expected number of orders. Each number of orders represents a group of 9 different instance settings with 3 different vehicle speeds and 3 different neighborhoods. The left y-axis represents the average delivery time per instance setting group. The values are represented by the bars in the graph. For the groups of instances, the right y-axis depicts the average thresholds in correspondence with the grey line.

We first analyze the average delivery times, the bars and the left y-axis. We observe relatively constant delivery times below two hours for instances with up to 600 orders and therefore not more than 100 orders per vehicle. We then observe an increase

[4] The standard deviation in improvements is 2.7% over π^{fast} and 23.3% over π^{late}.

as the workload becomes very high. The results suggest that we are able to serve up to 100 customers per vehicle on average in usually less than 2 h after the time of request. Recalling the literature on conventional same-day delivery where a vehicle is usually only able to serve 20 to 30 customers per day, this is a substantial increase in productivity. Thus, using pickup stations instead of individual delivery may allow significant reductions in required resources.

We now analyze the development of the threshold. We observe a constant increase in the threshold with increasing workload. With only a few orders per vehicle, the threshold is small leading to fast dispatches. This is reasonable because we have sufficient resources available while the potential for consolidation is relatively small. With increasing workload, our resources become scarce and both the potential and requirement for consolidation increases. Thus, the threshold increases as well.

The individual results in the Appendix show different delivery times for instance settings with the same workload. The differences result from varying neighborhood sizes and vehicle speeds. For a detailed analysis of the differences, we refer to Section A.3 in the Appendix. In summary, with larger neighborhood size and faster vehicles, the delivery time decreases because of relatively more consolidation opportunities and relatively more resources available.

As shown in this section, π^{PFA} outperforms the benchmark policy with respect to average delivery times. However, there are other measures for customer satisfaction. It may be that with our policy, a few customers have to wait exceptionally long while the vast majority is served fast. The few very dissatisfied customers may then abandon the business. The other policies may have longer average delivery times but may avoid very late services, especially policy π^{fast} dispatching a vehicle with goods whenever possible. To measure very late deliveries, for every instance setting, we calculate the average maximal delivery time a customer experienced. For the 1000 test instances, we calculate the average over the longest delivery time observed each day. We then calculate the average over all instance settings. The average maximal delivery time for π^{PFA} is 286.3 min, for π^{fast} 290.7 min, and for π^{late} 359.6 min. Thus, π^{PFA} also reduces the maximal delivery time compared to the benchmark policies.

For the maximal delivery times, we observe the same tendency as shown in Fig. 6 if we group instances by workload and calculate the average for each group. For the instance groups of 100 orders and even up to 600 orders, the average maximal delivery time is small and below 4 h. Thus, the resources are sufficient to deliver every parcel within reasonable time. With higher workloads, the maximal delivery times increase substantially. As an extreme, for the group of instances with 1000 orders, the average maximal delivery time is more than 8 h. In these cases, the available delivery resources become insufficient for acceptable service quality.

Table 3
Results for the Sensitivity Analysis.

	Vehicle Capacity			
Value	5	**10**	15	20
Average Delivery Time	327.6	**90.6**	68.0	67.2
		Station Capacity		
Value	10	**20**	30	40
Average Delivery Time	105.8	**90.6**	91.2	91.6
		Number of Vehicles		
Value	4	**6**	8	10
Average Delivery Time	208.9	**90.6**	59.7	53.6
		Pickup Time		
Value	**60**	120	180	240
Average Delivery Time	**90.6**	90.2	117.4	168.0

5.4. Sensitivity analysis

In this section, we analyze the dimensions of the MDP in detail. To this end, we vary number of vehicles, capacities of stations and vehicles as well as customer pickup times for the "average" instance setting. For our tests, we chose the instance setting with 600 orders, normal speed, and normal neighborhood. On average, the delivery time for this instance setting is 90 min which lays between 1 and 2 h delivery times, a common goal for many providers. We now vary the individual dimensions. For each variation, we repeat the tuning procedure described in Section 4. The results are depicted in Table 3. We depict the standard setting in bold.

First, we analyze the vehicle capacity. In our main tests, we assumed that a vehicle is able to carry up to 10 orders. In case, we reduce the capacity to 5, we observe a tremendous increase in delivery time. This indicates that exploiting the consolidation effect of pickup stations requires a sufficient vehicle capacity. Increasing the capacity to 15 reduces delivery time significantly. With a vehicle capacity of 20 similar to the station capacity, we observe stagnation. Thus, using vehicles with the same capacity as the stations does not add any significant benefit.

We observe a less distinct behavior with varying station capacity. Reducing the capacity to 10 leads to an increase in delivery time of about 15 min. Thus, even smaller stations may allow relatively fast same-day delivery. Increasing the station capacity any further than 20 does not add any benefit. We even observe a very small increase in delivery time which may result from our heuristic method.

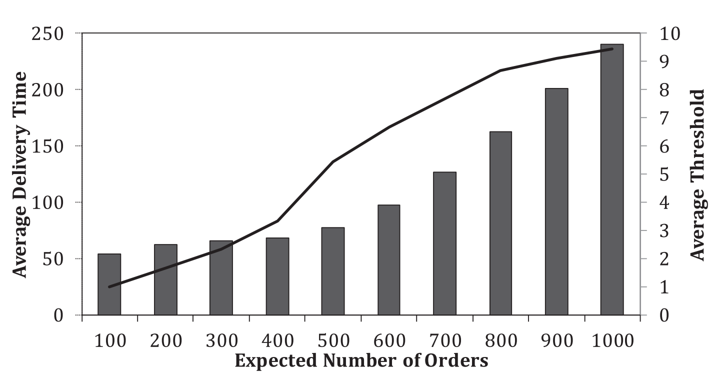

Fig. 6. Average Delivery Times and Thresholds over Number of Orders.

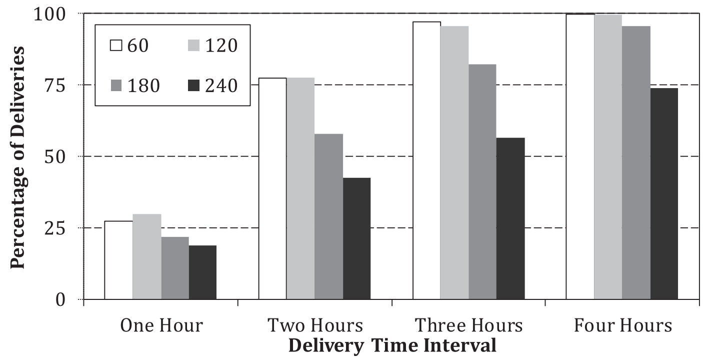

Fig. 7. Average Percentages of Delivery Time Intervals For Varying Customer Pickup Times.

Reducing the number of vehicles to 4 leads to substantially longer delivery times. The reduction leads a workload per vehicle of 150 orders which is too much for fast same-day delivery. This observation is in accordance to Fig. 6 where we could see that with more than 100 orders per vehicle, delivery times increase significantly. Also similar to Fig. 6, the delivery times decrease to around 60 min if the workload is further reduced in cases of 8 and 10 vehicles. We also observe a shift in the best threshold. With increasing numbers of vehicles, the suitable threshold decreases because resources are sufficient for fast delivery.

5.5. Customer pickup discipline

In the following, we analyze the impact of pickup times. In our main tests, we assumed that customers want their goods relatively fast and pick them up on average 60 min after they arrived at the pickup station. However, customers may be less disciplined in their pickup behavior. Because logistic service providers compete with other stakeholders such as car drivers, or shared-mobility providers for space in the city, the station capacities are limited. Longer pickup times may therefore block valuable locker space and impede the delivery process. To analyze the impact of later pickups, we test average pickup times of 120, 180, and even 240 min. The results are shown in the bottom of Table 3. We see that with 120 min pickup times, we achieve nearly the same results as with 60 min. With pickup times of 180 min, we observe an increase of about half an hour. In case, customers pick up their goods on average four hours after delivery, the delivery times increase substantially to nearly 3 h. That means that in case customers take a long time to pick up the goods, the service quality decreases.

The impact is visualized in Fig. 7. The graph shows the percentages of customers served within a certain time interval. On the x-axis, the time intervals are depicted. “One Hour” indicates the percentage of customers with delivery times of less than 60 min. “Four Hours” indicates the percentage of customers with delivery times less than 240 min. Each entry on the x-axis shows four different bars. Each bar represents an expected pickup time: 60, 120, 180, and 240 min. The white bars indicate the “normal” instance setting with expected pickup times of 60 min. We see that about 25% of customers have delivery times of less than 60 min, 75% have delivery times of less than 2 h, and nearly everyone is served after 3 h. We see a similar pattern for expected pickup times of 120 min. However, we then observe a tremendous drop for 180 and 240 min expected delivery time. If customers on average take 240 min to pick up their goods, about one quarter of customers does not receive their goods within 4 h. This observation leads to the managerial implication that it is very important to incentivize customers to pick up their goods relatively fast or to communicate that longer pickup times lead to less service quality. This phenomenon is also observed for conventional next-day delivery to pickup stations where DHL sends a frequent reminder to pick up the ordered goods.

5.6. Delivery times versus station preference

In this section, we address the tradeoff between the two goals of fast delivery and delivery (close) to the preferred station. To this end, we again draw on the instance setting with 600 orders, 6 vehicles, normal speed, and expected pickup times of 60 min. Dependent on the neighborhood, we observe a parcel migration between pairs of stations. The migration matrices can be found in the Appendix. In the following, we present a set of key figures to capture the tradeoff between delivery times and station preference.

We vary the neighborhood size and calculate three values, the average delivery time, the percentage of customers with goods delivered to their preferred station, and the average detour per customer from the preferred station to the station the goods were delivered to instead. For the detour, we assume the worst case meaning that the detour is from the preferred station to the station with the goods and back to the preferred station. In reality, the detour may be much smaller, for example, if the customer lives between both stations. Mathematically, the detour is calculated as $2 \times d^C(P^C, P^*)$ with P^* the station the goods were delivered to.

The results are depicted in Table 4. With increasing neighborhood sizes, we observe decreasing delivery times as well as increasing detour times and decreasing percentage of deliveries to the preferred stations. However, the delivery time reduction is relatively large compared to the increase in the detour times. For example, if allowing a large neighborhood, customers may receive their goods on average about 24 min earlier while only adding less than 3 min to their travel. A direct comparison between the two values is difficult because additional travel may be more inconvenient for the customer than waiting longer for the goods. However, the very large differences between the two values is an indicator for the potential value of delivery neighborhoods for both companies and customers.

Because the tradeoff may further be customer dependent, the obtained results can also be used as a starting point for demand

Table 4
Tradeoff between Average Delivery Time and Average Percentage of Deliveries to the Preferred Station.

Neighborhood	Delivery Time (min)	Detour Time (min)	Percentage
None	105.7	0.0	100.0%
Normal	90.6	1.3	77.6%
Large	81.4	2.9	62.4%

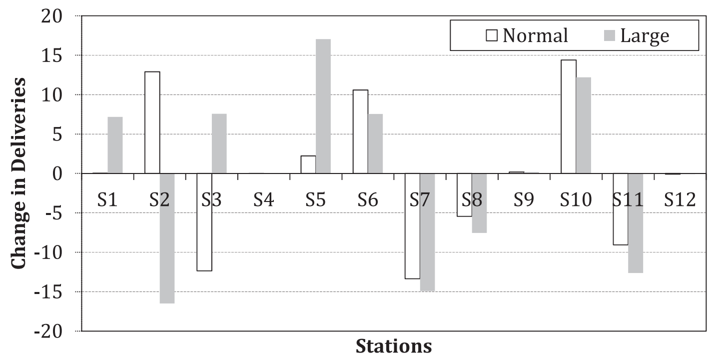

Fig. 8. Differences in Station Utilization for Normal and Large Neighborhood Sizes compared to No Neighborhood.

management, for example, for customer-dependent neighborhoods. Customers may chose between fast (and free) delivery to a station in a larger neighborhood. Customers may also be able to select a specific delivery station without any neighborhood at the expense of longer waiting times.

5.7. Station utilization and capacity setup

We now analyze how the neighborhood impacts the utilization of stations. This analysis is particularly valuable for the strategical and tactical planning levels of setting up a network of capacitated stations and defining a neighborhood for these stations. In this analysis, we examine the average number of deliveries per station for different neighborhoods and the aforementioned instance setting of 600 orders, 6 vehicles, normal speed, and pickup time of 60 min. In our instance settings we use equal probabilities for a station to be the preferred station of a customer. Thus, every station is preferred by on average 50 orders per day. In case of no neighborhood, these orders are delivered to the particular stations. Thus, every station gets 50 orders delivered per day on average. The values change once we include neighborhoods and allow delivery to adjacent stations. To analyze the changes, we calculate the difference of average delivered orders for every station and the normal and large neighborhood. A negative value indicates a decrease of deliveries compared to the no-neighborhood case. A positive value indicates that a station is utilized more than average.

We depict the results in Fig. 8. The x-axis indicates the stations 1 to 12. The y-axis shows the change in delivery. The white bars show the results for the normal neighborhood. The grey bars show the results for a large neighborhood. We now analyze the developments for three different sets of stations: Stations 5, 6, 7; Stations 8, 10, 11; and Stations 1, 2, 3:

Station 5, 6, and 7 are interchangeable for both normal and large neighborhoods. We observe that in both cases, Stations 5 and 6 are utilized more while Station 7 is utilized less.[5] Stations 5 and 6 are closer to the depot and are able to accommodate many orders for Station 7. This indicates that it may be reasonable to increase capacity for Stations 5 and 6 and reduce the capacity of Station 7 or even remove Station 7 entirely.

A similar behavior can be observed for Stations 8, 10, and 11. Station 10 becomes more utilized while Stations 8 and 11 show a reduced utilization. In this case, not the numbers of deliveries to the closest Station 8 is increased but to the station which is adjacent to more stations. This phenomenon is also often observed in close-enough vehicle routing. Station 10 is adjacent to both Stations 8 and 11. For strategical and tactical planning, it may therefore be reasonable to increase capacity for this station.

[5] The large increase for Station 5 and a large neighborhood results from the fact that even more stations are adjacent to Station 5 given a large neighborhood.

Eventually, we analyze the behavior for Station 1, 2, 3. We observe that for a normal neighborhood, Station 2 is utilized more than Station 3. Station 1 does not have any adjacent stations and therefore does not have a change in utilization for a normal neighborhood size. The Google Maps travel time from depot to Station 3 is larger than the travel time between the depot and Station 2. For a small neighborhood, the vehicles are dispatched to the closer station. Thus, Station 2 receives many parcels for Station 3. Interestingly, this behavior changes for a large-size neighborhood. Here, Station 3 receives more goods than Station 2. For a large neighborhood, deliveries to Station 2 become also eligible to be delivered to Station 1. Thus, many orders are delivered to Station 1 close to the depot. The remaining orders are consolidated and delivered to Station 3 which now also is eligible for deliveries destined to Station 5. This example indicates that (strategical) capacity considerations should be made in tandem with (tactical) neighborhood sizing decisions. If the providers wants to offer more customer-friendly small neighborhoods, different stations may be utilized than in the case that the providers aims on efficient delivery with large neighborhoods.

5.8. Heterogeneous demand

In the main study, we assume homogeneous demand over time to systematically analyze the problem dimensions. However, customers may order more frequently at certain times of the day. For such instances, having one policy-threshold for the entire day may therefore be insufficient. In this section, we analyze how heterogeneous demand impacts our service quality. We further extend our policy to enable time-dependent thresholds. We show that for homogeneous demand, our PFA performs well even with a single threshold while for heterogeneous demand, time-dependent thresholds are beneficial.

For the instances with heterogeneous demand, we assume two demand peaks, one in the morning and one in the afternoon. We split the 8 h-ordering phase in three segments, morning (2 h), noon (4 h), afternoon (2 h). During the peaks in the morning and afternoon, demand is four times as high compared to “off-peak” demand around noon. The first and last two hours have high demand with 80 expected orders per hour. The four hours in between have low demand with 20 expected orders per hour. This leads to instances with 600 expected orders.

Our PFA uses a single threshold to determine if a vehicle is dispatched. To account for heterogeneous demand, a time-dependent threshold may be beneficial. Thus, we split the time horizon into $p_{\max}$ periods, for example, hours. Each period $1 \leq i \leq p_{\max}$ has an individual threshold θ_i. With a maximal vehicle capacity of $\kappa^V_{\max} = 10$, this results in $10^{p_{\max}}$ potential policies and enumeration of all combinations is not possible anymore. To search the space of policies, we therefore adapt the method introduced in

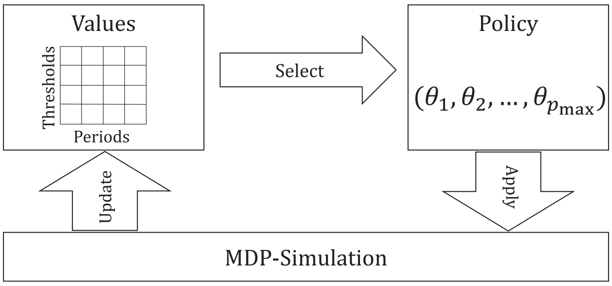

Fig. 9. Iteration of the Value Function Approximation.

Table 5
Average Delivery Times.

	PFA	Time-Dependent		
Periods	1	4	6	8
Homogeneous	90.6	89.1	89.1	90.1
Heterogeneous	123.7	120.9	116.6	113.9

Brinkmann et al. (2019). This method uses value function approximation (VFA, Powell 2011) and its procedure is illustrated in Fig. 9. The VFA approximates the "values" V, the expected sum of delivery times starting from the current period until the end of the day for each combination of period and threshold (i, θ_i). The values $V(i, \theta_i)$ are stored in a two-dimensional lookup table, one dimension for the periods $1, \ldots, p_{\max}$, the other for the thresholds $1, \ldots, \kappa^V_{\max}$. The VFA starts with an initial approximation of zero for all combination. It then iteratively selects a solution candidate as a $p_{\max}$-dimensional vector of thresholds $(\theta_1, \theta_2, \ldots, \theta_{p_{\max}})$ based on the current values. Next, it simulates the MDP with the corresponding policy for the 1000 training instances. It finally updates the values for the observed combinations. Over the iterations, the value approximations become more accurate and better policies are selected. For updates and exploration, we draw on the same tuning as presented in Brinkmann et al. (2019). Because the number of periods impacts both the accuracy of the policy and the size of the solution space, we test different numbers of periods, $p_{\max} = 8$ periods with length of one hour, $p_{\max} = 6$ periods of 1.5 h, and $p_{\max} = 4$ periods of two hours. We run 3000 VFA-iterations for each number of periods.

We compare the time-dependent policy to the PFA with a single threshold for two instance settings both with 600 expected orders, one with heterogeneous and one with homogeneous demand. The other instance parameters are 6 vehicles of capacity 10, station capacity of 20, normal vehicle speed, normal neighborhood, and expected pickup times of 60 min.

The results are depicted in Table 5. We show the average delivery times for homogeneous and heterogeneous demand and for policies with 1, 4, 6, and 8 periods. The policy with 1 period is equivalent to policy π^{PFA}. The first observation is that regardless the policy, delivery times are shorter for homogeneous demand. On average, delivery is made in about 90 min for homogeneous demand while the average delivery time for heterogeneous demand is about two hours. With heterogeneous demand, we observe peak hours with many orders resulting in a delivery backlog and longer delivery times.

We further observe that the differences of the policies is very small for homogeneous demand. The simple policy π^{PFA} with only a single threshold performs only slightly worse than the more sophisticated policies with time-dependent thresholds. The delivery time reduction of the more sophisticated policies is at most 1.7%. The results indicate that for homogeneous demand as assumed in our main study, a single threshold is very reasonable.

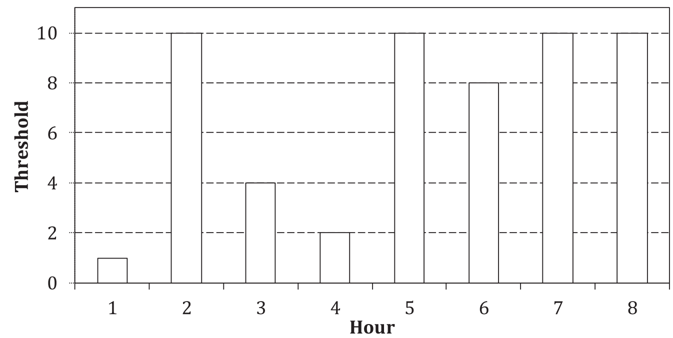

Fig. 10. Hourly Thresholds for Heterogeneous Demand.

For heterogeneous demand, we observe significant differences between π^{PFA} and policies with time-dependent thresholds. The best time-dependent policy is the policy with one threshold per hour, $p_{\max} = 8$. This policy allows for about 10 min faster delivery on average. This is a reduction of delivery time of 8.6%. Thus, if demand is heterogeneous over time, a more complex, time-dependent policy may be advantageous.

In the following, we show how the policy adapts to the heterogeneous demands. To this end, we depict the threshold per hour in Fig. 10. We observe that the threshold controls the resource utilization based on the expected current and future demand. In the beginning of the day, all vehicles idle at the depot. Resources are sufficiently available and the threshold is $\theta_1 = 1$ sending out every available vehicle. The expected demand is large in the first two hours. Thus, in the next hour, the threshold is high to enable consolidation and to account for the high workload. The expected demand in hours 3 to 6 is small. Sufficient resources for fast delivery are available and, in hours 3 and 4, the threshold is small sending out vehicles relatively fast. From hour 5 on, the threshold increases again in preparation of the high demand following in the final periods 7 and 8.

6. Conclusion

In this paper, we have proposed and analyzed a new delivery concept for same-day delivery. Instead of delivering goods individually to customers at their homes, deliveries are made to a nearby pickup station. This has two advantages. First, the delivery of a sets of orders to pickup stations allows for consolidation. Second, the delivery from goods between depot and pickup stations can be automated and conducted by autonomous vehicles. SDD to pickup stations may therefore resolve the issue of driver shortages and high costs while avoiding many technical issues currently unresolved for autonomous vehicles. In this paper, we have modeled the operational problem of dynamic dispatching between depot and stations as a Markov decision process. We have developed a runtime-efficient policy function approximation that allows for intuitive dynamic dispatching decisions in real-time. Our PFA balances the tradeoff between fast dispatches for fast deliveries and late dispatches for consolidation. For a variety of instance settings, we have shown that with our policy, a vehicle can serve about 100 customers per day within an average delivery time of less than 2 h. This is significantly more than the usual number of deliveries of about 20 to 30 reported in the same-day delivery literature (Ulmer, 2017a). We further have analyzed how the setup of vehicles and stations as well as customer behavior impact the delivery times. We also have provided managerial insight for the strategical and tactical planning level. We finally have presented means to extend the proposed method to capture heterogeneity in the customer demand.

We have presented a pioneering paper combining autonomous vehicles with same-day delivery and pickup stations. We have also presented the first paper combining pickup stations and autonomous vehicles as well as pickup stations and same-day delivery. Thus, there is substantial potential for future research in this domain. In the following, we present promising research directions for the presented problem as well as promising problem extensions.

For the problem, the test instances as well as the methodology may be extended. The customer demand may be based on real-world data, if available. Beside heterogeneity over time, this demand data could show heterogeneity in demand per station as well as differences in the expected pickup times. Furthermore, the neighborhood definition may be more complex. Customer may request delivery either to a station at work or at home. In that context, the PFA may be extended to meet the station heterogeneity. Suitable thresholds may not only vary over time but also per station. Another promising avenue could be the prediction of customer pickups. If it is likely that many goods are picked up at a station soon, the dispatching to this station may be postponed to allow the delivery of more parcels. In some cases, even “overbooking” stations may be suitable, i.e., sending parcels to the station even though the capacity is currently insufficient. This may be for example applicable if the station is relatively far away and the current probability of timely pickups is high.

The problem may be extended in several dimensions, for example, by considering heterogeneity in vehicles, stations, and customers. Autonomous vehicles may be heterogeneous in their speed and capacity. Small drones may be for example used to serve stations with low expected demand. Instead of only autonomous vehicles, also conventional vehicles may be considered. These vehicles may be able to serve several pickup stations in one trip and even serve some “convenient” customers directly at their homes. In that context, stations may be heterogeneous in their size and accessibility for autonomous vehicles. Some stations may be only accessible for conventional vehicles. Other stations may be larger because of heterogeneous demand. Larger stations may be further used as hubs for the conventional vehicles. The autonomous vehicles would then resupply the hubs while the conventional vehicles conduct the delivery trips to the individual customers or to other stations. Extending the problem poses several methodological challenges. For example, an orienteering problem has to be solved in real-time to account for subsequent visits of multiple stations. Furthermore, for two-echelon delivery, vehicles need to be synchronized to hand over the parcels at the stations.

Another promising modeling extension is to consider heterogeneity in the customers. Some priority customers may have narrow delivery deadlines and small pickup station neighborhoods or even demand direct delivery to their homes while conventional customers may not have any deadlines and are very flexible in their pickup station. In this context, revenue management may be applied to incentivize customers in their selection of pickup station, neighborhood, and deadline. Another implication of our study is that customer pickup times have a significant impact on the overall performance of our system. Thus, it may be beneficial to communicate estimated arrival times and potential pickup stations at time of the customer's order. This may allow for a faster and less volatile pickup increasing station capacity and system flexibility.

Acknowledgment

The authors like to thank Marcus Nolte from the Department Vehicle Electronics at the Technische Universität Braunschweig for his valuable advice on autonomous vehicles, particularly, on the elaborated tests already conducted in Braunschweig. The authors further thank the Associate Editor and the four Anonymous Referees for the constructive and detailed remarks.

Appendix

In the Appendix, we present the algorithmic details of the PFA. We further present an example for the tuning process and an analysis of the neighborhood and travel speed impact. Finally, we present individual data such as travel times as well as the individual results of our main tests.

A.1. Algorithm

Finally, we present the pseudo-code for our PFA in Algorithm 1. The algorithm is applied in each decision point. Input is the current state S_t and the threshold θ. Output of the algorithm is a decision x_t as a 3-tuple of the set vehicles to dispatch $\mathcal{V}^x$, the set of corresponding stations $\mathcal{P}^x$ and the sets of corresponding orders $\mathcal{C}^x$. The algorithm initializes by determining the set of available vehicles $\mathcal{V}$. The set $\mathcal{V}$ is derived by function *AvailableVehicles*(S_t). This function selects all vehicles v with $a_t^v = t$ in state S_t. The algorithm further initializes the components of the decision and the initial decision $x_t = (\emptyset, \emptyset, \emptyset)$. The initial decision represents “do nothing”.

Starting in line 8, the algorithm now subsequently checks for new dispatches - as long as vehicles are available and no

Algorithm 1: Threshold-Based Policy Function Approximation.

Input: State S_t, Threshold θ.
Output: Decision x_t.
1 // Initialization
2 $\mathcal{V} \leftarrow$ *AvailableVehicles*(S_t)// Filtering for Potential Vehicles
3 $\mathcal{V}^x \leftarrow \emptyset$ // Initialization of the Set of Dispatched Vehicles
4 $\mathcal{P}^x \leftarrow \emptyset$ // Initialization of the Set of Destinations
5 $\mathcal{C}^x \leftarrow \emptyset$ // Initialization of the Set of Shipped Goods per Vehicle
6 $x_t \leftarrow (\emptyset, \emptyset, \emptyset)$ // Initialization of the Current Decision
7 **while** $(\mathcal{V} \neq \emptyset)$ // As long as Vehicles are Available
8 **do**
9 $c_t^{P^*} \leftarrow -1$ // Maximum Number of Goods
10 $\mathcal{P}^{\text{candidates}} \leftarrow \emptyset$ // Potential Stations for Shipment
11 **for all** $P \in \mathcal{P}$ // Iterating through Stations
12 **do**
13 $c^P_{\text{eligible}} \leftarrow$ *NumberEligibleOrders*(P, S_t, x_t)// Filtering for Potential Orders
14 $c_t^P \leftarrow \min\{$*FreeCapacity*$(P, S_t, x_t), c^P_{\text{eligible}}\}$ // Maximum Shipments at Station P
15 **if** $(c_t^P > c_t^{P^*})$ **then** // New best Candidate
16 $c_t^{P^*} \leftarrow c_t^P$
17 $\mathcal{P}^{\text{candidates}} \leftarrow \{P\}$ // Station P is currently the only Candidate
18 **end**
19 **else if** $(c_t^P == c_t^{P^*})$ **then** // Similar Candidate
20 $\mathcal{P}^{\text{candidates}} \leftarrow \mathcal{P}^{\text{candidates}} \cup \{P\}$ // Station P is added as Candidate
21 **end**
22 **end**
23 **if** $(c_t^{P^*} \geq \theta)$ **then** // Add Dispatch, if Threshold is Satisfied
24 $v^* \leftarrow$ *GetFirstVehicle*$(\mathcal{V})$ // Select a Vehicle
25 $P^* \leftarrow$ *FilterBestStation*$(\mathcal{P}^{\text{candidates}}, S_t, x_t)$ // Select a Station
26 $C^* \leftarrow$ *FilterBestOrders*(P^*, S_t, x_t) // Select Orders
27 $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v^*\}$ // Remove Vehicle
28 // Update Decision Components
29 $\mathcal{V}^x \leftarrow \mathcal{V}^x \cup \{v^*\}$
30 $\mathcal{P}^x \leftarrow \mathcal{P}^x \cup \{P^*\}$
31 $\mathcal{C}^x \leftarrow \mathcal{C}^x \cup \{C^*\}$
32 $x_t \leftarrow (\mathcal{V}^x, \mathcal{P}^x, \mathcal{C}^x)$ // Update Decision
33 **end**
34 **else** // Termination
35 exit **while**
36 **end**
37 **end**
38 **return** x_t.

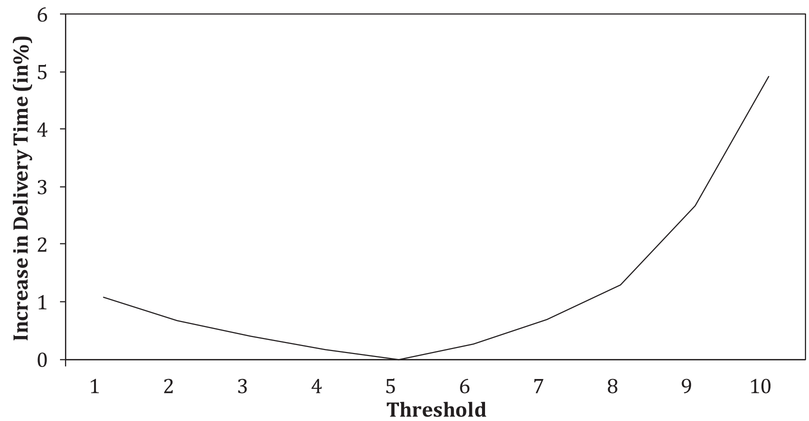

Fig. 11. Tuning Results.

termination criterion was met. It therefore iterates through all stations to find a set of candidate stations $\mathcal{P}^{\text{candidates}}$ for potential shipment. For each station, the algorithm determines the number of eligible orders for shipment c^{P}_{eligible} with function *NumberEligibleOrders*(P, S_t, x_t). This function depends on the station P and the set of orders that can be shipped from the depot in state S_t. Notably, the function also depends on the decision x_t currently considered because after each iteration orders may be already assigned to a dispatch and therefore not available in the next iteration. To determine the maximum number of potential shipments to a station, the algorithm now compares the number of eligible orders with the free capacity of station P. The free capacity is determined by function *FreeCapacity*(P, S_t, x_t). This function again depends on station, state, and decision because another vehicle may be planned to be dispatched to the station already in decision x_t. In line 15, the algorithm now checks if the currently found station allows for more shipments than every other previously found using variable $c^{P^*}_t$. If so, the new station becomes the currently only candidate and the variable $c^{P^*}_t$ is updated. In case that the new station is similar to existing stations (line 19), the station is added to the set of candidates.

Once the algorithm finished iterating through all stations, a new dispatch is determined in case the number of orders to ship $c^{P^*}_t$ is larger than the threshold θ (line 23). Else, the algorithm terminates (line 34). In case the threshold is satisfied, the algorithm selects the first vehicle v^* from the set of available vehicles $\mathcal{V}$ with function *GetFirstVehicle*$(\mathcal{V})$. It further selects the best station for dispatch. For the purpose of readability, the selection is condensed to function *FilterBestStation*$(\mathcal{P}^{\text{candidates}}, S_t, x_t)$. In case $\mathcal{P}^{\text{candidates}}$ contains only one station, the function selects this station. Else, the stations are hierarchically ranked with respect to the free capacity, then the travel time to the station, and finally the index of the station. Because the free capacity also depends on the already planned dispatches in x_t, the current decision x_t is also part of the function. The algorithm also determines the set of orders to dispatch, in case there are more orders available than capacity. To this end, the algorithm draws on function *FilterBestOrders*(P^*, S_t, x_t). This functions selects the orders by means of first-in-first-out. The algorithm then updates the decision x_t by adding v^*, P^*, and $\mathcal{C}^*$ to the corresponding components. It further removes vehicle v^* from the set of available vehicles.

Eventually, the algorithm terminates either when no vehicles are available anymore or when for the first time the maximum number of orders that can be shipped is lower than the threshold. It returns the decision x_t.

A.2. Tuning process

In the following, we give an example of the tuning process. For the purpose of illustration, we select the instance setting with 500 expected orders, 6 vehicles, normal neighborhood, and expected pickup times of 60 min. During the tuning process, for every threshold between 1 and 10, we calculate the average delivery times over the 1000 training instances. For this instance setting, the minimal value is achieved with a threshold of 5. To show the tradeoff depicted in Fig. 3, we normalize the values of the other thresholds to the value achieved with threshold 5. The results are depicted in Fig. 11. The x-axis represents the threshold. The y-axis shows the relative increase in delivery time compared to the best threshold. We observe that the pattern matches the assumptions made Section 4. Very fast dispatches and very late dispatches are both inferior to the “compromise” with threshold 5.

A.3. Impact of neighborhood and travel speed

To analyze the impact of neighborhood and travel speed to average delivery time and threshold, we calculate the average values for neighborhoods *none, normal*, and *large* as well as speeds *conservative, normal*, and *optimistic*. The results are depicted in Table A1. As expected, we observe increasing delivery times with decreasing neighborhood size and decreasing vehicle speed. Particularly, the vehicle speed leads to a substantial difference in delivery time. We also observe changes in the thresholds for varying neighborhoods and speeds. With increasing neighborhood size, the potential for consolidation is higher. Thus, the threshold is larger utilizing this potential. The speed impacts the time required for a dispatch. If a dispatch is more time-consuming, consolidation becomes more important. As a result, we observe a larger threshold.

A.4. Individual data

In this section, we present the travel time matrix in Table A2. We further present the individual results in Tables A3–A5. The tables contain the average delivery times, the maximal delivery times, and the improvement of policy π^{PFA} to the benchmark policies. The tables further contain the results for policy π^{time}. This policy dispatches a vehicle after waiting maximally a certain amount of time at the depot. If a vehicle can be dispatched fully loaded, it is sent earlier. The amount of time is again determined

Table A1
Average Delivery Time and Thresholds.

	Neighborhood			Speed		
	none	normal	large	conservative	normal	optimistic
Average Delivery Time	128.3	113.3	105.3	157.9	115.8	73.2
Average Threshold	4.1	6.1	6.4	6.4	5.6	4.5

Table A2
Travel Times from Google Maps between Stations and/or the Depot (Symmetrical).

Start Location	Pickup Station P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
Depot	1	7	8	8	13	12	14	13	14	15	14	14
P_1	0	5	4	6	9	12	10	13	13	13	13	12
P_2		0	3	6	8	11	8	11	11	12	12	12
P_3			0	7	5	8	6	9	8	9	12	12
P_4				0	11	14	11	12	10	11	10	10
P_5					0	3	2	5	12	5	8	16
P_6						0	3	4	11	4	7	15
P_7							0	5	12	6	8	16
P_8								0	10	3	6	13
P_9									0	7	7	9
P_{10}										0	3	11
P_{11}											0	8
P_{12}												0

Table A3
Individual Results: No Neighborhood.

		Average Delivery Time				Max Delivery Time				Improvement			
Orders	Speed	π^{PFA}	π^{fast}	π^{late}	π^{time}	π^{PFA}	π^{fast}	π^{late}	π^{time}	to π^{fast}	to π^{late}	to π^{time}	Threshold
100	Optimistic	48.4	48.4	244.1	48.4	171.7	171.7	525.5	171.7	0.0	80.2	0.0	1
200	Optimistic	59.6	59.6	150.5	59.6	204.7	204.7	437.3	204.7	0.0	60.4	0.0	1
300	Optimistic	63.6	63.8	114.5	63.8	196.6	197.0	324.4	197.0	0.3	44.4	0.3	2
400	Optimistic	65.8	66.0	95.8	66.0	189.2	188.1	263.3	188.1	0.3	31.3	0.3	2
500	Optimistic	67.4	67.6	84.8	67.6	184.7	183.1	225.6	183.1	0.4	20.5	0.4	3
600	Optimistic	69.7	70.0	79.1	70.0	181.2	180.3	201.2	180.3	0.4	11.9	0.4	3
700	Optimistic	78.6	79.1	84.0	79.1	187.4	188.4	194.3	188.4	0.6	6.4	0.6	3
800	Optimistic	99.9	100.9	104.8	100.9	236.2	237.2	242.7	237.2	1.0	4.7	1.0	5
900	Optimistic	127.2	128.8	131.3	128.3	295.9	297.3	300.9	297.2	1.2	3.1	0.8	5
1000	Optimistic	157.0	158.9	159.7	157.4	355.6	358.2	360.0	356.6	1.2	1.7	0.3	5
100	Normal	64.0	64.0	252.4	64.0	229.1	229.1	539.8	229.1	0.0	74.6	0.0	1
200	Normal	75.9	76.0	157.7	76.0	244.4	242.3	443.8	242.3	0.1	51.9	0.1	2
300	Normal	80.2	80.6	122.0	80.6	232.4	233.1	331.5	233.1	0.5	34.3	0.5	2
400	Normal	82.8	83.3	104.2	83.3	225.0	225.3	272.1	225.3	0.6	20.5	0.6	3
500	Normal	87.4	87.9	97.6	87.9	224.2	223.6	239.4	223.6	0.6	10.4	0.6	3
600	Normal	105.7	106.7	112.3	106.7	240.5	240.9	245.0	240.9	0.9	5.9	0.9	3
700	Normal	138.9	140.9	144.1	140.3	302.7	304.5	307.3	303.4	1.4	3.6	1.0	5
800	Normal	176.7	179.5	179.7	177.3	375.9	379.5	379.4	376.8	1.6	1.7	0.4	5
900	Normal	216.1	219.9	217.1	216.7	455.9	460.3	457.9	456.8	1.7	0.5	0.3	7
1000	Normal	255.9	262.0	255.9	257.2	538.6	544.6	538.6	539.4	2.3	0.0	0.5	10
100	Conservative	78.5	78.5	260.3	78.5	267.3	267.3	554.1	267.3	0.0	69.8	0.0	1
200	Conservative	91.0	91.3	164.6	91.3	278.7	277.4	450.2	277.4	0.3	44.7	0.3	2
300	Conservative	95.7	96.2	129.7	96.2	266.0	263.8	339.3	263.8	0.5	26.2	0.5	3
400	Conservative	100.3	101.1	115.2	101.1	259.7	259.5	284.6	259.5	0.8	12.9	0.8	3
500	Conservative	119.4	120.8	128.0	120.8	275.6	276.3	282.6	276.3	1.1	6.7	1.1	4
600	Conservative	157.7	160.2	163.2	159.1	338.6	340.4	342.4	339.2	1.6	3.4	0.9	5
700	Conservative	202.8	206.4	205.0	203.2	428.2	431.9	429.4	428.0	1.8	1.1	0.2	5
800	Conservative	248.2	254.9	248.5	249.6	521.9	529.4	522.5	524.1	2.6	0.1	0.6	9
900	Conservative	293.9	304.7	293.9	296.7	620.7	632.3	620.7	624.1	3.5	0.0	0.9	10
1000	Conservative	340.7	355.4	340.7	344.7	721.1	736.5	721.1	725.4	4.1	0.0	1.1	10

by sample average approximation using values of 5, 10, 15, 20, 25 and 30 min. The individual results of π^{time} are never better but often worse than π^{PFA}.

Furthermore, Tables A6 and A7 depict the parcel migration for normal and large neighborhoods for the instance setting with 600 orders, 6 vehicles, expected pickup time of 60 min. The value in row i, column j indicates the average number of parcels per day with preferred station i sent to station j. For instances without neighborhood, the matrix is empty because parcels are always delivered to the preferred station.

Table A4
Individual Results: Normal Neighborhood.

		Average Delivery Time				Max Delivery Time				Improvement			
Orders	Speed	π^{PFA}	π^{fast}	π^{late}	π^{time}	π^{PFA}	π^{fast}	π^{late}	π^{time}	to π^{fast}	to π^{late}	to π^{time}	Threshold
100	Optimistic	40.8	40.8	158.8	40.8	111.9	111.9	488.9	111.9	0.0	74.3	0.0	1
200	Optimistic	47.3	47.3	101.2	47.3	155.0	155.0	378.6	155.0	0.0	53.3	0.0	1
300	Optimistic	49.8	50.0	79.7	50.0	158.0	157.9	287.3	157.9	0.4	37.6	0.4	2
400	Optimistic	51.5	51.7	69.1	51.7	156.1	156.9	233.8	156.9	0.4	25.5	0.4	2
500	Optimistic	53.1	53.3	62.7	53.3	156.2	155.1	201.2	155.1	0.4	15.3	0.4	4
600	Optimistic	55.6	56.0	59.8	56.0	157.2	158.0	179.5	158.0	0.6	7.1	0.6	4
700	Optimistic	65.1	66.0	65.6	65.6	175.6	177.1	177.8	177.5	1.3	0.8	0.7	7
800	Optimistic	87.6	90.8	87.6	88.0	231.3	237.0	231.6	232.4	3.6	0.1	0.5	9
900	Optimistic	115.6	120.9	115.6	116.3	291.7	297.2	291.7	291.4	4.3	0.0	0.6	10
1000	Optimistic	145.5	153.0	145.5	146.6	349.5	356.9	349.5	349.6	4.9	0.0	0.7	10
100	Normal	52.3	52.3	164.8	52.3	157.1	157.1	494.6	157.1	0.0	68.3	0.0	1
200	Normal	59.1	59.2	107.3	59.2	189.8	187.7	383.8	187.7	0.2	44.9	0.2	2
300	Normal	62.2	62.5	86.1	62.5	190.2	189.4	292.8	189.4	0.4	27.7	0.4	2
400	Normal	64.8	65.2	76.3	65.2	189.5	189.5	240.3	189.5	0.6	15.0	0.6	3
500	Normal	70.0	70.8	73.5	70.8	196.9	197.2	213.2	197.2	1.1	4.7	1.1	5
600	Normal	90.6	93.4	90.8	90.8	226.9	230.1	227.5	227.5	3.1	0.2	0.3	9
700	Normal	125.1	132.0	125.1	126.4	291.5	300.2	291.5	292.7	5.2	0.0	1.1	10
800	Normal	163.1	173.7	163.1	165.2	364.2	376.5	364.2	366.7	6.1	0.0	1.3	10
900	Normal	202.9	215.9	202.9	204.9	442.5	457.0	442.5	444.8	6.0	0.0	1.0	10
1000	Normal	243.5	259.2	243.5	245.4	521.3	539.3	521.3	523.0	6.0	0.0	0.8	10
100	Conservative	62.8	62.8	170.3	62.8	193.5	193.5	500.3	193.5	0.0	63.1	0.0	1
200	Conservative	69.9	70.2	112.9	70.2	218.9	216.1	389.1	216.1	0.4	38.1	0.4	2
300	Conservative	73.7	74.3	92.3	74.3	219.2	218.0	298.8	218.0	0.8	20.1	0.8	3
400	Conservative	79.1	79.9	85.1	79.9	224.6	223.9	250.6	223.9	1.0	7.1	1.0	5
500	Conservative	101.4	105.1	101.6	101.8	261.2	265.1	260.5	261.1	3.5	0.3	0.5	9
600	Conservative	141.2	150.4	141.2	143.4	326.9	336.0	326.9	328.8	6.1	0.0	1.5	10
700	Conservative	185.5	199.6	185.5	189.3	412.3	428.3	412.3	416.3	7.1	0.0	2.0	10
800	Conservative	231.7	250.2	231.7	235.8	504.5	524.5	504.5	508.7	7.4	0.0	1.7	10
900	Conservative	279.4	300.6	279.4	282.9	601.6	624.1	601.6	604.6	7.1	0.0	1.2	10
1000	Conservative	327.6	351.9	327.6	330.9	699.6	724.6	699.6	702.4	6.9	0.0	1.0	10

Table A5
Individual Results: Large Neighborhood.

		Average Delivery Time				Max Delivery Time				Improvement			
Orders	Speed	π^{PFA}	π^{fast}	π^{late}	π^{time}	π^{PFA}	π^{fast}	π^{late}	π^{time}	to π^{fast}	to π^{late}	to π^{time}	Threshold
100	Optimistic	36.2	36.2	125.2	36.2	84.0	84.0	482.4	84.0	0.0	71.1	0.0	1
200	Optimistic	42.0	42.0	82.7	42.0	133.0	133.0	366.2	133.0	0.0	49.2	0.0	1
300	Optimistic	44.1	44.2	66.9	44.2	143.0	141.9	279.1	141.9	0.3	34.1	0.3	2
400	Optimistic	45.7	45.8	58.9	45.8	145.6	143.0	228.4	143.0	0.3	22.4	0.3	3
500	Optimistic	47.3	47.5	54.2	47.5	147.4	146.0	196.6	146.0	0.5	12.8	0.5	4
600	Optimistic	49.9	50.3	52.2	50.3	153.8	151.7	175.8	151.7	0.9	4.5	0.9	6
700	Optimistic	58.2	60.1	58.3	58.8	175.7	176.7	176.7	177.0	3.1	0.0	0.9	9
800	Optimistic	79.7	84.6	79.7	81.2	232.4	239.5	232.4	234.4	5.8	0.0	1.8	10
900	Optimistic	107.5	114.7	107.5	109.3	290.7	298.3	290.7	292.1	6.3	0.0	1.7	10
1000	Optimistic	137.7	147.3	137.7	139.4	350.4	360.7	350.4	350.7	6.5	0.0	1.2	10
100	Normal	45.9	45.9	131.2	45.9	120.7	120.7	488.6	120.7	0.0	65.0	0.0	1
200	Normal	52.3	52.3	88.8	52.3	173.4	165.7	372.3	165.7	0.1	41.1	0.1	2
300	Normal	55.0	55.2	73.1	55.2	173.6	173.5	285.3	173.5	0.4	24.7	0.4	2
400	Normal	57.5	57.8	65.6	57.8	177.9	176.4	235.4	176.4	0.4	12.3	0.4	4
500	Normal	62.6	63.4	64.0	63.4	193.2	189.8	208.4	189.8	1.2	2.2	1.2	7
600	Normal	81.4	86.0	81.4	84.2	228.8	233.9	228.8	232.2	5.4	0.0	3.3	10
700	Normal	115.9	124.7	115.9	120.1	293.7	302.1	293.7	297.9	7.0	0.0	3.5	10
800	Normal	154.5	166.9	154.5	158.6	364.8	378.8	364.8	369.0	7.4	0.0	2.6	10
900	Normal	194.9	209.9	194.9	198.2	442.5	459.5	442.5	446.0	7.2	0.0	1.7	10
1000	Normal	235.8	253.5	235.8	238.4	521.9	540.4	521.9	522.7	7.0	0.0	1.1	10
100	Conservative	55.1	55.1	136.7	55.1	154.7	154.7	494.8	154.7	0.0	59.7	0.0	1
200	Conservative	61.9	62.0	94.4	62.0	199.2	195.0	378.6	195.0	0.2	34.5	0.2	2
300	Conservative	65.6	65.9	79.1	65.9	205.5	202.9	292.2	202.9	0.5	17.1	0.5	3
400	Conservative	70.5	71.4	73.7	71.4	216.2	216.1	246.0	216.1	1.4	4.4	1.4	5
500	Conservative	90.4	96.2	90.4	94.6	263.1	270.2	263.1	269.0	6.0	0.0	4.4	10
600	Conservative	129.5	140.4	129.5	136.3	332.5	344.3	332.5	339.2	7.8	0.0	5.0	10
700	Conservative	174.0	189.9	174.0	181.2	416.2	434.5	416.2	422.3	8.4	0.0	4.0	10
800	Conservative	221.0	240.8	221.0	227.4	508.2	528.8	508.2	512.5	8.2	0.0	2.8	10
900	Conservative	269.2	291.8	269.2	274.3	602.8	626.2	602.8	606.2	7.8	0.0	1.9	10
1000	Conservative	318.1	343.2	318.1	321.8	700.9	726.6	700.9	702.0	7.3	0.0	1.2	10

Table A6
Migration Matrix: Large Neighborhood.

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	6.4	13.4	-	-	-	-	-	-	-	-	-
2	15.8	-	12.8	-	-	-	-	-	-	-	-	-
3	10.6	5.5	-	-	10.5	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	8.4	-	-	6.9	4.5	5.3	-	5.7	-	-
6	-	-	-	-	9.3	-	4.3	5.3	-	7.2	-	-
7	-	-	-	-	11.3	10.9	-	5.5	-	-	-	-
8	-	-	-	-	8.6	8.1	3.4	-	-	7.1	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	8.3	7.6	-	4.5	-	-	3.1	-
11	-	-	-	-	-	-	-	-	-	15.5	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-

Table A7
Migration Matrix: Normal Neighborhood.

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	8.0	-	-	-	-	-	-	-	-	-
3	-	21.1	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	14.2	5.9	-	-	-	-	-
6	-	-	-	-	11.8	-	5.0	-	-	-	-	-
7	-	-	-	-	11.1	13.3	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	14.4	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	8.8	-	-	6.3	-
11	-	-	-	-	-	-	-	-	-	15.0	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 52 (4), 965–981. doi:10.1287/trsc.2017.0791.

Arbanas, B., Ivanovic, A., Car, M., Haus, T., Orsag, M., Petrovic, T., Bogdan, S., 2016. Aerial-ground robotic system for autonomous delivery tasks. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 5463–5468. doi:10.1109/ICRA.2016.7487759.

Arbanas, B., Ivanovic, A., Car, M., Orsag, M., Petrovic, T., Bogdan, S., 2018. Decentralized planning and control for UAV–UGV cooperative teams. Auton. Robots doi:10.1007/s10514-018-9712-y.

Archetti, C., Feillet, D., Speranza, M.G., 2015. Complexity of routing problems with release dates. Eur. J. Oper. Res. 247 (3), 797–803.

Azi, N., Gendreau, M., Potvin, J.-Y., 2012. A dynamic vehicle routing problem with multiple delivery routes. Ann. Oper. Res. 199 (1), 103–112.

Basilico, N., Chung, T.H., Carpin, S., 2016. Distributed Online Patrolling with Multi-agent Teams of Sentinels and Searchers. Springer Japan, Tokyo, pp. 3–16.

Beirigo, B., Schulte, F., Negenborn, R., 2018. Dual-mode vehicle routing in mixed autonomous and non-autonomous zone networks. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1325–1330.

Bent, R.W., Van Hentenryck, P., 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Oper Res 52 (6), 977–987.

Bernau, V., Hampel, L., Wischmeyer, N., 2016. Her mit den Päckchen, aber bitte schnell und kostenlos. Süddeutsche Zeitung. URL http://www:sueddeutsche:de/wirtschaft/liefergesellschaft-alles-immer-bitte-sofort-1:3154867 [Online; accessed 20-September-2016].

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2019. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Comput. Oper. Res. 106, 260–279. doi:10.1016/j.cor.2018.06.004.

Bullo, F., Frazzoli, E., Pavone, M., Savla, K., Smith, S.L., 2011. Dynamic vehicle routing for robotic systems. Proc. IEEE 99 (9), 1482–1504.

Carlsson, J.G., Song, S., 2017. Coordinated logistics with a truck and a drone. Manage. Sci. 64 (9), 4052–4069.

Carrabs, F., Cerrone, C., Cerulli, R., Gaudioso, M., 2017. A novel discretization scheme for the close enough traveling salesman problem. Comput. Oper. Res. 78, 163–171. doi:10.1016/j.cor.2016.09.003.

Cheng, C., Adulyasak, Y., Rousseau, L.-M., 2018. Formulations and exact algorithms for drone routing problem. Working Paper.

Chong, C., 2017. Drone delivery to start on NUS campus by early next year. Bus. Times. URL http://www:businesstimes:com:sg/technology/dronedelivery-to-start-on-nus-campus-by-early-next-year [Online; accessed 17-October-2017].

Coelho, L.C., Cordeau, J.-F., Laporte, G., 2014. Heuristics for dynamic and stochastic inventory-routing. Comput. Oper. Res. 52, 55–67. doi:10.1016/j.cor.2014.07.001.

Condon, S., 2019. Fedex rolls out prototype of autonomous sameday bot. ZDNet. URL https://www:zdnet:com/article/fedex-rolls-out-prototype-ofautonomous-sameday-bot/ [Online; accessed 25-March-2019].

Crainic, T.G., Montreuil, B., 2016. Physical internet enabled hyperconnected city logistics. Transp. Res. Procedia 12, 383–398. doi:10.1016/j.trpro.2016.02.074. Tenth International Conference on City Logistics 17–19 June 2015, Tenerife, Spain.

Dayarian, I., Savelsbergh, M., Clarke, J.-P., 2018. Same-day delivery with drone resupply. Working Paper.

DHL, 2016. Successful trial integration of DHL parcelcopter into logistics Chain. URL http://www.dhl.com/en/press/releases/releases_2016/all/parcel_ecommerce/successful_trial_integration_dhl_parcelcopter_logistics_chain.html [Online; accessed 02-July-2018].

Ha, Q.M., Deville, Y., Pham, Q.D., HÃ, M.H., 2018. On the min-cost traveling salesman problem with drone. Transp. Res. Part C 86, 597–621. doi:10.1016/j.trc.2017.11.015.

Haas, I., Friedrich, B., 2017. Developing a micro-simulation tool for autonomous connected vehicle platoons used in city logistics. Transp. Res. Procedia 27, 1203–1210. doi:10.1016/j.trpro.2017.12.084. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4–6 September 2017, Budapest, Hungary.

Ham, A.M., 2018. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. Transp. Res. Part C 91, 1–14. doi:10.1016/j.trc.2018.03.025.

van Heeswijk, W.J.A., Mes, M.R.K., Schutten, J.M.J., 2019. The delivery dispatching problem with time windows for urban consolidation centers. Transp. Sci. 53 (1), 203–221. doi:10.1287/trsc.2017.0773.

Heutger, M., Kückelhaus, M., 2016. Self driving vehicles in logistics. a DHL perspective on implications and use cases for the logistics industry. DHL trend research. https://www.logistics.dhl/content/dam/dhl/global/core/documents/pdf/glo-logistics-insights-selfdrivingvehicles.pdf.

Klapp, M.A., Erera, A.L., Toriello, A., 2018a. The dynamic dispatch waves problem for same-day delivery. Eur. J. Oper. Res. 271 (2), 519–534. doi:10.1016/j.ejor.2018.05.032.

Klapp, M.A., Erera, A.L., Toriello, A., 2018b. The one-dimensional dynamic dispatch waves problem. Transp. Sci. 52 (2), 402–415. doi:10.1287/trsc.2016.0682.

Kolodny, L., 2017. Teleretail built a delivery robot to make on-demand logistics easy for small businesses. Techcrunch. URL https://techcrunch:com/2017/05/15/teleretail-built-a-delivery-robot-to-make-on-demandlogistics-easy-for-small-businesses/ [Online; accessed 02-July-2018].

Lowe, J., Khan, A.A., Bhatale, B., 2014. Same-day delivery: surviving and thriving in a world where instant gratification rules. Whitepaper 20. Cognizant.

Lunden, I., 2017. Amazon launches the hub, parcel delivery lockers for apartment buildings. Techcrunch. URL https://techcrunch:com/2017/07/27/amazon-launchesthe-hub-parcel-delivery-lockers-for-apartment-buildings [Online; accessed 02-July-2018].

McFarland, M., 2017. Dhl to test self-driving delivery trucks in 2018. CNN Bus.. https://money.cnn.com/2017/10/11/technology/future/dhl-autonomous-delivery-truck/index.html [Online; accessed 25-March-2019].

Mitchell, O., 2019. Amazon keeps on truckin' with autonomous vehicle investments. The Robot Report. https://www:therobotreport:com/amazon-autonomousvehicle-invest/ [Online; accessed 25-March-2019].
Murphy, M., 2017. Massive robot-powered drones may solve everything wrong with package delivery. QZ. https://qz:com/1163960/delivery-drone-startupelroy-air-announces-4-6-million-in-seed-funding-to-take-tothe-skies/ [Online; accessed 02-July-2018].
Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp. Res. Part C 54, 86–109. doi:10.1016/j.trc.2015.03.005.
NHTSA, 2017. Automated driving systems: a vision for safety. https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf.
Nothdurft, T., Hecker, P., Ohl, S., Saust, F., Maurer, M., Reschka, A., Böhmer, J.R., 2011. Stadtpilot: First fully autonomous test drives in urban traffic. In: Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. IEEE, pp. 919–924.
Persiel, S., 2017. DHL und Deutsche Post in Braunschweig - Paketzentrum von DHL. https://www.paketda.de/dhl-deutsche-post-braunschweig.html.
Powell, W., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd John Wiley and Sons, Hoboken, NJ, USA.
Ram, A., 2015. UK Retailers face high cost of online deliveries. Financ. Times. http://www:ft:com/cms/s/0/516aa75a-a04c-11e5-beba-5e33e2b79e46:html#axzz4CPxX2fjJ [Online; accessed 23-June-2016].
Reyes, D., Savelsbergh, M., Toriello, A., 2017. Vehicle routing with roaming delivery locations. Transp. Res. Part C 80, 71–91. doi:10.1016/j.trc.2017.04.003.
Rivera, A.E.P., Mes, M.R.K., 2017. Anticipatory freight selection in intermodal long-haul round-trips. Transp. Res. Part E 105, 176–194. doi:10.1016/j.tre.2016.09.002.
SAE International, 2016. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. 10.4271/J3016_201609
Savelsbergh, M., Van Woensel, T., 2016. 50Th anniversary invited article—city logistics: challenges and opportunities. Transp. Sci. 50 (2), 579–590.
Scherr, Y.O., Neumann-Saavedra, B.A., Hewitt, M., Mattfeld, D.C., 2018. Service network design for same day delivery with mixed autonomous fleets. Transp. Res. Procedia 30, 23–32. doi:10.1016/j.trpro.2018.09.004.
Scherr, Y.O., Saavedra, B.A.N., Hewitt, M., Mattfeld, D.C., 2019. Service network design with mixed autonomous fleets. Transp. Res. Part E 124, 40–55.
Stevenson, R., Black, T., 2018. Amazon spurs delivery startups with shot at $300,000 profit. Bloomberg. https://www:bloomberg:com/news/articles/2018-06-28/amazon-enlists-delivery-startups-to-help-expand-shipmentvolume [Online; accessed 29-June-2018].
Tipping, A., Kauschke, P., 2016. Shifting patterns, the future of the logistics industry. PriceWaterhouseCoopers, Phoenix.
Ulmer, M., 2017a. Delivery deadlines in same-day delivery. Logist. Res. 10 (3), 1–15.
Ulmer, M.W., 2017b. Dynamic pricing and routing for same-day delivery. Submitted
Ulmer, M.W., Thomas, B.W., 2018. Same-day delivery with heterogeneous fleets of drones and vehicles. Networks 72 (4), 475–505.
Ulmer, M.W., Thomas, B.W., Campbell, A.M., Woyak, N., 2017. The restaurant meal delivery problem: dynamic pick-up and delivery with deadlines and random ready times. Submitted
Ulmer, M.W., Thomas, B.W., Mattfeld, D.C., 2018. Preemptive depot returns for dynamic same-day delivery. EURO J. Transp. Logist. doi:10.1007/s13676-018-0124-0.
Vakulenko, Y., Hellström, D., Hjort, K., 2018. What'S in the parcel locker? exploring customer value in e-commerce last mile delivery. J. Bus. Res. 88, 421–427. doi:10.1016/j.jbusres.2017.11.033.
Verlinde, S., Rojas, C., Buldeo Rai, H., Kin, B., Macharis, C., 2018. E-consumers and their perception of automated parcel stations. In: City Logistics 3: Towards Sustainable and Liveable Cities, pp. 147–160.
Voccia, S.A., Campbell, A.M., Thomas, B.W., 2019. The same-day delivery problem for online purchases. Transp. Sci. 53 (1), 167–184. doi:10.1287/trsc.2016.0732.
Yuen, K.F., Wang, X., Ng, L.T.W., Wong, Y.D., 2018. An investigation of customers' intention to use self-collection services for last-mile delivery. Transp. Policy 66, 1–8. doi:10.1016/j.tranpol.2018.03.001.
Zhang, S., Ohlmann, J.W., Thomas, B.W., 2018. Dynamic orienteering on a network of queues. Transp. Sci. 52 (3), 691–706. doi:10.1287/trsc.2017.0761.
Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. Eur. J. Oper. Res. 265 (2), 765–778. doi:10.1016/j.ejor.2017.08.011.