Effective Management of the Curb Space Allocation in Urban Transportation System

by

Meigui Yu

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science (Data Science) in the University of Michigan-Dearborn 2019

Master's Thesis Committee:

Assistant Professor Armagan Bayram, Chair
Professor Brahim Medjahed
Assistant Professor Xi Chen

Acknowledgements

I want to take a moment and thank those who have helped me through the completion of my thesis.

Firstly, I express my gratefulness to my Professor Dr. Armagan Bayram of the College of Engineering and Computer Science at the University of Michigan – Dearborn. Her door is always open whenever I met difficulties while doing my thesis. She gave me guidance and support through every step of the project. Her advice, dedication, and expertise led to the success of this work.

I would also like to thank Dr. Xi Chen, who brought me to the Operations Research world. I also want to thank all the Professors or Lecturers who helped me throughout my master's program.

Finally, I would like to thank my parents for believing me, encouraging me throughout my master’s study.

Table of Contents

List of Tables

List of Figures

Abstract

Curb space management and traffic flow are two essential elements of the transportation system that interact with each other and affect the overall system performance. With the growth of new mobility operators and goods delivery, the demand for access to the curb space is increasing rapidly. Thus, the traditional use of curb space for parking only is challenged, and it becomes essential to manage the curb space effectively. Our study investigates the allocation of curb space for various uses (i.e., parking, pick-up/drop-off, and loading/unloading) so that the overall transportation system performance can be enhanced. We simulate the transportation system and analyze the interactions between traffic flow and curb space usage by investigating the impact of the allocated curb spaces for different uses on traffic congestion. We build an optimization model to determine dynamic curb space allocation decisions that ensure smooth traffic flow. Our objective is to maximize the cities' profit of curb space allocation decisions and minimizing the traffic delay. We further evaluate the value of dynamic curb space allocation policies over the fixed allocation policies and find that the dynamic policy can result in improvements in traffic delay.

Chapter 1: Introduction

1.1 Introduction and Motivation

Curb spaces have evolved so rapidly with the arrival of new mobility services and increased needs for goods delivery. Currently, curb spaces are not only used for parking but also used for pick-up/drop-off zone of ride-sharing services, bike share or scooter parking racks, delivery zones for online shopping companies, etc. Although due to the growing demand in ride-sharing services (i.e., Uber, Lyft, Chariot), the need for curbside parking has decreased, the need for other users of the curb spaces (i.e., pick-up, wait, drop-off) has risen. Further, the increasing demand in online shopping, which was supposed to reduce traffic jams by reducing individual trips to the stores (Hsiao, 2009), has resulted in an explosion in the trips made by delivery trucks (i.e., UPS, FedEx) and their use of the curb spaces. Hence, the concerns about traffic congestion have arisen, and it has found that cruising for parking spaces solely contribute to around 30% of the total traffic congestion in business areas during the rush hour (Shoup, 2006).

Similarly, it has found that the overall traffic delay from pick-up and delivery activities ranks third among all congestion-related events, indicating the magnitude of this traffic delay is more severe than the expectation (Han, et al., 2005). Moreover, the illegal parking in cities (i.e., blockages of bus lanes, bicycle facilities, and crosswalks by double-parked vehicles) has escalated, and it is reported that the delivery vans of companies such as FedEx and UPS have received millions of dollars of parking tickets due to the illegal parking in 2016 (Figliozzi & Tipagornwong, 2016). Thus, the inefficient use of the curb spaces can cause a potential safety hazard for people, traffic delays, and loss of city profits (Zalewski, et al., 2012), and it is very crucial for cities to utilize the curb spaces efficiently.

To make the transportation system more reliable, cities across North America are shifting curb spaces from solely parking lanes to flexible zones, where the use of the curb zones can vary dynamically during the day. For example, these flexible zones could shrink, grow, or be assigned to other purposes by considering varying demands for different usages. Some cities have adopted policies that define the use of curb spaces. For example, the city of Seattle uses flexible zones and assigns the curb spaces to different usages according to some predefined priorities. However, no standard methodology exists for cities to assess the potential for dynamic curb space allocation and the subsequent impacts of those changes. Also, despite the importance of curb space planning, the consideration of dynamic use of the curb spaces during the day limits its large scale adoption. In this thesis, we study the dynamic allocation of the curb spaces by the cities for different uses. We consider three possible usages of the curb spaces (i.e., parking, pick-up/drop-off, and loading/unloading). We address the benefits of dynamic curb space allocation by considering the interaction between the traffic and the curb spaces, and we develop answers to the following operational questions:

• Given the number of existing on-street parking spots inside a transportation network, what is the optimal dynamic curb space allocation policy which considers the flexible assignment of curb spaces for different uses (i.e., parking, pick-up/drop-off, and loading/unloading)?

• What is the value of dynamic curb space allocation policy in terms of vehicle traffic delay and vehicle drove distance?

To address these questions, we first build a macroscopic simulation model to capture the interaction between the transportation system and the curb space allocation policy. The macroscopic simulation model allows us to analyze several curb space allocation scenarios for different uses and observe the impacts of the model parameters (i.e., vehicle-free speed, traffic demand, cruising time, etc.) on the overall traffic flow. Second, we build an integer programming model by using the outputs of the macroscopic simulation model to determine the optimal dynamic curb space allocation policy among different uses of the curb space (i.e., parking, pick-up/drop-off, and loading/unloading). The objectives of the integer programming model are: (i) to maximize the cities' profit from parking, and (ii) to minimize the traffic delay. Due to the interactions in the macroscopic simulation model, we obtain a non-linear objective function, which makes the analyses further complicated. Hence, we propose two algorithms to solve the dynamic curb space allocation model efficiently. Finally, in our study, we consider both the fixed curb space allocation policy, in which the use of curb space is fixed over time and the dynamic allocation policy in which the use of curb spaces can vary over time. We further compare both policies to analyze the value of the dynamic curb space allocation implementation. We show that the flexible allocation of the curb spaces can yield a decrease in the traffic delay within the network.

The remainder of this thesis is structured as follows: In Chapter 1.2, we review the relevant literature. In Chapter 2, we describe our macroscopic simulation model and perform sensitivity analysis to validate the simulation model. In Chapter 3, we present the optimization model and perform numerical analysis to show our results from both macroscopic simulation and optimization models by using the proposed algorithms. Finally, our conclusions are outlined in Chapter 4.

1.2 Literature Review

In recent years, with the rapid growth of mobility services, the need for the effective use of curb space has attracted several researchers. Some of them study how cities manage their curb spaces and the existing approaches that are used for curb space management (Chang, 2009), (Schaller, et al., 2011), (Zalewski, et al., 2012). Some propose new policies to find solutions for mitigating traffic congestion. Among studies that focus on policy development in curb space management (Shoup, 2006) points out that the congestion within a network is mostly caused by parked vehicles and that the parking rate can be adjusted to decrease the traffic demand entering the network and to control the traffic delay better. In another similar study, (Downs, 2004) proposes policies to mitigate the traffic congestion, such as greatly expanding road capacity, using intelligent transportation system devices to speed traffic flow, and greatly expanding public transit capacity. These policies would be helpful if cities can afford the enormous cost and time for the changes, for example, new urban planning to expand road capacity and include high-occupancy vehicle lanes. However, most cities prefer a lower-cost strategy that takes a shorter time to see the effect. Thus, it is more practical and efficient to provide solutions by utilizing the current resources and allocating them efficiently for possible different uses. Researchers have also investigated drivers' parking and cruising behaviors and provided solutions related to parking fees and parking duration to mitigate the congestion and traffic delay (Calthrop & Proost, 2006), (Chang, 2009), (Lee, et al., 2017). However, these studies focus on high-level policies that are not necessarily based on any methodological framework/model.

Another stream of literature that is relevant to our study is on economics and traffic assignment. In this stream, studies investigate the interaction between parking and the traffic system and analyze the equilibrium of curbside parking (Arnott & Rowse, 2009), (Anderson & De Palma, 2004), (Arnott & Inci, 2006). Different from these studies, we consider the dynamics of the traffic system (i.e., time-varying conditions). We further build an optimization model to allocate the cities' curb spaces effectively. The studies related to the curb space management are almost solely about parking use, and few studies investigate other uses, such as pick-up/drop-off and loading/unloading. How cities manage curb spaces for primary uses (e.g., parking, loading/unloading) is studied by (Zalewski, et al., 2012). They propose three models related to curb space management planning, price regulation, and community strategies to help cities in curb space management policies and decision-making processes. Although the effect of the existing curb space management policies and the use of curb space for loading are discussed in the paper, no simulation models or optimization models are presented to further validate the efficiency of the proposed curb space management policies.

Several studies build multi-agent traffic simulation models to investigate the dynamics of the traffic system (Benenson, et al., 2008), (Chen & Cheng, 2010), (Schelenz, et al., 2014). Although multi-agent traffic simulation models allow the inclusion of personal preferences, driver behaviors, etc., they require detailed data for all specific conditions, and thus their results cannot be easily generalized. Also, the integration of the multi-agent simulation models with the optimization models would require more computational effort. More relevant to our study, (Cao & Menendez, 2015) build a macroscopic simulation model that analyzes the interaction between urban parking and the urban traffic systems and shows their effects on urban congestion. In a follow-up study, (Cao, et al., 2017) present a case study of an area within the city of Zurich, Switzerland, using their macroscopic simulation model and analyze the traffic performance measures (i.e., traffic delay, total distance) within the network. Different from them, we consider other uses of the curb space (i.e., pick-up/drop-off and loading/unloading) in addition to parking-only use and investigate the optimal curb space allocation by building an optimization model on the top of the macroscopic simulation model.

Chapter 2: The Simulation Model

This chapter comprises two main parts. First, we formulate our problem into different scenarios and define the system events and transition events that are used to update the number of vehicles in each time slice. Then we build the relationship between system events and transition events for the simulation model. Second, we perform sensitivity analysis under the current curb space management policy to observe how the parameters (i.e., vehicle speed, traffic demand proportion, cruising time, etc.) affect the traffic flow. Then, we show the influence of assigning some curb space for *PD* and *LU* uses to reduce the traffic delay.

2.1 Problem Formulation

In this section, we build on the study of Cao (Cao, et al., 2017) and develop a macroscopic simulation model to investigate the interaction between the transportation system and curb space allocation. Different from Cao (Cao, et al., 2017), we introduce additional system events by introducing new curb space uses (i.e., pick-up/drop-off, loading/unloading). We consider a relatively small urban area where all existing on-street public parking spaces are randomly distributed, and all vehicles drive on one lane and in the same direction. Also, we assume that all existing curb spaces are uniformly distributed, such that the drivers do not have a preference. We use *P*, *PD*, and *LU* to denote the cases of parking, pick-up/drop-off, and loading/unloading, respectively. A trip of a vehicle starts when a vehicle enters the urban network area, and it ends when the vehicle leaves the urban area. We assume that trips are uniformly distributed after the vehicles enter the network.

When a vehicle enters the network the following cases can occur: (i) the vehicle can go through traffic, (ii) the vehicle can search for a parking (*P*) spot, (iii) the vehicle can search for a pick-up/drop-off (*PD*) spot, (iv) the vehicle can search for a loading/unloading (*LU*) spot. We assume that only a proportion of traffic entering the network will look for curb space, the other traffic will go through the network after driving for a certain distance. Also, vehicles that look for a *P/PD/LU* spot may leave the network without accessing any curb space after cruising for a specific time. More specifically, as illustrated in Figure 2.1 and Figure 2.2, we consider three scenarios that can occur after a vehicle enters the network:

• Scenario 1: Vehicles that look for *P/PD/LU* spot enter the network and successfully access a curb space (Figure 2.1).

• Scenario 2: Vehicles that look for *P/PD/LU* spot enter the network and then leave the network after cruising for more than a certain time without accessing a curb space (Figure 2.2).

• Scenario 3: Vehicles that do not look for a curb space enter the network and go through the network (Figure 2.2).

Figure 2.1 Illustration of all states and transition events for scenario 1

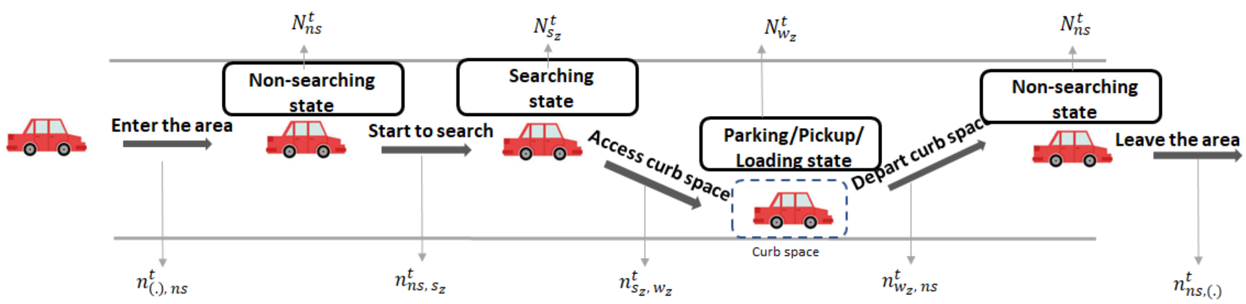
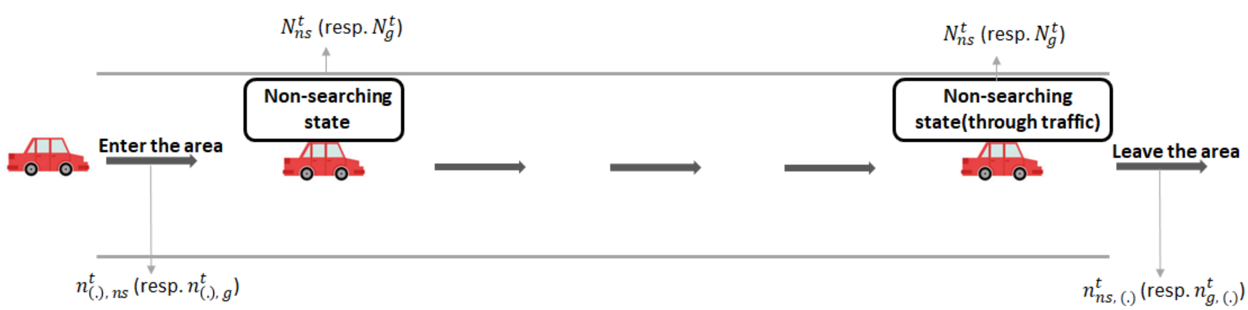


Figure 2.2 Illustration of all states and transition events for scenario 2 & 3



2.1.1 Definition of System Events and Transition Events

We define system events and transition events to describe the above three scenarios. Let $z \in \{P, PD, LU\}$ denote the different types of curb space usage and $\mathcal{J}$ be the set of system events. We use the following system events to simulate the vehicle movement:

1. Non-searching (ns): This state includes vehicles that are not searching for any spot. The vehicles may have either just entered the network or just departed from the curb space.

2. Searching (S_z): The vehicles in this state are cruising to find a curb spot $z \in \{P, PD, LU\}$.

3. Stationary (w_z): This state involves vehicles that have accessed a curb spot $z \in \{P, PD, LU\}$.

4. Going through traffic (g): In this state vehicles do not enter the searching state and go through the network.

During the simulation, we assume that there are $t \in \mathcal{T}$ time periods. In order to capture the changes in the number of vehicles, we define N_j^t to represent the number of vehicles in each system event $j \in \mathcal{J} = \{ns, s_z, w_z, g\}$ in time period t, and we define $n_{j,j'}^t$ to represent the number of vehicles transitioning from system event $j \in \mathcal{J}$ to system state $j' \in \mathcal{J}$ in time period t. All system events and transition events are defined in Table 2.1.

2.1.2 Update the Number of Vehicles in Each System Event

To capture the cumulative change of the number of vehicles in each system event, we build the relationship between system events and transition events. During a given time slice *t* (e.g., 1 min), we assume vehicles are driving at the same speed such that no overtaking is allowed in the network. Vehicle speed does not influence the total number of occupied curb spaces because a curb space always serves the first vehicle that passes by. Thus, we use the equations (2.1) - (2.4) to calculate the number of vehicles in each system event $j \in \mathcal{J}$. We note that $n_{(.),j}^t$ denotes the number of vehicles entering the network and $n_{j,(.)}^t$ denotes the number of vehicles leaving the network. Equations (2.1) - (2.4) define the number of vehicles in the states of non-searching, searching (i.e., for parking, for picking-up/dropping-off, for loading/unloading), stationary(i.e., *P, PD, LU*), and going through traffic, respectively, in time period *t*.

In equation (2.1), the number of "non-searching" vehicles consist of vehicles that enter the urban area and vehicles depart *P/PD/LU*, vehicles that start to search and all vehicles (no matter vehicles have accessed *P/PD/LU* or not) that leave the area.

$$N_{ns}^{t} = N_{ns}^{t-1} + n_{(.),ns}^{t-1} + n_{(.),g}^{t-1} + \sum_{z\in\{P,PD,LU\}} \left[n_{w_z,ns}^{t-1} - n_{ns,s_z}^{t-1}\right] - n_{ns,(.)}^{t-1} - n_{g,(.)}^{t-1} \tag{2.1}$$

In equation (2.2), the number of “searching” vehicles consists of vehicles that start to search for *P/PD/LU* and vehicles that access *P/PD/LU*.

$$N_{s_z}^{t} = N_{s_z}^{t-1} + n_{ns,s_z}^{t-1} - n_{s_z,w_z}^{t-1} \tag{2.2}$$

In equation (2.3), the number of *P/PD/LU* vehicles consists of vehicles that access *P/PD/LU* and vehicles that leave from *P/PD/LU*.

$$N_{w_z}^{t} = N_{w_z}^{t-1} + n_{s_z,w_z}^{t-1} - n_{w_z,ns}^{t-1} \tag{2.3}$$

In equation (2.4), the number of vehicles that go through traffic consists of vehicles that enter the area that go through it and vehicles that leave the area without *P/PD/LU* after driving for a certain distance.

$$N_{g}^{t} = n_{(.),g}^{t-1} - n_{g,(.)}^{t-1} \tag{2.4}$$

Table 2.1 Related states and transition events variables in a time period

Notation	Definition
N^t_{ns}	Number of vehicles in the state “non-searching” at the beginning of time period t.
$N^t_{s_p}$	Number of vehicles in the state “searching for P” at the beginning of time period t.
$N^t_{s_{pd}}$	Number of vehicles in the state “searching for PD” at the beginning of time period t.
$N^t_{s_{lu}}$	Number of vehicles in the state “searching for LU” at the beginning of time period t.
$N^t_{w_p}$	Number of vehicles in stationary state P at the beginning of time period t.
$N^t_{w_{pd}}$	Number of vehicles in stationary state PD at the beginning of time period t.
$N^t_{w_{lu}}$	Number of vehicles in stationary state LU at the beginning of time period t.
N^t_g	Number of vehicles that go through the traffic in the network at the beginning of time period t.
$n^t_{(.),ns}$	Number of vehicles that enter the network area and transition into the non-searching state during time period t.
$n^t_{(.),g}$	Number of vehicles that go through the traffic enter the area and transition to non-searching state during time period t.
$n^t_{ns,ns}$	Number of vehicles that can not enter the searching state after cruising more then a certain time during time period t.
n^t_{ns,s_p}	Number of vehicles that search for parking and transition from “non-searching” state to “searching for P” state during time period t.
$n^t_{ns,s_{pd}}$	Number of vehicles that search for a PD spot and transition from “non-searching” state to “searching for PD” state during time period t.
$n^t_{ns,s_{lu}}$	Number of vehicles that search for a LU spot and transition from “non-searching” state to “searching for LU” state during time period t.
$n^t_{s_p,w_p}$	Number of vehicles that search for a P spot and transition from “searching for P” state to stationary state P during time period t.
$n^t_{s_{pd},w_{pd}}$	Number of vehicles that search for a PD spot and transition from “searching for PD” state to stationary PD state during time period t.
$n^t_{s_{lu},w_{lu}}$	Number of vehicles that search for a LU spot and transition from “searching for LU” spot to stationary LU state during time period t.
$n^t_{w_p,ns}$	Number of vehicles that leave the P spot and transit into the “non-searching” state during time period t.
$n^t_{w_{pd},ns}$	Number of vehicles that leave the “PD” spot and transit into the “non-searching” state during time period t.
$n^t_{w_{lu},ns}$	Number of vehicles that leave the LU spot and transit into “non-searching” state during time period t.
$n^t_{ns,(.)}$	Number of vehicles that leave the area from the non-searching state during time period t.
$n^t_{g,(.)}$	Number of vehicles that leave the area from the state of go through the traffic during time period t.

2.1.3 Update the Number of Vehicles in Each Transition Event

In this section, we describe how we calculate the number of vehicles in each transition event during time slice *t*. First, we describe the assumptions used in the simulation model. Second, we describe the model inputs of the simulation model. Third, new variables are introduced to build up the equations. Finally, we describe how we update the number of vehicles in each transition event.

2.1.3.1 Assumptions

The network is assumed to be small and compact, and all existing parking spots are uniformly distributed such that there is no difference to the drivers' walking distance. Also, the parking rate inside the network should be identical to avoid personal preference.

We assume the network size, the vehicle arrival rates, the vehicle *P/PD/LU* duration distribution, the parking rate, and the number of existing on-street parking spots are known. We only consider vehicles that are heading for on-street *P/PD/LU* in this thesis. Thus, the proportion of vehicles that are heading for private parking, garage parking, and off-street parking are not considered in this thesis.

2.1.3.2 Model input

In Table 2.2, we define the inputs of the simulation model. For example, the length of the urban network, the vehicle arrival rate, the existing number of on-street parking spots, and the parking duration distributions are assumed to be known in the simulation model.

2.1.3.3 Intermediate variables

In order to capture the change in the number of vehicles between transitioning events during a given time period, we introduce some intermediate variables, as shown in Table 2.2.

Table 2.2 Intermediate variable definition

Notation	**Definition**
A_z^t	The number of available P/PD/LU spots at the beginning of time period t
k^t	Average traffic density in time period t
v^t	Average travel speed in time period t
d^t	Maximum drive distance of a vehicle in time period t
s^t	Spacing between vehicles that are searching for P/PD/LU at the beginning of time period t
m^t	Maximum number of vehicles that can pass by the same spot on the network during time period t
d_r^t	Remainder of the division $\frac{d^t}{s^t}$ when $d^t > s^t$

The number of curb spots available of type z in period t (i.e., A_z^t) equals the number of curb spots of type z minus the number of curb spots that are occupied in type z in period t. We define this relation by using Equation (2.5):

$$A_z^t = A_z - N_{w_z}^t \tag{2.5}$$

where $A_z^t \leq A_z$.

Let L be the length of the traffic network. In Equation (2.6), we define the average traffic density in period t (k^t) as the division of the total number of vehicles on the road at the beginning of time period t by the length of the network.

$$k^t = \frac{N_{ns}^t + N_g^t + \sum_{z \in \{P,PD,LU\}} N_{s_z}^t}{L} \tag{2.6}$$

In Equation (2.7), v^t denotes the average vehicle speed during time period t, and we calculate it based on a triangular fundamental diagram (FD) (Daganzo & Newell, 1995). To this end, we use k_c and k_j to denote the critical traffic density and the jam traffic density, respectively. We define Q_{max} as the maximum traffic flow rate that can be adopted on the network. We consider that congestion occurs if the traffic density for a given period is greater than the critical traffic density.

To calculate the average vehicle speed, we compare the current traffic density with the jam traffic density. For example, if k^t is not greater than k_j (i.e., traffic density in time period t is not greater than the jam density), we assume a free speed in the network, and we will use the FD methodology to update the travel speed during the time period t. Otherwise (i.e., if traffic density in time period t is greater than the jam density), we assume all vehicles are not able to move any farther in the network, indicating zero vehicle speed. During a given time period t, we assume all vehicles drive at the same speed such that no overtaking is allowed in the network, and curb space is always occupied by the first vehicle that passes by.

$$v^t = \begin{cases} \frac{Q_{max}}{k_c - k_j} \cdot (1 - \frac{k_j}{k^i}), & k^t \leq k_j \\ 0, & k^t > k_j \end{cases} \tag{2.7}$$

The maximum driven distance of a vehicle in time period t (d^t) is the multiplication of the vehicle speed in time period t by the length of the time period, and we define this relation through Equation (2.8).

$$d^t = v^t \cdot t \tag{2.8}$$

To calculate the distance between two consecutive vehicles in time period t (s^t), we divide the length of the network by the number of vehicles searching for *P/PD/LU* spots as shown in Equation (2.9).

$$s^t = \frac{L}{N^t_{s_z}} \tag{2.9}$$

In Equation (2.10), we describe m^t, which is the maximum number of vehicles that can pass by the same curb space in the network during time period t. We formulate m^t by using the maximum distance a vehicle can drive and the space between two consecutive vehicles in period t. We note that all curb spaces on the network could potentially be visited by $m^t - 1$ vehicles.

$$m^t = \left\lceil \frac{d^t}{s^t} \right\rceil \tag{2.10}$$

In Equation (2.11), d^t_r is formulated as explained by its definition when $d^t > s^t$.

$$d^t_r = d^t - \left\lfloor \frac{d^t}{s^t} \right\rfloor \cdot s^t \tag{2.11}$$

Table 2.3 Model input

Notation	Definition
N_{ns}^t	New arrivals to the network during time period t.
α^t	Proportion of new arrivals during time period t that will search for P.
β^t	Proportion of new arrivals during time period t that will search for PD.
γ^t	Proportion of new arrivals during time period t that will search for LU.
L	Length of the network.
A_p	Total number of existing P spots (for public use) in the network.
A_{pd}	Total number of existing PD spots (for public use) in the network.
A_{lu}	Total number of existing LU spots (for public use) in the network.
t_l	Length of a time period.
τ_p	P duration.
τ_{pd}	PD duration.
τ_{lu}	LU duration.
$f(\tau_p)$	The probability density function of P duration.
$f(\tau_{pd})$	The probability density function of PD duration.
$f(\tau_{lu})$	The probability density function of LU duration.
v	Free flow speed, i.e., maximum speed on the network.
Q_{max}	Maximum traffic flow rate that can be adopted on the network.
k_c	Critical traffic density on the network. If the traffic density is higher than this value, then congestion occurs.
k_j	Traffic jam density.
l_{ns}	Distance that must be driven by a vehicle before it leaves the area.
l_s	Distance that must be driven by a vehicle before it starts to search for P/PD/LU.
l_g	Distance that must be driven by a vehicle that goes through the traffic before it leaves the area.
l_w	Distance that must be driven by a vehicle before it leaves the area after leaving the stationary state.
N_{ns}^0	The initial condition of non-searching state.
$N_{s_p}^0$	The initial condition of searching for P state.
$N_{s_{pd}}^0$	The initial condition of searching for PD state.
$N_{s_{lu}}^0$	The initial condition of searching for LU state.
$N_{w_p}^0$	The initial condition of P state.
$N_{w_{pd}}^0$	The initial condition of PD state.
$N_{w_{lu}}^0$	The initial condition of LU state.

Assume a vehicle is located at x_c at the beginning of a time slice *t*. The locations of the vehicles behind it are $x_c - s^t$, $x_c - 2 \cdot s^t$, ..., $x_c - m^t \cdot s^t$. The maximum driven distance is from $x_c - (m^t - 1) \cdot s^t$ to $x_c + d_r^t$. Thus, one of the vehicles behind the x_c is not able to drive beyond $x_c + d_r^t$. Since we assume identical vehicle speed within the network, vehicle located in $x_c - m^t \cdot s^t$ is not able to drive beyond $x_c + d_r^t$. In other words, a maximum of m^t vehicles can pass by area within $[x_c,\ x_c + d_r^t]$, and a maximum of $m^t - 1$ vehicles can pass by area within $[x_c + d_r^t, x_c + s^t]$.

2.1.3.4 Enter the network

We first define the number of vehicles that enter the network (i.e., $n_{(.),ns}^t$ and $n_{(.),g}^t$). We consider that there is a probabilistic traffic demand that enters the network. Among those vehicles, we assume that a δ percentage of the vehicles will go through the traffic and leave the network directly after driving a distance of l_g^t during the time period *t*. The remainder of the vehicles (i.e., $(1 - \delta)$ percentage of the vehicles) will search for a curb spot. More specifically, vehicles will search for a *P, PD*, or *LU* spot with a percentage of α, β, and γ, respectively, where $\alpha + \beta + \gamma$ =1. However, we assume that if the vehicles cruise more than a certain time before entering the searching state, they will leave the area instead. We consider that α, β, γ and δ values are fixed throughout the simulation.

2.1.3.5 Start to search for *P/PD/LU*

After vehicles enter the network, they start to search for a *P/PD/LU* spot after driving a distance l_s^t during time period *t*. However, some vehicles leave the network without entering the searching state after they cruise for a certain time, and they leave the network after driving for a distance of l_{ns}^t during time period *t*. We denote vehicle cruising time as *CT* in the following equations. We formulate the number of vehicles that cannot enter the searching state in time period *t* after cruising for a certain time through Equations (2.12) and (2.13), where $\phi'_{ns,ns}$ defines the binary variables indicating whether these vehicles can drive the required distance to start searching.

$$n_{ns,ns}^t = \sum_{t'=1}^{t-CT} n_{(.),ns}^{t'} \cdot \phi'_{ns,ns} \qquad d^{t'} < l_s^{t'} \tag{2.12}$$

where

$$\phi'_{ns,ns} = \begin{cases} 1, & l_s^t > \sum_{j=t-CT}^{j=t} d^{t'} \\ 0, & otherwise \end{cases} \tag{2.13}$$

In Equation (2.12), $n_{(.),ns}^{t'}$ consists of vehicles that entered the area in any time period $t' \in \{1, \dots, t-CT\}$. In time period $t' \in [1,\ t-CT]$, the vehicles that satisfy the following conditions cannot enter the searching state: (i) vehicles that do not drive the required distance l_s^t and (ii) vehicles that have cruised for a certain time. We define these conditions through the above equations. We further formulate the number of vehicles that start searching for *P/PD/LU* spots during time period *t* after driving a certain distance l_s^t with Equations (2.14) and (2.15).

$$n_{ns,s_z}^{t} = \sum_{t''=t-CT}^{t-1} n_{(.),ns}^{t''} \cdot \phi'_{ns,s_z} \qquad d^{t''} < l_s^{t''} \tag{2.14}$$

where

$$\phi'_{ns,s_z} = \begin{cases} 1, & l_s^t \le \sum_{j=t''}^{j=t-1} d^j \text{ and } \sum_{j=t''}^{j=t-1} d^j \le l_s^t + d^{t-1} \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

In Equation (2.14), n_{ns,s_z}^{t} consists of vehicles that transit to a searching state in any time period between $t - CT$ and $t - 1$. We do not consider the vehicles that cruise more than the cruising time *CT*. In the time period $t'' \in [t - CT, t - 1]$, $n_{(.),ns}^{t''}$vehicles will search for *P/PD/LU* spots after entering the area. Two conditions must be satisfied to enter searching state: (i) the vehicles should drive a certain distance l_s^t to start searching, and (ii) they should not start searching in previous periods. In Equation (2.15), ϕ'_{ns,s_z} indicates whether $n_{(.),ns}^{t''}$ vehicles can drive the distance l_s^t within the cruising time *CT*.

2.1.3.6 Access *P/PD/LU*

Once vehicles drive enough distance to enter the searching state, they are able to access any curb space as long as there is a vacancy. However, we keep track of only the number of vehicles that can access curb space and not which vehicles. More specifically, we do not model the exact location of each vehicle and each curb space. Our goal is to observe how the curb space allocation decisions impact the overall traffic. Thus, we model the number of vehicles that access curb space and the number of spots that are occupied at time period *t*.

At the beginning of each time period, the number of vehicles searching for *P/PD/LU* spots and the number of available curb spaces are calculated in Equation (2.2) and Equation (2.5) respectively. We use the following two assumptions in the model: First, the locations of the available curb space are random at the beginning of each time period. Second, the locations of searching vehicles are uniformly distributed on the network at the beginning of each time period. The first assumption ensures stochasticity of the parking availability. The second assumption guarantees that the demand is homogeneously generated. The second assumption is necessary because if vehicles are located mostly within a few streets, the other available curb spaces will not be occupied even if they are vacant. Also, the model can provide an average amount of curb space being taken, and this average value is meaningful only when all searching vehicles are uniformly distributed in the network.

We use x to denote the curb space location. Assume a *P/PD/LU* spot is located at the location x_z, and the remaining *P*, *PD*, and *LU* spots are located at the location x_μ, for $\mu \in \{1,2,3,\dots,A_z^t - 1\}$ (i.e., there remain $A_z^t - 1$ spots for each curb space use z). We consider that the searching vehicles' initial positions are at the location x_c, for $c \in \left\{1,2,3,\dots,N_{s_z}^t - 1\right\}$. Then, we consider three different cases based on the relations between d^t, s^t, and L to calculate the number of searching vehicles that access a curb space for parking, picking-up/dropping-off, or loading/unloading.

- *Case 1:* if $d^t \in [0, s^t]$.

Under this scenario, the maximum driving distance of a vehicle (d^t) is shorter than the spacing between two consecutive vehicles (s^t). Therefore, no two vehicles' trajectories will ever overlap during a single time slice. As a result, a curb spot can be visited at most by one vehicle.

Then, there are two conditions to guarantee that this P/PD/LU spot at the location x_z becomes occupied during time slice t:

- *Condition 1:* The available spot at the location x_z must be within the reach of a vehicle. $x_z \in [x_c,\ x_c + d^t]$ for any $c \in \left[1,\ N_{s_z}^t\right]$. The probability is $\sum_{c=1}^{N_{s_z}^t} \int_{x_c}^{x_c+d^t} \frac{1}{L} d\, x_z$.

- *Condition2:* There should not be any other curb spaces between x_c and x_z. The probability is $\prod_{x_\mu=1}^{A_z^t-1} \left(1 - \int_{x_c}^{x_z} \frac{1}{L} dx_\mu\right)$.

Thus, the probability of random *P/PD/LU* spots been taken during time slice *t* is the product of these two probabilities defined under Condition 1 and Condition 2. The average number of *P/PD/LU* spots that are occupied during the time period *t* equals the multiplication of the number of available spots in each use (i.e., A_z^t) by the product of these two probabilities. We define this expression through Equation (2.16).

$$n_{s_z,w_z}^t = A_z^t \cdot \sum_{c=1}^{N_{s_z}^t} \int_{x_c}^{x_c+d^t} \frac{1}{L} d\, x_z \cdot \prod_{x_\mu=1}^{A_z^t-1} \left(1 - \int_{x_c}^{x_z} \frac{1}{L} dx_\mu\right) \tag{2.16}$$

- *Case 2:* if $d^t \in [s^t,\ L]$.

In this case, vehicles' trajectories can overlap, and a curb spot can be visited by more than one vehicle (although it only accommodates the first one). We define the probability of a curb spot at the location x_z being occupied during time period *t* through three sub-cases (i.e., $m^t > A_z^t, m^t = A_z^t, m^t < A_z^t$). We investigate the number of vehicles that transit from the searching state to the stationary state for each curb use type *z* (i.e., $N_{s_z,w_z}^t = A_z^t$) for each sub-case and describe the details below:

- *Sub-case 2.1:* if $m^t > A^t$. (i.e., the maximum number of vehicles that can pass by the same spot on the network during time period *t* is greater than the number of available spots in period *t*).

In this case, according to Equation (2.9) and Equation (2.10), $N_{s_z}^t > m^t$. Therefore, there is more parking demand than supply ($N_{s_z}^t > A_z^t$). Since any curb space in the network could be potentially visited by $m^t - 1$ vehicles *(*$m^t - 1 \geq A_z^t$), any available curb space can be taken by one of these vehicles. More specifically, in this case, there are too many vehicles searching and they drive a distance that is long enough to reach all available spots. Hence, all available curb spots will be taken, and still, some vehicles will remain searching at the end of the time period *t*. Then, the n_{s_z,w_z}^t is written in Equation (2.17).

$$n_{s_z,w_z}^t = \min\left\{A_z^i, N_{s_z}^t\right\} \tag{2.17}$$

- *Sub-case 2.2:* if $m^t = A^t$. (i.e., the maximum number of vehicles that can pass by the same spot on the network during time period *t* equals the number of available spots in period *t*).

If $x_z \in [x_c,\ x_c + d_r^t]$, a number of m^t cars could drive by that *P/PD/LU* spot at x_z. If *P/PD/LU* spot is located within this area, it will be taken. The probability of a curb spot located within this range and been taken is: $\sum_{c=1}^{N_{s_z}^t} \int_{x_c}^{x_c+d_r^t} \frac{1}{L} dx_z$.

If $x_z \in [x_c + d_r^t,\ x_c + s^t]$, a number of $m^t - 1$ car could drive by that *P/PD/LU* spot at x_z. Denote $P_{f(n=m^t-1)}$ as the probability of this curb spot not being taken, i.e., the probability that all the vehicles that could reach the location x_z park before arriving at x_z. The probability of a curb spot located within this range and been taken is: $\sum_{c=1}^{N_{s_z}^t} \int_{x_c+d_r^t}^{x_c+s^t} \frac{1}{L} \cdot \left(1 - p_{f(n=m^t-1)}\right) dx_z$.

Combing these two probabilities, N_{s_z,w_z}^t is written as Equation (2.18).

$$n_{s_z,w_z}^t = A_z^t \cdot \sum_{c=1}^{N_{s_z}^t} \left\{ \int_{x_c}^{x_c+d_r^t} \frac{1}{L} dx_z + \int_{x_c+d_r^t}^{x_c+s^t} \frac{1}{L} \cdot \left(1 - p_{f(n=m^t-1)}\right) dx_z \right\} \tag{2.18}$$

$$p_{f(n)} = \underbrace{\sum_{Z_n=n}^{A_z^t-1} C_{Z_n}^{A_z^t-1} \cdot \left(\int_{-(n-1)s^t}^{x_z} \frac{1}{L} dx_z \right)^{Z_n} \cdot \left(1 - \int_{-(n-1)s^t}^{x_z} \frac{1}{L} dx_z \right)^{A_z^t-1-Z_n}} \cdot \prod_{j=1}^{n-1} p_{f_j} \tag{2.19}$$

$$p_{f_j} = \sum_{z_j=j}^{z_j+1} C_{Z_j}^{Z_j+1} \cdot \left(\frac{\int_{-(j-1)s^t}^{x_z} \frac{1}{L} dx_z}{\int_{-j \cdot s^t}^{x_z} \frac{1}{L} dx_z} \right)^{z_j} \cdot \left(1 - \frac{\int_{-(j-1)s^t}^{x_z} \frac{1}{L} dx_z}{\int_{-j \cdot s^t}^{x_z} \frac{1}{L} dx_z} \right)^{z_j+1-z_j} \tag{2.20}$$

In Equation (2.19), n stands for the number of vehicles that can potentially reach x_z. Within these n cars, the probability that the furthest vehicle (to x_z) parks before it arrives at x_z is shown in term 1. The probability that the rest $n-1$ vehicles all park before they arrive at x_z is shown in Equation (2.20).

- *Sub-case 2.3:* if $m^t < A^t$.

Similar to sub-case 2.2. If $x_z \in [x_c, x_c + d_r^t]$, a number of m^t cars can drive by that *P/PD/LU* spot at x_z. If a *P/PD/LU* spot is located within this area, it will be taken. The probability is: $\sum_{c=1}^{N_{s_z}^t} \{ \int_{x_c}^{x_c+d_r^t} \frac{1}{L} \cdot \left(1 - p_{f(n=m^t)}\right) dx_z$.

If $x \in [x_c + d_r^t, x_c + s^t]$, a number of $m^t - 1$ car could drive by that *P/PD/LU* spot at x_z. Denote $P_{f(n=m^t-1)}$ as the probability of this parking spot not being taken, i.e., the probability that all the cars that could reach location x park before arriving at x_z. The probability is: $\sum_{c=1}^{N_{s_z}^t} \int_{x_c+d_r^t}^{x_c+s^t} \frac{1}{L} \cdot \left(1 - p_{f(n=m^t-1)}\right) dx_z$.

Combing these two probabilities, N_{s_z,w_z}^t is written as Equation (2.21).

$$n_{s_z,w_z}^t = \sum_{c=1}^{N_{s_z}^t} \{ \int_{x_c}^{x_c+d_r^t} \frac{1}{L} \cdot \left(1 - p_{f(n=m^t)}\right) dx_z + \int_{x_c+d_r^t}^{x_c+s^t} \frac{1}{L} \cdot \left(1 - p_{f(n=m^t-1)}\right) dx_z \tag{2.21}$$

- *Case 3:* if $d^t \in [L,\ \infty]$.

In this case, each vehicle can drive around the whole network at least once, so all vehicles will access curb space if there are enough curb spots. Otherwise, all curb space will be taken. n_{s_z,w_z}^t is written as Equation (2.22).

$$n_{s_z,w_z}^t = min\{ A_z^t, N_{s_z}^t \} \tag{2.22}$$

Since the computational cost of n_{s_z,w_z}^t is very large, so that we do some simplification and approximation to lower the cost (Cao & Menendez, 2015). For example, the simplification of case 1 is described as below:

$$\begin{aligned} n_{s_z,w_z}^t &= A_z^t * \sum_{c=1}^{c=N_{s_z}^t} \int_{x_c}^{x_c+d^t} \frac{1}{L} dx_z * \prod_{x_\mu=1}^{A_z^t-1} \left(1 - \int_{x_c}^{x_z} \frac{1}{L} dx_\mu\right) \\ &= A_z^t * \sum_{c=1}^{c=N_{s_z}^t} \int_{x_c}^{x_c+d^t} \frac{1}{L} * \left(1 - \frac{x_z}{L} + \frac{x_c}{L}\right)^{A_z^t-1} * dx_z \\ &= A_z^t * \sum_{c=1}^{c=N_{s_z}^t} \left(-\frac{1}{A_z^t} * \left(\left(1 - \frac{x_z}{L} + \frac{x_c}{L}\right)^{A_z^t}\right) |x_c + d^t, x_c\, |\right) \\ &= A_z^t * \sum_{c=1}^{c=N_{s_z}^t} \left(-\frac{1}{A_z^t} * \left(\left(1 - \frac{d^t}{L}\right)^{A_z^t} - 1^{A_z^t}\right)\right) \\ &= -N_{s_z}^t * \left(\left(1 - \frac{d^t}{L}\right)^{A_z^t} - 1^{A_z^t}\right) \\ &= N_{s_z}^t * \left[1 - \left(1 - \frac{d^t}{L}\right)^{A_z^t}\right] \end{aligned} \tag{2.23}$$

After simplification, we can further use the approximation methods proposed in paper System dynamics of urban traffic based on its parking-related-states (Cao & Menendez, 2015) to simplify further n_{s_z,w_z}^t. In this paper, we do not show the approximation process because it is pretty similar to that in the referring papers. Finally, n_{s_z,w_z}^t is written in Equation (2.24a) and Equation (2.24b).

$$n_{s_z,w_z}^t = \begin{cases} \begin{cases} N_{s_z}^t * \left[1-\left(1-\frac{v^t * t}{L}\right)^{N_{w_z}^t}\right], if t \in \left[0, \frac{L}{v^t * N_{s_z}^t}\right] \\ N_{w_z}^t + N_{s_z}^t \left[\frac{N_{w_z}^t}{N_{s_z}^t} - 1 + \left(1-\frac{1}{N_{s_z}^t}\right)^{N_{w_z}^t}\right] * log_{N_{w_z}^t} \frac{N_{s_z}^t}{N_{w_z}^t} * \frac{v^t * t}{L}, if t \in \left[\frac{L}{v^t * N_{s_z}^t}, \frac{L}{v^t} * \frac{N_{w_z}^t}{N_{s_z}^t}\right], if N_{w_z}^t \le N_{s_z}^t \\ N_{w_z}^t \qquad , if\ t \in \left[\frac{L}{v^t} * \frac{N_{w_z}^t}{N_{s_z}^t}, \infty\right] \end{cases} \quad (2.24a) \\ \begin{cases} N_{s_z}^t * \left[1-\left(1-\frac{v^t * t}{L}\right)^{N_{w_z}^t}\right], if t \in \left[0, \frac{L}{v^t * N_{s_z}^t}\right] \\ N_{s_z}^t \left[1 + \left(1-\frac{1}{N_{s_z}^t}\right)^{N_{w_z}^t} * log_{N_{s_z}^t} \frac{v^t * t}{L}\right] \qquad , if\ \ t \in \left[\frac{L}{v^t * N_{s_z}^t}, \frac{L}{v^t}\right], if\ N_{w_z}^t \ge N_{s_z}^t \\ N_{s_z}^t \qquad , if\ t \in \left[\frac{L}{v^t}, \infty\right] \end{cases} \quad (2.24b) \end{cases}$$

2.1.3.7 Depart *P/PD/LU*

As we know the number of vehicles that access *P/PD/LU* in all previous time slices, we can define the number of vehicles that transit from one of the stationary states to the non-searching state. We use the probability distribution function of the parking, picking-up/dropping-off, and loading/unloading durations. Equation (2.25) shows the number of vehicles that depart from stationary state *z* in time period *t*.

$$n_{w_z,ns}^t = \sum_{t'=1}^{t-1} n_{s_z,w_z}^{t'} \cdot \int_{(t-t')\cdot t_l}^{(t+1-t')\cdot t_l} f(\tau_z) d\, t_z \qquad (2.25)$$

In Equation (2.25), $n_{w_z,ns}^t$ consists of vehicles that accessed curb space in any time slice between 1 and $t-1$. Use t' to denote such time slice, $t' \in [1, t-1]$. Notice that the vehicles that access curb space during time slice *t* are not included, as they already experience one transition event during this time slice. The number of vehicles that accessed curb space in time slice t' is n_{s_z,w_z}^{t}. The probability that these vehicles depart parking in time slice *t* equals to the probability of the parking duration being between $(t-t)\cdot t_l$ and $(t+1-t')\cdot t_l, i.e., \int_{(t-t')\cdot t_l}^{(t+1-t')\cdot t_l} f(\tau_z) d\, t_z$.

2.1.3.8 Leave the network

Vehicles that do not access the curb space (i.e., $n_{(.),g}^{t}, n_{ns,ns}^{t}$), or that access and leave the curb space ($n_{w_z,ns}^{t}$), will leave the network after driving a certain distance. We use l_g^t and l_{ns}^t to denote the required distances that the vehicles need to drive to leave the network for different system events. Then we define the number of vehicles leaving the network at time period *t* with Equations (2.26) - (2.28).

$$n_{ns,(.)}^{t} = \sum_{t'=1}^{t-1}\left(n_{(.),g}^{t'} \cdot \phi_{g,(.)}' + \sum_{z\in\{P,PD,LU\}}\left(n_{w_z,ns}^{t'} \cdot \phi_{ns,(.)}'\right) + n_{ns,ns}^{t'} \cdot \phi_{ns,(.)}'\right) \tag{2.26}$$

where

$$\phi_{g,(.)}' = \begin{cases} 1, & l_g^t \le \sum_{j=t'}^{j=t-1} d^j \; and \; \sum_{j=t'}^{j=t-1} d^j \le l_g^t + d^{t-1} \\ 0, & otherwise \end{cases} \tag{2.27}$$

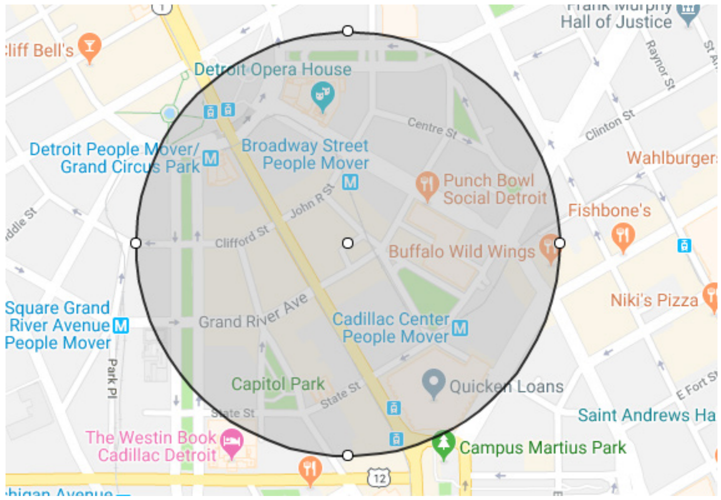
$$\phi_{ns,(.)}' = \begin{cases} 1, & l_{ns}^t \le \sum_{j=t'}^{j=t-1} d^j \; and \; \sum_{j=t'}^{j=t-1} d^j \le l_{ns}^t + d^{t-1} \\ 0, & otherwise \end{cases} \tag{2.28}$$

As shown in Equation (2.26), $n_{ns,(.)}^{t}$ consists of three parts, vehicles leave the network without accessing *P/PD/LU* spots, vehicles leave the network after *P/PD/LU*, and vehicles leave the network before they enter the "searching" state after a specific cruising time. $\phi'_{g,(.)}$ and $\phi'_{ns,(.)}$ are binary variables indicating whether these vehicles can leave the network at time slice *t*.

2.2 Numerical Study

In this section, we consider an urban traffic network located in downtown Detroit and conduct numerical experiments to validate the efficiency of the proposed simulation model. We select a network in the downtown Detroit area with a radius of 300 meters. In total, this network consists of 260 on-street curb spaces for public use (Parkopedia.com, 2019). First, we calculate the length of all streets inside this network that provide curb spaces for public use by using Google Distance API and the data provided from the website Parkopedia.com (Parkopedia.com, 2019). We further calculate the curb space width by using the Parking Area Design Report (WSDOT, 2003). Figure 2.3 displays the layout of the selected urban traffic network. This network contains 12 streets with a total length of 5.32 kilometers (calculated using the Google Distance API). We assume that each street has two directions and one lane per direction on average. Then, the total length of the network is 5.32 * 2 = 11.7 kilometers. Additionally, we study the rush-hour traffic in the downtown area, and we assume that the critical traffic density is k_c = 25 veh/km/lane and jam density is k_j=55 veh/km/lane (Cao and Menendez 2015).

Figure 2.3 Selected urban network in downtown Detroit area



We use the Regional Traffic Counts Database (SEMCOG, 2019) to estimate the approximate number of vehicles that enter the network within a given time period. This database provides the daily traffic of each street so that we can estimate the proportionate traffic demand of the streets that are in the selected network. The average vehicle speed in Detroit is about 40 KPH (kilometers per hour) without traffic, based on a Detroit city speed report (Kleint, 2011). We use an average speed of 30 KPH by considering the traffic in the downtown area during rush hours. We further perform sensitivity analysis on speed by using a speed range between 20 KPH and 40 KPH. Since all the existing on-street parking spots in the selected network are metered parking, we consider the metered parking duration for our setting. Based on the studies in the literature ((Adiv & Wang, 1987), (Gallo, et al., 2011), (Shoup, 2017)), we use gamma distribution to model the duration for parking, pick-up/drop-off, and loading/unloading. We use similar parameter values to those of the study of Adiv and Wang (Adiv & Wang, 1987), where the authors study metered on-street parking behavior by using both historical data and survey data from downtown Ann Arbor, Michigan. We include the figure of the probability distribution function of the parking duration in Figure 2.4. We estimate parameters of the LU duration distribution based on a survey conducted in a study about commercial vehicles' parking duration in New York City and its implications for planning (Schmid, et al., 2018). The duration of picking-up/dropping-off is expected to be shorter than loading/unloading goods, in general. Thus, we assume a shorter *PD* duration and estimate our parameters accordingly. All these known parameters related to the Detroit area are described in Table 2.3.

Figure 2.4 Illustration of traffic heading for the parking following the gamma distribution

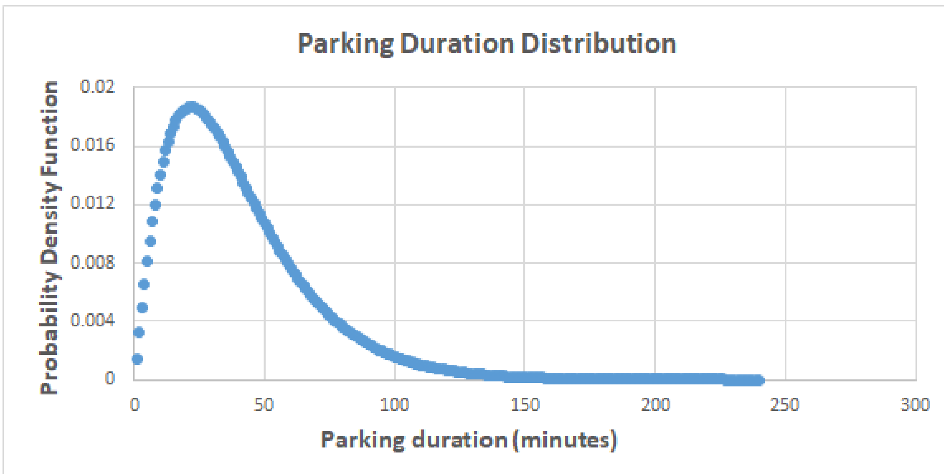


Table 2.4 Known parameters for the selected Detroit area

Notation	Definition	Unit	Value
L	Length of network	km	11.7
A	The number of existing on-street parking spots		260
k_c	Traffic density	veh/km/lane	25
k_j	Jam density	veh/km/lane	55
τ_p	Parking duration.	minute	gamma(2.195, 18.225)
τ_{pd}	PD duration.	minute	gamma(1, 8)
τ_{lu}	Lu duration.	minute	gamma(1, 15)

2.2.1 Sensitivity Analysis

In this section, we conduct our numerical experiments to observe the change in the traffic flow, traffic delay, and occupancy of curb space for different uses by considering several scenarios. We assume that the curb space allocation for different uses is given, and we investigate the optimal curb space allocation decisions in Chapter 3. In this section, we assume that the allocated curb spaces are proportional to the average demand ratio considered in the model. To this end, we consider that the percentages of the allocated curb spaces for parking, picking-up/dropping-off, and loading/unloading are 70%, 20%, and 10% respectively. Current parking policy in Detroit is static in this selected area, which means that the use of the curb space is fixed over time. Thus, we consider only a static curb space allocation policy in this section. We simulate the traffic system in Detroit for six hours (i.e., between 6:00 a.m. and 12:00 noon). We summarize the parameters used in the numerical analysis in Table 2.4. Since the sum of the demand proportions of *P*, *PD*, and LU should be equal to 1, we consider 18 combinations composed by $\delta, \alpha, \beta, \gamma$. For the traffic demand, vehicle speed, and cruising time we consider three possible values. Hence, in total, we analyze $18 * 3^3 = 486$ instances for the sensitivity analysis.

2.2.2 Traffic Delay and Vehicle Driven Distance Calculation

In previous section, we have generated 486 instances with different parameters to analyze how the traffic delay and vehicle driven distance changes under known curb space allocation decisions. We use one instance here as an example on how we calculate the traffic delay and

Table 2.5 Parameter setting for sensitivity analysis

Notation	Definition	Unit	Value	Distribution
T	Traffic demand entering the network.	veh	[3500, 4500, 6000]	
δ	Traffic demand entering the network and go through the network.	veh	[0.5, 0.6, 0.7]	
α	Traffic demand entering the network and headed to P.	veh	[0.6, 0.7, 0.8]	
δ	Traffic demand entering the network and headed to PD.	veh	[0.1, 0.2, 0.3]	
γ	Traffic demand entering the network and headed to LU.	veh	[0.1, 0.2, 0.3]	
v	The free flow speed of network (with traffic flow)	km/h	[20, 30, 40]	
CT	Cruising time	min	[5, 10, 15]	
l_g	Distance that must be driven by a vehicle that goes through the traffic before it leaves the area.	km	[0, 0.5]	uniform
l_s	Distance that must be driven by a vehicle before it starts to search for P/PD/LU.	km	[0, 0.5]	uniform
l_w	Distance that must be driven by a vehicle before it leaves the area after leaving the stationary state.	km	[0, 0.5]	uniform
l_{ns}	Distance that must be driven by a vehicle before it leaves the area.	km	[0, 0.5]	uniform

vehicle driven distance. We use the medium parameter values from Table 2.4 to generate the queuing diagrams for traffic heading for *P*, traffic heading for *PD*, traffic heading for *LU*, and traffic goes through the network from Figure 2.5 – Figure 2.8. Notice that we do not assign any curb space for *PD* or *LU* in this instance.

In Figure 2.5 – Figure 2.8, we illustrate the change in the cumulative number of vehicles that transit between system events over time for different use cases. For example, Figure 2.5 illustrates the change in the cumulative number of vehicles that transit between system events over time for parking use. The total number of vehicles that enter the network (i.e., the line "Enter the area"), that start searching for a parking spot (i.e., the line "Start searching for parking"), that leave the area after cruising for a certain time before entering the searching state (i.e., the line "Leave the area without parking (resp. pick -up/drop-off)"), that access the curb space (i.e., the line "Access parking (resp. pick-up/drop-off)"), that depart the curb space after parking (resp. pick-up/drop-off) (i.e., the line "Depart parking"), and that leave the network after parking (resp. pick-up/drop-o) (i.e., the line "Leave the area after parked"). Through Figure 2.5, we can calculate the average traffic delay and average vehicle driven distance. The area between the two curves is the total time vehicles spend within that state. The non-searching vehicle time contains two areas, the area between the curve "Enter the area" and "Start searching for parking" and "Leave the area without parking", and the area between the curve "Depart parking" and the curve "Leave the area after parked". The total non-searching parking delay equals the total time vehicles spend in the "non-searching" state minus the total time vehicles spend in the "non-searching" state when there has no congestion (i.e., the total time go through traffic vehicles spend when there is no congestion is $\frac{\sum_{t'=1}^{t} n_{(.),g}^{t'} \cdot l_g^{t'}}{v}$, the average travel speed remains free speed *v*). However, since we assume vehicles in searching for parking will access parking once there is available parking spot, the searching vehicle time is $\frac{\sum n_{s_z,w_z}^{t} \cdot d^t}{v^t}$. The searching delay will be the same as searching time. Notice that the traffic go through the network does not experience all the transition events, they leave the network after driving for a certain distance after entering the area. Thus, the total delay of traffic go through the network is the area between curve "Enter the area" and the curve "Leave the area" as shown in Figure 2.8. Thus, if there is congestion, this area should be greater than 0. The driven distance calculation in the "non-searching" state is based on the required distances we define (i.e., total driven distance for traffic go through the network $\sum_{t'=1}^{t} n_{(.),g}^{t'} \cdot l_g^{t'}$). While the driven distance in the searching state is $\sum n_{s_z,w_z}^{t} \cdot d^t$.

In Table 2.5, we present the minimum, average, and maximum traffic delay for different states among all scenarios (486 instances) under the current curb space management policy (assign all curb space for parking use only). We note that we let all vehicles leave the network even after the simulation ends. As shown, the total traffic delay per vehicle ranges from 586 to 910 minutes. The average delay time of searching for a curb space ranges between 93 and 305 minutes, while the delay time in the non-searching state ranges from 0.76 to 132 minutes. This varying range shows that it is important to have an efficient and dynamic curb space allocation policy that can change over time as a response to varying demand.

Table 2.6 Average vehicle time and traffic delay of $\alpha = 1,\ \beta = 0,\ \gamma = 0,\ \delta = 0.5$

	Average Delay Time Per Vehicle (minutes)				
Scenario	Non-searching state	Searching for Parking	Searching for Pick-up/ Drop-off	Searching for Loading/ Unloading	Total
Minimum	0.76	40.75	257.26	288.05	586.83
Average	61.9	93.33	272.08	305.14	732.45
Maximum	131.53	182.11	280.05	317.11	910.8

In Table 2.6, we present the minimum, average, and maximum values of average driven distance among all scenarios under the current curb space allocation policy. As shown in the table, we observe that among 486 scenarios, the minimum average driven distance is 2.05 kilometers and the maximum average driving distance 157.39 kilometers. The delay time and the driven distance are highly related to the allocated curb space for different uses.

Table 2.7 Average driven distance of $\alpha = 1,\ \beta = 0,\ \gamma = 0,\ \delta = 0.5$

	Average Driven Distance (km)				
	Non-searching state	Searching for P state	Searching for PD state	Searching for LU state	Total
Minimum	0.09	0.26	0.78	0.91	2.05
Average	0.17	1.06	6.13	10.21	17.57
Maximum	0.25	6.29	50.29	100.56	157.39

Figure 2.5 The number of vehicles transitioning between states (Parking case)

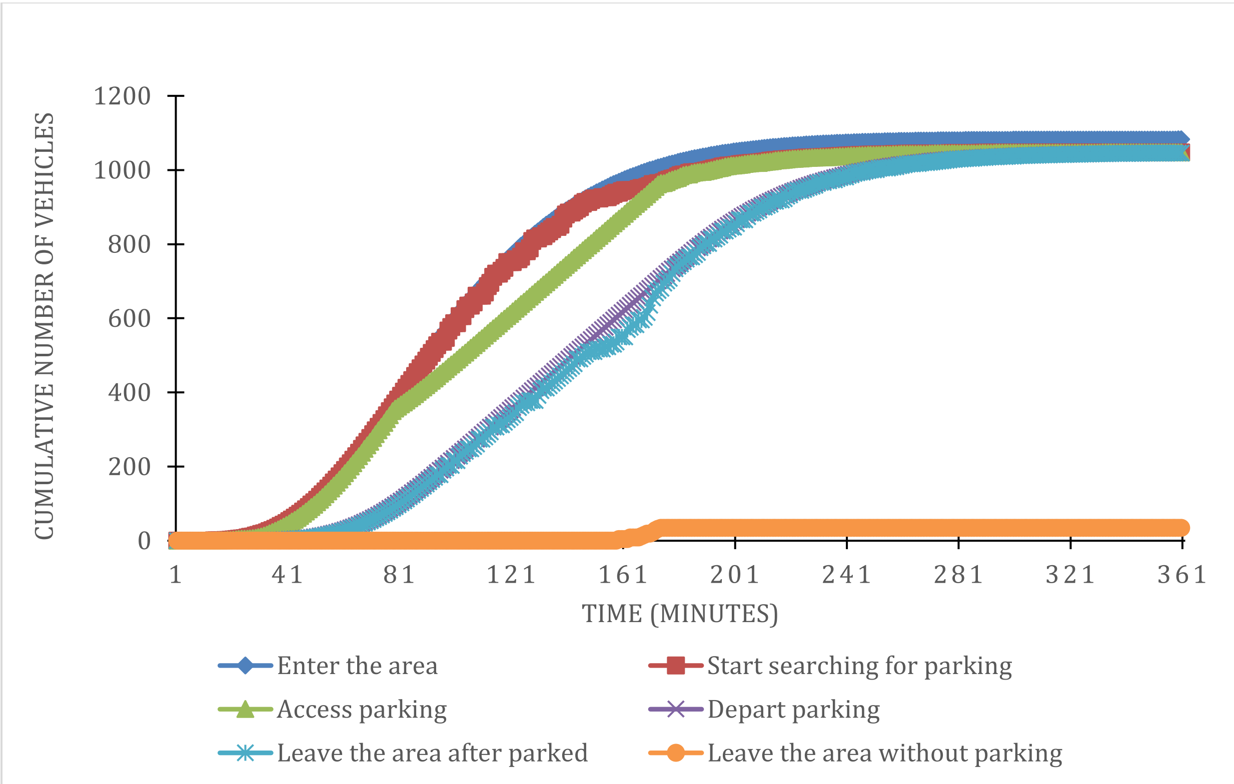


Figure 2.6 The number of vehicles transitioning between system events (PD case)

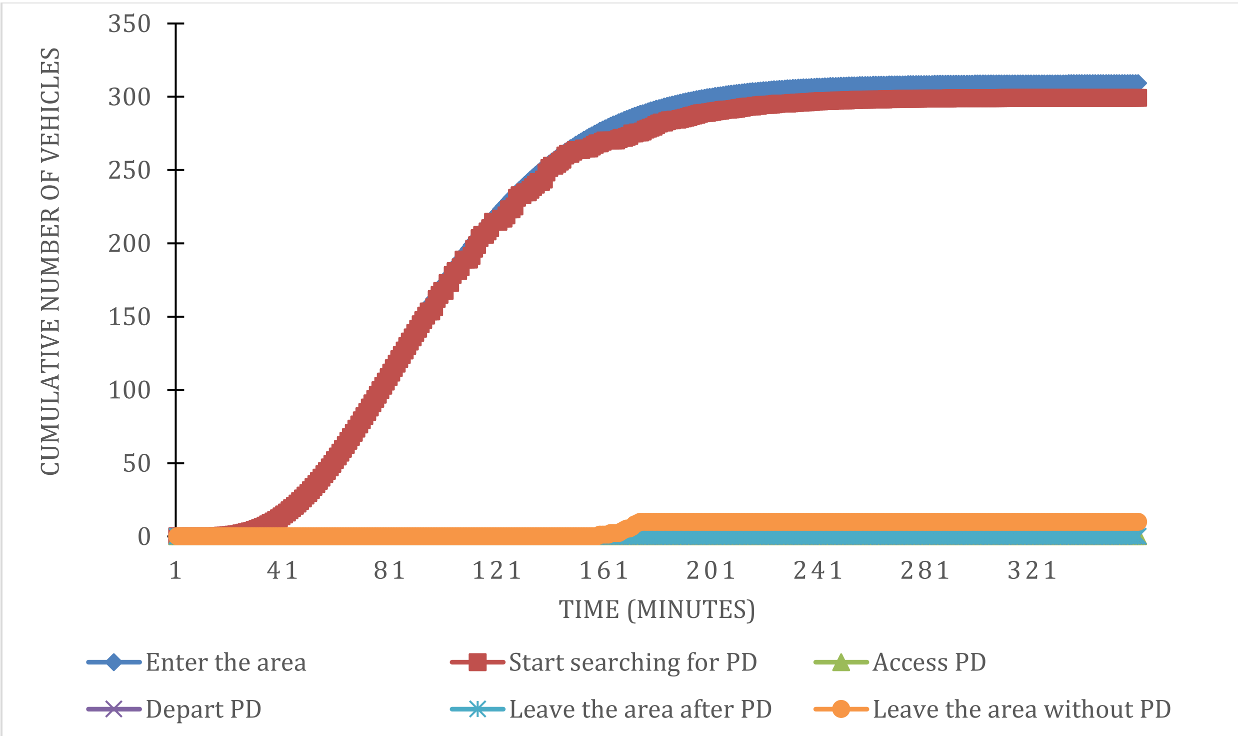


Figure 2.7 The number of vehicles transitioning between states (LU case)

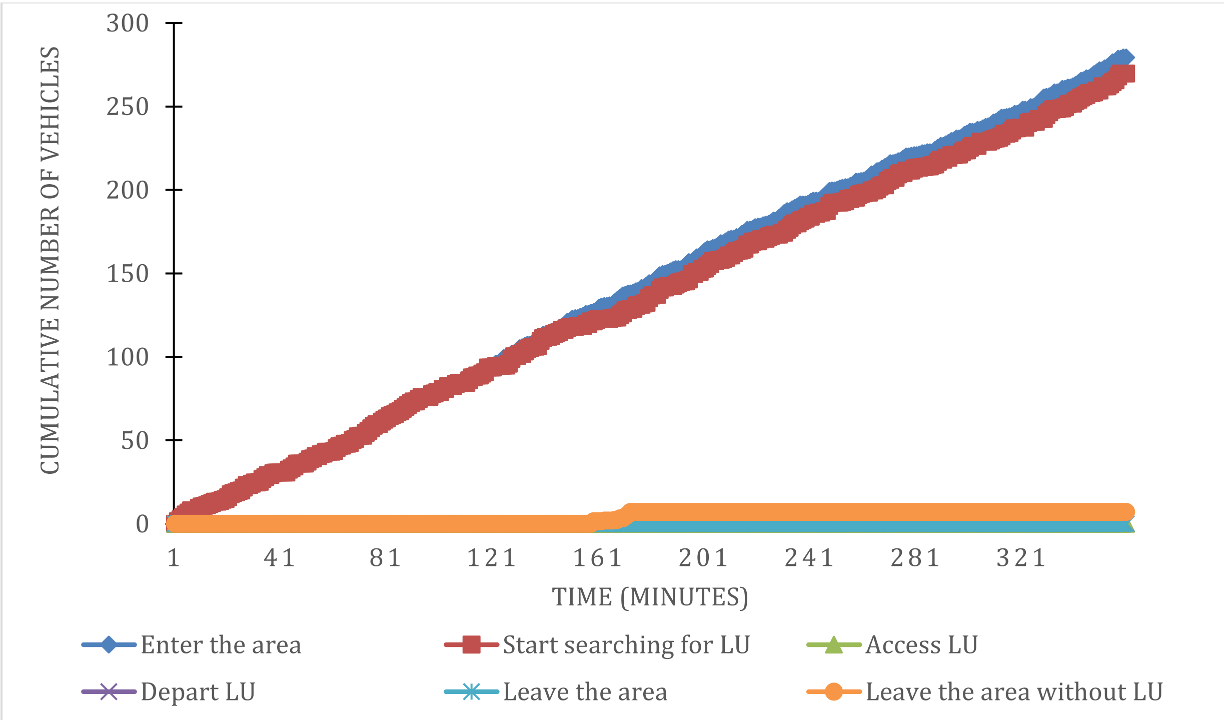
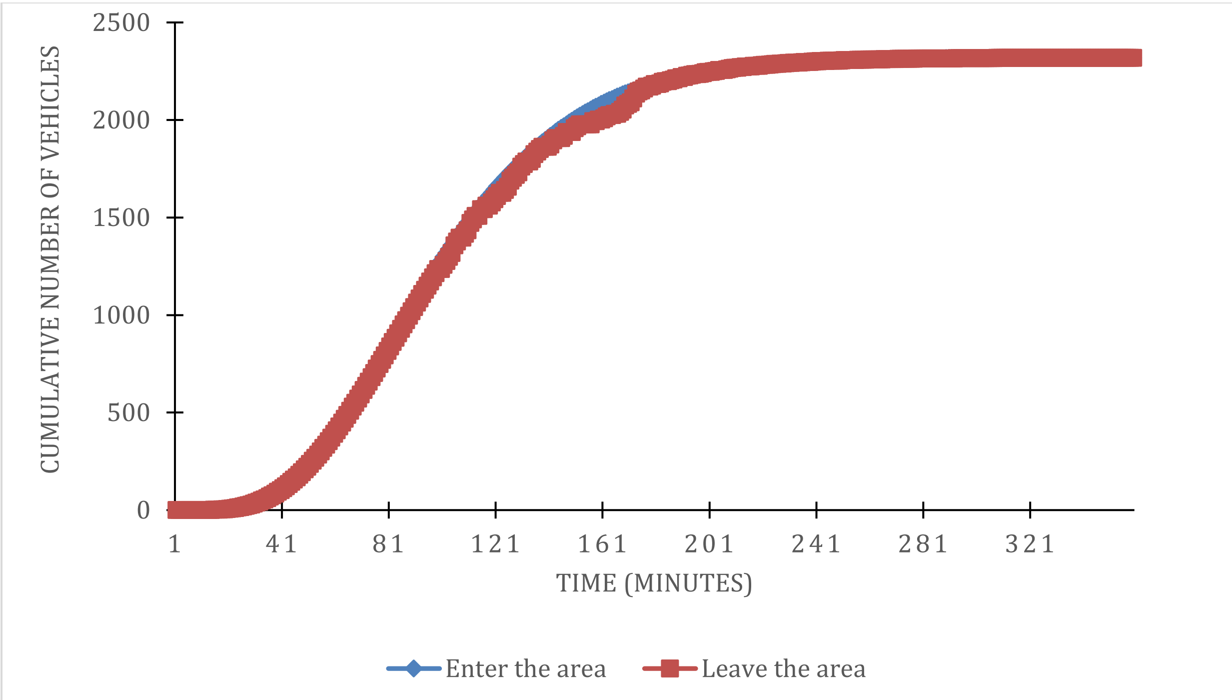


Figure 2.8 The number of vehicles transitioning between states (Through traffic case)



In Table 2.7, we present the minimum, average, and maximum trac delay for different states among all scenarios after we have assigned some curb space for *PD* and *LU* uses. As shown, the average delay per vehicle is from 70.52 to 561.91, the average delay time of searching for a curb space ranges between 32.81 and 111.09 minutes, while the delay time in the non-searching state ranges from 0.88 to 113 minutes. This varying range also shows that it is important to have an efficient and dynamic curb space allocation policy that can change over time as a response to varying demand.

In Table 2.8, we present the minimum, average, and maximum values of average driven distance among all scenarios. As shown, the minimum average driving distance is 8.88 kilometers and the maximum average driving distance 548.3 kilometers. The delay time and the driven distance are highly related to the allocated curb space for different uses. Hence, as a next step, we investigate efficient ways of allocating the curb space for different uses.

Table 2.8 Average vehicle time and traffic delay of $\alpha = 0.7,\ \beta = 0.2,\ \gamma = 0.1,\ \delta = 0.5$

	Average Delay Time Per Vehicle (minutes)				
Scenario	Non-searching state	Searching for Parking	Searching for Pick-up/ Drop-off	Searching for Loading/ Unloading	Total
Minimum	0.88	42.48	10.57	16.59	70.52
Average	36.52	111.09	32.81	36.82	217.24
Maximum	113.58	183.53	93.87	170.94	561.91

Table 2.9 Average driven distance of $\alpha = 0.7,\ \beta = 0.2,\ \gamma = 0.1,\ \delta = 0.5$

	Average Driven Distance (km)				
	Non-searching state	Searching for P state	Searching for PD state	Searching for LU state	Total
Minimum	0.11	1.13	3.4	4.23	8.88
Average	0.21	7.8	37.47	71.78	117.27
Maximum	0.28	21.92	175.38	350.72	548.3

Chapter 3: The Optimization Model

This chapter compromises three main parts. First, we build an optimal curb space allocation model and proposed two algorithms to solve the problem within half an hour quickly. Second, we compare the efficiency of the proposed algorithms with the non-linear solver's results. Third, we validate the efficiency of the proposed algorithms further using different starting points.

3.1 Model Formulation

In this section, we build a curb space allocation model by integrating the outputs of the simulation model. We develop an optimization model to allocate the curb space optimally among three different uses (i.e., *P*, *PD*, and *LU*). Given the total number of existing curb spaces, our goal is to maximize the total profit of an urban traffic network by allocating the available spaces for *P*, *PD*, and *LU* uses over time. First, we consider a static use of curb space by assigning a fixed allocation for parking, pick-up/drop-off, and loading/unloading. In practice, the curb space allocation strategies of cities are mostly static, where the use of the curb spaces is fixed. Indeed, currently, most of the curb spaces are used for solely parking.

To this end, we define ρ_p as the unit profit obtained from the parked vehicles and c_d as the unit cost of traffic delay. We further use σ^t to denote the total traffic delay in time period *t* due to the congestion. We note that we calculate the traffic delay through the simulation model, and the traffic delay varies as the curb space allocation for different uses changes. In our optimization model, we use A_z, which is the fixed number of curb spots allocated for curb use type z, as the decision variable of the model. Let M_A be the total curb space available. Then the optimization model for the static curb use can be defined as follows:

$$max \quad F(A_z) = \sum_{t=1}^{|\mathcal{T}|} N_p^t \cdot \rho_p - \sigma^t \cdot c_d \tag{3.1}$$

$$s.t. \sum_{z \in \{p,pd,lu\}} A_z = M_A \tag{3.2}$$

$$A_z \geq 0, \qquad \forall z \in \{p, pd, lu\}, t \in \mathcal{T} \tag{3.3}$$

In the above model, Equation (3.1) represents the objective function, which is the profit obtained from the curb space allocation decisions over all periods. The first term represents the profit earned from the parked vehicles over all periods, while the second term is the total cost due to the traffic delay. In constraint (3.2), we ensure that the total allocated spots for different uses should be equal to the total available curb space spots. Finally, constraint (3.3) defines the non-negativity constraints.

We further consider that the allocated curb spaces can be flexible and can change during the day by considering the demand of different curb uses. Hence, we define $h \in \mathcal{H}$ to represent the number of epochs where the number of allocated curb space for a different type of uses can change in each epoch h. We redefine the time as follows: $t \in \left\{1,2, \dots, \frac{|\mathcal{T}|}{|\mathcal{H}|}, \frac{|\mathcal{T}|}{|\mathcal{H}|} + 1, \dots, \frac{2 \cdot |\mathcal{T}|}{|\mathcal{H}|}, \dots, |\mathcal{T}|\right\}$. Then, our dynamic curb space allocation model can be defined as follows:

$$max \quad F(A_z^h) = \sum_{h=0}^{|\mathcal{H}|-1} \sum_{t=1+h\frac{|\mathcal{T}|}{|\mathcal{H}|}}^{(h+1)\frac{|\mathcal{T}|}{|\mathcal{H}|}} N_p^t \cdot \rho_p - \sigma^t \cdot c_d \tag{3.4}$$

$$s.t. \sum_{z \in \{p,pd,lu\}} A_z^h = M_A, \qquad \forall h \in \mathcal{H} \tag{3.5}$$

$$A_z^h \geq 0, \qquad \forall z \in \{p, pd, lu\}, h \in \mathcal{H} \tag{3.6}$$

where A_z^h represents the number of allocated curb space for the curb use z in epoch h. The dynamic curb space allocation model is similar to the static model. More specifically, equation (3.4) is used to define the profit function. Constraint (3.5) states that the allocated curb spots in each epoch equal to the total available capacity, and constraint (3.6) defines the non-negativity. The dynamic model allows the curb space allocation policy can change over time. This flexibility can ensure that the traffic delay within a specific time interval can be minimized as well as the curb space can be utilized to the most extent.

3.2 Heuristic Policy

The above curb space allocation model is challenging to solve as it requires the traffic delay output of the simulation model for all different curb space use configurations (A_z) to find the optimal setting. In our model, as the number of time periods and the number of available curb spots increase, it becomes intractable to compute the optimal objective function and find the optimal allocation policy. In this section, to address computational and practical challenges, we describe the simplistic curb space allocation heuristic 1 (CSAH1) and curb space allocation heuristic 2 (CSAH2). To this end, we consider $\omega \in \Omega$ iterations. Let $A_z^\omega, z \in \{p, pd, lu\}$ be the capacity of node k at iteration t. We further define $\Delta(F)$ to represent the change in the objective function as follows:

$$\Delta\left(F\left(A_p, A_{z'}, A_z\right)\right) = F\left(A_p, A_{z'}, A_z\right) - F\left(A_p - 1, A_{z'} + 1, A_z\right) \quad \forall z \in \{pd, lu\}, z' \in \{pd, lu\}\backslash z \qquad (3.7)$$

Then, we define the curb space allocation heuristic 1 as follows:

Algorithm 1 Curb Space Allocation Heuristic (CSAH1)

$\omega = 0, A_p \leftarrow M_A, A_{pd} \leftarrow 0, A_{lu} \leftarrow 0$
while $\sum_{z\in\{p,pd,lu\}} A_z^\omega \leq M_A$ **do**
  Calculate $\Delta\left(F\left(A_p^\omega, A_{z'}^\omega, A_z^\omega\right)\right) \quad \forall z \in \{pd, lu\}, z' \in \{pd, lu\}\backslash z$
  if $\Delta\left(F\left(A_p^\omega, A_{z'}^\omega, A_z^\omega\right)\right) \leq 0, \quad \forall z \in \{pd, lu\}, z' \in \{pd, lu\}\backslash z$ **then**
    break
  end if
  $z^* = argmax_{z'}\Delta\left(F\left(A_p^\omega, A_{z'}^\omega, A_z^\omega\right)\right), \forall z \in \{pd, lu\}, z' \in \{pd, lu\}\backslash z$
  $A_p^{\omega+1} \leftarrow A_p^\omega - 1, A_{z^*}^{\omega+1} \leftarrow A_{z^*}^\omega + 1,$
  while $\Delta\left(F\left(A_p^\omega, A_{z^*}^\omega, A_z^\omega\right)\right) > 0, \; \forall z \in \{pd, lu\}\backslash z^*$ **do**
    $A_p^{\omega+1} \leftarrow A_p^\omega - 1, A_{z^*}^{\omega+1} \leftarrow A_{z^*}^\omega + 1$
  end while
  $\omega \leftarrow \omega + 1$
end while
$A_z \leftarrow A_z^\omega$

In the curb space allocation heuristic1 (CSAH1), we assume all the current curb space are assigned to parking use. Then we calculate $\Delta\left(F\left(A_p, A_{z'}, A_z\right)\right)$ at each step and find the value of increasing the capacity of curb use type $z' \in \{pd, lu\}$ by one. We increase the allocated capacity of curb use type with the highest gain. We continue increasing the capacity by one for the same curb use till the increasing does not provide the sufficient gain (i.e., $\Delta\left(F\left(A_p^\omega, A_{z^*}^\omega, A_z^\omega\right)\right) > 0$). The algorithm stops when the allocated capacity reaches the available capacity or when adding one more capacity for all appointments yields a negative profit gain. In CSAH1, we only consider checking the first derivative in the stopping criteria. We further consider adding more condition to the stopping criteria in CSAH2, where we consider checking both the first derivative and the second derivative.

We define the curb space allocation heuristic 2 as follows:

Algorithm 2 Curb Space Allocation Heuristic (CSAH2)

$\omega = 0, A_p \leftarrow M_A, A_{pd} \leftarrow 0, A_{lu} \leftarrow 0$

while $\sum_{z \in \{p,pd,lu\}} A_z^{\omega} \leq M_A$ **do**

Calculate $\Delta\left(F\left(A_p^{\omega}, A_{z'}^{\omega}, A_z^{\omega}\right)\right) \quad \forall z \in \{pd, lu\}, z' \in \{pd, lu\} \backslash z$

if $\Delta\left(F\left(A_p^{\omega}, A_{z'}^{\omega}, A_z^{\omega}\right)\right) \leq 0, \quad \forall z \in \{pd, lu\}, z' \in \{pd, lu\} \backslash z$ **then**

break

end if

$z^* = argmax_{z'} \Delta\left(F\left(A_p^{\omega}, A_{z'}^{\omega}, A_z^{\omega}\right)\right), \forall z \in \{pd, lu\}, z' \in \{pd, lu\} \backslash z$

$A_p^{\omega+1} \leftarrow A_p^{\omega} - 1, A_{z^*}^{\omega+1} \leftarrow A_{z^*}^{\omega} + 1,$

while $\Delta\left(F\left(A_p^{\omega}, A_{z^*}^{\omega}, A_z^{\omega}\right)\right) \leq \Delta\left(F\left(A_p^{\omega} - 1, A_{z^*}^{\omega} + 1, A_z^{\omega}\right)\right), \quad \forall z \in \{pd, lu\} \backslash z^*$ **do**

$A_p^{\omega+1} \leftarrow A_p^{\omega} - 1, A_{z^*}^{\omega+1} \leftarrow A_{z^*}^{\omega} + 1$

end while

$\omega \leftarrow \omega + 1$

end while

$A_z \leftarrow A_z^{\omega}$

CSAH2 is similar to CSAH1, but unlike curb space allocation heuristic 1 where we only check the first derivative, we review both the first derivative and the second derivative while updating the capacity in CSAH2. Thus, we continue increasing the capacity by one for the same curb use till the increasing does not provide the sufficient gain (i.e., $\Delta\left(F\left(A_p^{\omega}, A_{z^*}^{\omega}, A_z^{\omega}\right)\right) \leq \Delta\left(F\left(A_p^{\omega} - 1, A_{z^*}^{\omega} + 1, A_z^{\omega}\right)\right)$). The algorithm stops when the allocated capacity reaches the available capacity or when adding one more capacity for all appointments yields a negative profit gain.

3.3 Numerical Analysis

This section comprises three main parts. First, we describe our parameter setting and present the results of the optimization model for diverse scenario for the small case. Second, we describe our parameter setting and present the results of the optimization model for diverse scenario for the Detroit case. Third, we further validate the efficiency of the proposed heuristics using different starting points.

The unit profit from the parked vehicles and unit cost of traffic delay are given before the optimization of the dynamic curb space allocation decision. On-street parking fees vary depending on the region. For example, the on-street parking fee in Detroit ranges between \$1/h and \$2/h (ParkDetroit.us, 2019). However, the selected traffic network has the same parking fee which is \$2/h. Hence, we use a fixed parking rate (i.e., \$0.025/min) in the optimization model. We further use \$0.217/min as a delay cost, which is defined and described in detail in the "INRIX Global Traffic Scorecard" (Cookson & Pishue, 2018).

3.3.1 Parameter Setting and Optimized Results for the Small Case

In this section, we investigate how curb space spots should be allocated among different uses. Considering the scale of the considered urban network, it is not tractable to solve the large-scale setting optimally for varying settings. Hence, in order to examine the efficiency of the proposed algorithm according to the optimal solution, we first consider a small setting. In the small setting, we consider a small urban network with a total network length of 1 km and total simulation time of three hours. Since the network is small, we further adjust the demand in the network and consider two different values for the demand (i.e., 600 and 800). We summarize the parameters that are different from the real setting in Table 3.1. Similar to the sensitivity part, $\alpha + \beta + \gamma = 1$ yields 4 combinations. We have two different values for δ and traffic demand which also yields 4 combinations. We notice that we consider a flexible curb space allocation policy that the total time length will be split into 4 epochs. Thus, in total, we have 64 instances with different parameter settings for the small case. The parameter values of the small case are listed in Table 3.1.

Table 3.1 Parameters used in the optimization model for small case

Notation	Definition	Unit	Value
L	Total network length	km	1
$\mathcal{T}$	Total time length	hour	3
$Demand$	Traffic demand entering the network.	veh	600; 800
δ	Traffic demand entering the network and go through the network.	veh	0.5; 0.7
α	Traffic demand entering the network and headed to P.	veh	0.6; 0.8
β	Traffic demand entering the network and headed to PD.	veh	0.1; 0.2; 0.3
γ	Traffic demand entering the network and headed to LU.	veh	0.1; 0.2; 0.3
v	The free flow speed of network (with traffic flow)	km/h	30
CT	Cruising time	min	5

We first solve the optimal curb space allocation for all instances of the small case using a nonlinear solver (Scipy package in Python) and the proposed algorithm. We note that we also enumerate the potential solutions to find the optimal solution. Since the nonlinear solver gets the optimal solution faster than the enumeration, we used nonlinear results in the comparison. In Table

3.2, we present the average process time of the algorithms and the average percent objective gap. We calculate the percent objective difference between different algorithms by using the following formula:

$$\text{Percent Objective ap} = \frac{\text{Objective Value of the Optimal Solution} - \text{Objective Value of the CSAH1/CASH2}}{\text{Objective Value of the Optimal Solution}} \quad (3.8)$$

We note that we do not limit the run time of the nonlinear solver for the small case and we output the optimal curb space allocation decision. The comparison results for the small-scale setting are shown in Table 3.2. According to our results, the proposed algorithm is ten times faster than the NLS solution for both CSAH1 and CSAH2. The percent objective gap between the NLS solution is -0.59% for both CSAH1 and CSAH2, which indicates that the proposed algorithms have good performance for small-scale settings. The CSAH1 and CSAH2 can reach a better solution more time-efficient compared to the NLS.

Table 3.2 Comparison of the proposed and the optimal solutions for the small case

	Optimal	CSAH1	CSAH2
Average process time (min.)	38.86	3.33	3.52
Average % objective gap	-	0.59%	0.59%

3.3.2 Parameter Setting and Results for the Detroit Case

As a next step, to examine the efficiency of the proposed algorithms, we also include a Detroit case. We consider the large-scale setting defined in Table 3.3. Same as the small case, we have 64 instances for the Detroit case. For all instances defined, we compare the solution of the nonlinear solver (NLS) with the proposed algorithm solution. It takes a long time to solve the problem (usually more than three hours) using the existing nonlinear solver. Thus, we use the proposed algorithms to obtain a near-optimal solution within half an hour quickly.

Table 3.3 Parameters used in the optimization model for the Detroit case

Notation	Definition	Unit	Value
L	Total network length	km	11.6
$\mathcal{T}$	Total time length	hour	6
$Demand$	Traffic demand entering the network.	veh	4500; 6000
δ	Traffic demand entering the network and go through the network.	veh	0.5; 0.7
α	Traffic demand entering the network and headed to P.	veh	0.6; 0.8
β	Traffic demand entering the network and headed to PD.	veh	0.1; 0.2; 0.3
γ	Traffic demand entering the network and headed to LU.	veh	0.1; 0.2; 0.3
v	The free flow speed of network (with traffic flow)	km/h	30
CT	Cruising time	min	10

We note that we limit the run time of the nonlinear solver to around one hour for each instance and report the best results obtained. The comparison results for the large-scale setting are shown in Table 3.4. According to our results, the CSAH2 is three times faster than the NLS solution, the CSAH1 is two times faster than the NLS solution. The percent objective gap between the NLS solution is 1.18% and 5.47% for CSAH2 and CSAH1 respectively, which indicates that the CSAH2 has better performance for large-scale settings than CSAH1. The CSAH2 can reach a better solution more time-efficient compared to the NLS.

Table 3.4 Comparison of the proposed and the optimal solutions for the large-scale setting

	Optimal	CSAH1	CSAH2
Average process time (min.)	63.29	32.22	18.99
Average % objective gap	-	5.27%	1.18%

3.3.3 Other Optimization Results

We further investigate the average traffic delay that is obtained by using the NLS and the CSAH1 and CSAH2 for the large-scale setting. In Table 3.5, we compare the average vehicle delay time in different system events for the NLS, the CSAH1, the CSAH2, and a fixed-allocation policy (FAP), which is discussed in Chapter 2 (i.e., 70% for parking, 20% for pick-up/drop-off, and 10% for loading/unloading). For the NLS, the CSAH1, and the CSAH2, we consider both the static and the dynamic curb space allocation policies in comparison. We calculate the percent change in the average traffic delay with respect to FAP and use the following equation for calculation:

$$\text{Percent Objective ap} = \frac{\text{Vehicle Delay of the Proposed Policy} - \text{Vehicle Delay of the FAP}}{\text{Vehicle Delay of the Proposed Policy}} \quad (3.9)$$

As shown in Table 3.5, the total average delay time per vehicle in the FAP is greater than the NLS, CSAH1, and CSAH2. The NLS results in lower traffic delay in all system events compared to the FAP. The CSAH1 and CSAH2 both yields lower traffic delay than the FAP in some system events. In addition, we see that the dynamic curb space allocation policy yields lower average traffic delay per vehicle than the static curb space allocation policy for NLS, CSAH1, and CSAH2.

To analyze the benefit of the dynamic allocation policy with respect to the static allocation policy, we also compare the average objective function values over all instances in Table 3.6. We find that the dynamic allocation policy yields higher profit than the static policy by more than 20% for NLS, CSAH1, and CSAH2 on average, indicating the benefit of the dynamic curb space allocation policy.

Table 3.5 Comparison of average vehicle delay time

Policies	Non-searching state (min.)	Searching for P state (min.)	Searching for PD state (min.)	Searching for LU state (min.)	Total (min.)	Percent gap from FAP
FAP	50.38	126.66	38.79	49.04	264.86	-
Optimal-Static	22.90	95.48	21.45	23.93	163.76	-61.74%
Optimal-Dynamic	22.34	94.40	20.93	24.12	161.79	-63.71%
CSAH1-Static	27.68	92.78	60.30	26.57	207.34	-27.75%
CSAH1-Dynamic	26.85	92.85	45.22	25.82	190.73	-38.87%
CSAH2-Static	24.40	88.56	50.89	60.50	224.35	-18.06%
CSAH2-Dynamic	23.74	88.41	45.26	54.10	211.52	-25.22%

3.2.4 Proposed Algorithm Starting Point Validation

In previous section, we proposed algorithms to solve the problem efficiently using a x = 260, y = 0, z = 0 (x is parking spot, y is PD spots, and z is LU spots) starting point. The reason for selecting this starting point is that we can infer vehicles that heading for parking is the most based on the provided parameters (in real-world networks, we usually have the most vehicles seeking a parking spot compared with *PD* and *LU*). Thus, it is reasonable to select x = 260, y = 0, z = 0 as the starting point for both nonlinear solver and the proposed algorithms. However, we also prove the efficiency of the proposed algorithm using different starting points, when the most traffic heading for *PD* and when the most traffic is heading for *LU*.

In this section, we randomly select 8 instances out of the 64 instances from Chapter 3.3.2 to do the validation. Recall that we assume at least 60% of traffic is heading for parking in previous sections. Thus, we assume at least 60% is heading for *PD* or *LU* while we are using $x = 0$, $y = 260$, z = 0 as the starting point and x = 0, y = 0, z = 260 as the starting point respectively.

Table 3.6 Comparison of objective value

	Nonlinear Objective			CSAH1 Objective			CSAH2 Objective		
Instance	**Static**	**Dynamic**	**%gap from static**	**Static**	**Dynamic**	**%gap from static**	**Static**	**Dynamic**	**%gap from static**
1	27725.15	21808.96	-27.13%	98621.51	93048.63	-5.99%	26911.56	22383.16	-20.23%
2	126305.21	125117.64	-0.95%	135340.83	132955.97	-1.79%	142682.19	131521.52	-8.49%
3	26700.69	21904.61	-21.90%	26810.62	21921.17	-22.30%	26831.03	21975.80	-22.09%
4	122348.67	121634.45	-0.59%	126350.87	125280.89	-0.85%	135975.23	132273.04	-2.80%
5	26148.11	22810.09	-14.63%	26736.61	22184.65	-20.52%	26275.35	21365.69	-22.98%
6	119836.88	117488.48	-2.00%	120767.64	117980.74	-2.36%	118363.63	125472.69	5.67%
7	9022.24	6271.51	-43.86%	9025.84	6246.43	-44.50%	9040.10	6246.76	-44.72%
8	16055.15	12152.60	-32.11%	16110.75	12183.18	-32.24%	16194.80	12276.40	-31.92%
9	9156.22	6380.95	-43.49%	9101.01	6317.88	-44.05%	9134.96	6376.33	-43.26%
10	15994.82	12103.54	-32.15%	15960.27	12139.90	-31.47%	15978.67	12170.38	-31.29%
11	9343.97	6570.47	-42.21%	9231.60	6438.21	-43.39%	9263.96	6472.63	-43.13%
12	15962.87	12313.72	-29.63%	15829.32	12063.72	-31.21%	15975.74	12282.07	-30.07%
13	94633.79	92885.79	-1.88%	94899.87	93111.19	-1.92%	95168.51	93364.28	-1.93%
14	142442.13	141200.34	-0.88%	145669.25	141300.21	-3.09%	147893.42	145370.64	-1.74%
15	12888.04	9347.77	-37.87%	12917.93	9379.90	-37.72%	12917.93	9404.64	-37.36%
16	26157.54	20241.65	-29.23%	25112.18	20340.56	-23.46%	25260.82	20576.94	-22.76%
		Average	**-22.53%**		Average	**-21.68%**		Average	**-22.44%**

In Table 3.7, we present the starting point validation result for the small case. The percentage gap from NLS is 0.59% for both CSAH1 and CSAH2 while validating $x = 260$, $y = 0$, $z = 0$ starting point, however, their CPU time is10X faster than NLS. While validating $x = 0$, $y = 260$, $z = 0$ starting point, both CSAH1 and CSAH2 outperform NLS in CPU time and solution efficiency. While validating $x = 0$, $y = 0$, $z = 260$ starting point, it is obvious that CSAH1 has a better solution than CSAH2 in solution efficiency, but the solving speed of CSAH2 is 8 times faster than CSAH1.

Table 3.7 Starting point validation for the small case

	Nonlinear	CSAH1			CSAH2		
Starting point	CPU time (seconds)	CPU time (seconds)	%time saving	%gap from nonlinear	CPU time (seconds)	%time saving	%gap from nonlinear
(260, 0, 0)	2331.6	199.8	-1066.97%	0.59%	201	-1060.00%	0.59%
(0, 260, 0)	1660.25	544.75	-204.77%	-0.86%	797.38	-108.21%	-4.48%
(0, 0, 260)	1491.75	797.38	-87.08%	-4.48%	189.63	-686.68%	3.25%

In Table 3.8, we present the starting point validation result for the Detroit case. As we can see, a significant drop in CPU time for the proposed algorithms for starting point $x = 0$, $y = 260$, $z = 0$ and $x = 0$, $y = 0$, $z = 260$ compared with NLS. CSAH2 yields a better solution with a 2.3 times faster solving speed than NLS while validating $x = 260$, $y = 0$, $z = 0$ starting point. Both CSAH1 and CSAH2 perform slightly better than NLS by -0.39%, but CSAH1 is more efficient in CPU time while validating $x = 0$, $y = 260$, $z = 0$ starting point. CSAH1 performs better in CPU time and percentage gap from NLS while validating $x = 0$, $y = 0$, $z = 260$ starting point. Therefore, we conclude that the proposed algorithms are robust in both CPU time and solution quality in solving the small case and real-world case.

Table 3.8 Starting point validation for the Detroit case

	Nonlinear	CSAH1			CSAH2		
Starting point	CPU time (seconds)	CPU time (seconds)	%time saving	%gap from nonlinear	CPU time (seconds)	%time saving	%gap from nonlinear
(260, 0, 0)	3797.4	1933.2	-96.43%	5.27%	1139.4	-233.28%	1.18%
(0, 260, 0)	3695.4	1981.88	-86.46%	-0.39%	2958.13	-24.92%	-0.39%
(0, 0, 260)	2887.38	2283.13	-26.47%	-0.39%	3571.75	19.16%	-0.38%

Chapter 4: Conclusion and Discussion

Curb space management for different uses is essential for smooth traffic, especially during rush hours in urban areas. A dynamic curb space allocation for different usages ensures a flexible, less costly traffic compared with fixing the curb space for parking use only or fixing the different usages of curb space all the time. The numerical results demonstrate that the dynamic curb space allocation outperforms the static curb space allocation in both small urban network settings and large urban network settings. This dynamic curb space allocation policy also contributes to reducing the traffic delay caused by those vehicles that are heading for PD/LU spots. Moreover, the demand for a smart city planning that includes an intelligent curb space planning is growing as the development of ride-sharing and autonomous vehicles. Thus, it is necessary to utilize the curb space for multiple uses and apply the dynamic curb space allocation along with time changes.

In this study, we build a transportation system simulation model to analyze the interaction between traffic delay and other parameters (i.e., vehicle speed, traffic demand, cruising time, etc.). We also observe how the traffic delay changes when we assign all curb space for parking use and assign the curb space for P, PD, and LU uses. Then we use the simulation outputs as inputs to build an optimization model that maximizes the total profit (maximal parking revenue and minimal traffic delay). Since it overall takes more than three hours for existing nonlinear solvers to obtain the optimal solutions, thus, we proposed algorithms to obtain a near-optimal solution efficiently. We compare its performance with the nonlinear solver's. It shows that the proposed algorithm is a more practical procedure that outperforms the existing nonlinear solver in both CPU time-saving and percentage gap from the nonlinear solver. The proposed algorithms require much less computational efforts but yield an even better result than nonlinear solver. We further show that the proposed algorithms are an efficient way to solve real-world instances in a reasonable time. While there is more traffic heading for parking than PD/LU, Algorithm 1 takes about 32 minutes to solve the Detroit case with a 5.27% gap different from the nonlinear solver's solution while Algorithm 2 takes about 19 minutes to solve the Detroit case with a1.18% gap different from the nonlinear solver's solution on average. It is indicating that Algorithm 2 is a more efficient approach to solve real-world instances.

As part of future research, first, we can let a proportion of vehicles that are searching for curb space leave the network after they have spent a specific time in the searching state. Second, the solutions given using the optimization model does not indicate which exact spot should be assigned for which use, it only provides an overview of how to efficiently allocate the curb space for different uses that reduce the traffic delay. Thus, we can extend the existing models or build a new model to determine which spots are assigned for which use in real-world implementations. As a third extension, we can include the parking rate as a variable in the model instead of a known value, such that the parking rate also changes along with time changes.

Appendix

Table A1: Solution details of the nonlinear solver

Instances	Objective value	Parking spot	PD spot	LU spot	Optmized time slot	Total demand	Proportion	CPU time
1	27725.15	210.55	40.50	8.96	(0, 360)	4500	(0.6, 0.3, 0.1, 0.5)	3675
2	6164.01	205.56	45.23	9.21	(0, 120)	4500	(0.6, 0.3, 0.1, 0.5)	3903
3	14783.98	210.19	39.52	11.21	(120, 240)	4500	(0.6, 0.3, 0.1, 0.5)	3812
4	860.98	206.84	41.09	12.07	(240, 360)	4500	(0.6, 0.3, 0.1, 0.5)	3793
5	126305.21	184.86	62.57	12.57	(0, 360)	6000	(0.6, 0.3, 0.1, 0.5)	3956
6	12558.28	179.19	58.23	22.58	(0, 120)	6000	(0.6, 0.3, 0.1, 0.5)	3955
7	52532.53	186.08	61.82	12.10	(120, 240)	6000	(0.6, 0.3, 0.1, 0.5)	3702
8	60026.83	186.96	59.49	13.54	(240, 360)	6000	(0.6, 0.3, 0.1, 0.5)	4002
9	26700.69	214.90	26.36	18.73	(0, 360)	4500	(0.6, 0.2, 0.2, 0.5)	3769
10	5987.50	209.49	32.35	18.17	(0, 120)	4500	(0.6, 0.2, 0.2, 0.5)	3769
11	14534.66	215.84	26.70	17.46	(120, 240)	4500	(0.6, 0.2, 0.2, 0.5)	3784
12	1382.45	217.07	23.50	19.43	(240, 360)	4500	(0.6, 0.2, 0.2, 0.5)	3878
13	122348.67	193.97	40.97	25.06	(0, 360)	6000	(0.6, 0.2, 0.2, 0.5)	3946
14	11867.76	194.03	41.65	24.33	(0, 120)	6000	(0.6, 0.2, 0.2, 0.5)	3988
15	50530.65	194.43	33.98	31.59	(120, 240)	6000	(0.6, 0.2, 0.2, 0.5)	3676
16	59236.03	193.45	41.24	25.31	(240, 360)	6000	(0.6, 0.2, 0.2, 0.5)	3681
17	26148.11	218.33	13.62	28.04	(0, 360)	4500	(0.6, 0.1, 0.3, 0.5)	3693
18	5838.16	216.49	16.28	27.23	(0, 120)	4500	(0.6, 0.1, 0.3, 0.5)	3822
19	14295.10	218.19	15.06	26.74	(120, 240)	4500	(0.6, 0.1, 0.3, 0.5)	3828
20	2676.83	201.78	32.87	25.35	(240, 360)	4500	(0.6, 0.1, 0.3, 0.5)	3806
21	119836.88	191.55	28.52	39.93	(0, 360)	6000	(0.6, 0.1, 0.3, 0.5)	4000
22	11392.02	200.99	22.90	36.10	(0, 120)	6000	(0.6, 0.1, 0.3, 0.5)	3942
23	47377.77	201.52	18.70	39.78	(120, 240)	6000	(0.6, 0.1, 0.3, 0.5)	3995
24	58718.69	198.76	19.95	41.28	(240, 360)	6000	(0.6, 0.1, 0.3, 0.5)	4010
25	9022.24	226.75	27.57	5.68	(0, 360)	4500	(0.6, 0.3, 0.1, 0.7)	3810
26	3541.32	224.96	29.56	5.48	(0, 120)	4500	(0.6, 0.3, 0.1, 0.7)	3822
27	3533.52	227.85	26.66	5.48	(120, 240)	4500	(0.6, 0.3, 0.1, 0.7)	3801
28	-803.33	226.27	28.12	5.61	(240, 360)	4500	(0.6, 0.3, 0.1, 0.7)	3690
29	16055.15	218.94	33.77	7.30	(0, 360)	6000	(0.6, 0.3, 0.1, 0.7)	3882
30	4954.62	213.52	39.19	7.29	(0, 120)	6000	(0.6, 0.3, 0.1, 0.7)	3759
31	8135.59	219.46	33.27	7.26	(120, 240)	6000	(0.6, 0.3, 0.1, 0.7)	3811
32	-937.61	231.11	20.62	8.27	(240, 360)	6000	(0.6, 0.3, 0.1, 0.7)	3739
33	9156.22	227.09	21.92	10.98	(0, 360)	4500	(0.6, 0.2, 0.2, 0.7)	3685
34	3482.69	228.38	20.70	10.92	(0, 120)	4500	(0.6, 0.2, 0.2, 0.7)	3794
35	3528.11	231.03	18.04	10.93	(120, 240)	4500	(0.6, 0.2, 0.2, 0.7)	3897
36	-629.85	217.46	31.45	11.08	(240, 360)	4500	(0.6, 0.2, 0.2, 0.7)	3694
37	15994.82	220.31	25.12	14.57	(0, 360)	6000	(0.6, 0.2, 0.2, 0.7)	3736
38	4870.71	219.21	26.22	14.57	(0, 120)	6000	(0.6, 0.2, 0.2, 0.7)	3669
39	7960.68	223.58	21.85	14.57	(120, 240)	6000	(0.6, 0.2, 0.2, 0.7)	3718
40	-727.85	231.62	13.72	14.67	(240, 360)	6000	(0.6, 0.2, 0.2, 0.7)	3845
41	9343.97	227.41	16.04	16.55	(0, 360)	4500	(0.6, 0.1, 0.3, 0.7)	3682
42	3448.34	220.71	22.55	16.74	(0, 120)	4500	(0.6, 0.1, 0.3, 0.7)	3659
43	3588.46	231.17	12.44	16.39	(120, 240)	4500	(0.6, 0.1, 0.3, 0.7)	3652
44	-466.34	226.99	16.09	16.92	(240, 360)	4500	(0.6, 0.1, 0.3, 0.7)	3664
45	15962.87	223.74	14.40	21.85	(0, 360)	6000	(0.6, 0.1, 0.3, 0.7)	3694
46	4782.96	223.43	13.25	23.32	(0, 120)	6000	(0.6, 0.1, 0.3, 0.7)	3638
47	7800.76	227.06	11.09	21.85	(120, 240)	6000	(0.6, 0.1, 0.3, 0.7)	3706
48	-269.99	215.49	22.65	21.86	(240, 360)	6000	(0.6, 0.1, 0.3, 0.7)	3850
49	94633.79	235.03	15.94	9.04	(0, 360)	4500	(0.8, 0.1, 0.1, 0.5)	3826
50	9601.32	235.39	14.83	9.79	(0, 120)	4500	(0.8, 0.1, 0.1, 0.5)	3861
51	38847.26	235.56	15.27	9.18	(120, 240)	4500	(0.8, 0.1, 0.1, 0.5)	4050
52	44437.22	235.54	15.25	9.22	(240, 360)	4500	(0.8, 0.1, 0.1, 0.5)	3656
53	142442.13	210.83	34.45	14.72	(0, 360)	6000	(0.8, 0.1, 0.1, 0.5)	3811
54	15020.77	210.90	36.44	12.66	(0, 120)	6000	(0.8, 0.1, 0.1, 0.5)	3752
55	59429.54	208.91	39.22	11.87	(120, 240)	6000	(0.8, 0.1, 0.1, 0.5)	3875
56	66750.03	229.64	18.43	11.93	(240, 360)	6000	(0.8, 0.1, 0.1, 0.5)	3861
57	12888.04	245.42	9.09	5.48	(0, 360)	4500	(0.8, 0.1, 0.1, 0.7)	3842
58	4112.77	244.22	10.27	5.50	(0, 120)	4500	(0.8, 0.1, 0.1, 0.7)	3733
59	6284.12	245.99	8.54	5.47	(120, 240)	4500	(0.8, 0.1, 0.1, 0.7)	3800
60	-1049.12	248.97	5.41	5.62	(240, 360)	4500	(0.8, 0.1, 0.1, 0.7)	3727
61	26157.54	221.39	24.85	13.75	(0, 360)	6000	(0.8, 0.1, 0.1, 0.7)	3499
62	5877.42	239.59	13.08	7.33	(0, 120)	6000	(0.8, 0.1, 0.1, 0.7)	3850
63	14019.80	243.00	10.32	6.68	(120, 240)	6000	(0.8, 0.1, 0.1, 0.7)	3769
64	344.43	244.71	6.81	8.48	(240, 360)	6000	(0.8, 0.1, 0.1, 0.7)	3852

Table A2: Solution details of CSAH2

Instances	Objective value	Parking spot	PD spot	LU spot	Optmized time slot	Total demand	Proportion	CPU time
1	26911.56	206	42	12	(0, 360)	4500	(0.6, 0.3, 0.1, 0.5)	1529
2	6167.58	206	45	9	(0, 120)	4500	(0.6, 0.3, 0.1, 0.5)	2054
3	15140.26	206	42	12	(120, 240)	4500	(0.6, 0.3, 0.1, 0.5)	1482
4	1075.31	206	42	12	(240, 360)	4500	(0.6, 0.3, 0.1, 0.5)	1164
5	142682.19	239	18	3	(0, 360)	6000	(0.6, 0.3, 0.1, 0.5)	740
6	12407.85	185	60	15	(0, 120)	6000	(0.6, 0.3, 0.1, 0.5)	1795
7	55853.63	218	36	6	(120, 240)	6000	(0.6, 0.3, 0.1, 0.5)	1209
8	63260.04	218	36	6	(240, 360)	6000	(0.6, 0.3, 0.1, 0.5)	1212
9	26831.03	215	27	18	(0, 360)	4500	(0.6, 0.2, 0.2, 0.5)	1382
10	5990.93	209	33	18	(0, 120)	4500	(0.6, 0.2, 0.2, 0.5)	1792
11	14545.25	215	27	18	(120, 240)	4500	(0.6, 0.2, 0.2, 0.5)	1433
12	1439.62	215	24	21	(240, 360)	4500	(0.6, 0.2, 0.2, 0.5)	1484
13	135975.23	236	18	6	(0, 360)	6000	(0.6, 0.2, 0.2, 0.5)	1157
14	11895.61	194	39	27	(0, 120)	6000	(0.6, 0.2, 0.2, 0.5)	2298
15	55526.58	236	18	6	(120, 240)	6000	(0.6, 0.2, 0.2, 0.5)	1129
16	64850.85	236	18	6	(240, 360)	6000	(0.6, 0.2, 0.2, 0.5)	1018
17	26275.35	221	12	27	(0, 360)	4500	(0.6, 0.1, 0.3, 0.5)	1172
18	5850.01	215	18	27	(0, 120)	4500	(0.6, 0.1, 0.3, 0.5)	1749
19	14235.91	221	12	27	(120, 240)	4500	(0.6, 0.1, 0.3, 0.5)	1218
20	1279.77	221	9	30	(240, 360)	4500	(0.6, 0.1, 0.3, 0.5)	1208
21	118363.63	206	18	36	(0, 360)	6000	(0.6, 0.1, 0.3, 0.5)	2129
22	11340.99	206	18	36	(0, 120)	6000	(0.6, 0.1, 0.3, 0.5)	1707
23	51421.35	233	9	18	(120, 240)	6000	(0.6, 0.1, 0.3, 0.5)	956
24	62710.35	233	9	18	(240, 360)	6000	(0.6, 0.1, 0.3, 0.5)	1252
25	9040.10	227	27	6	(0, 360)	4500	(0.6, 0.3, 0.1, 0.7)	1142
26	3542.12	224	30	6	(0, 120)	4500	(0.6, 0.3, 0.1, 0.7)	1269
27	3544.74	227	27	6	(120, 240)	4500	(0.6, 0.3, 0.1, 0.7)	987
28	-840.10	239	15	6	(240, 360)	4500	(0.6, 0.3, 0.1, 0.7)	676
29	16194.80	218	33	9	(0, 360)	6000	(0.6, 0.3, 0.1, 0.7)	1231
30	4960.54	212	39	9	(0, 120)	6000	(0.6, 0.3, 0.1, 0.7)	1371
31	8235.44	218	33	9	(120, 240)	6000	(0.6, 0.3, 0.1, 0.7)	1277
32	-919.58	230	21	9	(240, 360)	6000	(0.6, 0.3, 0.1, 0.7)	820
33	9134.96	230	18	12	(0, 360)	4500	(0.6, 0.2, 0.2, 0.7)	1022
34	3484.29	227	21	12	(0, 120)	4500	(0.6, 0.2, 0.2, 0.7)	1202
35	3548.42	230	18	12	(120, 240)	4500	(0.6, 0.2, 0.2, 0.7)	875
36	-656.39	236	12	12	(240, 360)	4500	(0.6, 0.2, 0.2, 0.7)	885
37	15978.67	221	24	15	(0, 360)	6000	(0.6, 0.2, 0.2, 0.7)	1322
38	4875.31	218	27	15	(0, 120)	6000	(0.6, 0.2, 0.2, 0.7)	1443
39	7994.74	224	21	15	(120, 240)	6000	(0.6, 0.2, 0.2, 0.7)	983
40	-699.67	230	15	15	(240, 360)	6000	(0.6, 0.2, 0.2, 0.7)	840
41	9263.96	233	9	18	(0, 360)	4500	(0.6, 0.1, 0.3, 0.7)	905
42	3440.48	230	12	18	(0, 120)	4500	(0.6, 0.1, 0.3, 0.7)	1070
43	3561.05	233	9	18	(120, 240)	4500	(0.6, 0.1, 0.3, 0.7)	941
44	-528.91	236	6	18	(240, 360)	4500	(0.6, 0.1, 0.3, 0.7)	850
45	15975.74	224	12	24	(0, 360)	6000	(0.6, 0.1, 0.3, 0.7)	1221
46	4791.45	221	15	24	(0, 120)	6000	(0.6, 0.1, 0.3, 0.7)	1361
47	7943.79	224	12	24	(120, 240)	6000	(0.6, 0.1, 0.3, 0.7)	1188
48	-453.17	230	6	24	(240, 360)	6000	(0.6, 0.1, 0.3, 0.7)	933
49	95168.51	239	12	9	(0, 360)	4500	(0.8, 0.1, 0.1, 0.5)	1057
50	9619.98	233	18	9	(0, 120)	4500	(0.8, 0.1, 0.1, 0.5)	1535
51	39079.10	239	12	9	(120, 240)	4500	(0.8, 0.1, 0.1, 0.5)	1098
52	44665.19	239	12	9	(240, 360)	4500	(0.8, 0.1, 0.1, 0.5)	1010
53	147893.42	254	3	3	(0, 360)	6000	(0.8, 0.1, 0.1, 0.5)	604
54	14947.28	236	12	12	(0, 120)	6000	(0.8, 0.1, 0.1, 0.5)	1073
55	61543.99	254	3	3	(120, 240)	6000	(0.8, 0.1, 0.1, 0.5)	597
56	68879.36	254	3	3	(240, 360)	6000	(0.8, 0.1, 0.1, 0.5)	609
57	12917.93	245	9	6	(0, 360)	4500	(0.8, 0.1, 0.1, 0.7)	693
58	4118.03	242	12	6	(0, 120)	4500	(0.8, 0.1, 0.1, 0.7)	761
59	6315.42	245	9	6	(120, 240)	4500	(0.8, 0.1, 0.1, 0.7)	600
60	-1028.80	248	6	6	(240, 360)	4500	(0.8, 0.1, 0.1, 0.7)	520
61	25260.82	242	9	9	(0, 360)	6000	(0.8, 0.1, 0.1, 0.7)	609
62	5897.38	236	15	9	(0, 120)	6000	(0.8, 0.1, 0.1, 0.7)	889
63	14172.11	242	9	9	(120, 240)	6000	(0.8, 0.1, 0.1, 0.7)	605
64	507.44	242	9	9	(240, 360)	6000	(0.8, 0.1, 0.1, 0.7)	612

References

Adiv, A. & Wang, W., 1987. On-street parking meter behavior. *Transportation Quarterly,* 41(3).

Ajeng, C. & Gim, T.-H. T., 2018. Analyzing on-Street Parking Duration and Demand in a Metropolitan City of a Developing Country: A Case Study of Yogyakarta City, Indonesia. *Sustainability,* 10(3), p. 591.

Alarcón, G. A., Torres, C. F. & Gómez, L. E., 2002. Global optimization of gas allocation to a group of wells in artificial lift using nonlinear constrained programming. *Journal of energy resources technology,* 124(4), pp. 262-268.

Anderson, S. P. & De Palma, A., 2004. The economics of pricing parking. *Journal of Urban Economics,* 55(1), pp. 1-20.

Arnott, R. & Inci, E., 2006. An integrated model of downtown parking and traffic congestion. *Journal of Urban Economics,* 60(3), pp. 418-442.

Arnott, R. & Rowse, J., 1999. Modeling parking. *Journal of urban economics,* 45(1), pp. 97-124.

Arnott, R. & Rowse, J., 2009. Downtown parking in auto city. *Regional Science and Urban Economics,* 39(1), pp. 1-14.

Benenson, I., Martens, K. & Birfir, S., 2008. PARKAGENT: An agent-based model of parking in the city. *Computers, Environment and Urban Systems,* 32(6), pp. 431-439.

Berthon, P. & Guion, C., 1989. *System for controlling metered parking.* s.l.:Google Patents.

Byrd, R. H., Gilbert, J. C. & Nocedal, J., 2000. A trust region method based on interior point techniques for nonlinear programming. *Mathematical programming,* 89(1), pp. 149-185.

Byrd, R. H., Hribar, M. E. & Nocedal, J., 1999. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization,* 9(4), pp. 877-900.

Calthrop, E. & Proost, S., 2006. Regulating on-street parking. *Regional Science and Urban Economics,* 36(1), pp. 29-48.

Cao, J. & Menendez, M., 2015. System dynamics of urban traffic based on its parking-related-states. *Transportation Research Part B: Methodological,* Volume 81, pp. 718-736.

Cao, J., Menendez, M. & Waraich, R., 2017. Impacts of the urban parking system on cruising traffic and policy development: the case of Zurich downtown area, Switzerland. *Transportation,* pp. 1-26.

Chang, T., 2009. On-street parking management and pricing study.

Chen, B. & Cheng, H. H., 2010. A review of the applications of agent technology in traffic and transportation systems. *IEEE Transactions on intelligent transportation systems,* 11(2), pp. 485-497.

Cookson, G. & Pishue, B., 2018. Inrix global traffic scorecard. *Intelligence That Moves the World. United States: INRIX RESEARCH.*

Cullinane, K. & Polak, J., 1992. Illegal parking and the enforcement of parking regulations: causes, effects and interactions. *Transport Reviews,* 12(1), pp. 49-75.

Daganzo, C. & Newell, G., 1995. Methods of analysis for transportation operations. *Institute of Transportation Studies, University of California at Berkeley.*

Downs, A., 2004. *Traffic: Why It's Getting Worse, What Government Can Do,* s.l.: s.n.

Figliozzi, M. & Tipagornwong, C., 2016. *The Impact of Last Mile Parking Availability on Commercial Vehicle Costs.* s.l., s.n., pp. 1-4.

Gallo, M., D'Acierno, L. & Montella, B., 2011. A multilayer model to simulate cruising for parking in urban areas. *Transport policy,* 18(5), pp. 735-744.

Gould, N. I. M. & Leyffer, S., 2003. An introduction to algorithms for nonlinear optimization. In: *Frontiers in numerical analysis.* s.l.:Springer, pp. 109-197.

Han, L. D., Chin, S.-M., Franzese, O. & Hwang, H., 2005. Estimating the impact of pickup-and delivery-related illegal parking activities on traffic. *Transportation Research Record,* 1906(1), pp. 49-55.

Hsiao, M.-H., 2009. Shopping mode choice: Physical store shopping versus e-shopping. *Transportation Research Part E: Logistics and Transportation Review,* 45(1), pp. 86-95.

International Transport Forum, 2017. *The Shared-Use City: Managing the Curb,* s.l.: s.n.

Kelley, C. T., 2003. *Solving nonlinear equations with Newton's method.* s.l.:Siam.

Kleint, J., 2011. *City Speed.* s.l.:s.n.

Lee, J. B., Agdas, D. & Baker, D., 2017. Cruising for parking: New empirical evidence and influential factors on cruising time. *Journal of Transport and Land Use,* 10(1).

Lee, J. T., Ryoo, M. S., Riley, M. & Aggarwal, J. K., 2009. Real-time illegal parking detection in outdoor environments using 1-D transformation. *IEEE Transactions on Circuits and Systems for Video Technology,* 19(7), pp. 1014-1024.

Litman, T., 2009. Transportation cost and benefit analysis. *Victoria Transport Policy Institute,* Volume 31.

Li, Y., Zhang, Y., Kong, N. & Lawley, M., 2016. Capacity planning for long-term care networks. *IIE Transactions,* 48(12), pp. 1098-1111.

Nourinejad, M., Wenneman, A., Habib, K. N. & Roorda, M. J., 2014. Truck parking in urban areas: Application of choice modelling within traffic microsimulation. *Transportation Research Part A: Policy and Practice,* Volume 64, pp. 54-64.

ParkDetroit.us, 2019. *Park Detroit.* s.l.:s.n.

Parkopedia.com, 2019. *Parkopedia.* s.l.:s.n.

Saranow, J., 2006. Carpooling for Grown-Ups—High Gas Prices, New Services Give Ride-Sharing a Boost; Rating Your Fellow Rider. *Wall Street Journal.*

Schaller, B. et al., 2011. *Parking pricing and curbside management in New York City,* s.l.: s.n.

Schelenz, T., Suescun, Á., Wikström, L. & Karlsson, M., 2014. Application of agent based simulation for evaluating a bus layout design from passengers' perspective. *Transportation Research Part C: Emerging Technologies,* Volume 43, pp. 222-229.

Schmid, J., Wang, X. C. & Conway, A., 2018. Commercial vehicle parking duration in New York City and its implications for planning. *Transportation Research Part A: Policy and Practice,* Volume 116, pp. 580-590.

SEMCOG, 2019. *Southeast Michigan Council of Governments.* s.l.:s.n.

Shoup, D., 2017. *The high cost of free parking: Updated edition.* s.l.:Routledge.

Shoup, D. C., 2006. Cruising for parking. *Transport Policy,* 13(6), pp. 479-486.

Vanderbei, R. J. & Shanno, D. F., 1999. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications,* 13(1-3), pp. 231-252.

Wächter, A. & Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming,* 106(1), pp. 25-57.

WSDOT, 2003. *Parking Area Design report.* s.l.:s.n.

Zalewski, A. J., Buckley, S. M. & Weinberger, R. R., 2012. *Regulating curb space: developing a framework to understand and improve curbside management,* s.l.: s.n.