December 2018

# A Microscopic Simulation Laboratory for Evaluation of Off-street Parking Systems

Yun Yuan
*University of Wisconsin-Milwaukee*

# A MICROSCOPIC SIMULATION LABORATORY FOR EVALUATION OF OFF-STREET PARKING SYSTEMS

by

Yun Yuan

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin-Milwaukee

December 2018

# ABSTRACT

# A MICROSCOPIC SIMULATION LABORATORY FOR EVALUATION OF OFF-STREET PARKING SYSTEMS

by

Yun Yuan

The University of Wisconsin-Milwaukee, 2018
Under the Supervision of Professor Yue Liu

The parking industry produces an enormous amount of data every day that, properly analyzed, will change the way the industry operates. The collected data form patterns that, in most cases, would allow parking operators and property owners to better understand how to maximize revenue and decrease operating expenses and support the decisions such as how to set specific parking policies (e.g. electrical charging only parking space) to achieve the sustainable and eco-friendly parking.

However, there lacks an intelligent tool to assess the layout design and operational performance of parking lots to reduce the externalities and increase the revenue. To address this issue, this research presents a comprehensive agent-based framework for microscopic off-street parking system simulation. A rule-based parking simulation logic programming model is formulated. The proposed simulation model can effectively capture the behaviors of drivers and pedestrians as well as spatial and temporal interactions of traffic dynamics in the parking system. A methodology for data collection, processing, and extraction of user behaviors in the parking system is also developed. A Long-Short Term Memory (LSTM) neural network is used to predict the arrival and departure of the vehicles. The proposed simulator is implemented in Java and a Software as a Service (SaaS) graphic user interface is designed to analyze and visualize the simulation results. This study finds the active capacity of the parking system, which is defined as

the largest number of actively moving vehicles in the parking system under the facility layout. In the system application of the real world testbed, the numerical tests show (a) the smart check-in device has marginal benefits in vehicle waiting time; (b) the flexible pricing policy may increase the average daily revenue if the elasticity of the price is not involved; (c) the number of electrical charging only spots has a negative impact on the performance of the parking facility; and (d) the rear-in only policy may increase the duration of parking maneuvers and reduce the efficiency during the arrival rush hour. Application of the developed simulation system using a real-world case demonstrates its capability of providing informative quantitative measures to support decisions in designing, maintaining, and operating smart parking facilities.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ABS | Agent-Based Simulation |
| AIGA | American Institute of Graphic Arts |
| API | Application Programming Interface |
| ARIMA | Auto-Regression Integrated Moving Average |
| ARCH | Auto-Regressive Conditional Heteroskedasticity |
| BJS | Bureau of Justice Statistics |
| CA | Cellular Automaton |
| CART | Classification and Regression Trees |
| CCDC | Capital City Development Corp |
| CNTK | Microsoft Cognitive Toolkit |
| DES | Discrete-state Event-driven Simulation |
| GRU | Gated Recurrent Units |
| GUI | Graphic User Interface |
| ID3 | Iterative Dichotomiser 3 |
| IRNN | Identity Recurrent unit Neural Network |
| JVM | Java Virtual Machine |
| KPI | Key Performance Indicators |
| LTSM | Long-Short Term Memory |
| MPH | Mile Per Hour |
| MCM | Monte Carlo Method |
| MVC | Model-View-Controller |
| NDRC | National Development and Reform Commission |
| NHTSA | National Highway Traffic Safety Administration |
| NSC | National Safety Council |
| OOP | Object Oriented Programming |
| PROLOG | LOGic PROgramming language |
| REST | Representational State Transfer |

| | |
|---|---|
| RNN | Recurrent Neural Network |
| SaaS | Software as a Service |
| SD | System Dynamics |
| SVM | Support Vector Machine |
| SUMO | Simulation of Urban MObility |
| TDM | Transportation Demand Management |
| UML | Unified Modeling Language |
| XNA | Xinhua National Agency |

# ACKNOWLEDGMENTS

Undertaking this Ph.D. has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

I take this opportunity to thank my advisor, Dr. Yue Liu, for his invaluable advice and guidance throughout my doctoral study.

I gratefully acknowledge the funding received towards my Ph.D. through the Dean's Fellowship.

My deepest gratitude goes to my dissertation committee members, Dr. Jie Yu, Dr. Xiao Qin, Dr. Chao Zhu and Dr. Lingqian Hu. Their technical expertise and help were always available when needed.

Thanks to Dr. Matthew Petering, Dr. Christine Cheng, Dr. Tian Zhao, Dr. John Boyland, Dr. Jun Zhang, and Dr. Kishor Kundan for their kindly instructions in building my interdisciplinary knowledge.

Thanks to my colleagues Dr. Xin Li, Dr. Peng Li, Dr. Yaojun Wang, Dahai Han, Xinyu Liu, Weijie Tan and Zihao Jin for their friendly help.

Thanks to my friends Dr. Anni Huang, Tianfang Li, Xiaopeng Min, Tianyao Ding, Xiaowei Hu and Jie Qu for their technical comments and kind support.

Thanks to my master advisors Dr. Yang Liu and Dr. Sihong Zhu.

Thanks to my instructor and friend Elizabeth W. Lewis.

I would like to express special thanks to Liang Wang and Liqi Mao for their invaluable help in the implementation and debugging of the computer programs and Dr. Hao Wang for industrial and professional advices.

Finally, I wish to express my heartfelt thanks to my family for the affection and encouragement.

# 1 Introduction

## 1.1 Background

Since an average car spends 95% of a day or 23 hours per day parked, parking spaces occupy a huge amount of land use and could possibly cause congestions, emissions, noise and accidents in the urban area.

Since vehicles should be parked at both origins and destinations of trips, the number of parking spots is estimated more than twice of the car ownership. For urban areas, increasing demands in parking spaces challenge the limited land use. In Los Angeles, 14 percent of incorporated land or 200 square miles is tied to parking, the lane use is 1.4 times more area than that is devoted to roads, and 18.6 million parking spots are dedicated to storing 5.6 million vehicles. In developing countries, the parking cost is increasing sharply. In Guangzhou, China, the price of purchasing a parking space is about $114,000, and the public parking fee is about $2.29 per hour in the daytime in 2016. Contending such issue, authorities would like to promote parking facility development (NDRC of P. R. China, 2015, 2016), and a considerable number of new parking spaces are planned to be constructed in a few years. For example, there are estimated 600,000 constructing parking spaces and $2.5 billion budget raised in Guangzhou (XNA, 2016). 30,000 parking spaces in Chongqing, 17,959 in Suzhou, 14,000 in Quanzhou, and 10,000 in Qingdao would be constructed. However, the increasing parking facility supply would encourage private car ownership and cannot satisfy the demand. Thus, a parking supply/demand report in Boise, ID (CCDC, 2015) suggested the following five strategies related to addressing parking demand growth: (a) better utilization of existing parking, (b) implementation Transportation Demand Management (TDM) initiatives, (c) examining parking regulations, (d) examining parking rates, (e) building additional parking spaces. Modern techniques, such as parking demand management, parking reserving system, and parking guide system, are developed by engineers and researchers to improve the efficiency of the parking facilities.

Since on-street/curb-side parking causes higher externalities, such as congestion, space occupation, reduced safety and so on (Feitelson and Rotem, 2004), over 80% of the existing and most of constructing parking spaces are for off-street parking. However, the “last-mile problem” exists in parking systems, which may ruin the experience of drivers and reduces the efficiency and safety of parking systems. Thus, it comes to be critical to evaluate the performance to improve the operation of the existing parking facilities and the design newly-planned ones.

To improve the efficiency of the off-street parking lot, newly-opened public parking facilities are equipped with intelligent managing devices, such as indicators, sensors, indoor positioning system and guiding information distributors for advanced operational requirements. New techniques potentially relieve management problems such as:

(a) insufficient information by traffic signs, warning signs, convex traffic mirrors, changeable/variable message signs, in-lot parking guidance systems and so on;
(b) lacking queuing estimation during peak hours by parking space monitoring, parking behavior learning;
(c) lacking guidelines for designing dimensions of spaces, aisles, and entrances.

However, there lacks a comprehensive microsimulation tool to estimate the efficiency-related outputs of adding new devices.

Safety concerns draw increasing attention in the recent studies. Even though vehicles in parking facilities have a low speed (5 MPH to 10 MPH), National Safety Council (NSC) found on average at least 60,000 people were injured and 500 or more died in the 50,000 plus crashes in parking lots and garages every year in the U.S., 20% of accidents involving fatalities and injuries occurred within parking facilities and 14% of all claims of auto damage involved collisions therein (NSC, 2016).

Parking systems involve the transition between static state and moving the state in compact space. Therefore, parking lots have more dilemma zones and blind spots than urban streets. The situations accounting for accidents include: (a) when looking for parking spaces, distracted drivers leave traffic in danger; (b) obstacles block both drivers' and pedestrians' visions; (c) the space between the vehicle and surroundings is much narrower than that on road, which needs advanced driving experiences; and (d) distracted walking pedestrians have unpredictable and misleading behaviors.

In view of such situation, new design concepts are desired, such as consciousness-raising traffic markings and accessible pedestrian design (e.g. in Shanghai Hongqiao International Airport, China, see Wang, 2016), where solid arrows are routes for pedestrian, and dotted ones are for vehicles. There needs a micro-simulation tool to aid designers with evaluating the consciousness-raising and accessible pedestrian designs.

For the non-traffic safety issue, the off-street parking facilities are so poor-slight enclosed areas with dark stairwells, high walls, structural columns that attracted crimes, unfortunately. According to the National Crime Victimization Survey in 2015 (BJS, 2016), more than 10% of all property crimes (such as theft), and more than 7% of all violent crimes (such as assault, rape, and robbery) occurred in parking facilities. Thus, sufficient monitoring systems and security systems should be provided to deter crimes to ensure safety and security. However, there lack tools for parking lot managers to identify the major and minor security problems to implement effective parking lot solutions.

Modern parking facilities satisfy more specific customers, such as women, electrical vehicles owners, and car sharers. The establishment of the reserved spots for specific users is empirical and lacks quantitative analytics.

Women/female only parking spaces were originally designed in 1990 in Germany. Such parking spot sign includes high heels, Venus symbol, Victorian women, or American Institute of Graphic Arts (AIGA)-style gender symbol, etc. In some German states, women’s parking spaces must be marked as such, should be near the facility entrance, must be monitored by a security guard or camera. In Hebei, China, parking spaces for women have been established in shopping centers, which are between 3.2 to 3.3 meters wide to allow car door to be fully opened. Women-only parking spots have been in widespread use in South Korea since 2009. In Seoul, Korea, pink “she-spots” are designed near destination for being more conductive. With a similar purpose, parking spots are reserved for expected parents in various countries. In other countries, similar designs are made with a pelican sign for expected mother/parent only and pregnancy only.

For serving an increasing number of electrical vehicles, plug-in recharging stations are usually set up with parking spaces. For increasing mobility with existing parking facilities, the reserved parking spaces for car-sharing (such as Zipcar and peer-to-peer car-sharing), ridesharing (such as Uber and Lyft with multiple customers), park and rides, and car-pooling. For example, carshare-only parking spaces were established at ten metro stations in Los Angeles in 2015, CA. However, there lacks a quantitative tool for evaluating the planning of parking spaces of specific types, therefore the impact and technical reasoning for designing such specific spots are not clear.

In view of the efficiency and safety issues, the existing parking simulation system studies contribute to (a) evaluating parking guidance system (Li, 2016) and smart parking system (Chaniotakis and Pel, 2015), (b) aiding design (Yue and Young, 2005), and (c) demand management (Waerden, 2003, 2005).

However, the previous microscopic modeling frameworks didn't consider the emerging requirements of parking facility design, management, evaluation: (a) park space sharing, (b) serving specific types of vehicles to promote green traffic, such as plug-in electrical vehicle recharging devices and reserved parking spaces for hybrid vehicles, (c) multi-purpose parking space usage for promoting shared mobility, such as car sharing and carpooling, (d) ancillary services, such as car washing, (e) mixed-use parking for cars, motorcycles and bikes, (f) multi-design park-and-ride facility, and (g) automated parking facility.

## 1.2 Research objective

The primary objective of this dissertation is to develop an overall operational framework embedded with a set of integrated simulation models for designing, maintaining, operating in urban parking systems. This research is expected to assist responsible agencies, planners and operators in generating effective simulation models and experiments under various scenarios. More specifically, this research contributes to:

1. Developing agent-based representation of the spatial and temporal interactions between components in parking systems due to time-varying demands;
2. Developing a comprehensive simulation model to capture the interactions and dynamics of users and environments within parking systems;
3. Proposing a methodology for data collection, processing in parking systems that can

extract the temporal and spatial distribution of user behavior within urban parking systems;

4. Reporting quantitative measures to support decisions in designing, maintaining, operating parking systems within parking facilities;
5. Illustrating the proposed methodology through a real-world case study to help planners and operators to best apply the proposed framework.

## 1.3 Thesis outline

This chapter illustrates the research framework of the proposal and the interrelations between its principal components.

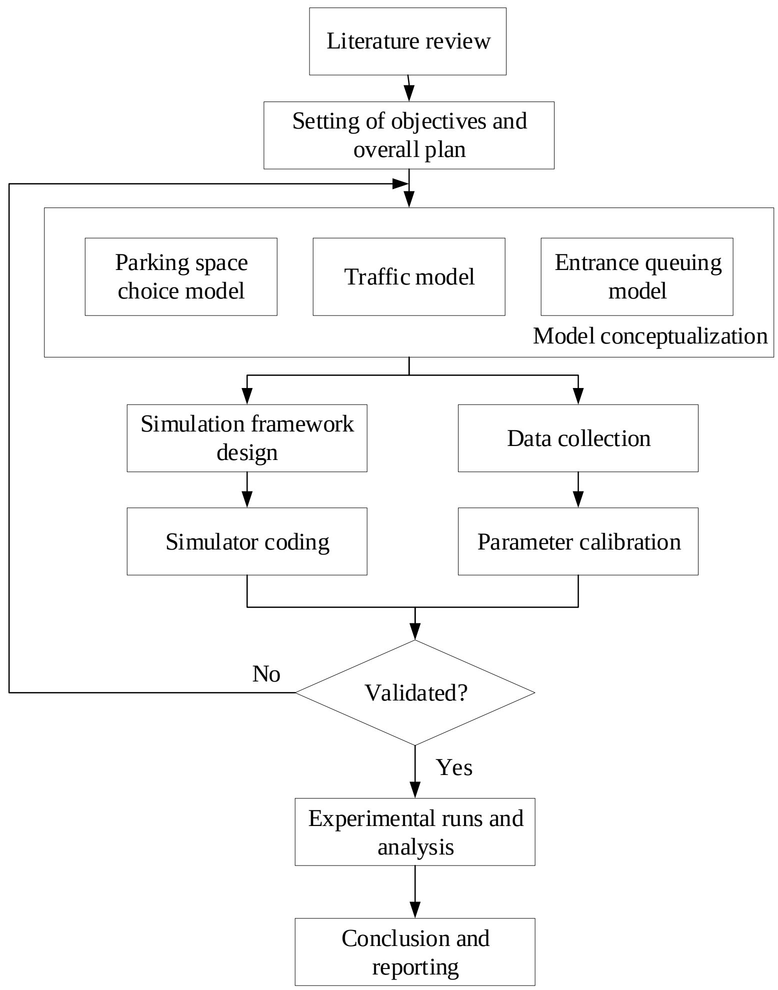



Figure 1-1. The proposed research framework

To address the critical issues listed in Chapter 1, this proposal has divided the research efforts into the following primary tasks:

Task 1: Perform a comprehensive review of relevant research for parking behaviors and simulation system design.

Task 2: Propose an agent-based simulation framework for the parking logic programming model.

Task 3: Present a local binomial choice model to capture the cruising and searching parking behavior.

Task 4: Code the simulation engine in Java programming language.

Task 5: Develop a Software as a Service (SaaS) and a web-based Graphic User Interface (GUI) to visualize the movement of drivers and pedestrians within a parking lot.

Task 6: Real-world data collection and calibration, then possibly modify the simulation model.

Task 7: Perform result analysis and draw a conclusion.

Base on the proposed research objective, the chapters of this dissertation are organized as follows:

(a) **Chapter 1 Introduction** outlines the existing problems in the state of practice and motivation of this research with respects to challenging parking demand, concerning efficiency and safety issues, and emerging new types of reserved parking spaces.

(b) **Chapter 2 Literature review** presents a comprehensive review of relevant research, including parking behavior models, parking facilities design, and parking system simulation.

(c) **Chapter 3 Parking system simulation framework** illustrates the modeling framework of the proposed research including: (1) Process for evaluation and design refinement, (2) Parking system simulator design, (3) Simulation output and measure of effectiveness, (4) System integration, and (5) Software architecture.

Section 3.2 shows the proposed parking system model consists of simulations of multi-agent choice model, randomness, system dynamics, processes, and rules. Section 3.5 shows a simulation engine structure and a Software as a Service design and implementation.

(d) **Chapter 4 A Microscopic agent-based parking system simulator** illustrates details about elements, data collection and processing procedures of parking systems and their interactions: (1) testbed, (2) data collection, (3) descriptive analysis, (4) demand distribution calibration and experiments, (5) predicting dynamic demand, (6) mathematical notation, (7) modeling traffic dynamics, and (8) modeling entity behavior.

(e) **Chapter 5 System application** illustrates the system application of the proposed simulator. The structure consists of charting and visualization and simulation-aided design.

Section 5.2 shows the following addressed design concerns smart check-in device, flexible pricing policy, special parking spot, and reverse parking policy.

(f) **Chapter 6 Summary and conclusion** draws research conclusion and indicates expected future work.

# 2 Literature Review

## 2.1 Parking facility design, evaluation, and management

In the conventional methodology, parking lot design follows the infrastructure guideline and manuals (Weant, 1987; Wekerle and Whitzman, 1995; Chrest et al., 2012; Yang, 2003; Shao et al., 2016). The manuals of parking transportation design are nationally applicable and suitable, while most of the guidelines are localized in a city domain (e.g. City of Philadelphia, PA, 2010; City of Solana Beach, CA, 2012) or in more limited areas. In developing countries, transportation engineers investigate the proper methods to provide enough parking resources. In state-of-practice, mature parking programs across the US are moving to a new phase aiming to improve their communities and stimulate economic development opportunities (CCDC, 2012). Table 2-1 shows some cases of various parking design approaches in the US.

In the literature, Prevost (1985) modeled the on-street parking transportation. Iranpour and Tung (1989) proposed the parking lot optimal design method to maximize efficiency. In the modern parking planning and management, the researchers revisit the parking design theory and paradigm to expand the role of the parking storage. To address the environmental concern, Rushton (2001) investigated low-impact parking lot design to reduce runoff and pollutant loads. Ben-Joseph (2012) suggested to rethinking the parking lot design and culture and showed parking lots can be aesthetically pleasing, environmentally and architecturally responsible, and used for something other than car storage. Jin (2003) investigates the practices of parking lot planning in Guangzhou, China. Barone (2013) showed possible applications of intelligent parking management system in smart cities. In the US, the parking spaces are oversupplied due to the traffic pattern and the car ownership. Abdelfatah and Taha (2014) proposed a mathematical model to maximize the capacity of the parking lot with given land use. However, the existing studies do not involve the quantitative simulation study of the parking design and management. In the state of practice, the animation of parking facility simulation is used as an intuitive representation for design aiding but not a quantitative tool for performance evaluation. There lacks a comprehensive tool for aiding design and operational strategies, especially for the mixed-use parking facilities (see Table 2-1).

Table 2-1 Advanced design approaches for mixed-use parking facilities

| Parking design approach | Example |
| --- | --- |
| Book-ended with other uses | Spring Street Garage,<br>City of Greenville, SC |
| Wrapped with other uses | 15th & Pearl Street Garage,<br>City of Boulder, CO |
| Stacked between other uses | Wynkoop Garage,<br>LoDo District Downtown, Denver, CO |
| Below with other uses | Terrance at Riverplace,<br>City of Greenville, SC |

For parking facility layout design, Computer-Aided Design (CAD) tools such as AutoCAD and ParkCAD are widely used in the state of practice. There exist a rich set of simulators for traffic simulation and animation. However, there’s no dedicated commercial software for the parking facility analysis purpose.

AutoCAD vehicle tracking module is a full function parking facility layout design software, vehicle movement animation for testing potential obstacles. The adding a single-sided group of spots, adding a two-sided group of spots, adding aisles along with a line, adding a single aisle, connecting two aisles, breaking two aisles, editing an aisle, editing a single spot, editing parking island.

ParkCAD is a professional AutoCAD plugin developed by Transoft Solutions for parking facility layout design. It supports adding a two-sided group of spots, adding angled spots, adding strip curb, and testing rich standard compliance.

## 2.2 Modeling parking behaviors

Designing efficient and safety parking systems is a vital transportation research topic. Researchers are dedicated to modeling parking-related behaviors.

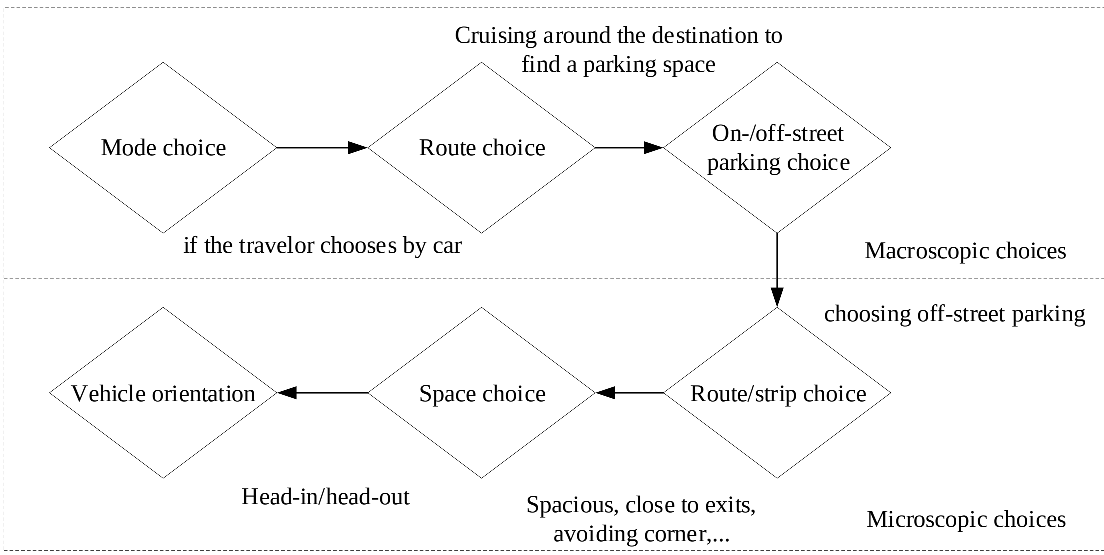



Figure 2-1. Travelers' parking choice process

Figure 2-1 shows that parking-related behaviors can be modeled as a consequent multiple-stage choice process, including on-/off-street parking choice, parking lot choice, parking space choice, vehicle orientation choice (i.e. head-in or rear-in), route choice within the parking lot. These choices can be classified into two categories: (a) macroscopic choices (b) microscopic choices. Figure 2-1 shows the factors investigated in the literature. Researchers found drivers prefer indoor parking spaces closer to the destination, less walking, equipped with an intelligent guidance system, and easier to find a space. Table 2-2 summarizes the attributes considered in parking lot choice modeling. It is surprising however reasonable that the macroscopic choices are based on microscopic factors, such as walking distance, intelligent guidance system, chance to find a space. These findings drew attention to microscopic researches.

For determining the stochasticity of the parking space choice behavior, Cassady and Kobz (1998) presented probabilistic strategies for parking space selecting behavior, which shows the preference of parking spaces in parking lots. Arnott and Rowse (1999) found complex nonlinearity in the parking space searching behavior.

Table 2-2. Attributes considered in parking lot choice modeling

| Reference | Fee | Walk Distance | Access Time | Search Time | Duration | Driver Age | Lot Type | Fine | Purpose | Parking Guidance System | Occupancy | The chance to find a space |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gillen (1978) | √ | √ | √ | √ | | √ | | | | | | |
| Kanafani (1983) | √ | | | | √ | | | | | | | |
| Hunt (1988) | √ | √ | √ | √ | | | √ | | | | | |
| Axhausen and Polak (1991) | √ | √ | √ | √ | | | | √ | | | | |
| Hunt and Teply (1993) | √ | √ | √ | √ | | | √ | | | | | |
| Lambe (1996) | √ | √ | √ | | | | | | | | | |
| Tompson and Richardson (1998) | √ | √ | √ | √ | √ | | | √ | | | | |
| Dell'Orco et al. (2003) | √ | √ | √ | √ | | | | | √ | | | |
| Bonsall and Palmer (2004) | √ | √ | √ | | | | | | | √ | | |
| Ruisong et al. (2009) | √ | √ | | | | √ | | | | | √ | |
| Caicedo (2010) | | √ | | √ | | | | | | | | |
| Van der Waerden (2012) | √ | √ | √ | √ | √ | √ | √ | | √ | √ | √ | √ |
| Shaaban and Pande (2016) | | √ | | | | | √ | | | √ | | √ |

## 2.3 Parking simulation systems

Simulation (or Monte Carlo Method) is an approximation of the real world. Due to sufficient complexity of the stochastic system, it may be the only feasible way to perform quantitative assessment numerically. The simulation generates possible behaviors from the simulation model and collects statistics from these records to estimate the performance measures.

Since the parking process in parking systems cannot be simply described by probability distributions, simulation methodology is used to describe these complicated behaviors. Parking simulation is the imitation of the locations and entities in parking lots to evaluate and improve the performance of parking lots.

With regards to the considered scope of choices, parking simulation can be macroscopic or microscopic: (a) macroscopic simulation visualizes parking lot choice, route choice to parking lot or road-side parking space to analyze the competitive relations between parking facilities; and (b) microscopic simulation focuses on route choice, parking space choice, vehicle orientation (head-in/rear-in) choice in the parking lot, pedestrian behavior and so on.

In microscopic behavior modeling, drivers show preferences for certain spaces in parking lots. The website Wikihow shows experienced drivers would like to park in spaces without other cars parked aside, and experienced drivers prefer rear-in vehicle orientation because this orientation is easy for leaving (Wikihow contributors, 2018). There lacks study on evaluating how such preferences impact designing, operating, and maintaining parking lots.

### 2.3.1 Macroscopic parking simulation models

Macroscopic simulation describes parking lot choice, route choice to the parking lot or road-side parking space, many scholars studied parking lot choice. Based on parking behavior studies, researchers proposed various kinds of the simulator to capture these behaviors and evaluate the impact on the performance of the planned parking lot deployment.

Benenson et al. (2008) proposed an agent-based model to evaluate search time, walking distance, and parking costs over different driver group with self-organizing parking agents. The simulation system PARKAGENT by Benenson et al. (2008) captures availability for both on-street and off-street parking spaces but cannot describe the microscopic movements and behaviors with parking systems. Levy et al. (2013) proposed an analytical model called PARKANALYST alongside the PARKAGENT to analyze the impact of occupancy rate and demand-to-supply ratio on cruising for parking. Levy et al. (2015) applied the PARKAGENT to estimate the effectiveness of planned parking facilities and showed the potential benefits of using an intelligent parking guidance system. PARKAGENT was a parking searching tool for estimating the effectiveness of planned parking facilities for different development scenarios in the area and assessing electronic signage system that directs drivers to available parking facilities.

Spitaels et al. (2009) proposed a macroscopic parking behavior simulation system for assessing the parking management strategies to support sustainable parking policymaking. SYSTAPARK captures aggregated cruising flow of cars, which can investigate the externalities of cars cruising for on-street spaces around the parking destinations.

Dieussaert et al. (2009) developed an agent-based model for simulating parking search, where the movement of the car when searching for a parking place is determined by a search strategy and translated into cellular automata movements.

Obdeijin (2011) developed an S-Paramics-based tool to simulate parking guidance systems and applied the tool to analyze the performance of road-side variable message board for parking vacancy.

Waraich and Axhausen (2012) and Waraich et al. (2012) presented an agent-based parking lot choice model to illustrate the overall simulation can react to spatial differences in parking demand and supply. Horni et al. (2013) reported the development of their model using cellular automaton (CA) approach integrating MATSIM and tested the software in a real-world scenario for the town center of Zürich.

Guo et al. (2013) developed an agent-based transportation model of a university campus, primarily focusing on vehicle-related travel and the associated parking search progress and integrated the proposed model with TRANSIMS and MOVE2010 emissions model. Beheshti (2015) presented a hybrid approach for combining agent-based and stochastic simulations to forecast transportation patterns and parking lot utilization on a large university campus.

The macroscopic simulation considered the traffic flow redistribution and the impact of on-street cruising on the traffic congestion and pollution, however, failed to capture microscopic driver behaviors, maneuvers, and vehicle movements within parking facilities, which have an essential impact on the macroscopic behavior.

### 2.3.2 Microscopic parking simulation models

Macroscopic behavior researches only consider parking lot choice and ignore parking behavior modeling within the parking lot. For better designing, managing, and maintaining the parking lot, microscopic behaviors are investigated to capture the choices within the lot such as how motorists select a parking space and how cars will move across the parking lot. For this research topic, scholars have proposed many models to illustrate the dynamics of these behaviors.

For off-street parking simulation, Young (1986), Young and Thompson (1987a, 1987b) developed a rule-based parking model to evaluate the means of quantifying measures of performance. Yue and Young (1996) proposed a second version of their microscopic simulation system, which provided a quantitative measurement of performance of the existing layout designs such as parking lot utilization, average travel time and degree of conflicts. The set of attributes considered in their simulation system include travel time to the parking place, walking time form the parking place to the desired destination, ease of parking, ease of exit from vehicles, and available shade. Young (2000) distinguishes five types of parking models, namely parking-design models, parking-allocation models, parking-search models (both in parking lots and in a street network), parking-choice models, and parking-interaction models.

However, only simple deterministic models are used to describe the behavior in these studies which has deficiencies as follows: (a) failing to detail cars' and pedestrians' movements, and influence of obstacles; (b) ignoring pedestrians' interaction; (c) lacking to consider the vision of drivers.

For modeling off-street parking space choice behavior, Thompson and Richardson (1998) proposed a conceptual framework with respect to the parking behavior in parking lot, which took the state of the parking system (e.g. number of vacant space), individual parking spaces (e.g. the distance from a parking space to the entrance, to the pedestrian exit, and to the payment device), and the characteristic of the motorist (e.g. gender, age, type of car, car occupancy). Based on this framework, van der Waerden et al. (2003) proposed a nested logit model for space choice behavior in the parking lot and calibrated their model with real-world data. The results show a substantial degree of heterogeneity in parking choice behavior. Based on their model, Vo et al. (2016) developed a multi-agent-based simulation tool to demonstrate its capability of studying driver movements across parking lots where vehicle travel time and parking occupancy indicators were integrated to investigate the efficiency of the parking. Zhao et al. (2017) followed the Vo's study and proposed a framework for optimizing parking management based on microscopic simulation systems.

However, these studies just focus on ad hoc parking behavior model and fail to

(a) describe queuing at entrances and exits which impact the real-time capacity;

(b) evaluate the parking guidance system.

For the on-street parking facilities, Ukpong et al. (2007) developed a traffic model so as to display the travel time of traversing vehicles with and without the presence of on-street parking in VISSIM-ENVPRO software, which carried out a comparison in emission levels between specified road networks.

To evaluate the parking guidance systems, Li (2014) used real-world data to evaluate the performance for the parking guidance information system. However, the Li's model only applied to a tree-like in-lot network and would fail to incorporate networks with general topological structure. Yuan and Liu (2014) implemented Vehicle Generation Model and Car-Following Model, vehicle parking behavior, such as individual vehicles parking, and leaving principle and multiple parking in VISSIM, the commercial microscopic simulation environment.

Table 2-3 summarizes the design criteria of the parking lot simulation systems in the literature. The existing studies have failed in the following key concerns: (a) agent-based model captures complex behaviors, (b) machining learning techniques driver behavior data mining, (c) general modeling templates since existing studies are ad hoc, hard for calibration, and (d) a comprehensive modeling and testing environment.

Table 2-3. Design criteria of the parking lot simulation systems in the literature

| Reference | Scope | | Detail level | | Parking guidance system | Pedestrian |
|---|---|---|---|---|---|---|
| | Micro | Macro | On-street | Off-street | | |
| Benenson et al. (2008) | √ | | √ | √ | | |
| Levy et al. (2013) | √ | | √ | | | |
| Levy et al. (2015) | √ | | √ | | √ | |
| Spitaels et al. (2009) | | √ | √ | | | |
| Obdeijin (2011) | | √ | √ | | √ | |
| Waraich et al. (2012) | | | | | | |
| Waraich and Axhausen (2012) | | | | | | |
| Horni et al. (2013) | | | | | | |
| Guo et al. (2013) | | | | | | |
| Beheshti (2015) | | | | | | |
| Young (1986) | √ | | | √ | | |
| Young and Thompson (1987a, 1987b) | √ | | | √ | | |
| Young and Taylor (1991) | √ | √ | √ | √ | | |
| Yue and Young (1996) | √ | | | √ | | √ |
| Young (2000) | √ | | | √ | | |
| Yang and Weng (2005) | √ | √ | √ | √ | | |
| van der Waerden et al. (1997) | | | | | | |
| van der Waerden et al. (2003) | √ | | | √ | | |
| Vo et al. (2016) | √ | | | √ | | |
| Ukpong et al. (2007) | | | | | | |
| Li (2014) | √ | | | √ | | |
| Yuan and Liu (2014) | √ | | | √ | | |

In view of the deficiencies of the previous microscopic systems, existing studies are case-specific and hard for calibration. The proposed simulation system includes agent-based model captures complex pedestrian behavior, driver behavior data mining with modular machining learning techniques, and provides a comprehensive modeling and testing environment.

### 2.3.3 Existing commercial and open-source parking simulation systems

In VISSIM, the parking spot choice is captured by a fixed logit model involving parking cost (from zone property parking fee), attractiveness, direct distance between parking lot and the destination zone's center of gravity, general cost of best route from current vehicle position, availability of free parking spaces, index of the vehicle type, index of the decision situation (departure, routing decision). VISSIM have great animation rendering module in both 2-dimensional and 3-dimensional environments for transportation and could be a visualizer of the results of the proposed model.

Simulation of Urban Mobility (SUMO, Krajzewicz, et al., 2012) is a free and open traffic simulation suite which is designed by Institution of Transportation System, German Aerospace Center and is available since 2001. The crosses of parking spot links and aisle links are modeled as intersections.

Figure 2-2 shows a sample surface parking lot modeling in SUMO. Each spot and cross are modeled as intersections. Such analog would complicate the problem and the continuous traffic simulation would not implement the specific rules and traffic environment in the parking facilities. The visualization of such intersection is confusing and messy.

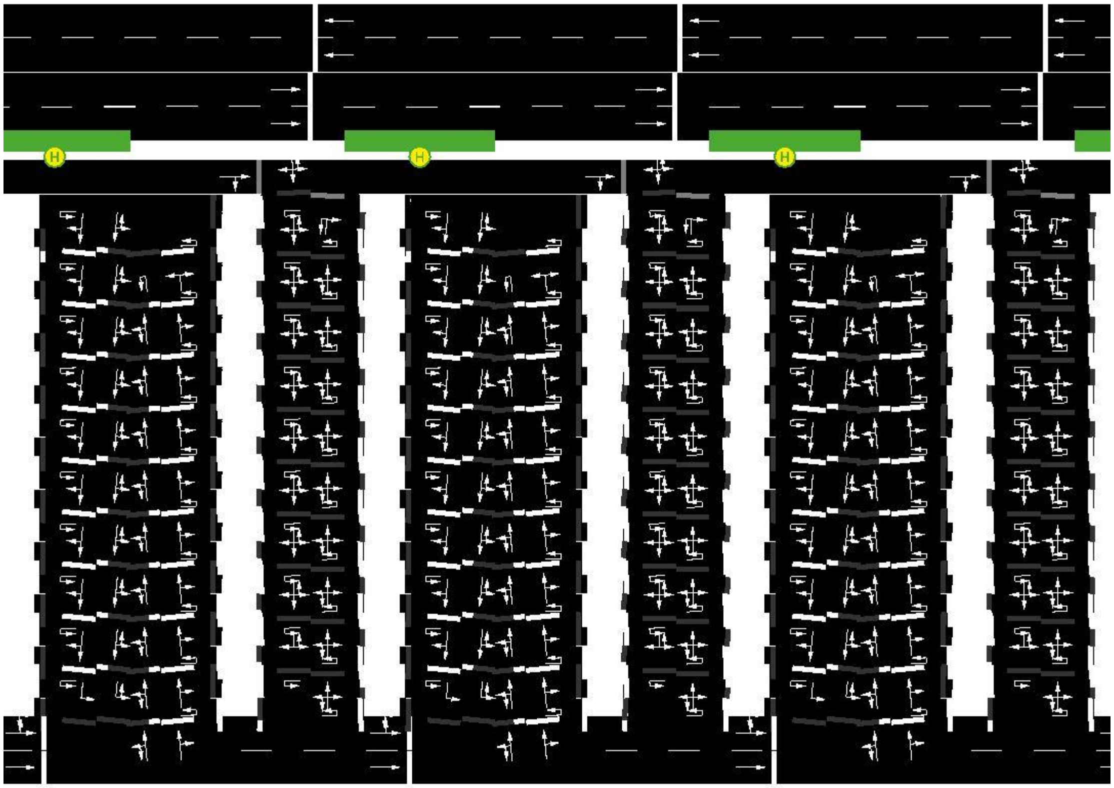


Figure 2-2 A screenshot of the parking lot layout modeling in SUMO

For behavior process modeling, there exist 18 commercial discrete event simulators and 11 open-source ones. As a typical discrete event simulator, ProModel, the commercial industrial simulation system, also allows modeling of continuous processes which is developed by ProModel, Inc. However, ProModel is not designed for traffic simulation. The en route movement doesn’t have a psychical model.

A pilot study is conducted on modeling a campus parking lot on ProModel. Figure 2-3 shows the layout of the studied parking facility built in ProModel.

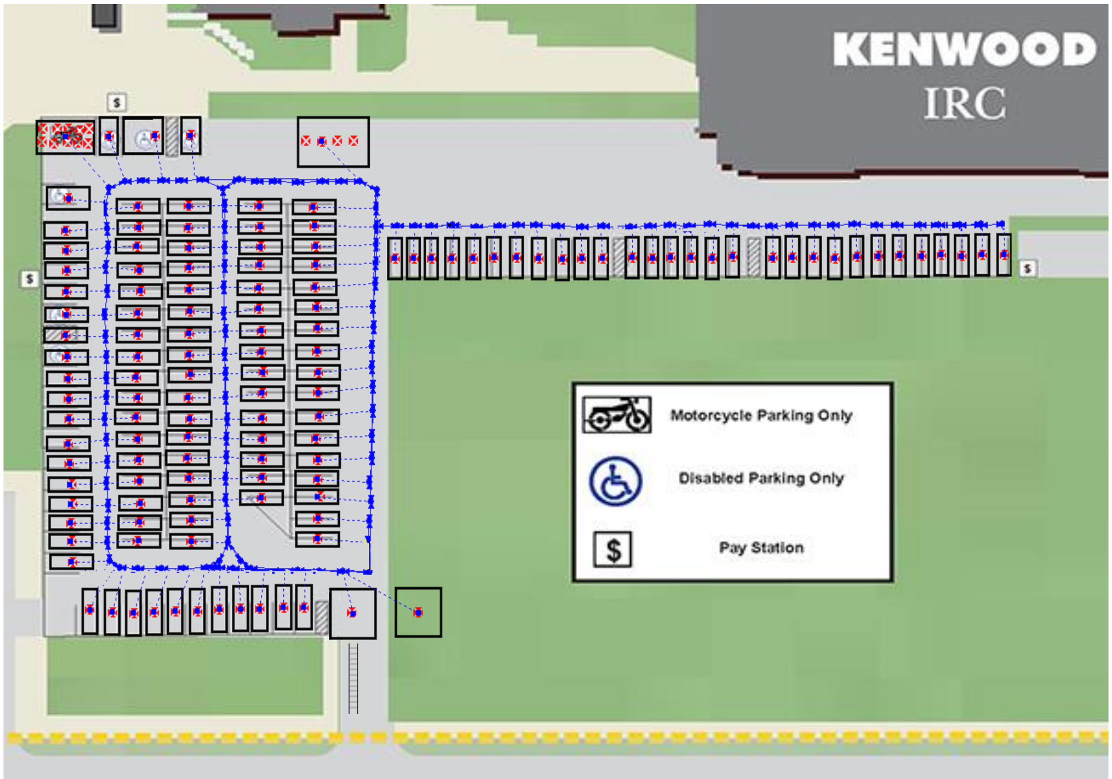



Figure 2-3 A sample of the parking lot layout modeling in ProModel

In ProModel, the spots and entrances are modeled as locations and the aisles are modeled as paths, vehicles are modeled as resources, drivers are modeled as entities, and the arrival and exiting patterns are modeled with the empirical distributed process.

In view of the limitations of the existing traffic simulators, the parking system has unique traffics dynamics and user behavior pattern which should be modeled and implemented in the self-programmed simulation system.

# 3 A parking system simulation framework

## 3.1 Process for evaluation and design refinement

The parking system evaluation and design refinement follow the procedure:

(a) Overview and planning a practical objective

(b) Collecting and preprocessing data

(c) Evaluating performance measures

(d) Investigating the relationship between these measures and operation/maintaining factors

(e) Drawing the conclusion and reporting

In the state-of-practice parking infrastructure design and management, the simulation does not have a critical role in extracting but display animations and renderings. The proposed off-street parking simulation system would provide a more quantitative tool to aid the designer and the manager. With the proposed simulator, the design and the management of parking facilities involve the following trial-and-error procedure: (a) establishing a simulation model, (b) evaluating the potential outcomes, (c) changing parameters and settings, (d) comparison to the former plan, and (f) making the decision.

## 3.2 Parking system simulator design

Modeling the parking system simulator incorporates the common features with the traffic simulators such as multiagent-based simulation, random variables and system dynamics, and the specific features in contrast to the traffic simulators such as modeling processes and rules. This section elaborates the key features of the parking system simulator modeling and construction.

### 3.2.1 Simulation of multi-agent

The simulation is one of the best strategic and tactical design-support technologies for the complex, dynamic and stochastic system (Siebers and Aickelin, 2008). The informative simulation modeling depends on the proper design of abstraction and simplification.

Based on the event organization, the simulation methods can be classified into two categories, the continuous and the discrete. The continuous-state simulation is applicable to systems with the continuous state space and typically differential equations, such as physical motion equations.

Based on the behavior organization, the simulation modeling in Operational Research can be classified into three categories: Discrete Event Simulation (DES), System Dynamics (SD), and Agent-Based Simulation (ABS). DES and ABS usually describe the decision processes at the microscopic level.

The discrete-state event-driven simulation models the systems with finite discrete states and events. Typical DES systems model deterministic resources without performance variation and pro-active behavior. Technically, the DES method maintains a list of events, by adding new events and eliminating the finished events for a given horizon of time.

There exist two mechanisms of DES capturing the processes of the simulation system: (a) fixed-time-stamp advance (b) variable-time-stamp advance (Davidsson, 2000). Table 3-1 shows the advantages and disadvantages of the two mechanisms. Since this research focuses on non-differential finite-state systems, the parking system is handled by the fixed-time-stamp advance mechanism in this study.

Table 3-1 Comparison of procedures for executing DES models

| | Fixed-time-stamp advance (time driven) | Variable-time-stamp advance (event driven) |
|---|---|---|
| Advantages | Good for all models where most events happen at fixed increments of time (e.g., gate-level simulations). Has the advantage that no "future event list" needs to be maintained. | Periods of inactivity are skipped over, models with a bursty occurrence of events are not inefficient. |
| Disadvantages | Can be inefficient if events occur in a bursty manner, relative to time-step used. | If event times are general (have memory) then "future event list" is needed. |

This study leverages the ABS framework in view of the advantages over the DES. (a) In comparison to the DES method, the agent-based simulation initially models agents as a cellular automaton, which is able to model systems with heterogeneous, autonomous and pro-active entities (Siebers, 2007). The parking system fits well in the framework in agent-based simulation due to the nature of nonlinearity and heterogeneousness. In this study, the entities are modeled autonomously with multiple goals of a process and intermediate goals to justify the process of the method. For example, the vehicle is modeled with a state transition process without external stimuli from the simulation environment. (b) ABS supports distributed computation naturally. Since each agent is typically implemented separately, the different agents are able to be encapsulated to a process or thread for better performance and scalability. This study takes advantages of this feature to implement concurrent computation with the Akka framework. (c) ABS has the capability of incorporating various modeling paradigms. In this study, the logic programming is used to model the behavior of agents, which is diverged from Situated automata which is originally proposed by Kaelbing (1986). The logic programming produces an inside-out technique to manipulate the attributes of the agents to provide a proof-of-concept prototype before creating a full-scale experiment. For example, in logic programming the state of the vehicle $C_1$ is accessed via the query $state(C_1)$ while in the Objective Oriented Programming

(OOP) the state is accessed via the attributes in the vehicle class.

For the multi-agents design of the proposed simulation, there exist two main kinds of agents: (a) entity, the movable agents with complex pro-active behaviors; (b) location, the unmovable agents with passive behaviors. The entity may occupy the location. In the industrial simulation system, ProModel (Harrel et al., 2004), the entity and the location are abstract models of general industrial processes. In the agent-based simulator NetLogo (Tisue and Wilensky, 2004), the entity and the location are modeled as the *turtle* and the *patch*.

### 3.2.2 Simulation of randomness

The Monte Carlo Method (MCM) is based on the combination of stochastic processes. The randomness is a critical factor for the universal simulation system. The proposed parking simulation system is designed to generate random variables from parametric distributions (e.g. normal, exponential distributions) and the nonparametric methods (e.g. histogram, kernel density estimation).

The proposed system facilitates the end-to-end calibration and sampling of the hypothetical distributions. To calibrate the random variables generators, the parametric distributions and the nonparametric distributions should be identified. For the parametric method, the probability distribution is identified by $\chi^2$(chi-square) test for the hypnosis that the observations are from a distribution of the parameters. For the non-parametric method, the probability distribution is captured by the model-free data-driven methods, such as empirical distribution and kernel density estimation. When planning new parking facilities, the field survey data are not available. The data from neighboring facilities of the similar type or hypothetical probability distributions can be used.

In comparison to the limited stochastic distribution of the previous studies, the contribution is that the proposed set of the probability distribution is rich enough to capture the variability of stochastic variables in the parking system. In addition, the proposed framework is capable to use extensions for newly developed stochastic methods.

### 3.2.3 Simulation of system dynamics

For a general parking simulation modeling and system design, there exist the following phenomena:

a) *Blowing-up*

In the blowing-up system, entities arrive faster than they depart without end and the servers cannot serve sufficiently, and the entities do not “appear” or “disappear” when in the system. In the steady system, the expected time an entity spends in the system, the expected number of entities in the system and the expected inter-arrival time of entities into system follow the Little’s Law. The Little’s Laws tells us that the average number of entities in the system equals the effective arrival rate times the average time that an entity spends in the system. For the parking system, the queuing of vehicles getting in and out the parking lot in a surged demand, the quantitative metrics are blowing-up and cannot reveal the true performance.

b) *Parallel and series*

The serial servers process entities one-by-one. Any entity cannot skip the step and go to the next step. The parallel servers can provide equivalent service redundantly. A compound system may have parallel parts and serial parts. For example, in the parking system, the parking lot entrances, the parking spots are parallel, respectively.

c) *Blocking*

In the serial servers, one in-process entity may occupy a server and block the waiting others from being processed. For example, the aisles in one strip are serial since vehicles should be blocked by the front entities.

d) *Variability*

The simulation system mimics the real-world stochastic situations and the system is subject to some level of variability. For example, the parking duration for each vehicle varies randomly, which may affect the parking policy and management.

e) *Buffer*

Buffers are spaces between locations for temporary storing waiting for entities to relieve blocking. In the parking system, the double-lane aisles provide buffers for the opposite vehicles while the single-lane aisles have limited space and shall be one-way.

f) *Aggregation*

Two entities may combine into one entity and show the aggregated behavior. For example, when the pedestrian gets on the vehicle, the two entities (the pedestrian and the vehicle) are aggregated to one vehicle.

g) *Warming-up*

When the simulation starts, the system is not empty. A preparing stage of simulation is used to restore the state of the system. For example, at the starting time, the parking lot should be initialized to replicate the utilization of parking spots. In the warming-up, the measures are not meaningful due to the missing information of the existing entities.

### 3.2.4 Simulation of processes

One of the key tasks for building the parking simulation system is to identify the general processes in the parking systems. According to the observation of the real-world parking systems, the parking simulation system has two critical differences from the road traffic simulation system. First, the entities in the road traffic have only one process that is traveling from the origin to the destination via the transportation network, while the parking system has structural multiple processes.

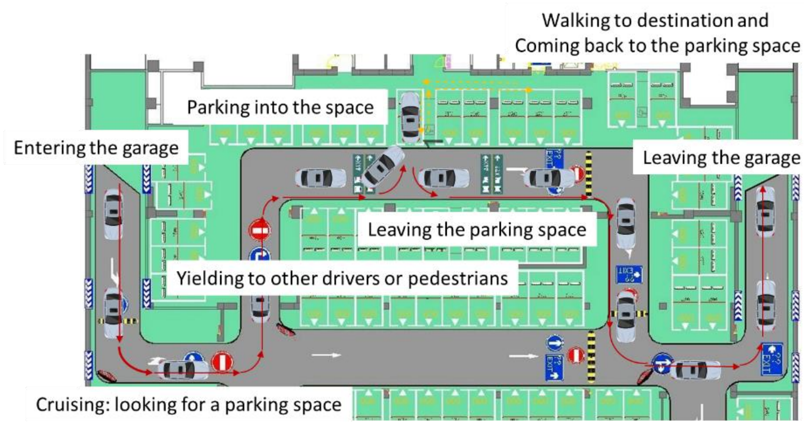



Figure 3-1 A sample parking process

Figure 3-1 illustrates an example of the processes in detail, where the red solid line is for the vehicle trajectory and the orange dash line is for the pedestrian trajectory. Figure 3-1 shows the parking procedure at least involves the following steps: (a) entering the garage, queuing, and paying the parking fee or checking the seasonal permit at the entrance; (b) cruising and looking for a parking space; (c) during the cruising process, yielding to other vehicles or pedestrians; (d) completing the parking maneuver into the spot, where the position could be rear-in or head-in; (e) the passenger(s) would get off the vehicle and walk to the pedestrian exit; (f) preparing to unpark after the passengers are back to the vehicle and completing the maneuver if there exists an acceptable gap; (g) cruising and driving to the exit of the lot; and (h) leaving the parking lot.

Discrete event parking simulation models are potentially the most realistic replication of parking systems because such models can capture the decision-making and interactions of the system components with other elements of the parking system in small time intervals (Young

2000, 2001). Note that the DES is capable to model processes for replicating the entity behaviors in the parking system.

As a pioneering work, Young (1990) presented a comprehensive flow chart of model development process which included (a) the determination of the problem to be addressed, (b) the clarification of objectives to be achieved, (c) the criteria to be used to measure the effectiveness of achieving these objectives, (d) the methods to collect data and calibrate, verify, and validate parameters with real-world situations, and (e) the initial applications of the model. This study follows this methodology of simulation modeling and proposes unique traffic dynamics for off-street parking inner network.

Young and Weng (2005) reviewed and summarized discrete event parking simulation models of on-street parking systems. Their models replicated the parking and traffic in a general parking simulation framework for traffic dynamics, drivers' decision-making processes, and outcomes. The traffic dynamics could be described with:(a) speed, acceleration, and braking, (b) car-following, (c) lane changing, (d) overtaking, (e) gaps and gap acceptance in traffic, (f) signalized intersection behavior, (g) parking and unparking procedures. The decision-making processes could be illustrated with (a) interaction between road users, (b) route choice, (c) total trip consideration, (d) driver risk. The outcomes may involve (a) energy consumption, (b) emissions, (c) noise levels, (d) community impacts. In comparison to their on-street parking system, the off-street parking system simplifies the speed, acceleration, and braking, car-following, lane changing and overtaking since (a) the speed limit with the parking facilities is usually set to10MPH or 15MPH and (b) the lane changing and overtaking are not applicable for a compacted inner network with one lane for one movement direction in the parking facility.

The flowchart presented by Yue and Young (1997) showed a fundamental procedure for the microscopic simulation of parking facilities. The procedure involves entities movement, maneuvers, and unparking decisions processes parallelly, which successfully capture the traffic features with the parking lot. However, fails to incorporate the decision of drivers, such as parking spot choice and parking route choice.

In view of such deficiency, Thompson and Richardson (1998) proposed a framework incorporating parking space choice behavior modeling. This framework illustrated a principle process of the off-street parking behavior which was extended by researchers of driver behaviors in the off-street parking facilities.

Vo et al. (2016) presented a preliminary case study for parking systems with an agent-based modeling and simulation tool, NetLogo (Tisue and Wilensky, 2004). However, their work only considered parking space choice behavior for a special case, which would not incorporate various behavior patterns. Case-specific behavior modeling would have deficiencies in considering extensive behavior factors and evaluating the parking guidance system.

Li (2016) proposed a simulation model to capture the off-street parking spot choice behavior with the multinomial logit model. Li's model was calibrated using real-world data are used for evaluating the performance of parking guidance system. An agent-based simulation model is used for implementing in Repast S environment. Li's model captured both the parking spot choice and the entity movement. However, using a case-specific parking choice model, drivers should make the strip choice and then the spot choice in Li's model. This deficiency limits the application in parking lots with none tree-like topology. And it's difficult to fully evaluate the model without more technical details about the traffic model and visualization.

Vo et al. (2016) and Li (2016) contributed to agent-based parking simulation models, however, proposed case-specific and non-scalable models. In view of these deficiencies, the proposed off-street parking simulation model captures both choices, traffic movements in the parking lot with a general topology. The proposed simulation model captures the behavior processes of entities with a state machine. The entities have states with choices and states without choices.

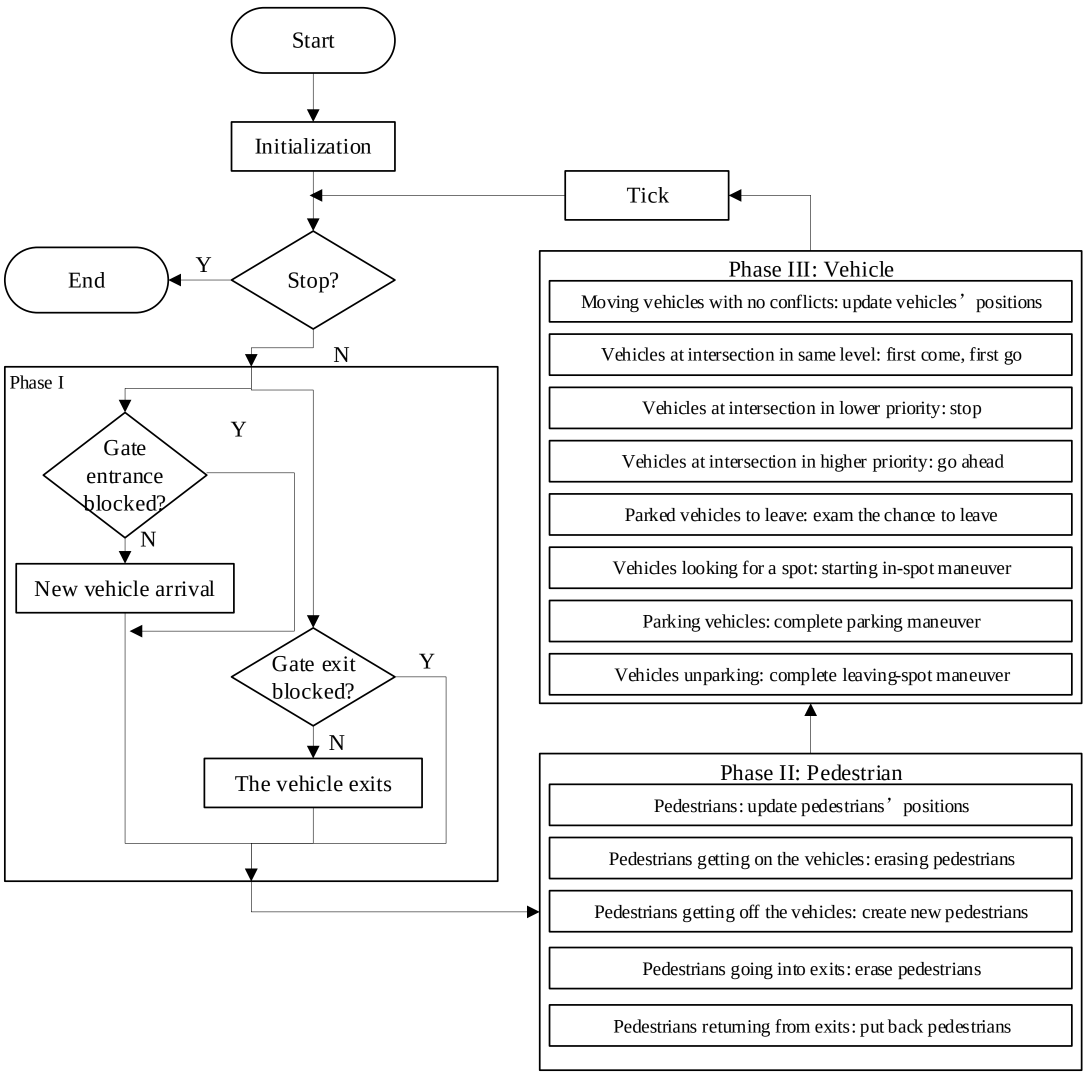



Figure 3-2 The proposed simulation procedure for off-street parking facilities

Figure 3-2 shows the parking procedure includes three phases: (I) arrival and departure (II) pedestrian behavior (III) vehicle behavior. In Phase I, new vehicles are created if the current simulation time reaches the new arrival time. The pedestrian Phase II has precedence over the vehicle Phase III because the vehicles should yield to pedestrians in parking facilities. The proposed procedure is easy to parallel for implementing concurrent computing.

The contribution includes that the proposed system presents the extended process modeling integrating the pedestrians' behaviors and queuing at the entrances and exits which are not covered by the previous studies.

### 3.2.5 Simulation of rules

The second critical difference from the road traffic is that the parking simulation model solves a rule-based logic problem since many interactivities and blocking checkpoints make event triggering mechanism overcomplicated to handle. The entity movement and state transition subject to logic constraints which is not explicitly modeled in the road traffic simulation systems.

For the parking system simulation, the traffic dynamics involves how to find the feasible next state of the system subject to a rule-based moving logic. A logic-based approach is required to stack up a scalable set of rules. In the literature, the logic-based approaches are studied by two communities: (a) operations research and (b) artificial intelligence.

A logic-based approach to operations research was first discussed by Hammer and Rudeanu (1968) and Granot and Hanmmer (1971). The methods are classified into three categories: (a) mixed logic linear programming (Jeroslow 1987, 1989; Hooker et al., 1994, 1999; Raman and Grossmann, 1993, 1994a, 1994b, 1994); (b) disjunctive programming (Balas, 1975, 1977, 1979); and (c) combining logic and linear programming (KcAloon and Tretko, 1995; Tretkoff, 1996; Barth, 1995).

From the aspect of the artificial intelligence, the logic programming is introduced by

Colmerauer (1973, 1986) and Kowalski (1974), allow one to formulate a problem in a subset of first-order logic or Horn clause logic. The logic programming language, Prolog (Clocksin and Mellish, 2012) has two major updates to facilitate the constraint programming: III the advent of constraint programming (Colmerauer, 1990) and IV the approximation of non-linear constraints (Colmerauer, 1996; Colmerauer et al., 2010).

In a compact traffic infrastructure, deadlocks would occur when two vehicles are moving in the opposite direction on a one-lane aisle. The deadlocks would cause severe congestion in the parking garage. Thus, the parking facilities are designed carefully to avoid the potential deadlocks. However, the parking simulation system does not avoid the deadlock but find a possible recovery plan to resolve the deadlock. The deadlock recovery problem varies from the traditional game, Huarongdao or Klotski to the modern machine scheduling models.

To solve this hard problem, a Prolog-like Domain Specific Language (DSL) called PICAT was proposed by Zhou and Kjellerstrand (2014), Zhou (2016) and Zhou et al. (2017). PICAT provides a language level methodology to address the deadlock issue. Inspired by Zhou's idea, this study also proposes a first-order logic rule system for the parking simulation. The specific rules are modeled for the entity behaviors and traffic dynamics.

## 3.3 Simulation output and measures of effectiveness

The simulation system is designed to output historical records and quantitative measures for aiding design, management, and maintenance. Note that the applicability and deliverability of these outputs are subject to the quality and quantity of data, and the proposed system supports to add more suggestive outputs.

### 3.3.1 Measures of efficiency

In the proposed system, the following measures of efficiency are incorporated.

- The average time a vehicle moves from the entrance to a parking space in the parking lot for measuring the searching time of the vehicles.

$$AvgVehArr = \frac{\sum_{v \in V} vehtoexit_v}{|V|} \tag{1}$$

where $vehtoexit_v$ denotes the searching time of the vehicle $v$.

- The average time a pedestrian moves from the vehicle to the exit for pedestrians

$$AvgPedExt = \frac{\sum_{p \in P} vehtoexit_p}{|P|} \tag{2}$$

where $vehtoexit_p$ is the time pedestrian $p$ in set of pedestrians P moves from the vehicle to the exit.

- Number of moving/parked vehicles for each time interval (per day/week/month)

$$NumMovVeh_t = \sum_{v \in V} Moving_{vt} \tag{3}$$

$$NumParVeh_t = \sum_{v \in V} Moving_v \tag{4}$$

where $moving_v$ is 1 if the vehicle $v$ is moving, 0 otherwise.

- Number of walking pedestrians for each time interval (per day/week/month)

$$NumWalPed = \sum_{t \in T'} numped\_t \tag{5}$$

where $numped_t$ is the number of in-system pedestrians at time t, and T′ denotes the set of the timestamps of the period.

- Utilization of each parking space

$$Uti_s = \frac{\sum_{t \in T} occ_{st}}{|T|} \tag{6}$$

where $occ_{st}$ is 1 if the spot $s$ is occupied at time $t$, 0 otherwise.

- Turnover rate of each parking space

$$Turnover_s = \frac{\sum_{s \in S} numuse_s}{|T'|} \tag{7}$$

where $numuse_s$ is the total number of the spot $s$ is used and $T'$ is the set of the time period.

- Utilization of each aisle location

$$Uti_a = \frac{\sum_{t \in T} occ_{at}}{|T|} \tag{8}$$

where $occ_{at}$ is 1 if the aisle a is occupied at time $t$, 0 otherwise.

- Utilization of aisles by vehicles for measuring routing pattern for vehicles

$$UtiVeh_a = \frac{\sum_{t \in T} occbyveh_{at}}{|T|} \tag{9}$$

where $occbyveh$ is 1 if the aisle a is occupied by vehicles at time $t$, 0 otherwise.

- Routing pattern for pedestrian, utilization of aisles by vehicles

$$UtiPed_a = \frac{\sum_{t \in T} OccupancyByVeh_{at}}{|T|} \tag{10}$$

### 3.3.2 Measures of safety

The parking facilities have safety concerns including theft, vandalism, robbery and vehicle collisions since the compact space blocks the vision and lacks sufficient protections. Note that this study involves only the traffic incidents instead of crime within the parking facilities.

In the literature, Gettman and Head (2003) indicated that on-street parking (parallel and double) parking create conflict situations, lane-changes, etc. in the real world and have a significant safety impact. Simulations that model on-street parking maneuvers are preferred. Jason and Jung (1984) showed that parking spaces are a major safety problem for this special population; if a disabled person parks far away from a place of employment, he or she might have to cross busy thoroughfares and require help up. Charness et al. (2012) showed that the most likely reason for the differential crash types in parking lots for older compared to younger pedestrians probably lies in the reduced speed with which older pedestrians can react to hazardous events. Yue and Young (1998) proposed a parking simulator Parksim2 to measure safety in parking lots.

The conventional measures are listed as follows.

- The major reason for accidents within parking lots

$$MajRea = argmax(A \cup S) \tag{11}$$

where $argmax$ is the function to find the index of the maximal number, $A$ is the set of the aisles, and $S$ is the set of parking spots. Note that this measure is derived from the historical accident record data.

- Accident frequency across each layout

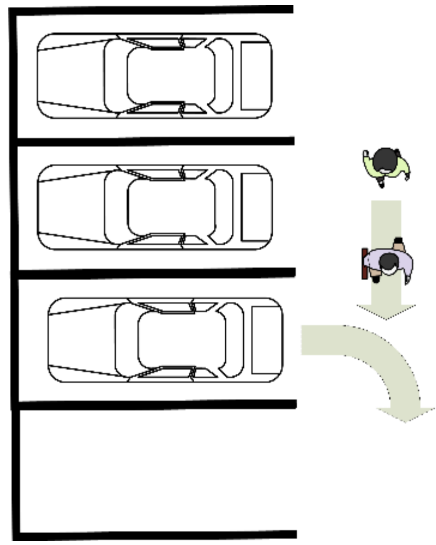
Note that this measure is derived from the historical accident record data.

In the parking simulation methods, traffic safety is not well investigated in the previous works. In this study the traffic safety is evaluated in the following measures:

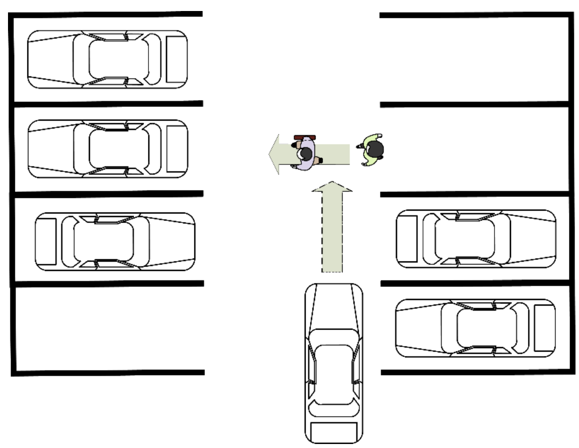
(a) *Pedestrian-vehicle weaving duration.* When the vehicles and pedestrians are moving in the parallel directions, the pedestrians may weave with the pedestrians due to the narrow aisle.

$$AggPedVehWea_a = \sum_{p \in P} PedVehWea_{apvt}, \forall a \in A \tag{12}$$

where $a$ denotes the aisle in the set of aisles $A$.



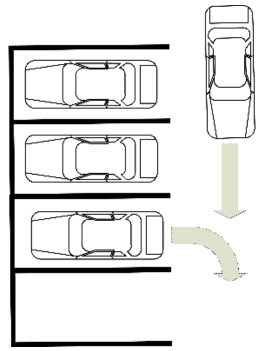
(a) before the parking spot



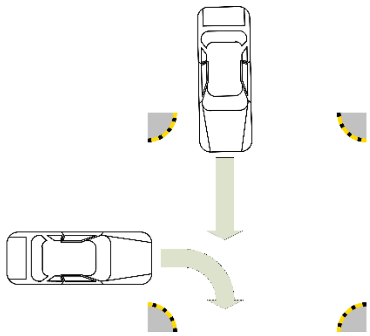
(b) pedestrian across the aisle

Figure 3-3 Pedestrian-vehicle weaving duration in multiple scenarios

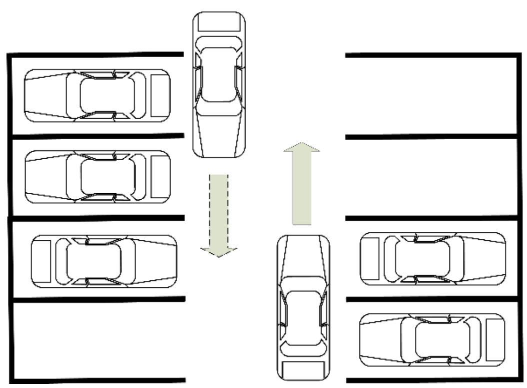
(b) *Vehicle-vehicle weaving duration.* When the vehicles are merging or turning at intersections they may weave with the other vehicles. The weaving zone and duration are measures for parking infrastructure safety evaluation.



(a) before a parking spot



(b) at an intersection

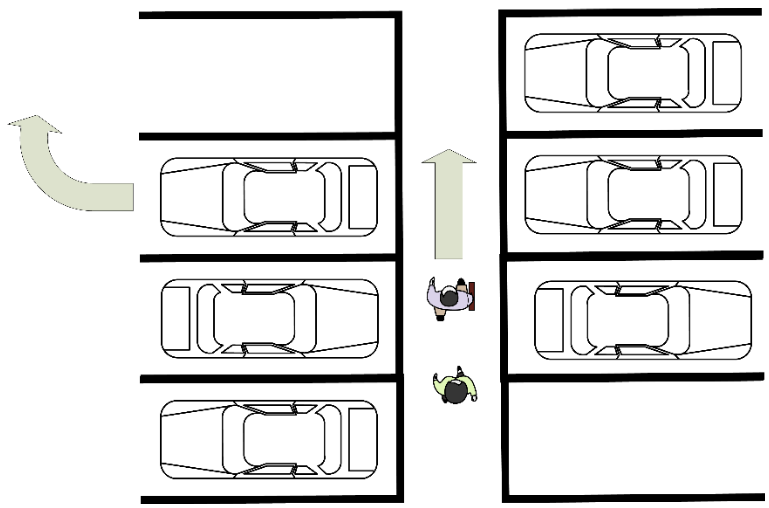


(c) moving in the opposite direction in a narrow aisle

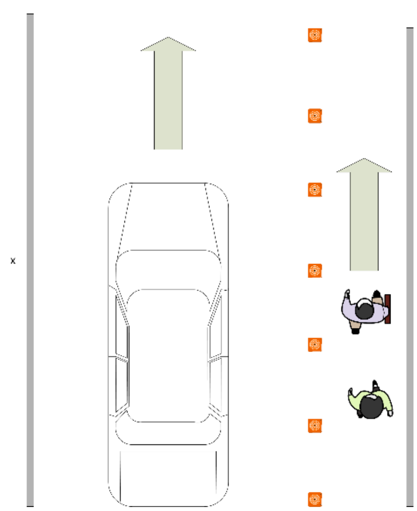
Figure 3-4 Vehicle-vehicle weaving duration in multiple scenarios

(c) *Reversing blind zone weaving duration.* When the vehicle reverses to unpark from a parking spot, the vehicle may weave with the passing-by pedestrians. This weaving spot and duration are measured for safety concerns.

Note that when the pedestrians move on the protected sidewalk, it does not count for the weaving between pedestrians and vehicles as shown in the following figures, where the orange cones are for the barrier of the protected sidewalk. These designs consume a portion of the land use and reduce the space efficiency and profit but would grant pedestrians the reserved right-of-way to eliminate the weaving of vehicles and pedestrians.



(a) The mid-strip sidewalk



(b) The barrier sidewalk

Figure 3-5 The protected sidewalk designs

Contending the safety issues, the safety design of the parking facilities could be improved by the following strategies. These strategies are not implemented in the proposed system but can be an extension within the proposed simulation framework.

(a) *Lighting*. It is suggested that all the pathways should be sufficiently covered with motion-sensor controlled lighting and cameras should be installed in the high-crime areas for video surveillance.

(b) *Clear-span construction*. To avoid possible collisions due to the narrow space, the building designers should reduce the numbers of the columns within the parking facilities for better visibility to minimize the potential hiding places.

(c) *Pedestrian accessible structure*. The parking facility designer should consider glass-backed elevators and open stairs for an open environment.

## 3.4 System integration

### 3.4.1 System structure

In modern software engineering, the user-orientated application is developed in the Software as a Service (SaaS) framework. Its advanced features benefit users including free for installation, up-to-date, cross-platform, and user-friendly. This structure continuously delivers state-of-art methods to users for testing and production. Based on the feedback from the user, the developer of the system is able to improve the simulation system and GUI for better user experience. Figure 3-6 shows the SaaS structure of the proposed parking simulation system.

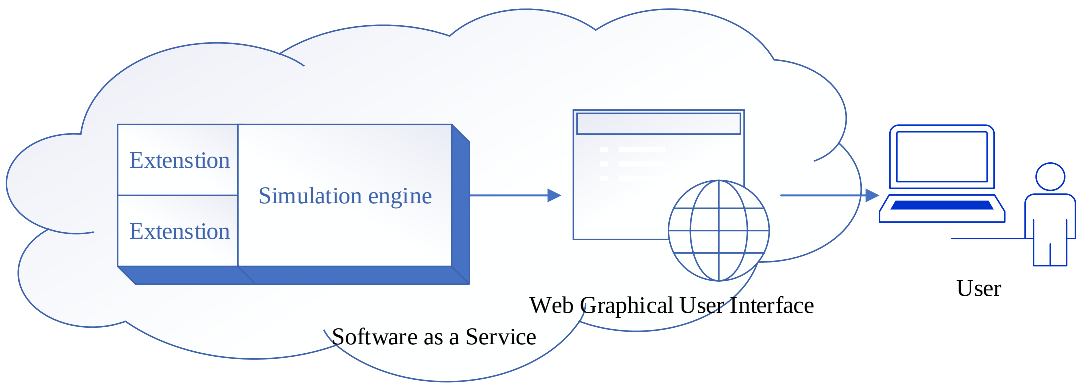



Figure 3-6 The software as a service framework for the proposed system.

From the view of software development, maintainability and extendibility are critical for a comprehensive system. Thus, the submodules should be decoupled. To replicate the parking mechanism, the critical problem is how to model choice behavior coupling with the traffic dynamic model. Figure 3-7 shows the proposed system includes the following modularized components: machine learning and deep learning, concurrent programming, and logic programming.

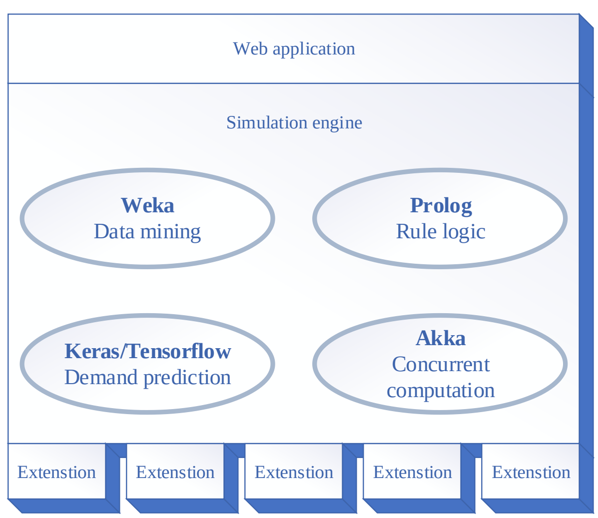



Figure 3-7 Decoupled the mechanism of the parking system

The implementation of the sketch-up simulation engine complies Object Oriented

Programming (OOP) principle. All entities and locations of the simulation system are modeled as serializable classes. Figure 3-8 shows a class diagram in the Unified Modeling Language (UML) of the proposed structure.

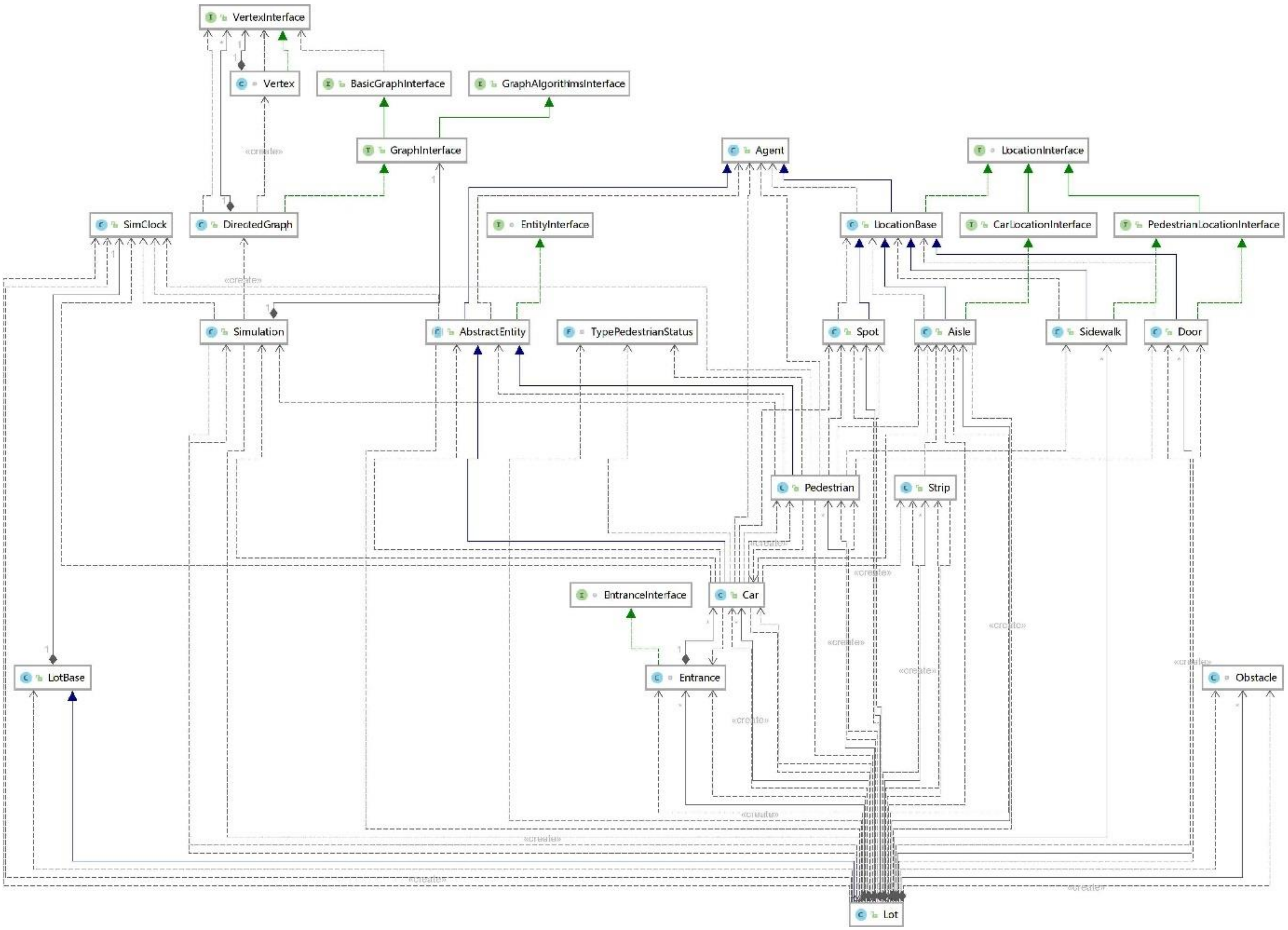



Figure 3-8 The class diagram of the core of the simulation engine in UML

### 3.4.2 Machine learning and deep learning

The proposed parking simulation system has adopted big data mining techniques to calibrate the demand arrival and departure and drivers and pedestrians decision-making models.

The conventional statistical methods capture the temporal patterns with time series models, such as Auto-Regression Integrated Moving Average (ARIMA). However, the estimation of time series models is not asymptotically efficient, the linear models cannot perform accurate prediction due to the heteroskedasticity (heterogeneousness of variation), and the nonlinear models such as Auto-Regressive Conditional Heteroskedasticity (ARCH) would (Hamilton, 1994; Wu and Min, 2005). The estimation of these models is considerable time-consuming for large-scale scenarios and the strong assumption such as normal randomness should be tested. In comparison to traditional time series models, the deep neural networks can facilitate the nonlinearity of the parking demand and the big data set.

In the literature, the Recurrent Neural Network (RNN), Long-Shor Term Memory (LTSM), Gated Recurrent Units (GRU) and Identity Recurrent unit (IRNN) neural network show great potential in capturing the temporal pattern in big data (Hochreiter and Schmidhuber, 1997; Ger et al., 1999; Graves, 2012; Gal and Ghahramani, 2016).

Based on these pioneer studies, RNN, LSTM, GRU are applied in long-term and short-term traffic prediction. Ma et al. (2015) employed the LSTM in traffic speed prediction using remote microwave sensor data. Tian and Pan (2015) used an LSTM approach for short-term traffic forecast. Zhao et al (2017) applied the LSTM in short-term traffic forecast. Fu et al. (2016) applied LSTM and GRU in traffic flow prediction. Yu et al. (2017) showed the deep approaches are able to predict traffic states under the extreme conditions. Chen et al. (2016) predicted traffic congestion with LSTM using online open data. Duan et al. (2016) showed predicted travel time with the LSTM networks. Vinayakumar et al. (2017) showed LSTM performed well in comparison to the other RNN methods. Zhuo et al. (2017) combined the LSTM and DNN and showed improved effectivity and accuracy.

To fast implement the proposed neural network, the state-of-art deep learning libraries are employed. To facilitate the time-dependent demand prediction, a Keras/Tensorflow library is used for implementing the Long-Short Term Memory (LSTM) neural network in demand forecasting. Tensorflow is an open source software library for high-performance numerical computation developed by the Google Brain team (Abadi et al., 2016). Keras (Chollet et al., 2015) is a high-level neural networks API on top of deep learning libraries including TensorFlow, CNTK (Microsoft Cognitive Toolkit, Seide and Agarwal, 2016), and Theano (Al-Rfou et al., 2016). Note that there is no single software tool that can outperform others (Shi et al., 2016). The proposed method can be fully implemented in other libraries.

This study used a scalable parking space choice behavior model to capture the drivers and pedestrian's parking behaviors and state-of-art machine learning techniques to calibrate the parameters, such as:

(a) Decision tree and random forest

The decision tree uses a tree-like model of decision and their possible consequences. Each node of the flowchart-like structure represents a yes-or-no question on an attribute, each branch represents the outcome of the test, and each leaf node represents a possible decision result. There exist several algorithms to build the decision tree such as Classification and Regression Trees (CART, Breiman et al., 1984) with Gini Index, and Iterative Dichotomiser 3 (ID3, Quinlan, 1986) with entropy function and information gain. The information gain is the measure of the difference in entropy from before to after the set $S$ is split based on an attribute A as defined in the following equation.

$$IG(A,S) = H(S) - \sum_{t\in T} p(t)H(t) \tag{13}$$

where $H(S)$ denotes entropy of set $S$, $T$ denotes the set of branches created from splitting $S$, $t$ denotes the subset of $S$ and $p(t)$ represents the proportion of the number of elements in $t$. The decision tree method has been applied to the parking simulation by Vo et al. (2016) and Li (2016). The random forest is constructed by a multitude of the decision tree to correct the overfitting on their training set (Hastie et al., 2008).

(b) Support vector machine

The Support Vector Machine (SVM) is one kind of supervised learning model for classification and regression analysis. The training of the SVM classifier amounts to minimizing the following quadratic programming problem.

$$min\ \frac{1}{n}\sum_{i-1}^{n} max(0,1 - y_i(wx_i - b)) + \lambda||w||^2 \tag{14}$$

subject to

$$y_i(wx_i - b) \geq 1 - \zeta_i, \forall i = 1,2, \dots, n \tag{15}$$

$$\zeta_i \geq 0, \forall i = 1,2, \dots, n \tag{16}$$

where $x_i, y_i$ denote the attributes and the label of the sample, $\zeta_i$ represents the smallest feasible nonnegative number, $\lambda$ is the coefficient of the margin hardness, $w$ is the slope and $b$ is the intercept of the linear margin function. The SVM can efficiently perform a non-learn classification using the customized kernel function to address the nonlinearity in the parking simulation system.

(c) Logistic regression

The logistic model or logit model or multinomial logit model is a widely-used statistical and machine learning model. The probability for choosing the specific option $i$ is the *softmax* function which is shown in the following equation.

$$Pr_i = \frac{e^{u_i}}{\sum_{k \in K} e^{u_k}} \tag{17}$$

where $Pr_i$ is the probability to choose the option $i$, the set $i$ is the set of potential options, $u_i, u_k$ are the utility of the corresponding option which is defined as the sum of weighted attributes.

The logistic model is also employed in parking choice models. Ji et al. (2009) proposed the multinomial logit model to capture the parking spot choice behavior from the global vision of the parking guidance system. Note that the multinomial logistic model for the parking spot choice is used when the decision maker considers the options at the same time. However, when the cruising-and-searching drivers do not have the full information of all potential parking spots, the local vision of the parking spots should be considered.

(d) Multilayer Perceptron

The multilayer perceptron is a class of feedback artificial neural network and is recognized as the vanilla neural network. It consists of three layers of nodes or perceptrons: the input layer, the hidden layer and the output layer, of which the weights of connections are trained by the backpropagation algorithm in a supervised learning manner. The linear activation function is applied to all neurons. In the extended version of the multilayer perceptron, the alternative nonlinear activation functions include the *logistic*, *rectifier* and *softplus* functions.

To implement the classification training and predicting in the proposed system, Weka (Hall et al., 2009; Frank et al., 2016), the open source machine learning and data mining library, is used for data processing, parameter calibration and the classification of parking spot choice. Weka contains tools for data pre-processing, classification, regression, clustering, and visualization.

The contribute is to investigate the application of the machine learning and deep learning techniques in the modeling of a parking spot, route choice behavior with classification methods and time-varying demand with recurrent neural network prediction methods.

### 3.4.3 Concurrent programming

When the scale of the parking facility is large, the simulation engine would be considerably time-consuming. Thus, the proposed system has a parallel computing version to speed up computing with the concurrent strategy.

For developing the concurrent simulation engine, the concurrent actor model is used for actor-based concurrent computing. Akka is a free and open-source toolkit and runtime simplifying the construction of concurrent and distributed applications on the Java Virtual Machine (JVM).

Actors are defined as computational elements for the concurrent computation. The actors wrap the non-concurrent data structures and interact with other actors via messages. The proposed implementation creates rules to organize blocking operations and addresses the concerns: (a) how to synchronize the actor actions in a concurrent environment, and (b) how to implement the mechanism of parking simulation, respectively. To incorporate the concurrency in the agent level, this study has the following findings:

(a) The synchronize protocol is for each time stamp, sending "move" messages to each actor and waiting until receiving feedback from all.

(b) The vehicle should ask the next location for occupancy before making movements. Then the "vision" (the driver may see only a subset of actors) and "choice" (the driver may make decisions from the environment information) behavior.

A sample case for illustrating the running process of the concurrent programming is shown in Appendix A. Figure 3-9 shows the sequence diagram of the proposed system in Unified Marked Language (UML) where the red rectangles represent the actors, the black baskets represent the procedures and loops, the solid arrows represent instant communication between actors and the dashed arrows represent the delayed messages between actors. In Figure 3-9, the domain objects and agents in the proposed simulation model are capsuled into actors, such as *lotActor, timeActor, locationActor, sourceActor, sinkActor, entityActor,* and *choiceActor*. The *lotActor* is the main node to manage the whole actor network, the *timeActor* maintains a simulation clock to synchronize all of the agents, the *entityActor* and *locationActor* are two main types of agents, the *sourceActor* and *sinkActor* captures the queuing model at the entrance and the exits, respectively, and the *choiceActor* is used to extract attributes and make decisions such as the parking spot. This sample presents the procedure of creating a network, creating entities, updating the state of entities for the parking simulation. This case demonstrates the actor structure in the parking simulation modeling and is extended to the full functional concurrent simulation engine.

In comparison to the nonconcurrent implementation, the concurrent implementation would reduce the running time in large-scale cases on multiple multicore workstations and reduce the occupancy of the system resource by allowing nonactive actors sleep such as parked vehicles. The contribution is to explore the capability of modeling the agent-based parking simulation model in a concurrent mechanism to achieve better computational performance and extensibility.

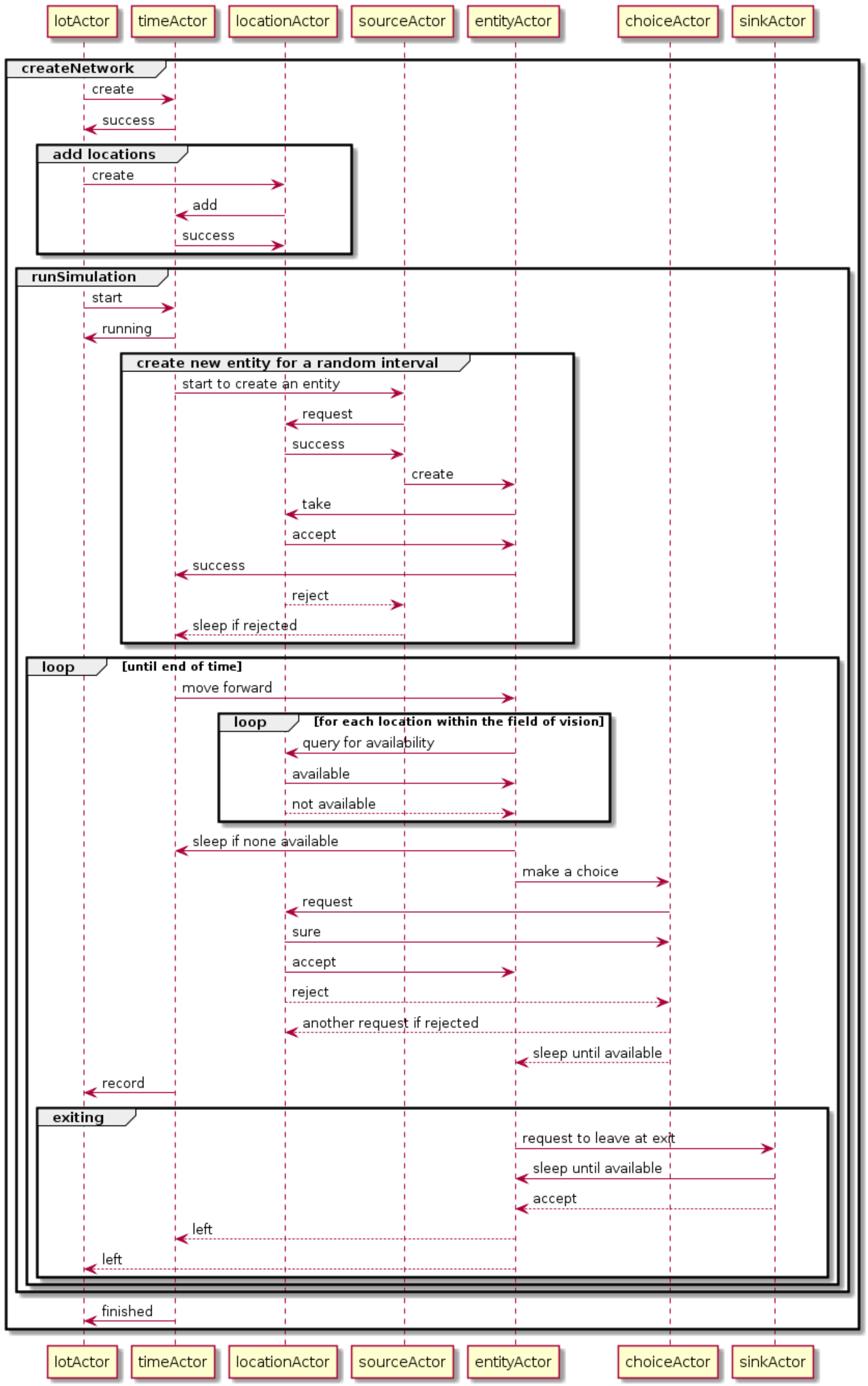



Figure 3-9 The sequence diagram in Unified Marked Language

### 3.4.4 Logic programming

As illustrated in Chapter 2, state transition and traffic dynamics of the parking system are rule-based, which is significantly different from the traditional traffic system. The motivation for using logic programming is to define the logical condition clearly. If the comprehensive parking system is modeled with the $n$ if-clause, there exist $2^n$ cases to be covered, which is impossible to implement and can hardly be maintained. Thus, the rule-based simulation model needs a scalable extendable framework instead of nested if-else-then rules due to the complexity concern. The proposed system presents a Prolog-like domain specific language to scale up the rule modeling.

Prolog is a modern logic programming language to present the first-order logic where the rules and states of entities and locations are programmed as $theories$ and $atoms$, respectively. Thus, the additional rules can be organized in Prolog instead of if-else clauses.

Prolog could figure out the entity movement deadlock in the language level since Prolog interpreters incorporate logic programming solvers. To solve the rules, Prolog introduces an advanced pattern-matching mechanism called unification. Two terms unify if there is some way of binding the variables that make them identical. For instance, given $parent$ is a user-defined predicate of arity 2, $parent(adam, Child)$ and $parent(adam, seth)$ unify by binding the variable $Child$ to the atom $seth$. One may check the rules by querying via unification. For example, the query $parent(adam, X)$ is used to find the solution $X = seth$. The logic programming semantics makes the declarative model language to avoid side-effects and keep interdependencies.

This study uses the syntax of Prolog to present the first-order logic model in the pseudocode, which does not require Prolog as the implementation language but declares a provable model for the rule-based parking simulation modeling. Note that $:-$ (turnstile) is the Horn clause operator of Prolog, _ (underscore) means any possible value, capitalized letters represent variables, the rule consists of the head expression, the turnstile, and the tail expression, and the tail expression may include multiple expressions separated with a comma. If the tail expression unifies with the head expression, the rule is satisfied and has true value. To find a feasible solution, the meaning of the Horn clause is that if the tail expression is true, the head expression is true. The built-in predicates, $retract$ and $assert$, are used for deleting and creating facts, respectively.

In this study, the parking mechanism rules are modeled in logic programming and are defined in Prolog-like equations which may be immediately runnable in Prolog interpreter with trivial supplemental codes.

A sample code is shown in Appendix B. The computational result shows the Prolog solver is able to deal with the proposed rule-based logic representation of the entity state machine and traffic dynamics. However, the original Prolog interpreters are in low computational efficiency.

Note that the first-order logic and the Prolog are only for representing rules for developing the parking simulation model, and the developed model could be implemented in any programming language. This study employs a Prolog implementation in Java and proposed a Domain Specific Language (DSL) to address the efficiency issue of the native Prolog interpreter.

In this study, new predicates are designed to better fit the domain used to extend the Prolog language. The predicate $vision\ (location, orientation)$ would detect the potential location and entities within the vision of the driver located in the parameter location and facing the direction the parameter orientation. The predicate $connect\ (location, action, goal)$ would construct the network by indicating the location, potential actions, and goals. The predicate $move\ (entity, location)$ would update the location and state of the entities. The predicate $choose\ (entity, locations)$ would generate a random choice based on proposed choice models.

The contribution is that this study originally proposes the logic programming model for modeling the entity state transition in the simulation system and the traffic dynamics in the parking system.

## 3.5 Software architecture

This section provides an overview of the proposed system from the aspect of software engineering.

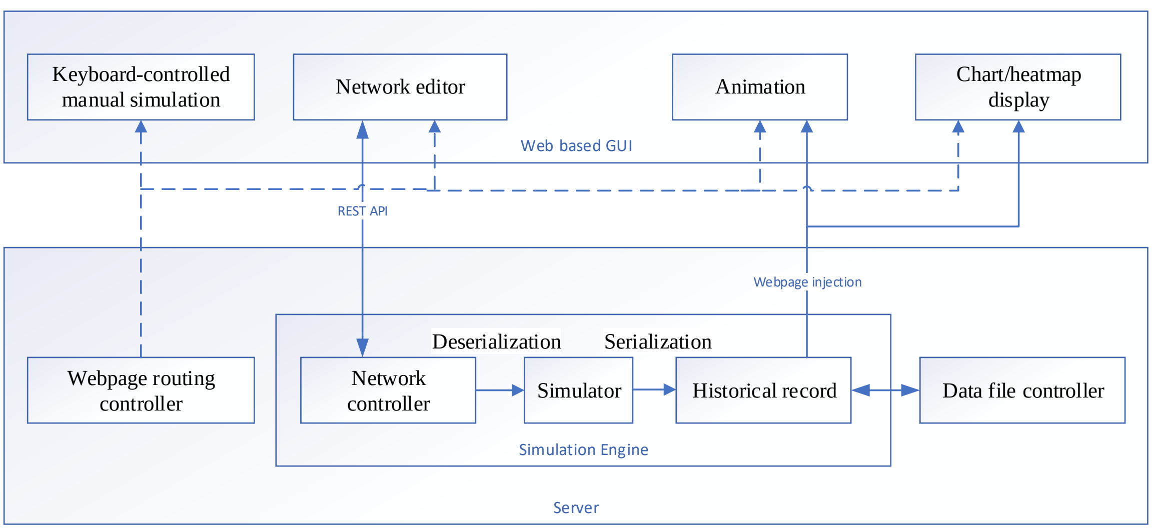



Figure 3-10 The software architecture of the proposed SaaS system

Figure 3-10 shows the software architecture of the proposed system, where the solid arrow is for the transfer of data, and the dashed line is the for transfer of webpage. The proposed system integrates four modules: (a) Keyboard-controlled manual simulation, (b) Network Editor,

(c) Animation, and (d) Result Display. In the Keyboard-controlled manual simulation module, the user can play a parking simulation minigame in a two-dimensional layout with the physical engine Box2D, and a script of the maneuvers of the vehicles can also provide the potential vehicle trace with the parking facility. In the Network Editor, the user can leverage the web-based GUI to overlay the locations and links on the parking facility layout. The Network Editor support the editing of the simulation model such as adding or removing the locations (including spots, aisles, pedestrian exits, entrances, obstacles, etc.), the customer type and distribution settings, the arrival, and departure distribution settings, and the simulation configurations. In the Result Display, the historical record can be illustrated in charts such as bar chart, line chart, and heat map with the Google Charts and Vue frontend libraries.

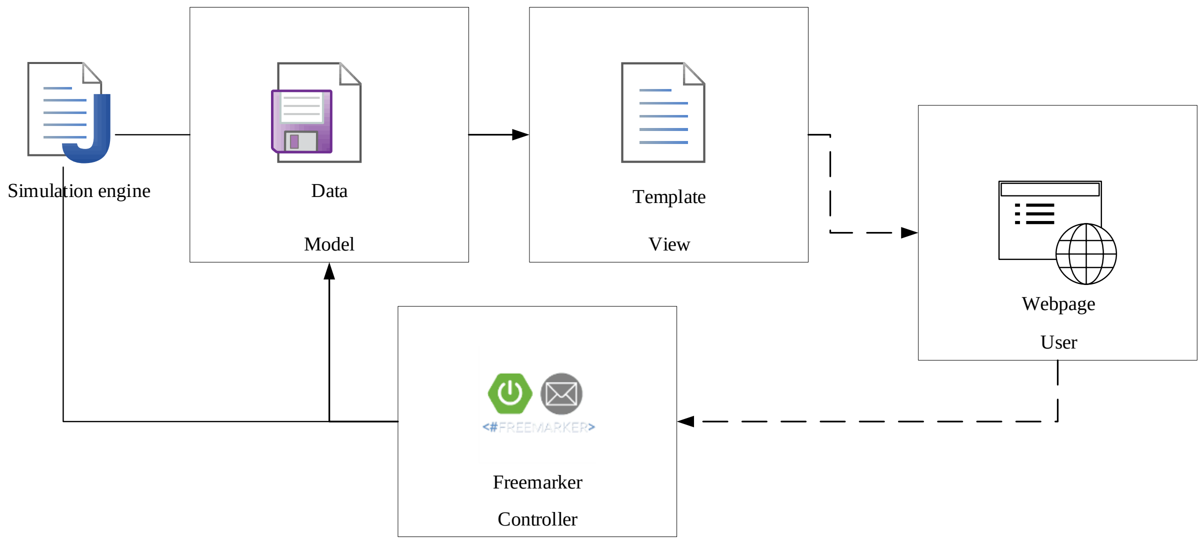



Figure 3-11 The server-side application structure

Figure 3-11 shows that the server-side application incorporates the webpage routing controller, the simulation engine, and the data file controller. The controllers support the Representational State Transfer (REST) utilities of the server, where the resources are accessed via Hypertext Transfer Protocol (HTTP) API.

The coding of the proposed system includes multi-languages (a) Java for the server-side application and the simulation engine, and (b) JavaScript/HTML5/CSS3 for the browser-side application and the user interface. The proposed system is implemented in the cutting-edge programming techniques: (a) Software-as-a-Service (SaaS) browser-server software structure, where the most updated features can be feed to the users, (b) serialization/deserialization for the persistence of memory in object-oriented programing, where the deserialization means the structural data and record files of inputs and outputs in JSON/XML format are converted to memory objects and the serialization means the conversion in the other direction, (c) the Spring Framework for Model-View-Controller (MVC) development framework, where the web services and inversion of control container for the Java platform is managed, and (d) Website Templates Injection, where the templates are composed with Apache FreeMarker and the data in the webpage are changed in the server side when requested. These features enable the flexibility and expandability of the proposed system.

# 4 A microscopic agent-based parking system simulator

## 4.1 Testbed

To describe the capability and modeling of the proposed system, the illustration of the proposed microscopic agent-based simulation model is aided by the studied real case. In this case, the studied data are collected from transaction record and field survey. Based on the solid data, the parameters are calibrated to evaluate the performance of the proposed model. Figure 4-1 shows a satellite photo of the University of Wisconsin-Milwaukee (UWM) Sciences surface parking lot, which is used to illustrate the proposed method in this section.

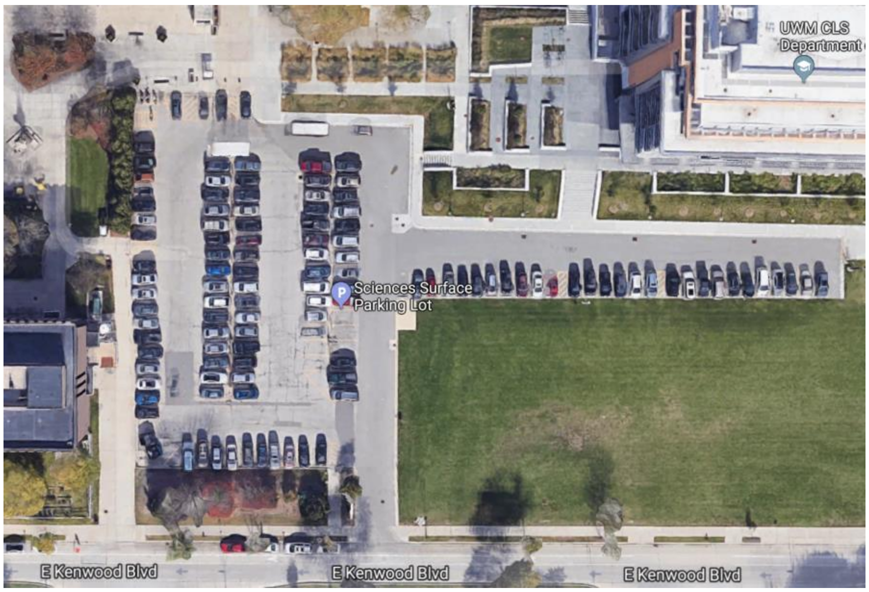



Figure 4-1 The satellite photo of the Sciences Surface Parking Lot

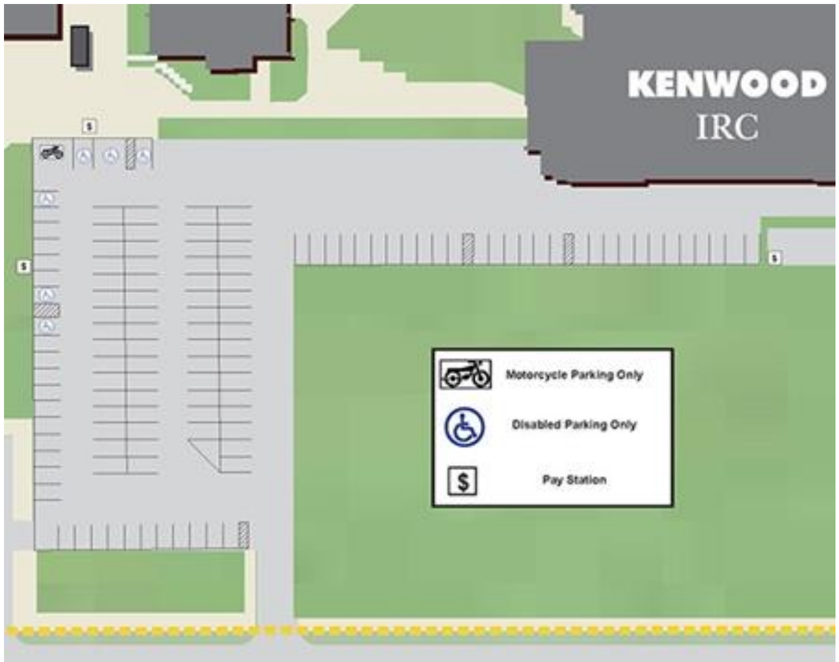



Figure 4-2 The layout of the studied UWM Sciences Surface Parking Lot

The Transportation Department provides the layout of the parking lot. Figure 4-2 shows the layout of the UWM Sciences surface parking lot, which is used as the case to illustrate the proposed method. The to-scale layout of the parking lot is imported as the background picture of the simulation network. The simulation network can be established with the web-based editor. Figure 4-3 shows the simulation network for the case established in the web-based editor, where the rectangle is for spots, the square is for aisles, the edge is for connections between locations, the solid arrow is for the entrance, the walking icon is for pedestrian exits, and the wheelchair icon is for the handicapped-only spot. These spots and aisles can be drawn either one by one or by batch.

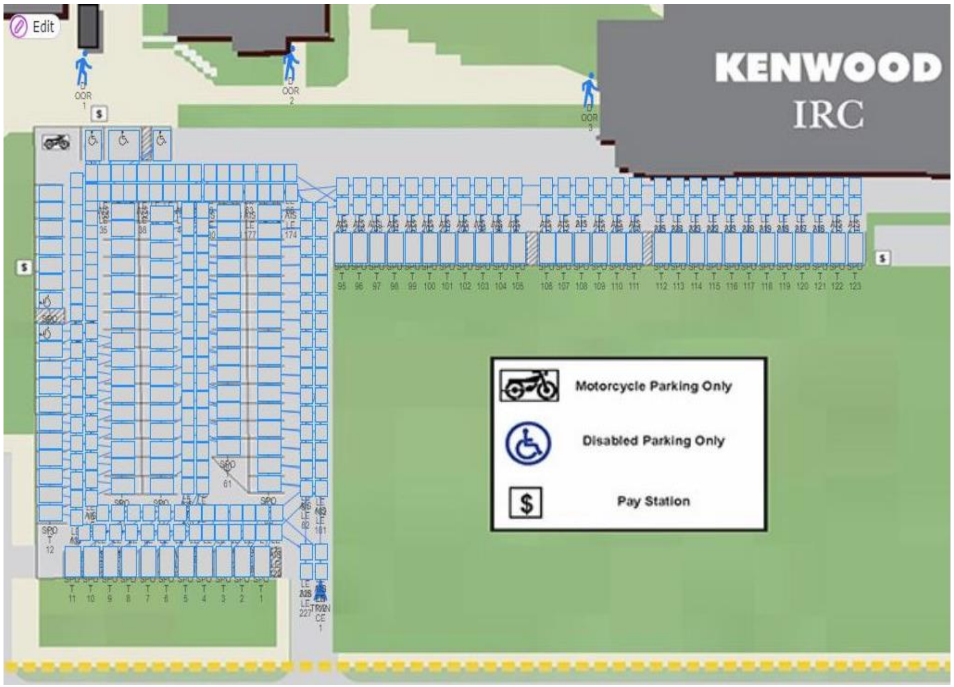



Figure 4-3 A Screenshot of the sample parking facility modeling in the proposed system

## 4.2 Data collection

The accuracy depends on the quality of the specification of the model and the accuracy with which data used to calibrate the validate them can be collected (Young et al, 1989). Table 4-1 shows the general information collected in the parking lot survey.

Table 4-1 Parking lot survey items

| Category | Attributes | Data Type | Item |
|---|---|---|---|
| Basic | Type of the parking lot | Multiple choices | Mixed-use |
| | | | Public |
| | | | Commercial |
| | | | Curb-side |
| | | | Occupying sidewalk |
| | | | Under-interchange |
| | | | Residential |
| | | | Parking-and-ride |
| | Management | Filling the blank | Name of managing corporation |
| | | | Type of the managing corporation |
| | Infrastructure | Filling the blank | Guiding sign |
| | | | Automated charging device |
| | | | Pedestrian management |

| | | | |
|---|---|---|---|
| | | | Others |
| | Pricing | Filling the blank | Long-term seasonal permit<br>Short-term permit<br>Time-of-day ticket<br>Available period |
| Building | Building basic | Filling the blank | Type of ownership<br>Construction year<br>Total area |
| | Propensity | Filling the blank | Number of beds if the building belongs to hospitals<br>Number of rooms if the building belongs to hotels<br>Number of faculty and students if the building belongs to schools<br>Number of tables if the building belongs to restaurants<br>Number of seats if the building belongs to stadiums |
| Supply | Number of spaces | Filling the blank | Number of spaces<br>open spaces<br>the total area of spaces |
| | Number of constructions | Filling the blank | Number of surface spaces<br>Number of underground spaces<br>Number of automated spaces |
| Demand | Peak hour demand | Filling the blank | Total number of vehicles<br>Parked vehicles in the spaces<br>Traffic influence<br>Emergency aisles<br>The position of parking (grassland, sidewalk, aisle, open space, etc.) |
| | Overnight demand | Filling the blank | Total number of parked vehicles at night |
| | Operation | Filling the blank | Turnover rate<br>Average parking time<br>utilization |
| Environment | Photo and video | Filling the blank | On-site pictures<br>Pictures of surroundings<br>Videos of traffic status |

The raw real-world data include the transaction records from the University Transportation Department (see a sample in Appendix C) and the field survey data (see Appendix D). The data spans from October 2016 to June 2017. service. In this case, there’re two kinds of parking service customers in term of payment method: (a) ticket(drive-in) user, paying the parking fee each service, and (b) permit (reserved) user, paying for a seasonal permit for weekly or monthly or semesterly. In some other cases, there may be a non-reserved seasonal user.

Table 4-2 Data format of the parking facility ticket transaction data

| Attribute | Description | Example |
|---|---|---|
| Gross Amount | the prepaid amount of parking fee | $2.00 |
| Net Amount | the actual amount of parking fee | 2 |
| At ID | the ID of the parking ticket machine | 0 |
| Rate | the rate of parking fee, varying between normal customers and the handicapped | 1 |
| Ticket Type | entry or exit | Entry |
| Transaction Type | the normal or handicapped customer | Normal |
| Ticket Number | the label of the ticket | 39177 |
| Entering Time | the timestamp when the vehicle arrives | 2017/05/03 08:47:00 |
| Exiting Time | the timestamp when the vehicle exits the garage | 2017/05/03 09:18:00 |
| Transaction Number | the label of this transaction | 494 |
| Device name | the label of the ticket machine | Lubar Pay Station |

Table 4-3 Data format of the parking facility permit usage data

| Attribute | Description | Example |
|---|---|---|
| Card Number | the label of the customer | 104509 |
| Date and Time | the timestamp when the customer arrives | 5/3/2017 9:31 |
| Reader Label | the label of the machine | Lubar Rev. Entry |
| Lot Label | the label of the garage | 3 |
| Direction | in or out the garage | In |
| Result | the result of checking the validity of the customer | Valid Access |

| | | |
|---|---|---|
| Allowed | whether the customer is allowed to the garage | Yes |

Sample transaction data are attached in Appendix C. The field survey data include parking choice behavior data and demand data. Sample transaction data are attached in Appendix D. The choice behavior file is in CSV format. The default attributes are defined in Table 4-4. One can define the own attribute and use the defined data file to calibrate the model. The prediction of the calibrated model is subject to simple modification of the simulation code.

Table 4-4 Parking space choice behavior data collection

| Attribute name | Definition | Value Type | Example |
|---|---|---|---|
| walkingdistance | walking distance in meter from the spot to a pedestrian exit | Double | 10 |
| traveldistance | driving distance from the entrance to the spot | Double | 20 |
| lanestatus | if the strip in front of the spot is occupied or clear | String | UNOCCUPIED, OCCUPIED |
| spotstatus | if the left or right neighbor of the spot is occupied | String | RIGHT, LEFT, CLEAR |
| class | the choice for the spot, 1 for chosen, 0 for unchosen | Integer | 0, 1 |

In view of the collected data, the parking lot demand can be measured with the following quantities: the number of arrivals within a period, the number of departures within a period, the inter-arrival time, the inter-departure time, and the parking duration. With the advances in the parking management and data collection systems, there exist more convenient technologies, such as precise vehicle positioning system, than the manual counts and drawing. The proposed data processing method would fit the various data sources to fulfill the calibration of the parking simulation model.

## 4.3 Descriptive analysis

The parking service shows long-term heterogeneousness in arrival and exiting distributions as well as the short-term variability. To simplify the problem, the distributions are assumed to be homogeneous in each time interval. In the parking simulation system, the arrival and exiting distribution should be labeled by time period. Ignoring this pattern would induce errors in the simulation. For example, if the daily distributions are used as the inputs for a morning peak hour simulation, the results would underestimate turnover rate and active vehicles and overestimate the occupancy.

The parking demand on campus has great season-dependent patterns. The following demand patterns are observed: (a) The gap between semesters has much lower demand in parking on campus and the demand is significantly impacted by the semester and vacation time. (b) The daily distribution of parking demand is more retractable than the monthly and weekly distribution. (c) The number of ticket user is considerably greater than the permit user. This is reasonable since most of the customers of the on-campus garages are college students. The propensity of parking on campus is high, but the users are not willing to buy permits since the permit is expensive than the ticket if they don’t have to park on campus every day.

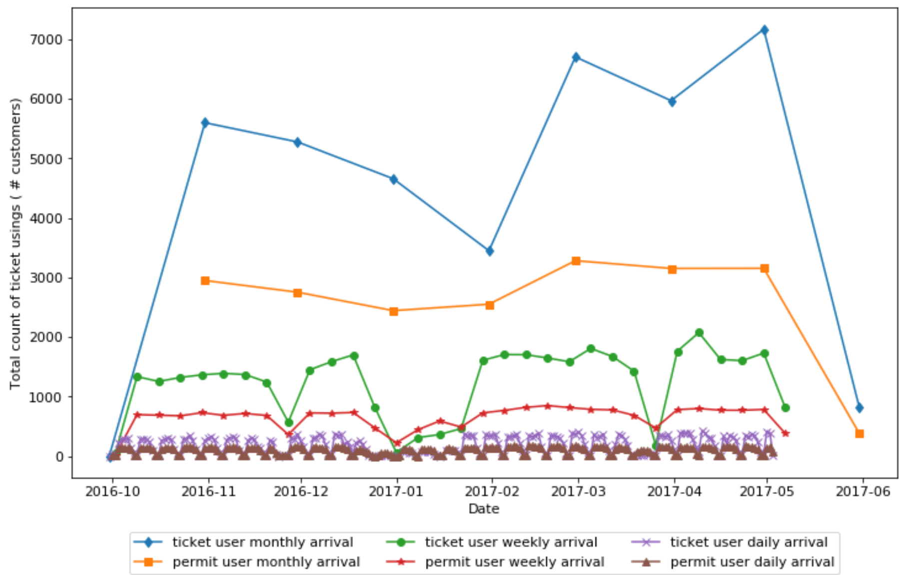



Figure 4-4 Time-dependent arrival pattern for ticket and permit users

Figure 4-5 and Figure 4-6 shows the daily arrival distribution of ticket users during two weeks in the middle of the semester. The weekly demand pattern is further addressed. There're two kinds of demand patterns: two-peak pattern from Monday to Thursday and one-peak from Friday to Sunday.

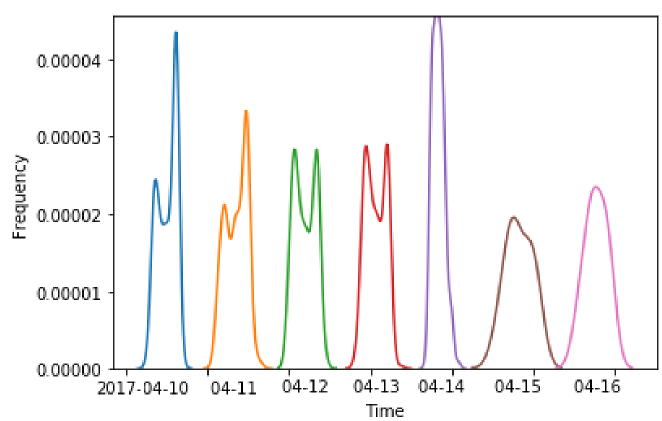



Figure 4-5 The daily arrival distribution of ticket users during the week 2017-04-10 to 2017-04-16

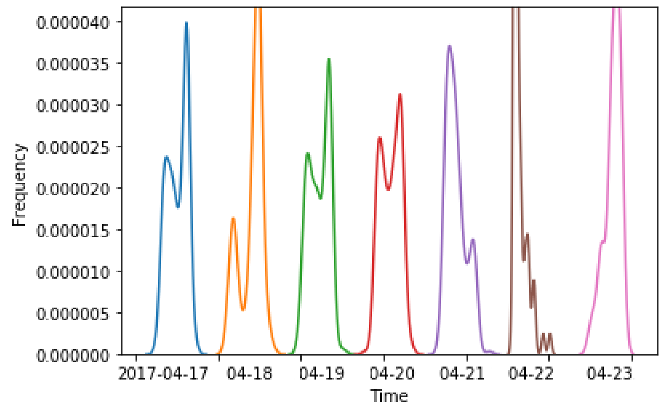



Figure 4-6 The daily arrival distribution of ticket users during the week 2017-04-17 to 2017-04-23

The daily parking demand pattern depends on neighboring points of interests (POI) such as office, retail, theater, hotel, hospital, and university. Figure 4-7 illustrates a one-day arriving and departure patterns of the ticket and permit users in a garage on campus. Figure 4-7 shows the following descriptive analysis: (a) the one-day arrival distribution is centralized in the a.m. peak hours. (b) The permit and ticket users exiting are distributed heterogeneously, and the permit users park significantly longer in the garage than the ticket user. (c) The arrival and exiting peaks are overlaps in the a.m. peak hours.

From the interview survey to the manager of the University Transportation Department, the following ideas are learned: (a) the garage manager would prefer to sell more permits than tickets since permits are prepaid and fewer efforts are needed to manage the permit users. (b) To encourage the permit users, the permit users are guaranteed to have a vacant spot when arriving. (c) To achieve this goal, the garage manager should reserve several spots in the a.m. peak hour, especially in the rainy morning on Mondays. Thus, the proposed simulation system could help the garage manager to investigate the peak hour arrival and exiting pattern and help decision making.

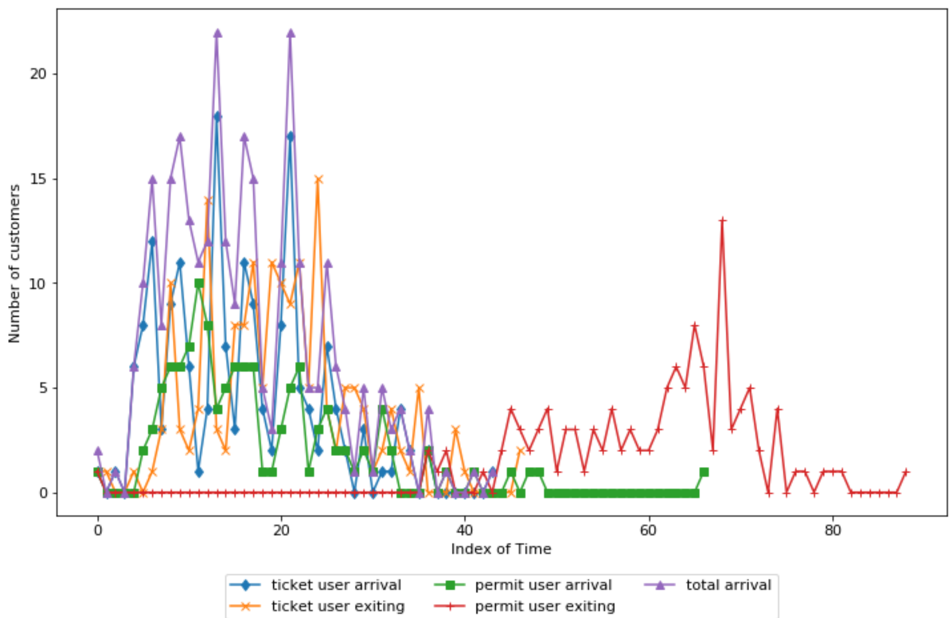
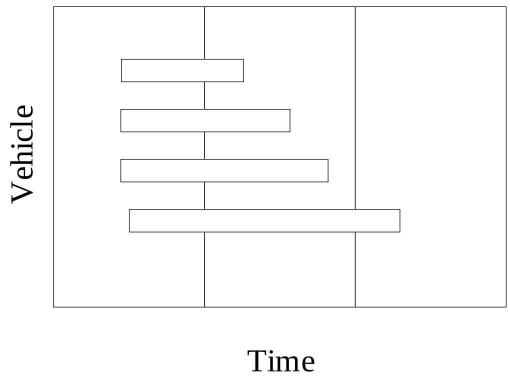
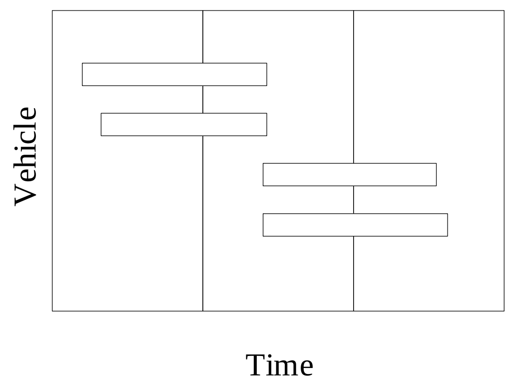



Figure 4-7 Daily arrival and exiting counts for the ticket and permit users on 2017-04-10

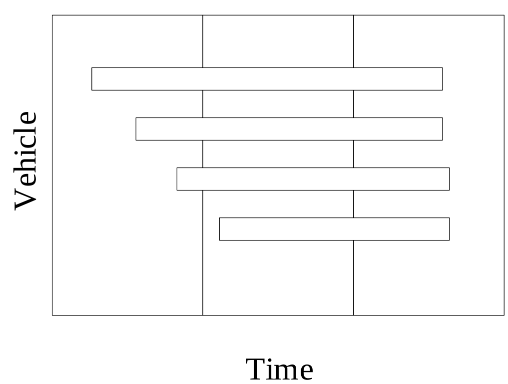
From the daily arrival and existing record, the demand pattern is learned to explain the appearance of peaks in arrival and existing counts. Figure 4-8 shows the demand diagram in various scenarios, where the x-axis is for the time span, the y-axis is for the label of the arrival vehicle, each bar represents a vehicle arriving at the head of the bar and exiting at the tail of the bar. Note that the demand pattern depends on the neighboring land use of the parking lot, and in this case, the demand pattern is impacted by the university travel pattern. Figure 4-8 (a) shows the daily demand pattern with a morning peak around 9:00 a.m. to 10:00 a.m. because most classes are scheduled starting during this period from Monday to Thursday. Figure 4-8 (b) shows the demand pattern with one peak around noon from Friday to Sunday. Figure 4-8 (c) shows the demand pattern with a surged exiting peak on the special event day. For example, the building holds the Poster Competition which attracts far more vehicles than the daily vehicle counts. When the event is held in the other building, the parking manager can refer to the history parking demand pattern for the special event parking management. Figure 4-8 (d) shows the parking duration of the seasonal permit users, which implies a long parking duration and normally distributed arrival and leaving counts on the whole day.



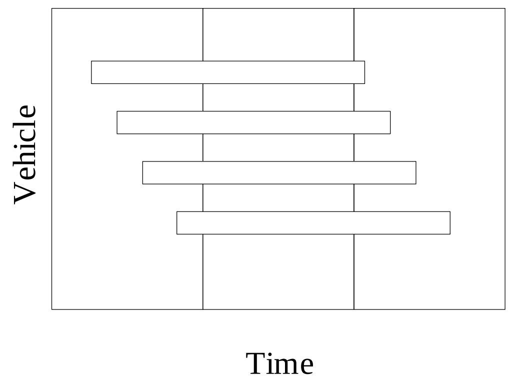

(a) The morning peak pattern




(b) The mid-day peak pattern




(c) The special event peak pattern




(d) The permit user pattern
Figure 4-8 The parking demand pattern of the studied case

## 4.4 Demand distribution calibration and experiments

This section shows a real case for calibration of the hypothetical probability parameters. To collect the data, a camera is set in front of the IRC building, where arriving area, exiting area and route of the car in the parking place can be recorded very clearly. The 1-hour video is recorded from 8:30 am to 9:30 am on one Wednesday morning, when is the busiest hour during the week. After recording, data need to be sorted. The arrival time of each car is recorded then the inter-arrival time is calculated. During 1-hour observation, there are totally 71 cars arrived including 32 passing-by cars, 1 motorcycle, and 38 normal cars.

In order to find which distribution the inter-arrival time follows, a histogram is created to make the assumption of the distribution of the data. Figure 4-9 shows that it is assumed that the inter-arrival time follows the exponential distribution, where the blue curve is drawn from the possible exponential distribution.

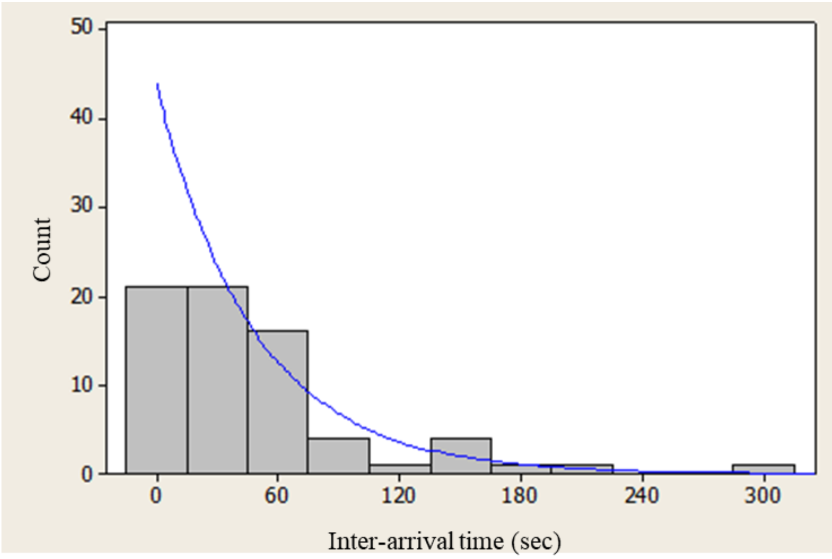
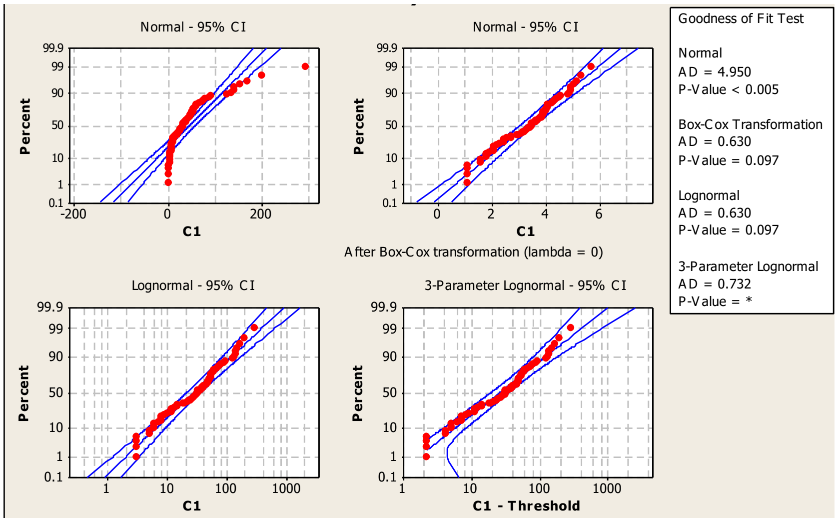
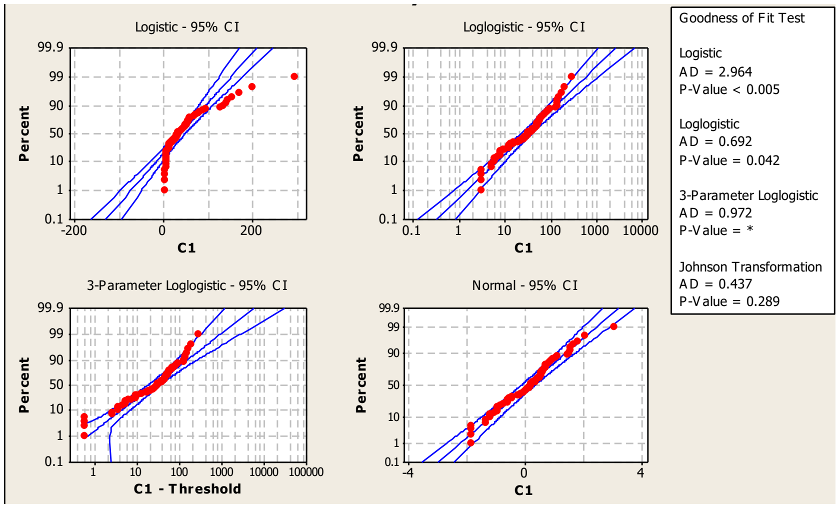



Figure 4-9 The histogram of the inter-arrival duration

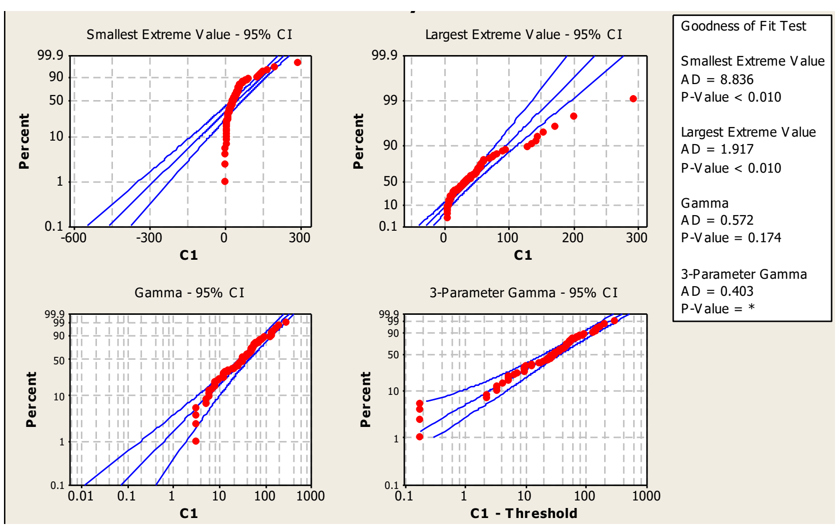
A $\chi^2$ goodness of fitting test is performed to check whether the data follows an exponential distribution. Figure 4-10 shows the results of the goodness of fitting test, which shows that the inter-arrival is distributed exponentially with the mean 48.01 sec.



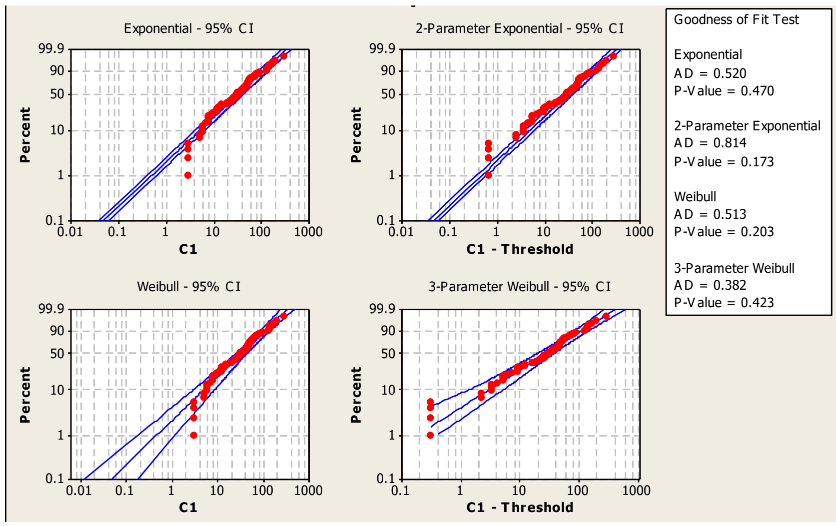

(a) Rejecting the normal, Box-Cox, lognormal, 3-parameter lognormal distribution assumption




(b) Rejecting the logistic, log-logistic, 3-parameter log-logistic, Johnson Transformation distribution assumption




(c) Rejecting the smallest extreme value, large extreme value, Gamma, 3-parameter Gamma distribution assumption




(d) Accepting the exponential distribution assumption and rejecting the others

Figure 4-10 The goodness of fitting of the interarrival distribution

The parking duration is tested with the distribution identification analysis, however, the results show that the parking duration of cars doesn't follow any hypothetical probability distribution. Figure 4-11 shows the histogram of the used empirical distribution. The proposed simulation system is capable to calibrate the empirical distribution on the raw data.

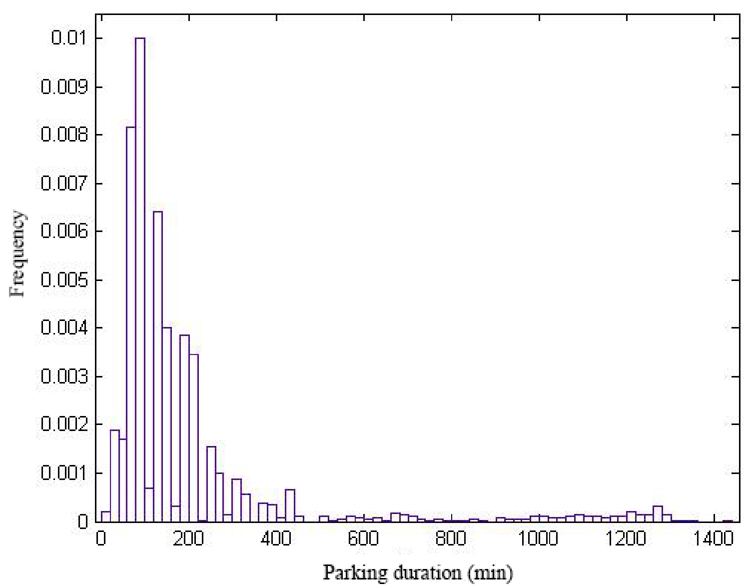



Figure 4-11 The empirical distribution of the parking duration

In the experimental part, the calibrated model is tested with various inter-arrival time distribution parameter values. Various mean values of exponential distribution are tested for experiments. Supposing there's a special event holding in the near buildings, demand may increase sharply and cause problems in a real-world situation. Figure 4-12 shows the maximal number of moving vehicles increases when the mean of interarrival distribution increases, and converges to the value 10 vehicles if the mean of the interarrival distribution is lesser than 24.

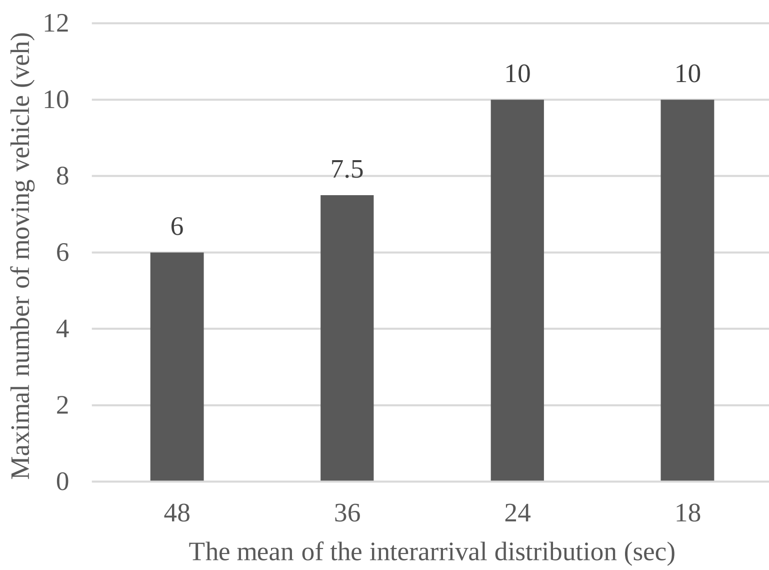



Figure 4-12 The experiment results for finding the active capacity

The active capacity is defined as the maximal number of concurrently moving vehicles indicates the arrival pattern captured by the simulation model. If the maximal number of concurrently moving vehicles is too large, deadlocks may occur. From this experiment, the active capacity of this parking lot is found to be 10 vehicles, however, the active capacity in the real-world scenarios is lesser than the simulated result due to the external factors such as weather.

## 4.5 Predicting dynamic demand

In the literature, Caicedo et al. (2012) proposed a method for predicting real-time parking space availability in intelligent parking reservation systems, which was based on a calibrated discrete choice model for selecting parking alternatives to allocate simulated parking requests, estimate future departures, and forecast parking availability.

From the descriptive analysis, the demand for parking facilities is found to have the following features: (a) The demand for parking facilities is stable in a short period (e.g. peak hour). To address this feature, the hypothetical distributions can be calibrated of multi-period data after the simulation starting from not empty (i.e. warming-up period). The demand for parking facilities is dynamic from time to time in the long period. Regarding this aspect, the prediction of series applies with model-free techniques and times series analysis with assumptions.

Due to the heterogeneousness, the arrival and exiting counts vary from time to time and from day to day, and the parameters of the stochastic process cannot be regressed asymptotically. The deep learning approaches are able to overcome these issues with sufficient data. Figure 4-13 shows the neural network structure for predicting the parking arrival and departure counts, where the tuples in the right side refer to the shapes of input and output tensors in each layer. The employed neural network is sequentially constructed, where the input layer is followed by four LSTM layers, and there is one dense (i.e. full-connected) layer connecting the LSTM layers for outputting the predicted results. The neural network has the following settings: (a) the size of each LSTM layer is 256, which is determined by rule-of-thumb, (b) the activation function of each LSTM layer is *softsign* (Bergstra et al., 2009), which is chosen by trial-and-error method, (c) the loss function is the mean squared error, which is determined in view of the real value output, and (d) the optimization algorithm is selected as *Adam* (Kingma and Ba, 2014). The *softsign* activation function is shown in the following equation.

$$f(x) = \frac{x}{1+|x|} \tag{18}$$

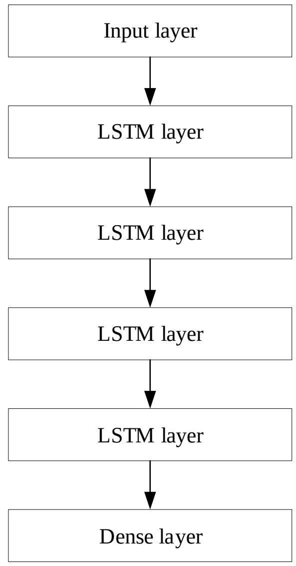
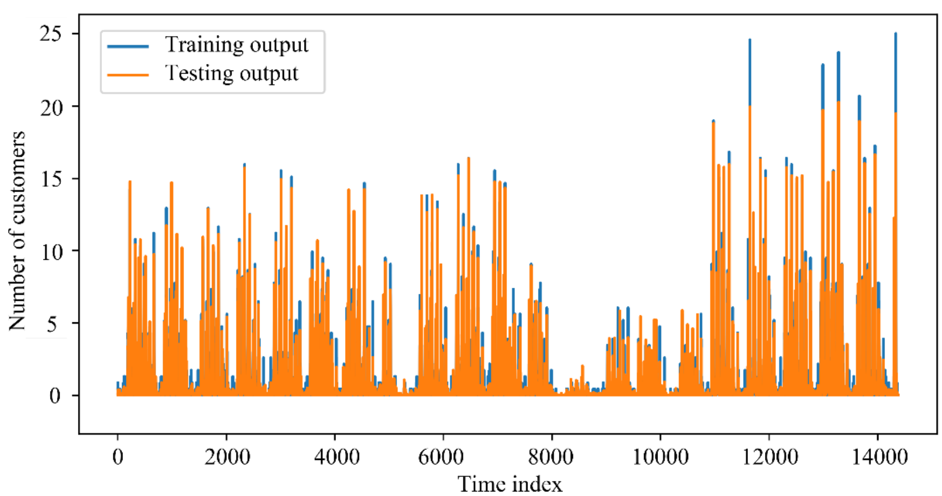
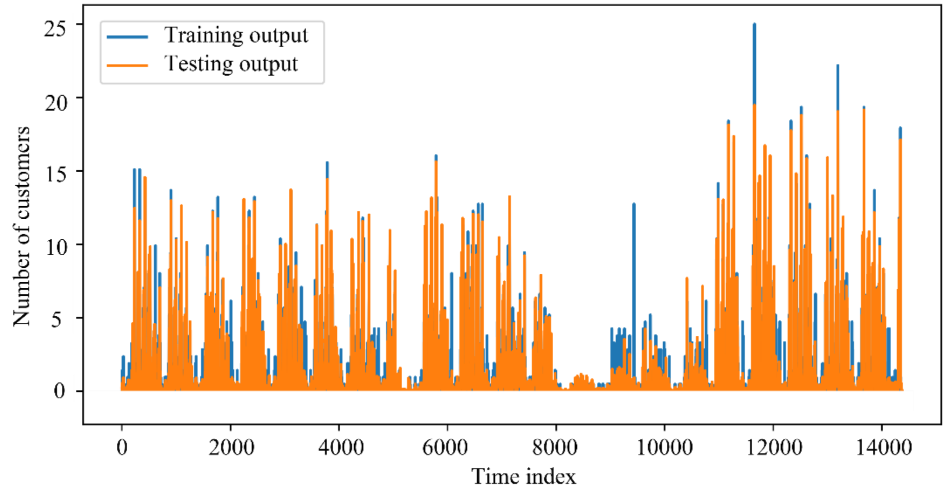



Figure 4-13 The network structure for parking arrival and departure counts

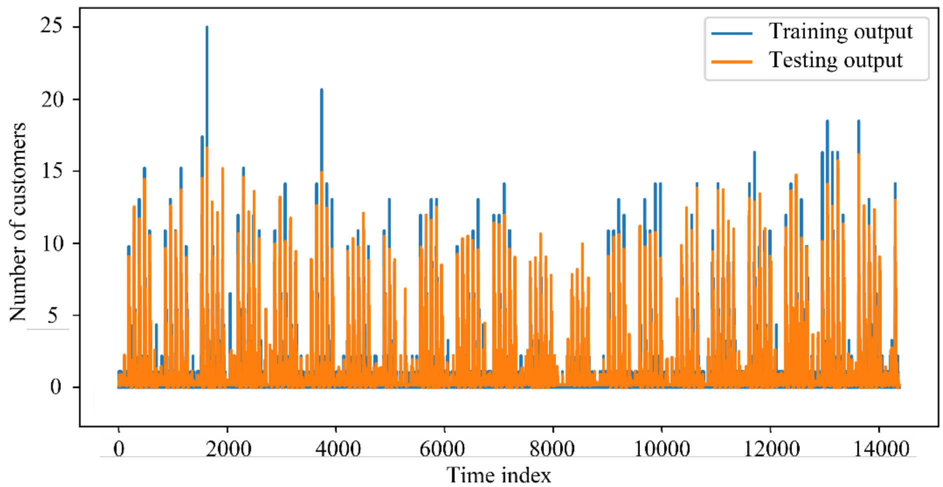
The data set is split into the training set and the validation set. Figure 4-14 shows the trained results and the tested results. The R-squared measure of the training set is 0.91, and one of the testing set is 0.90. The results prove the effectiveness of the proposed LSTM neural network.



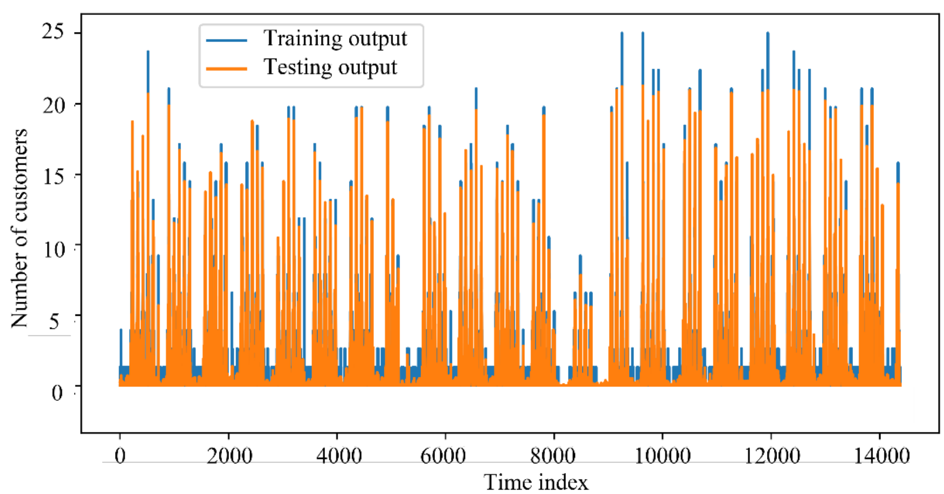

(a) Permit user arrival counts




(b) Permit user exiting counts




(c) Ticket user arrival counts




(d) Ticket user exiting counts

Figure 4-14 The prediction of arrival and exiting counts of customers in 15 minutes intervals.

The previous parking simulation system did not consider the impact of the varying demand to the operation of the parking facilities. This section contributes to a dynamic demand prediction for the parking simulation in view of the short-term system variability. This study identifies this critical problem for preparing best preplans of the parking facility management. The online prediction and simulation method would benefit the development of the time-sensitive plans, such as the temporal permit-only policy of the parking facilities.

## 4.6 Mathematical notation

This section formulates a general logic programming model for the parking simulation problem. This study uses the following notation:

$\mathbb{M}$ = space of entity movements;

$\mathbb{P}$ = space of entity processes;

$\mathbb{S}$ = space of agent attributes;

$\mathbb{T}$ = space of time;

$\mathcal{A}$ = set of agents;

$\mathcal{E}$ = set of entities;

$\mathcal{V}$ = set of vehicles;

$\mathcal{P}$ = set of pedestrians;

$\mathcal{L}$ = set of locations;

$a$ = index of the agent;

$\boldsymbol{w}$ = the vector of random variables;

$t$ = a state of time;

$s$ = a state of the agent;

$p$ = a process of the agent;

$\mathcal{M}$ = a movement of the agent;

$n$ = the number of random variables;

$\mathcal{F}$ = a state machine;

$\mathcal{G}$ = a choice model;

$\mathcal{H}$ = a performance measure;

In this study, the agents are defined as the interactive physical objects with dimensions, the entities are defined as the agents with actions and birth-death processes, and the locations are defined as the agents can be affected by other agents, are only created at the beginning of the simulation and never die.

The attributes of an agent include the length, width, height, three-dimensional position, orientation etc., and can be further described by the Equation (31).

$$\mathbb{S} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R} \times \cdots \times \mathbb{R} \tag{19}$$

Equation (32-35) show the definition of the process space and the movement space in the proposed simulation model. The transitions of the states in those spaces are defined in the process diagram in the next section. The time-space is defined as the discrete simulation time span. The space of processes and the space of agent movements are shown in Equation (32)-(33).

$$\mathbb{P}_{\text{vehicle}} = \{arriving, entering, cruising, reneged,$$

$$intospot, parked, unparking, leaving, left\} \tag{20}$$

where the process $arriving$ means the vehicle arrives at the entrance of the parking lot and joins the end of the entry queue; the process $entering$ means the informed drivers decide the destination parking spot at the entering in the system and travel in the shortest path; the process $cruising$ means the uninformed drivers move and search the current vision for the potential spot; the process $reneged$ means the driver of the last arrival is not willing to wait in the entry queue and leaves the entry queue after waiting for a time interval; the process $intospot$ means the vehicle moves from the aisle into the spot and stops the engine; the process $parked$ means the vehicle is parked in the spot till the duration of being parked is up; the process $parking$ means driving into the parking spots from the aisle, the process $unparking$ means driving out of the parking spot and merging into the aisle traffic; the process $leaving$ means the vehicle heads the exit of the parking lot and moves across the parking lot, and; the process $left$ means the vehicle has left the system and is moved to the historical list.

$$\mathbb{P}_{\text{pedestrian}} = \{no\_opt, to\_door, to\_be\_back, to\_vehicle, pass\_by\} \tag{21}$$

where the $no\_opt$ means the pedestrian is moving around in the system without destination such as children playing in the parking lot, $to\_door$ means the pedestrian is moving towards the pedestrian exit, $to\_be\_back$ means the pedestrian is temporarily not in the system until the parking duration is used up, $to\_vehicle$ means the pedestrian returns to the system and walks towards the parked vehicle, and $pass\_by$ means the pedestrian is moving across the parking lot without the use of the parking spot.

$$\mathbb{M}_{\text{vehicle}} = \{arrive, yield, forward, turn, reverse\} \tag{22}$$

$$\mathbb{M}_{\text{pedestrian}} = \{arrive, yield, forward, turn, random\_moving, on\_route, merge\} \qquad (23)$$

The random vector $\boldsymbol{w}$ incorporates the random demand distribution and the random choice models for parking space choice and path choice. And $n$ is the dimension of the random vector such that $n = |\boldsymbol{w}|$.

The state machine is defined of the mapping of the state-action space to the state-action space, as shown in Equation (36).

$$\mathcal{F}: \mathbb{T} \times \mathbb{S} \times \mathbb{P} \times \mathbb{M} \times \mathbb{R}^n \to \mathbb{T} \times \mathbb{S} \times \mathbb{P} \qquad (24)$$

The state machine is applied to all the entities, as shown in Equation (37), where the letter with the prime symbol indicates the state in the next time step.

$$(t', s', p') \leftarrow \mathcal{F}_e(t, s, p, m|\boldsymbol{w}) \quad \forall e \in \mathcal{E} \qquad (25)$$

The interaction with other agents is defined as the functions of the state-action-random space as shown in Equation (38).

$$[(t'_e, s'_{e'}, p'_{e'}), (t'_e, s'_e, p'_e)] \leftarrow \mathcal{I}_{ee'}[(t_e, s_e, p_e), (t_{e'}, s_{e'}, p_e)] \quad \forall e, e' \in \mathcal{E} \qquad (26)$$

Choice models are defined as the machines for generating random variables and process as shown in Equation (39).

$$\mathcal{G}: \mathbb{T} \times \mathbb{S} \times \mathbb{P} \to \mathbb{R}^n \times \mathbb{P} \times \mathbb{M} \qquad (27)$$

The performance measurements are defined as the functions for evaluating the system for given input settings as shown in Equation (40).

$$\mathcal{H}: \mathbb{T} \times \mathbb{S} \times \mathbb{P} \to \mathbb{R} \qquad (28)$$

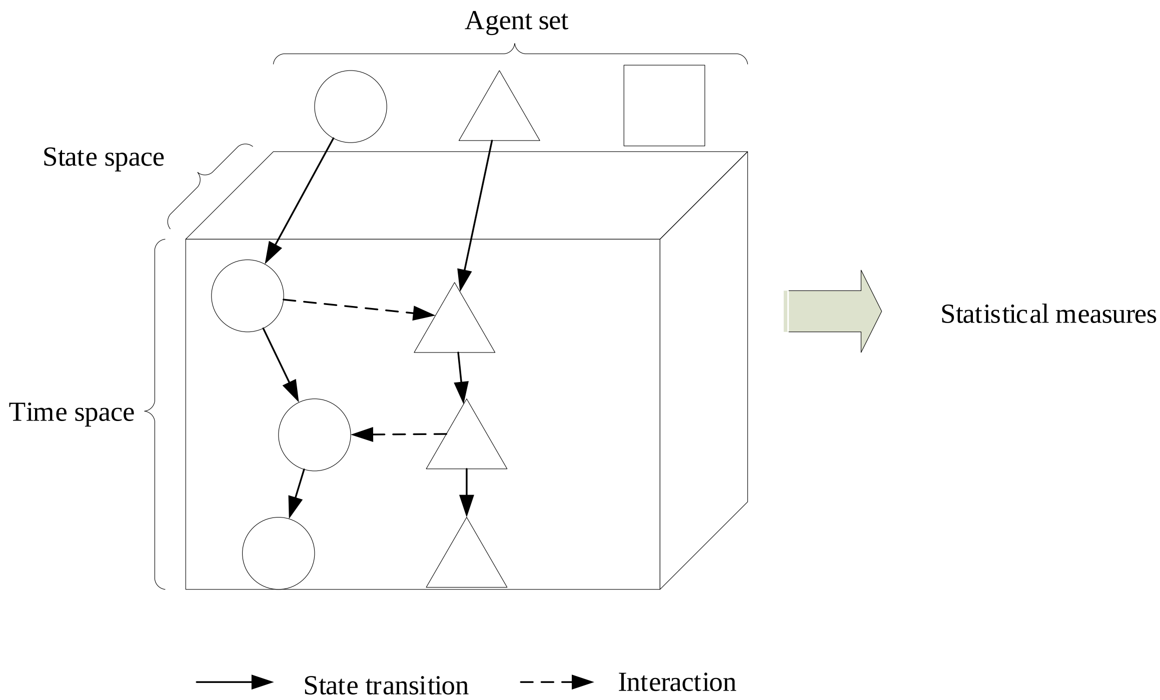



Figure 4-15 The simulation problem diagram

Figure 4-15 shows the simulation problem is defined to explore the high-dimensional space repeatedly, record the explored path and output the measures. The objective of the simulation problem is to replicate the real situation on an appropriate detailed level and to predicate possible outputs with regards to various scenarios. The statistical simulation result may help the engineers and managers to optimize the design, operation and management strategies.

## 4.7 Modeling traffic dynamics

### 4.7.1 Network representation

This section presents the network representation and the physical mechanism in the parking simulation. In the literature, the representations of simulated traffic networks are classified into two categories: the discrete-link network and the continuous-link network.

Inspired by the finite element theory, the Cellular Automata (CA) has emerged as a discrete approach for modeling complex behavior in the microscopic simulation (Levy, 1992; Wolfram, 1994). In comparison to the continuous model, the advantage of using CA is that the entities can be modeled with intuitive behavioral rules and the CA models are easily implemented and run efficiently on the large-scale network. CA microsimulation has been successfully applied to modeling vehicular flows, car-following, and pedestrian flow, and is proven to be a sufficient approximation of complex traffic flow (Nagel and Rasmussen, 1994; Paczuski and Nagel, 1995; Nagel, 1996, 1998; Santé et al., 2010).

In the previous parking simulation systems, the macroscopic models employed the continuous-edge networks since they captured the traffic flow instead of individuals. In the microscopic studies, the CA network is employed to describe the occupancy of the location and the location structure with the parking facilities. The off-street parking modeling involves the specific subdomain of the mixed vehicle and pedestrian traffic modeling. Vo et al. (2016) and Zhao et al. (2017) used the cellular network representation to model the vehicle movement within parking facilities. However, CA models are not integrated with the choice models and their models have limited representability of the details such as vehicle orientation.

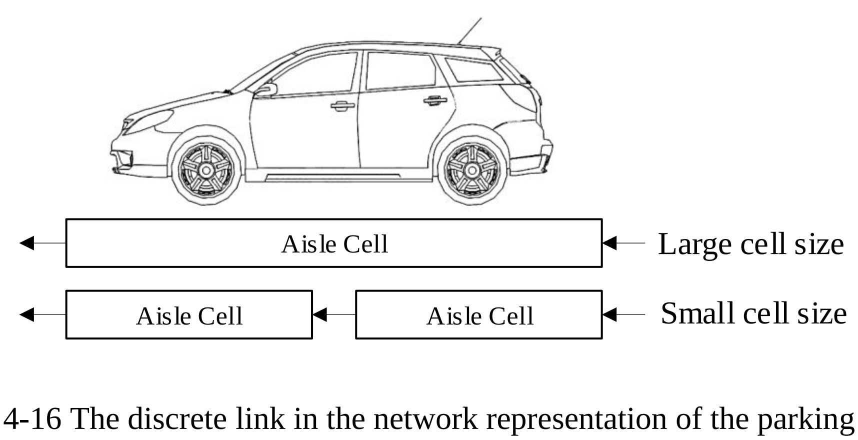



Figure 4-16 The discrete link in the network representation of the parking facility

This study uses a CA network representation for the physical network in the parking facilities. Figure 4-16 shows the roads in the parking facility are divided into pieces called "*aisle*", where the length of the large cell is about 4 to 6 meter. The size of the aisles can be determined by rule of thumb. Fine-grain simulation results can be derived from a smaller cell size of aisles, the trade-off is that finer location dimensions result in more computation and additional complexity of the simulation system. The vehicles would occupy one or more pieces of aisles. Given the speed of the vehicle is limited to about 15MPH in the parking facility, the traffic parameters such as the speed and traffic flow can be simplified to constants. The parking spaces are model as discrete cells of locations called "*spots*". The exclusive pedestrian way is separated and protected by barriers are called "*sidewalks*".

The entities can move from one location to another neighboring location. The occupancy of locations can be denoted by binary attributes. If a location is occupied by a car, other entities cannot move into this location anymore. If a location is occupied by a pedestrian, other pedestrians can move into this location until the number of pedestrians is lower than the location capacity. All the locations are connected by weighted directed edges. The edge represents one-step movement from the tail to the head. The edges can be blocked by the occupancy of other overlapping edges or locations.

In comparison to the previous studies, the proposed method contributes to creating a cellular network with connected cells of various sizes. This representation also facilitates the collection of the performance measures in the parking system by dividing the continuous space into finite elements. In such a manner, the traffic dynamics in the cellular network can be described with the rule-based movement logic.

### 4.7.2 Modeling the entity movement

Based on the cellular network representation, the traffic dynamics can be modeled with the logic programming techniques. In logic programming, the facts are a set of given feasible conditions, the rules are logical propositions and constraints, and the logical inference can be conducted to find the solution subject to a set of facts and rules. The logic programming model is to formulate a set of facts and rules to describe the domain problem (i.e. the parking simulation system).

To model the domain concepts in the parking simulation, the following predicates are defined to formulate the rule-based model. In the first place, the predicates are defined to construct the in-lot network. The predicate $connect\ (CL, A, NL)$ would construct the network by indicating the current location $CL$, potential actions $A$, and the next location $NL$. The predicate $in\ (E, L)$ would find the position $L$ of the entity $E$ as well as the occupancy of location. Secondly, the predicates are defined to model the entity behavior. The predicate $create\ (E)$ would generate an entity $E$ or a location $L$ in the system. The predicate $vision\ (L, O, E)$ would detect the locations $L$ in the vision of the entity $E$ facing the direction orientation $O$. The predicate $choose\ (S, LIST)$ would generate a random choice $S$ out of a candidate list $LIST$ based on proposed choice models.

The following rules define the creating of vehicles and pedestrians.

$$create\_veh(E){:-}\,create(E), assert(vehicle(E)) \tag{29}$$

$$create\_ped(E){:-}\,create(E), assert(pedestrian(E)) \tag{30}$$

The following rules are defined to build an inner network, where the predicate $spot(L)$ means whether the location $L$ is of the type $spot$ and so as the other location type predicates.

$$create\_spot(L){:}-create(L), assert(spot(L)) \tag{31}$$

$$create\_aisle(L){:}-create(L), assert(aisle(L)) \tag{32}$$

$$create\_entrance(L){:}-create(L), assert(enrance(L)) \tag{33}$$

$$create\_exit(L){:}-create(L), assert(exit(L)) \tag{34}$$

$$create\_door(L){:}-create(L), assert(spot(L)) \tag{35}$$

$$create\_sidewalk(L){:}-create(L), assert(sidewalk(L)) \tag{36}$$

The predicate $vacant\ (E, A, CL, NL)$ refers to checking the connectivity and the vacancy of next location $NL$ of the entity $E$ taking the action $A$ at the current location $CL$. The checking rule is defined as the following rule.

$$vacant(C, A, CL, NL){:}-in(C, CL), connect(CL, A, NL), not(in(\_, NL)) \tag{37}$$

where $C$ denotes the index of the vehicle, $A$ is the action of the vehicle, $CL$ represents the current location of the vehicle, and $NL$ denotes the next position the vehicle moves to when taking the action $A$.

The predicate $move\ (E, A, CL, NL)$ would update the position state of the entity. The movement of the vehicle $C$ with the action $A$ from the position $CL$ to $NL$ is defined by the following rule, where the predicate $f(C, NL)$ refers to the process state machine of the entity $C$ moving into the location $NL$.

$$move(C, A, CL, NL){:}-vacant(C, A, CL, NL), choose(C, NL),$$

$$retract\big(in(C, CL)\big), assert\big(in(C, NL)\big), f(C, NL) \tag{38}$$

$$move(C, A, CL, NL) \colon - vacant(C, A, CL, NL), route(C, NL),$$

$$retract\big(in(C, CL)\big), assert\big(in(C, NL)\big), f(C, NL) \tag{39}$$

The following rules are examples for adding vehicle movement logic. If the vehicle is in the process state $parked$, the movement of the parked vehicle is defined by the following rule. With this rule, the parked vehicle does not move.

$$move(C, \_, \_, NL) \colon - \, in(C, spot(\_)), f(C, NL) \tag{40}$$

For example, supposing the goal is to let all vehicles parked, the predicate $solve$ is defined to find all vehicles parked by the following rule, where the predicate applies to each element of the collection $LIST$.

$$solve(LIST) \, : - \, in(LIST, spot(\_)) \tag{41}$$

In this example, if some vehicles are not parked, the following rule means to try $move$ recursively, where the predicate $clock()$ refers to moving to the next time step.

$$solve(LIST) \colon - \, move(LIST), solve(LIST), clock() \tag{42}$$

This example of the predicates $move, solve$, and $f$ show the if-else condition logic can be fully described by stacking up rules instead of nesting conditions. In comparison to the nesting conditions, the movement logic built on this feature is more flexible and scalable.

The following equation defines a rule to find a feasible next movement of the entities, where $C$ represents the entity, $A$ denotes the action, and $X$ denotes the potential destinations.

$$next\_move(C, A, X) \colon - in(C, Y), connect(Y, A, X), not(in(\_, X)) \tag{43}$$

The following equation defines a rule that finds all possible movements and choose a random movement for the entity, where the predicate $next\_move$ finds the possible next movement of the entity $C$, $LIST$ is the list of the possible movements, the built-in predicate $choose$ selects the one of the candidates with the hypothetical probability or empirical distribution.

$$random\_move(C){:}-findall(X, next\_move(C,\_,X), LIST), choose(Y, LIST), f(C,Y), retract(in(C,\_)), assert(in(C,Y)) \quad (44)$$

The logic model of the entity process transition is detailed in the next section. Appendix A presents a sample code in Prolog for elaborating the idea of modeling logic-based traffic dynamics.

### 4.7.3 Modeling queuing

The parking lots can be modeled as a multi-server queuing system since each parking spot is a parallel server. In the literature, Ceballos and Curtis (2004) investigated queuing in parking facilities analyzed the multi-server queuing models and traffic simulation at toll and exit areas to capture the queuing at entrances and exits. Ratliff et al. (2016) modeled the urban parking system as a set of parallel queues and investigated the user equilibrium and system optimal equilibrium of arriving drivers. Thompson and Bonsall (1997); Waterson et al., (2001) found PGI systems reduced parking facility queue lengths and marginal system-wide travel time. Figure 4-17 shows the general M/M/N queuing model of the parking facilities, where the traffic model and the parking space choice model are modeled between the toll plaza and the parallel multi-servers.

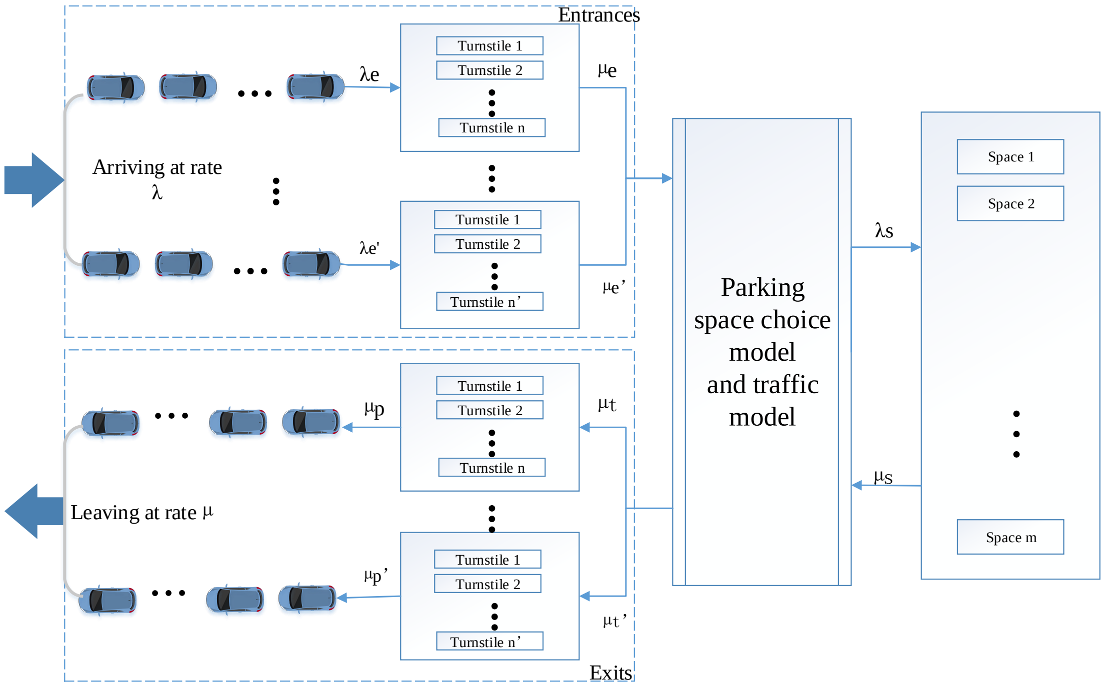



Figure 4-17 Queuing model diagram

However, the queuing at the toll plaza does not only the conventional queuing model but also is impacted by the parking spot choice model and the movement in the in-lot network. There're significant differences between the M/M/N queuing model and the parallel parking spots: (a) The parking space choice model creates queues since the vehicle may not use the first available server but rejects the feasible spot and looks for other options. (b) And the traffic model creates natural physical queues when the vehicle blocks the other vehicle physically. Figure 4-18 shows the proposed model involves that the queuing at the parking facility toll plazas blocks the traffic in the inner network. Thus, the proposed simulation method outperforms the analytical queuing models in capturing the complicated mechanism.

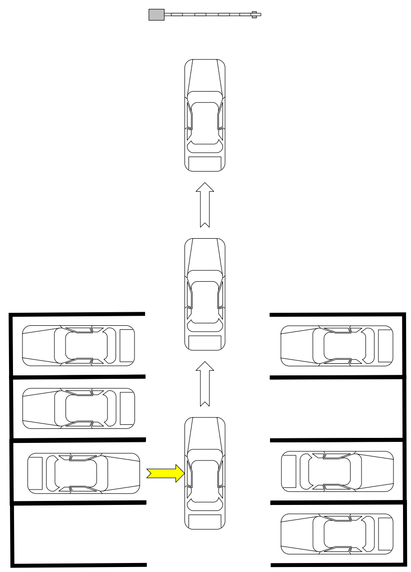


Figure 4-18 The exit queue blocking the inner network

The toll plazas are queuing areas and natural bottlenecks in the traffic system. If the entrance queue exceeds the capacity, the road traffic would be impacted by the low efficiency of the parking tolling. And the drivers who cannot wait for a time interval would renege. If the exit queue exceeds the capacity, the inner network of the parking facility would be severely jammed. Majid et al. (2016) investigated the impact of various arrival patterns on the queue at toll plazas.

In comparison to the existing studies, this study models the queuing at toll plazas with queuing as well as spillback to the inner network. The contribution is modeling a mixed queuing and spillback at the entrance of the parking facility, which captures the interactions between traffic dynamics and queue theory models.

## 4.8 Modeling entity behavior

This study presents a state machine for modeling entity behaviors incorporating both the position state and the process state. In this section, the process machine is used to capture the process transition of the entities in the parking facilities. The process machine is formulated in the logic programming method. The process machine $\mathcal{F}$ of the entity $E$ entering the location $L$ is denoted as the predicate $f(E, L)$. The related rules are defined for the process transitions with the label of the entity $E$ and the goal location $L$.

Figure 4-19 shows the entity process machine for both the *Vehicle* and the *Pedestrian*. Predicates are defined for each process in the process space $\mathbb{P}$, such as "cruising" and "parked".

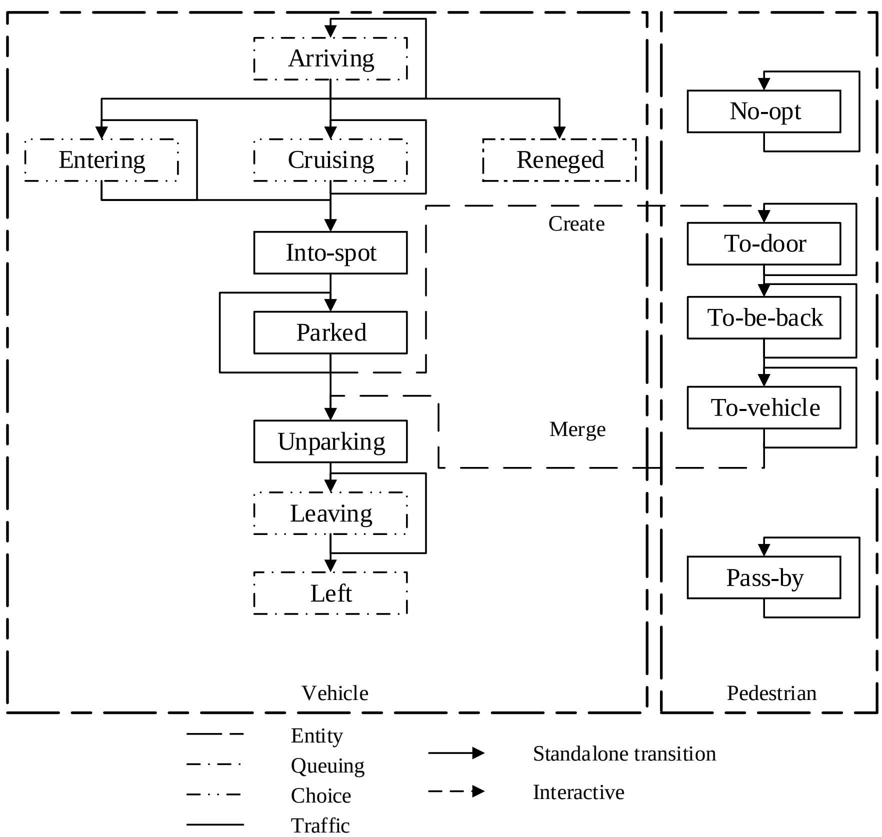



Figure 4-19 Parking simulation entity process transition

This study incorporates two kinds of parking behavior patterns regarding informed drivers and uninformed drivers. The informed drivers include the guided drivers, the special parking spot users, and the daily users who are familiar with the situation of the parking lot. The informed drivers are supposed to travel in the shortest path from the entrance to the destination parking spot as well as from the used spot to the exit of the parking lot. The uninformed driver is defined as the driver who finds the parking spot in a cruising-and-searching manner. For both kinds of drivers, the arriving transitions are defined as the following rules, where the predicate $informed$ is a label for the informed drivers and the predicate $arrive(C, L)$ is defined for the vehicle arrivals in the next section.

$$f(C, L){:}-arriving(C), arrive(C, L), informed(C),$$

$$retract\big(arriving(C)\big), assert(entering(C)) \tag{45}$$

$$f(C, L){:}-arriving(C), arrive(C, L), not(informed(C)),$$

$$retract\big(arriving(C)\big), assert(cruising(C)) \tag{46}$$

If the vehicle in the arriving queue does not enter the parking facility in a limited time, the vehicle would renege as the following rule, where the predicate $qt(C)$ means the queuing time of the vehicle $C$.

$$f(C, L){:}-arriving(C), qt(C),$$

$$retract(arriving(C)), assert\big(reneged(C)\big) \tag{47}$$

For the uninformed driver behavior, the transition from the process "*cruising*" to the process "*intospot*" is defined in the following rule. The meaning of this rule is that if the vehicle $C$ is cruising and will move into a parking spot $L$, the state of the vehicle is changed to "*intospot*".

$$f(C, L){:}-cruising(C), spot(L)$$

$$retract(cruising(L))),\ assert(intospot(L)) \tag{48}$$

The following rule determines that the process "*cruising*" is not changed when the vehicle is moving into an aisle.

$$f(C, L): -cruising(C), not(spot(L)) \quad (49)$$

For the informed driver behavior, the entering transition is defined by the following rules, where the predicate $route(L)$ refers to the optimized route to the selected destination.

$$f(C, L): -entering(C), route\ (C, L), spot(L), not(in(L))$$

$$retract(entering(C)), assert(intospot(C)) \quad (50)$$

$$f(C, L): -entering(C), route(C, L), not(spot(L)) \quad (51)$$

$$f(C, L): -intospot(C), spot(L), assert\big(parked(C)\big), retract\big(intospot(C)\big) \quad (52)$$

where the predicate $intospot$ checks whether the process of the vehicle $C$ is equal to *intospot*. The following two rules define that the parked vehicle waits for the counting down time.

$$f(C, L): -parked(C), cd(C), assert\big(unparking(C)\big), retract\big(parked(C)\big) \quad (53)$$

$$f(C, L): -parked(C), not(cd(C)) \quad (54)$$

where the predicate $cd(C)$ means the counting down time of the vehicle $C$ and checks whether the parked duration of the vehicle $C$ is equal to the generated one. If the condition is true, the vehicle should unpark. If not, the vehicle $C$ should continue until reaching the generated parking duration.

$$f(C, L): -unparking(C), aisle(L), assert\big(leaving(C)\big), retract\big(unparking(C)\big) \quad (55)$$

where the predicate $aisle(L)$ checks whether the location $L$ is an aisle.

$$f(C, L){:}-leaving(C), exit(L), retract(leaving(C)), assert(left(C)) \quad (56)$$

$$f(C, L){:}-leaving(C), not(exit(L)) \quad (57)$$

$$f(C, L){:}-left(C) \quad (58)$$

The pedestrian process transitions in the process space $\mathbb{P}_{\text{pedestrian}}$ are defined in the following rules.

$$f(P, L){:}-todoor(P), door(L), retract(todoor(P)), assert(tobeback(P)) \quad (59)$$

$$f(P, L){:}-todoor(P), aisle(L) \quad (60)$$

$$f(P, L){:}-todoor(P), sidewalk(L) \quad (61)$$

$$f(P, L){:}-tobeback(P), cd(P), retract(tobeback(P)), assert(tovehicle(P)) \quad (62)$$

$$f(P, L){:}-tovehicle(P), spot(L), retract\big(tovehicle(P)\big), merge(P, L) \quad (63)$$

$$f(P, L){:}-tovehicle(P), aisle(L) \quad (64)$$

$$f(P, L){:}-tovehicle(P), sidewalk(L) \quad (65)$$

where the predicate $cd(P)$ refers to the counting down time of the left time and checks whether the left time of pedestrian $P$ is equal to the generated one, and the predicate $merge(P)$ is to merge the pedestrian $P$ into the corresponding vehicle. If the pedestrian counting down $cd(P)$ is used, the vehicle counting down $cd\ (C)$ does not apply.

The following rules define the looped pedestrian process transition along with the random movement.

$$f(P, \_){:}-noopt(P), random\_move(P) \quad (66)$$

$$f(P,L){:}-passby(P), route(P,L) \tag{67}$$

where $P$ is the variable of the pedestrian and the processes $noopt, passby$ are not related to the type of the location.

With the presented process machine $f(C,P)$, the vehicle and pedestrian parking processes are modeled with logic programming technique. In comparison to the flow chart, the logic programming can determine where the rules are well-defined rigorously. This study contributes to (a) employing a parking spot choice classifier to capture the parking space choice behavior and an intersection classifier to capture the route choice behavior; (b) modeling driver behavior type with informed (e.g. handicapped, guided, women, etc.), uninformed (e.g. uniformly distributed); and (c) modeling interactions with pedestrians.

### 4.8.1 Modeling entity intermediate process behavior

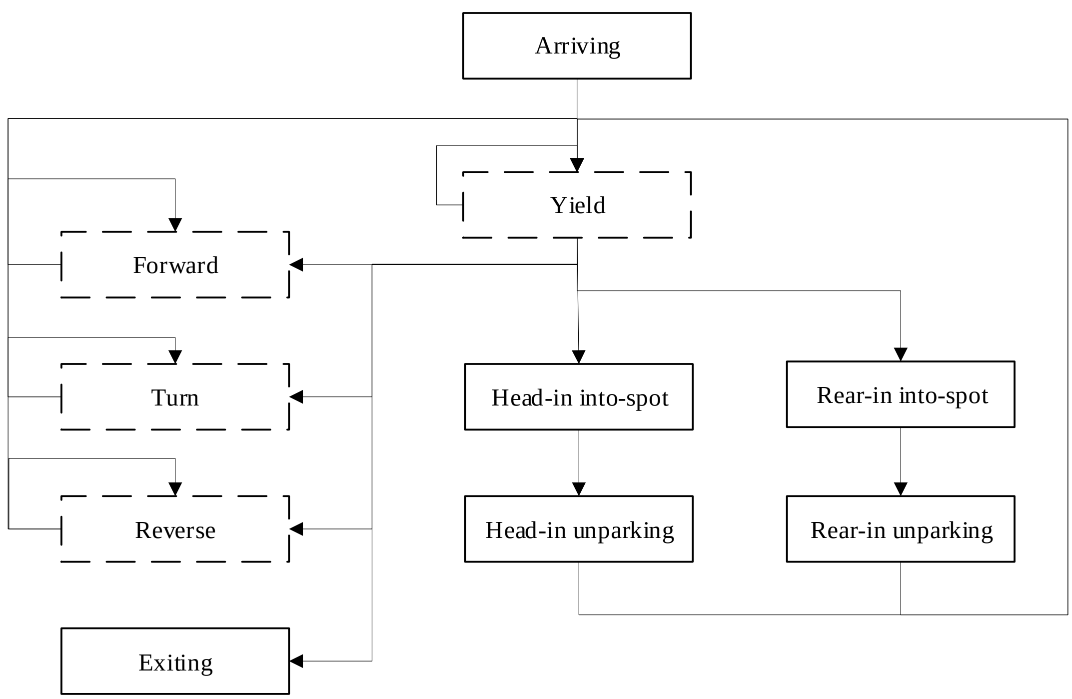



Figure 4-20 Vehicle movement behavior modeling diagram

The state transition in the multidimensional space integrates the process transition and the position movement. In the state transition, there exist several intermediate movement processes. The extraction of the intermediate movement processes is used to derive the statistical measures for aiding the design and management of the parking facility. For example, the critical safety measure weaving duration is collected when the vehicle is yielding the pedestrians and other vehicles. Figure 4-20 presents the vehicle traffic rules, where the dashed rectangle represents intermediate states in the vehicle parking process. To concatenate the intermediate process with the defined state transition, the definitions are listed as follows:

- *Arrive.* When the coming vehicle arrives at one of the entrances of the parking facility, it joins entering the queue and waits until the finishing paying the ticket or checking the seasonal permit and getting the right-of-way. The following rules show that if the entrance location $L$ is not occupied, the vehicle $C$ should be put in location $L$; otherwise, nothing should be done.

$$arrive(C, L){:}-\ queue(C), entrance(L)\ , not(in(\_, L)),\ \ assert(in(C, L)) \tag{68}$$

$$arrive(C, L){:}-in(\_, L) \tag{69}$$

where the predicate $queue(C)$ is to check if the head is $C$ and pop it.

- *Yield.* The vehicle stops and yields when a conflicting entity has the right-of-way. The following rule means if the location $L$ is occupied, then the vehicle $C$ should yield the right-of-way.

$$yield(C, L){:}-\ in(\_, L) \tag{70}$$

- *Forward.* When the front strip is clear and there’s not a coming pedestrian, the vehicle can move on its path forward. The following rule shows the

$$forward(C,L){:}-move(C,Forward,\_,\_) \quad (71)$$

- *Turn.* When a vehicle arrives at an intersection or a turning aisle, the vehicle turns if it has the right-of-way.

$$turn(C,L){:}-move(C,Left,\_,\_),in(\_,L) \quad (72)$$

$$turn(C,L){:}-move(C,Right,\_,\_),in(\_,L) \quad (73)$$

- *Reverse.* The vehicle goes into a narrow strip when another vehicle goes in the opposite direction, there would create a deadlock. In this situation, the blocking vehicle should reverse to eliminate the deadlock.

$$reverse(C,L){:}-move(C,Back,\_,\_) \quad (74)$$

- *Parking and Unparking*. The vehicle moves into the parking spot and finishes the parking maneuver. The driver can choose head-in or rear-in parking. If the head-in is chosen, the corresponding rear-out unparking maneuvers should be performed. The constraints of head-in are defined by the following rules, where the semicolon (;) refers to the "or" logic.

$$intospot(C){:}-assert(headin(C)),move(C,(Left;Right),\_,\_) \quad (75)$$

$$unpacking(C){:}-headin(C)\ ,move(C,(Left;Right),\_,\_) \quad (76)$$

This study further extended the framework of parking behavior process considering pedestrian activities because the safety concerns raise and the deficiency of ignoring pedestrians limits the application of the simulation methodology in parking facility design and the performance evaluation.

The proposed pedestrian behavior model incorporates (a) interaction with vehicles (b) moving logic across the parking lot: walking, wandering. Figure 4-21 shows the pedestrian intermediate process diagram.

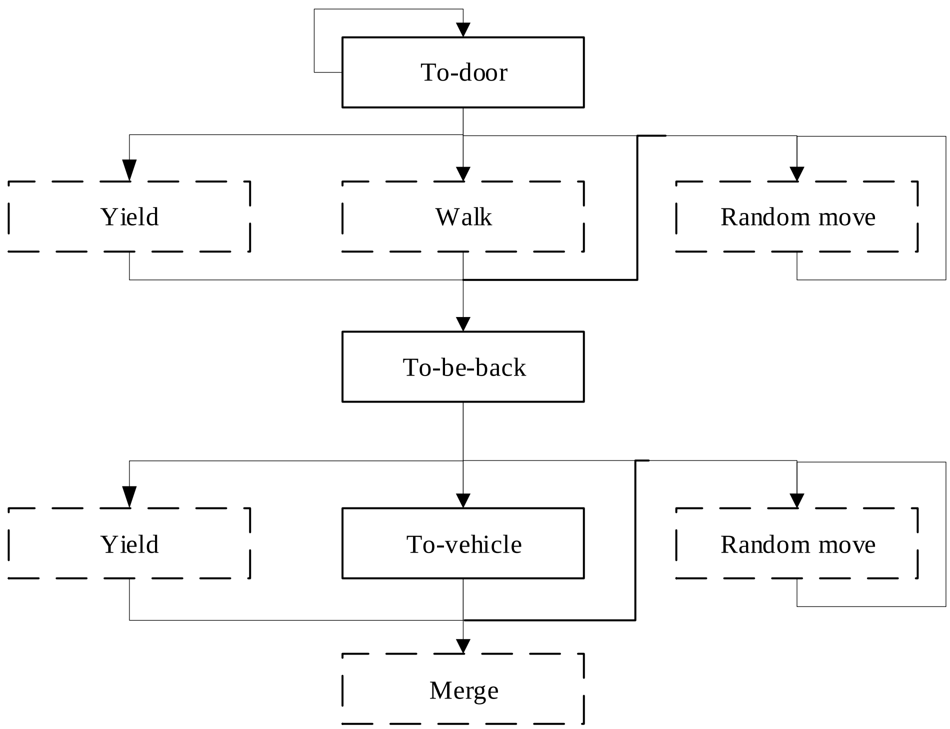



Figure 4-21 Pedestrian intermediate process diagram

Figure 4-21 uses the following definitions of intermediate processes, where the solid rectangles are for the process state, and the dash rectangles are for the intermediate process.

- *Yield.* The pedestrian yields to traffic when the next location on the route is occupied by vehicles, which is defined by the following rule.

$$yield(P, L) :- in(\_, L), route(L) \tag{77}$$

- *Walk.* When the front location is clear and there's not a coming vehicle, the pedestrian can move on its path forward. The following rule defines the forward logic

$$walk(C, L) :- move(C, (Forward; Left; Right), \_, \_), route(L) \quad (78)$$

- *Merge.* When the pedestrians arrive at their vehicles, they get on the vehicles. The following rule defines the merge logic, where the predicate $belong(P, C)$ refers to when the pedestrian $P$ is from the vehicle $C$.

$$merge(P, L) :- belong(P, C), in(C, L), retract(in(P, \_)) \quad (79)$$

### 4.8.2 Modeling driver parking spot and route choice modeling

In the literature, the previous studies utilized logit models to capture parking space choice behavior. van der Waerden et al. (2003) proposed a tree-like process and a nested logit model, where the drivers decide parking strip and parking spot sequentially. In this setting, the drivers should have a global view of the parking facility with the guidance system. Vo et al. (2016) proposed a decision tree to capture the parking choice model, however, didn't provide details about their method, model calibration and how to apply their choice model in the simulation environment. However, their model could not apply when the drivers don't have such a view and cruise for available spaces. Li (2016) employed a similar process to evaluate the parking guidance system. In Li's specific case, the drivers don't need to have a global view since the layout of the parking facility has a tree-like topology. Ji et al. (2009) put forward the key factors of parking space choice include walking distance, cruising distance, distance to monitors (safety concern), state of the lane to the parking space (strip occupancy), sunlight shelter, state of available parking space (side spot occupancy). Chen et al. (2011) proposed a parking space choice model with a fuzzy set based on Ji's model.

However, the previous studies have the critical deficiencies: (a) their models do not apply when the drivers are not well-informed and cruise for available spaces; (b) their models are case-specific and hard to calibrate in the other parking facilities. When designing the parking simulation system, it shows an incompatibility between the parking spot choice model and the movement behavior due to the separation in the behavior study. In the parking simulation, the integration of the choice models and the movement behaviors is critical to replicate the driver behavior.

This study incorporates two kinds of parking choice patterns regarding the informed drivers and uninformed drivers, respectively: (a) uninformed driving which is defined as cruising and searching without guiding system, (b) informed driving which is defined driving with the aid of parking guiding information, or driving to reserved spaces directly. For the guided driving, the proposed system assumes the drivers have full knowledge of the parking facilities such as the position of the parking space, and the shortest path to the destination.

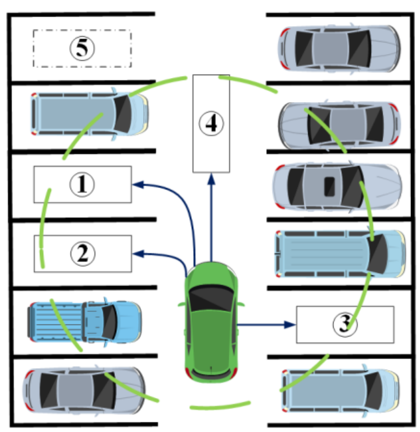



Figure 4-22 The diagram of vehicle's parking spot choices

Figure 4-22 shows an example of the cruising and searching behavior without guiding system, where the car in the aisle is looking for the potential parking spot, the dashed circle is the vision of the driver, the numbered rectangles are the feasible options for the cruising vehicle, the point dash rectangle is out of the vision of the driver. Note that the vision could be in any shape depending on the graphical calculation for the visible area of the driver, in this study the vision is simplified as a circle in order to model the localized parking spot choice. The difficulty of the previous studies is that the multinomial logit model is not capable to predict the choice out of the available parking spots such as label 1, 2, 3 and the next aisle such as label 4 since the aisle location does not have the attributes of the parking spots.

To address this issue, this study uses the binary classification methods such as the binomial logit model to predict whether or not to take each option, where the predicted target "one" denotes taking this option, otherwise rejecting this option. The proposed parking spot choice model assumes the driver should make the decision one-by-one without aftereffect. The following rule defines the choice behavior of the entities, where the variable $LIST$ refers to the list of options, the built-in predicate $gs(L, LIST)$ extracts the attributes of the options and makes predictions of the chosen spot $L$.

$$choose(L, LIST){:}-vision(C, LIST), gs(L, LIST), spot(L) \tag{80}$$

In the parking spot choice model, the critical attribute of the potential spot is the walking distance from the parking spot to the pedestrian exit or the walking destination in the parking lot. The other attributes such as gender are case-specific and should be developed with local survey data.

The route choice behavior is modeled regarding the information perceived by the drivers as well. In the route choice model, the critical attribute is the length of the potential path. If the parking guidance system is applicable, the occupancy of the path should be considered. For the informed drivers, the first $k$ shortest paths within the parking are modeled in the following rule, where the predicate $gr(R)$ is the choice model of the routes and the predicate $path(R, L)$.refers to whether the location $L$ is on the path $R$.

$$route(L):-gr(R),path(R,L) \tag{81}$$

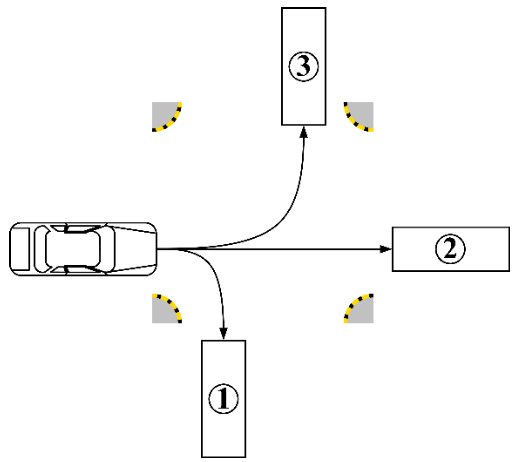



Figure 4-23 The diagram of the route choice at the intersection

For the uninformed drivers, the vehicle route in the parking lot is not decided at the beginning of the vehicle entering the system but at the intersections. Figure 4-23 shows the diagram of the route choice at the intersection. In the route choice model, the driver should choose the desired direction for choosing at the intersection. According to the calibration results and the previous studies, the critical attribute is the aisle occupancy of the strip. If at least one of the aisles is occupied, the probability of choosing this strip reduces. If the variable message sign provides the spot occupancy of the strip, the route choice behavior should include the spot occupancy as attributes. The strip is optional, where the strip consists of a set of aisles and spots. The strip choice model is integrated into the model by the following rule, where the predicate $ga(R)$ is the choice model of the routes and the predicate $instrip(S,L)$.refers to whether the location $L$ is on the strip $S$.

$$choose(L,LIST):-ga(L,S,LIST),instrip(S,L),aisle(L) \tag{82}$$

In the previous studies, the vision and the scope of the feasible parking spot are not considered. The choice classifier models are calibrated separately and are able to be used to produce predictions as hyperparameters. To address the incompatibility of the movement and the choice model, the proposed model has the advanced features: (a) cooperating with the traffic dynamics model without the topology dependence, (b) incorporating the parking behavior subject to the parking guidance system, (c) easy to be calibrated and tuned. The deficiency of the proposed choice model includes the assumption that the options are standalone decisions and can be captured by the binary classification model.

# 5 System application

## 5.1 Charting and visualization

The proposed system incorporates the following charting and visualization modules: (a) curves for temporal measures, (b) heatmaps for spatial measures (van der Waerden et al., 2003), (c) histograms for outputs and measures, and (d) animation for micro-behaviors. Animation provides an intuitive understanding and insight into system dynamics rather than just predicting the output of the studied case to support decisions (Yuan and Liu, 2014; Vo et al., 2016). In the proposed parking simulator, the outputs are visualized in preliminary web-based animates and are able to be present with more user-friendly animation engine. In the proposed parking simulation system, the curve, histogram plotting, and animation are designed to produce a flexible ready-to-deliver application for the full-scale real-world scenarios.

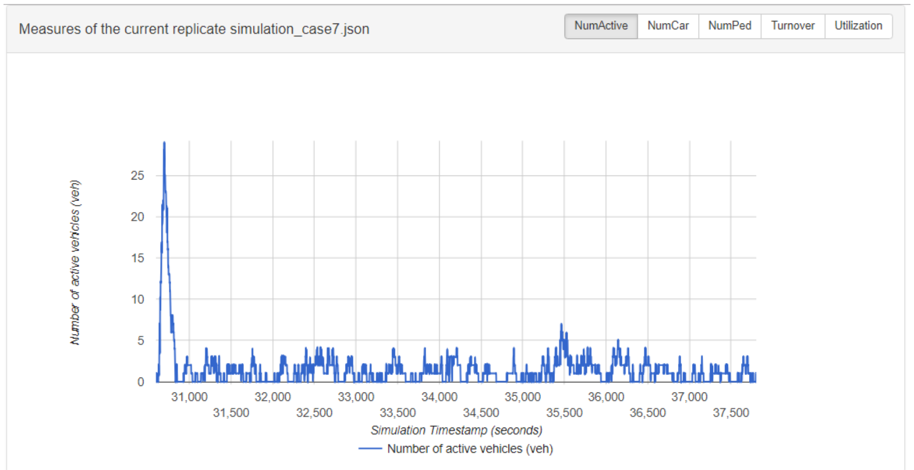
The simulation settings can be inputted via dialogs in the web Graphical User Interface (GUI) or Representational State Transfer (REST) API. The arrival and departure distributions are inputted as CSV files in the simulation model. The initial occupation rate is set to 0.3 according to the field survey data. The system yields the simulation results in 10 seconds. The history of all entities and locations is recorded for further analyses. From the historical record, the simulation system can extract informative Key Performance Indicators (KPI).

To visualize the result of the simulation, the charts for the efficiency and the safety on the system level, the location level, and entity level can be outputted via the web GUI. Figure 5-1 shows a heatmap for the utilization of the locations, where the higher utilization is in red, and the lower utilization is in green. It shows the spots and aisles closer to the pedestrian exits are more frequently used, which is consistent with the choice behavior model. In the spot choice behavior model, the walking distance from the spot to the pedestrian exit or destination has the largest weight. Note that this behavior feature is founded in most of the parking facilities but may not be homogeneous in every parking facility.

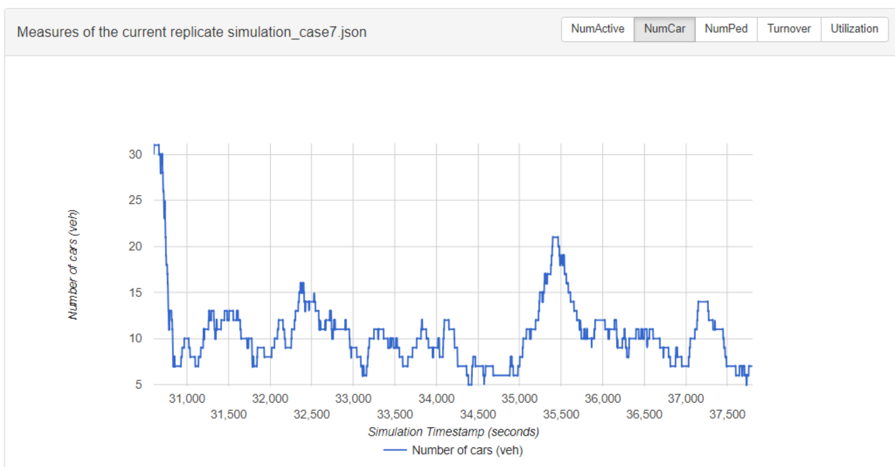

Figure 5-1 A sample case for location measures in the proposed system

Figure 5-2 shows plots of the disaggregated key performance indicators of the simulation. The peak in the beginning in Figure 5-2(b) is due to the initialization for the nonempty facility. This period is critical for replicating the system dynamics from a state when the system is not empty. The warming-up process would create the equivalent number of vehicles and initialize the state of each entity. In the final report, the warming-up period should not be counted for the overall performance measure.

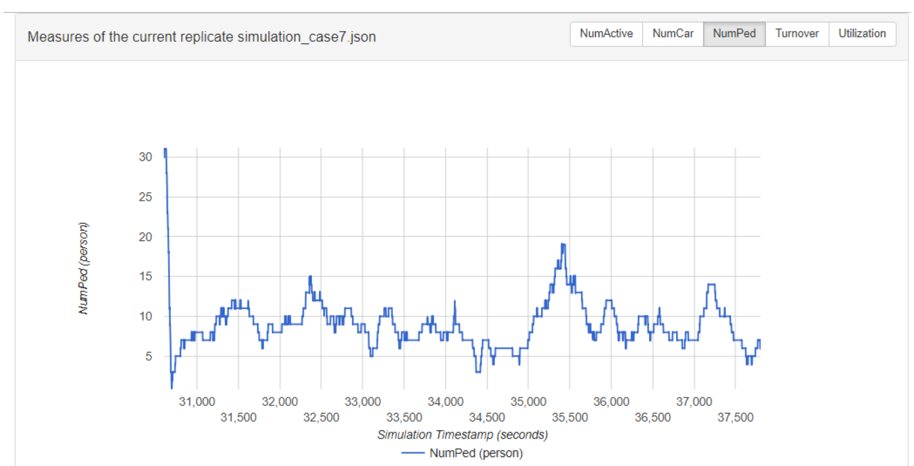
The critical finding of this study is the active capacity of the parking system. The active capacity is defined as the largest number of actively moving vehicles in the parking system. If one vehicle is waiting for any possible movement towards the destination or the intermediate destination, the vehicle is not active. The parking simulator finds the active capacity of the existing facility and the planning layout. Note that the active capacity is essentially an attribute of the parking facility layout subject to environmental parameters such as the arrival distribution and departure distribution. During the peak hour or special event, the arrival or the departure rate is considerably greater than the designed capacity, and the entrances would be jammed, and the actual throughput of the system would be lower than the active capacity. The design of parking facility involves compromise of the limitation of land use, the settings of entrances, the number and geometry of spots. The parking simulator helps the designers to detect the potential design deficiency. The active capacity is an indicator for potential which may cause a deadlock or reduce the throughput of the parking facility. To find the critical active capacity, more replicates should be tested for finding the critical blowing-up point of the simulated parking facility. The arrival and departure pattern should be calibrated with the land use and customer source surrounding. With the fixed arrival and departure pattern, the active capacity finds the constant throughput of a layout. A layout with more spots is desired since it would provide more servers, however, may also create more blockages and reduce the active capacity and the user experience. The desired layout should compromise the efficiency of the land use and the traffic throughput during the peak hour.



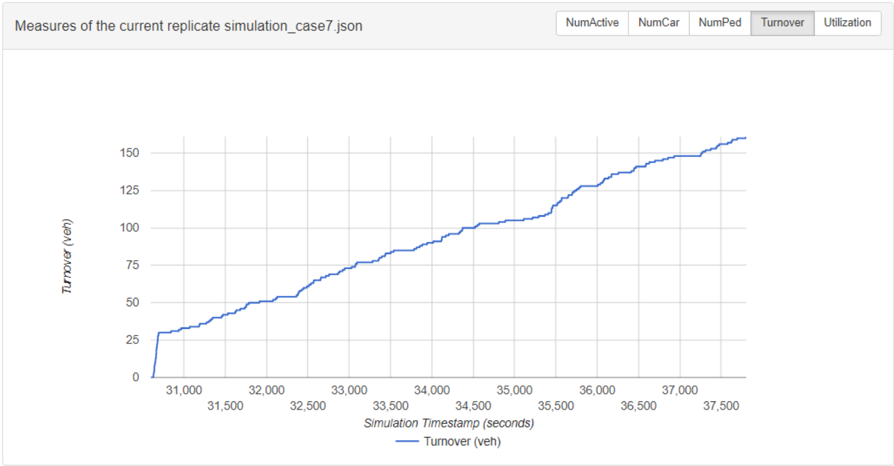

(a) Number of active vehicles in the system



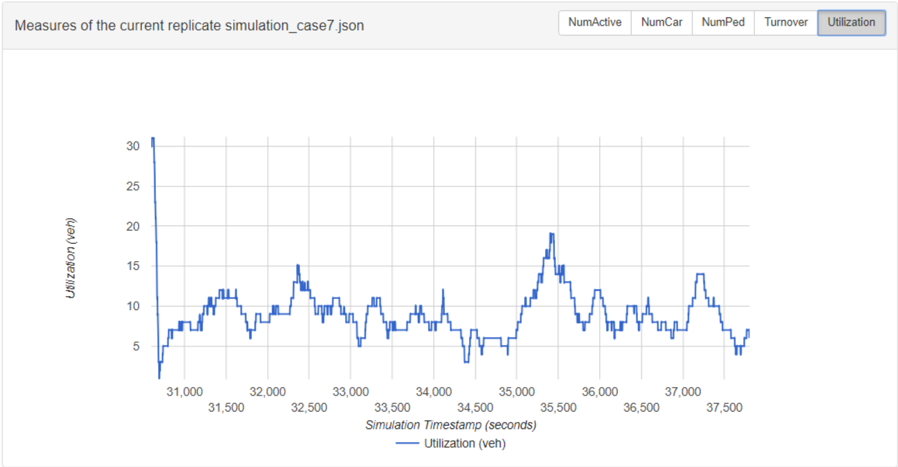

(b) Number of cars in the system




(c) Number of pedestrians in the system

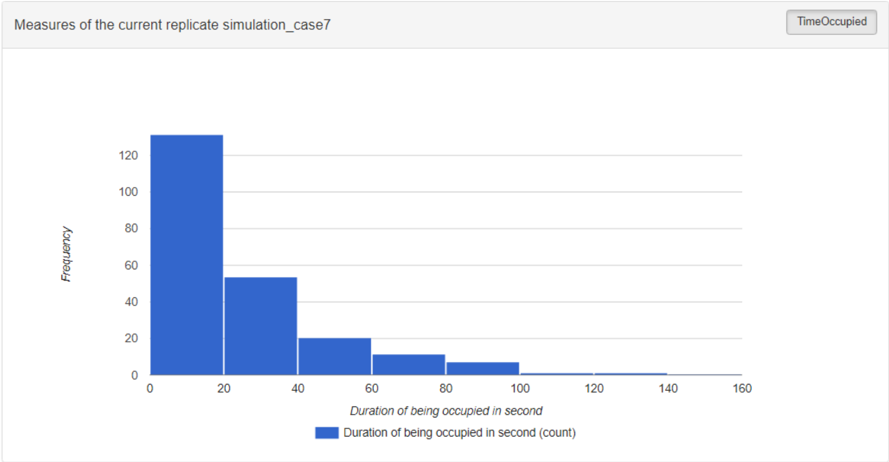



(d) Number of turnovers in the system

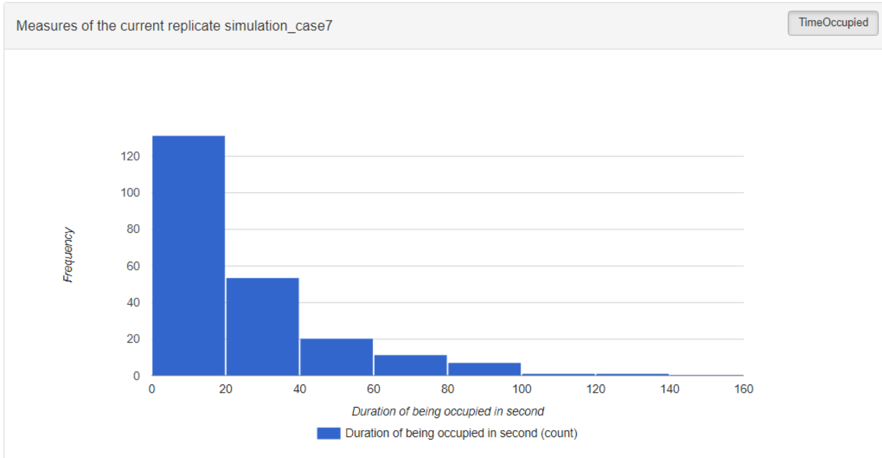



(e) The utilization of parking spots in the system

Figure 5-2 Overall key performance indicators of the proposed system

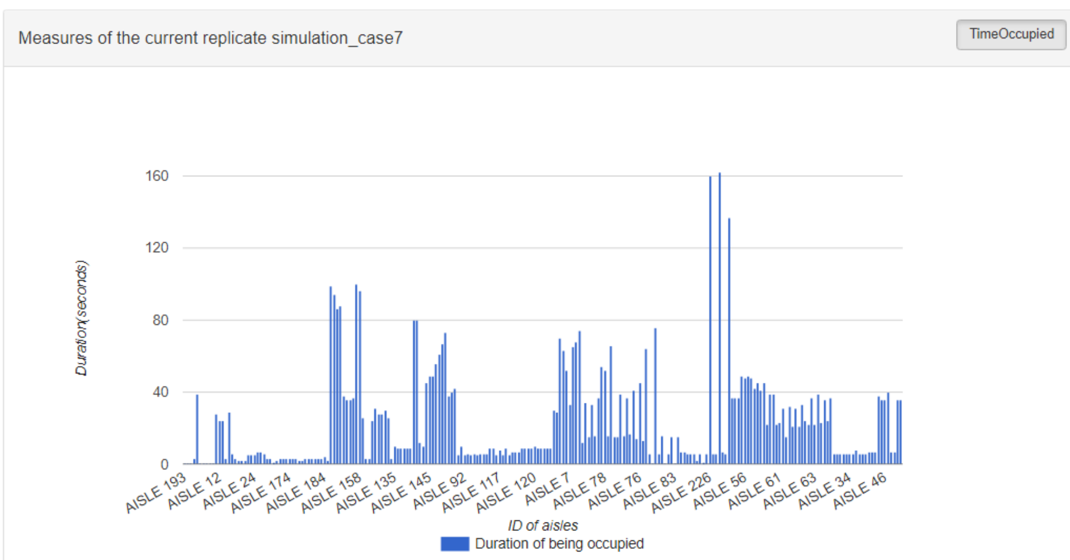



(a) the histogram of the duration of being occupied spot

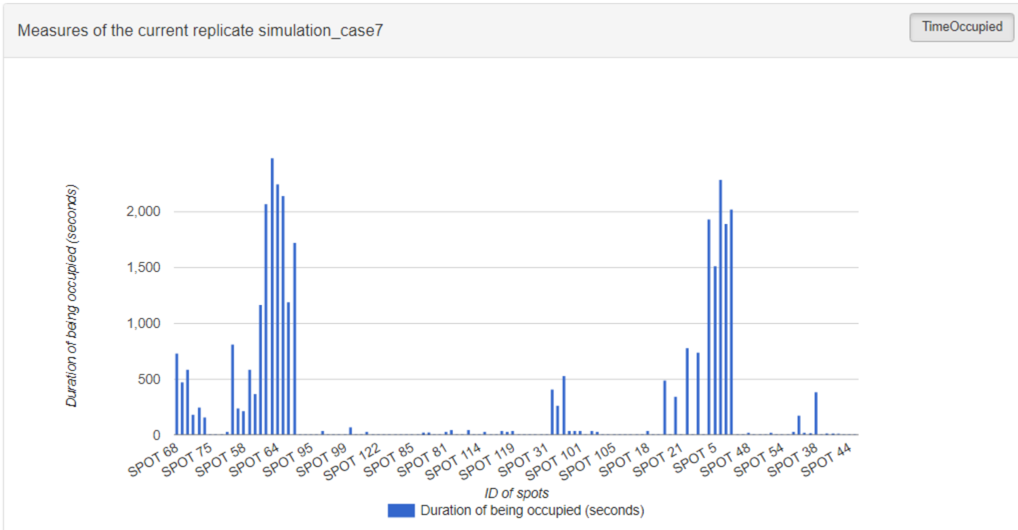



(b) the histogram of the duration of being occupied spot

Figure 5-3 The histograms of the proposed system

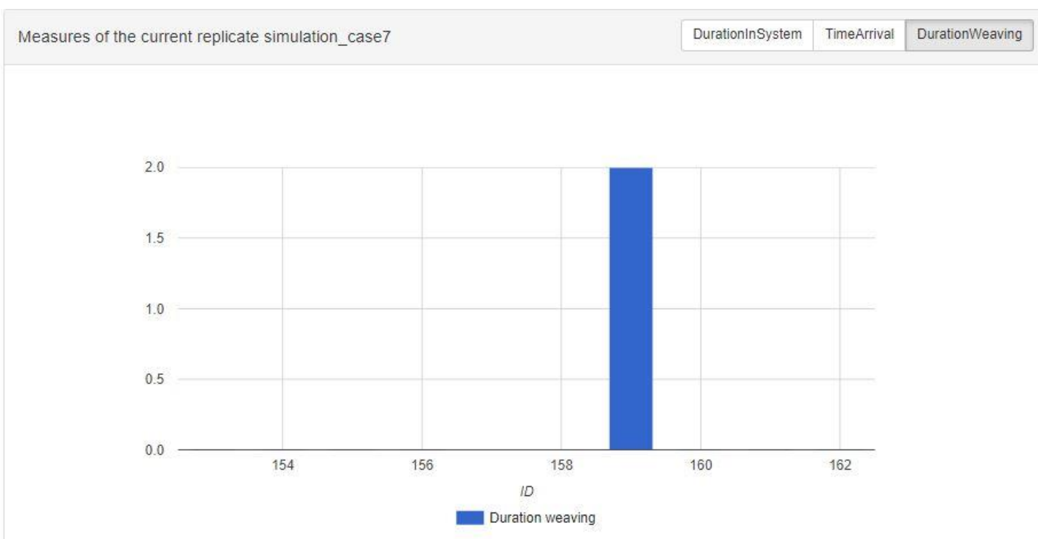



(a) The occupied time of the aisles

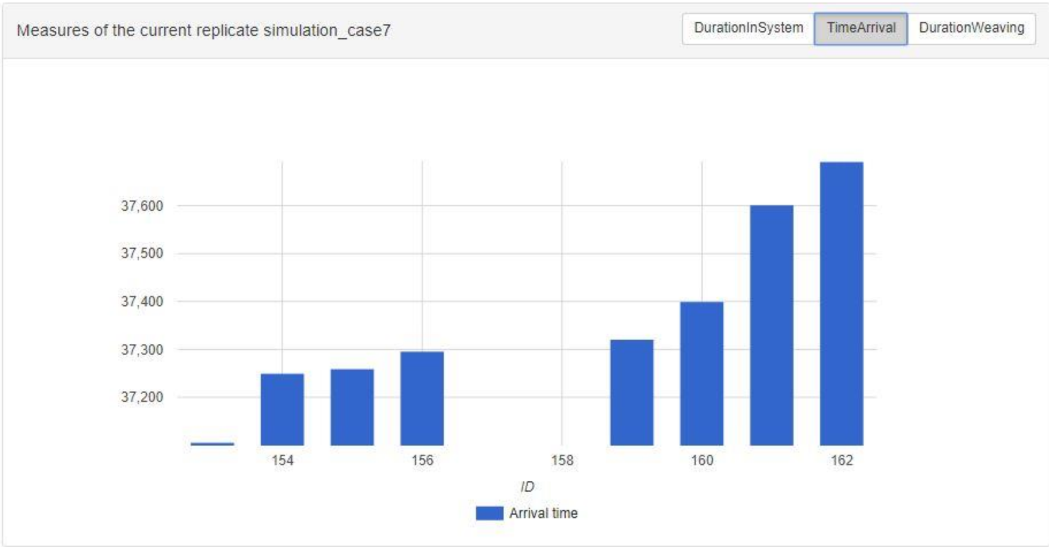



(b) The occupied time of the spots

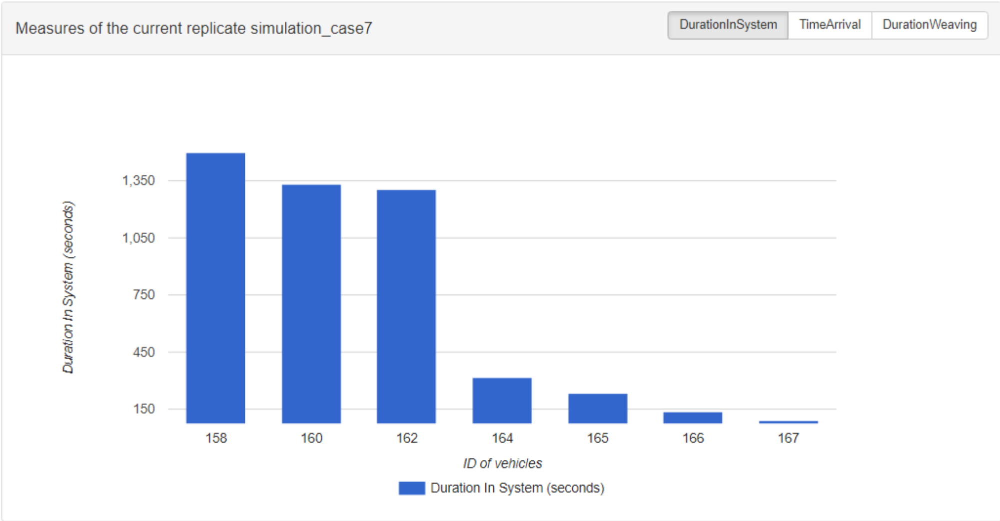
Figure 5-4 The key performance indicators of locations in the proposed system




(a) The duration of vehicle-pedestrian weaving in the system

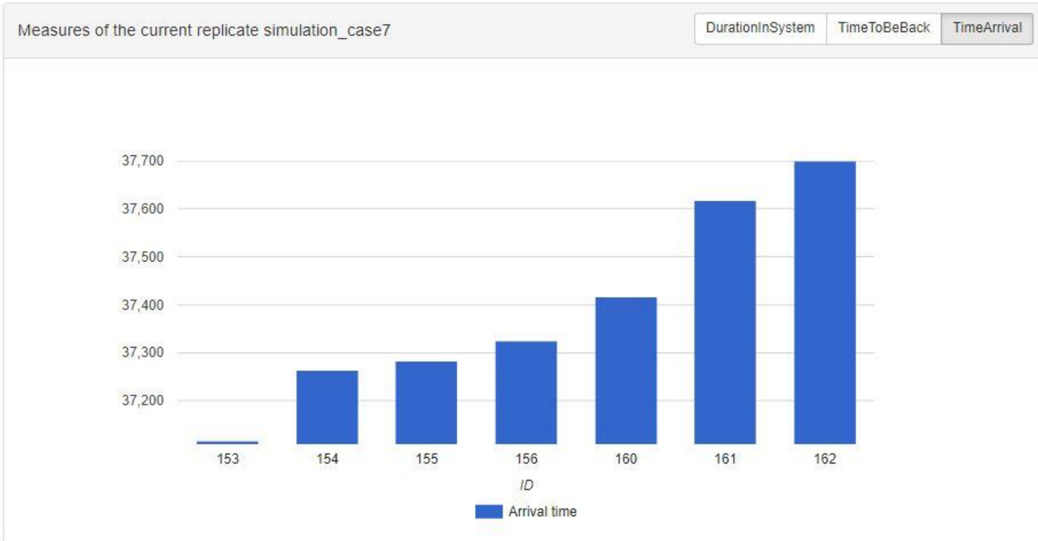



(b) The time of arrival for vehicles in the system

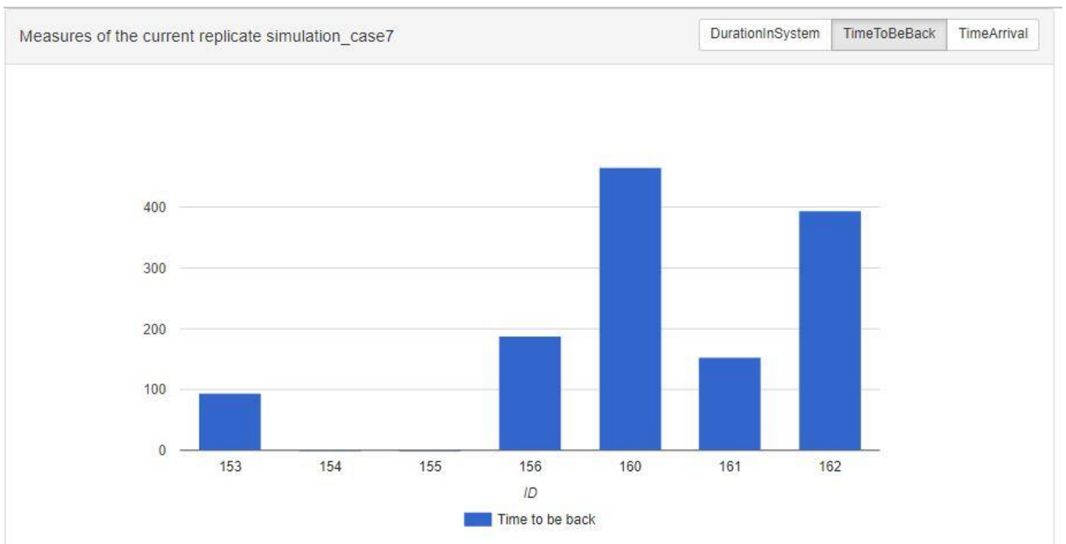



(c) The duration of vehicles in the system

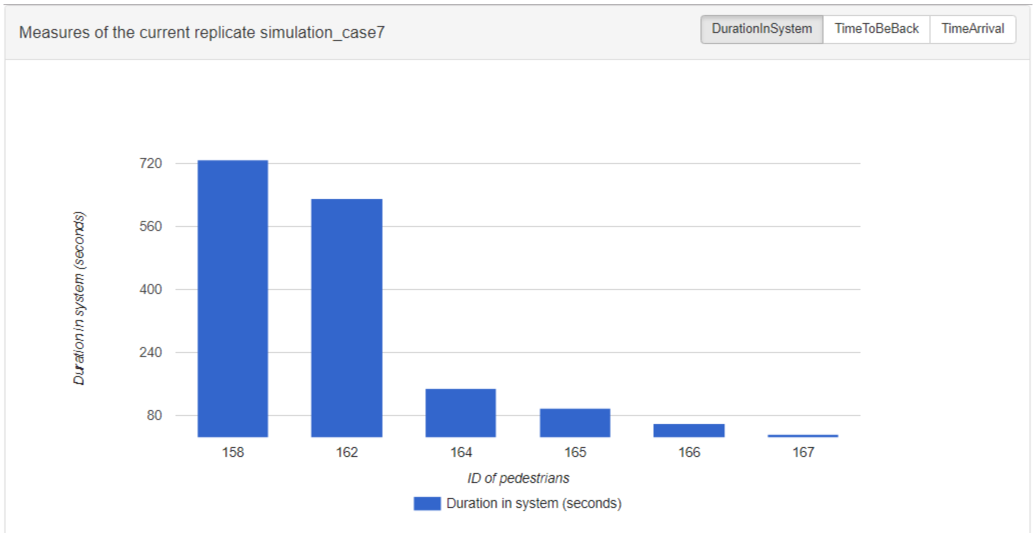
Figure 5-5 The key performance indicators of vehicles in the system




(a) The arrival time of pedestrians




(b) The time to be back from the exit of pedestrians




(c) The duration of pedestrians in the system

Figure 5-6 Pedestrian key performance indicators in the proposed system

## 5.2 Simulation-aided design

The proposed simulation system provides a new methodology for the parking lot design and optimization. In the traditional methodology, the designer does not have a dedicated tool for the simulation analysis to test the potential outcomes given the infrastructure configurations and assumptions. The proposed system can provide informative KPIs to aid the design of smart parking facilities with multiuse, automated spots, shared spots, electrical-charging spots, etc. The simulation-aided design incorporates a forward-back process. The designer iterates the draft design runs the simulation to extract the feedbacks and revises the design until the final design.

The design criteria include (a) maximizing the efficiency of land use, (b) fulfilling the requirements of the standard and the regulations, (c) providing the vision for the safety concern, and (d) supporting the development of smart parking facilities.

For illustrating the proposed system, the following cases and discussions address the critical practical problems: (a) The evaluation of the smart check-in device, (b) the evaluation of flexible pricing policy, (c) the evaluation of the special parking spots, and (d) the evaluation of reverse parking policy. Note that to simplify the scenarios, the testing has the assumption that the inter-arrival time of vehicles is normally distributed, and the inter-departure time of vehicles is exponentially distributed, and each testing case is repeated for 100 replicates for stable average outputs.

### 5.2.1 The evaluation of the smart check-in device

In the real-world case, the UWM Transportation Department plans to install the smart check-in devices at the entrances of the parking facilities. The smart device can recognize the license plate number to automatically check-in and check-out when the vehicle arrives and departs. The involvement of this device has several benefits: (a) This device can reduce the duration for check-in and check-out by simplifying the pass checking produce. (b) The customers don't have to bring the identification pass for this service. (c) The license plate number identification can avoid seasonal parking pass fraud. In the analysis of the benefit of this device, the simulation system helps the evaluation of the impact of the new device. The device is assumed to reduce the mean of service time in the arrival queue and the departure queue. In the settings of the simulation system, the arrival distribution supposing the arrival and the departure distributions and the layout configurations are not changed. The arrival and departure entrance queuing duration distribution are used as the inputs for smart check-in devices for the benchmarking. Figure 5-7 shows the results of comparing the scenarios across multiple interarrival with and without the smart check-in device, where the primary y-axis is for the average daily revenue and the secondary y-axis is for the average waiting time of vehicles. In Figure 5-7, the device does not have a great impact on the performance of the parking facilities. Note that the revenue counts only the hourly ticket payments and does not include the seasonal permit incomes.

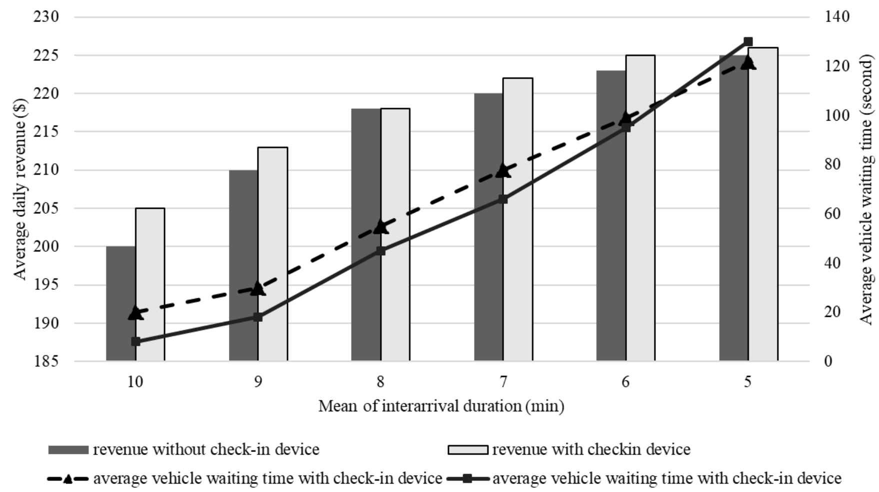



Figure 5-7 The comparison of facility performance with and without the smart checking device

The operational concerns focus on how to improve the service level and customer experience of the parking infrastructure. Note that the customer experience cannot be simply modeled with simulated quantitative measures due to the external impact factors. The smart check-in devices save the customers' time when rushing for parking, and customers enjoy the techniques of using intelligent devices and are willing to pay for the service fee. Those factors cannot be tracked by the proposed system.

### 5.2.2 The evaluation of the flexible pricing policy

To maximize the profit of the parking service, the more flexible pricing strategies can be tested. The critical operational concern is the impact of price. The impact of the change in hourly price or seasonal permit depends on the elasticity of the parking fee and the traffic mode choice of the travelers. According to the survey of the Transportation Department, the parking service on campus is of low elasticity and faculty, crew, and students have strong propensity to drive to the campus. The income of the parking sector is used to support the non-profitable sectors such as the transit sector, the Be On the Safe Side (BOSS) program, etc. Increasing the parking price is definitely possible to earn more profit, which would induce the complaints from customers. Thus, the pricing of the parking service also depends on the negotiation between the parking operators and the stakeholders.

For testing the impact of pricing of the parking system, the flexible pricing policy is defined as that the price of spots with higher utilization is greater than the price of spots with lower utilization. The customers prefer the spots closer to the pedestrian exits. Thus, the flexible pricing policy would balance the geometric distribution of the occupied parking spots. Figure 5-8 compares the average daily revenue of the scenarios with and without the flexible pricing policy. Figure 5-8 shows when the inter-arrival time is lower than 8 min, the flexible pricing policy has greater revenue, and when the inter-arrival time is greater than 8 min, the increment of average daily revenue is marginal. Note that the revenue does not include the incomes from the seasonal permit users and this numerical test does not involve the price elasticity which means the inter-arrival and inter-departure time do not change over the price changes.

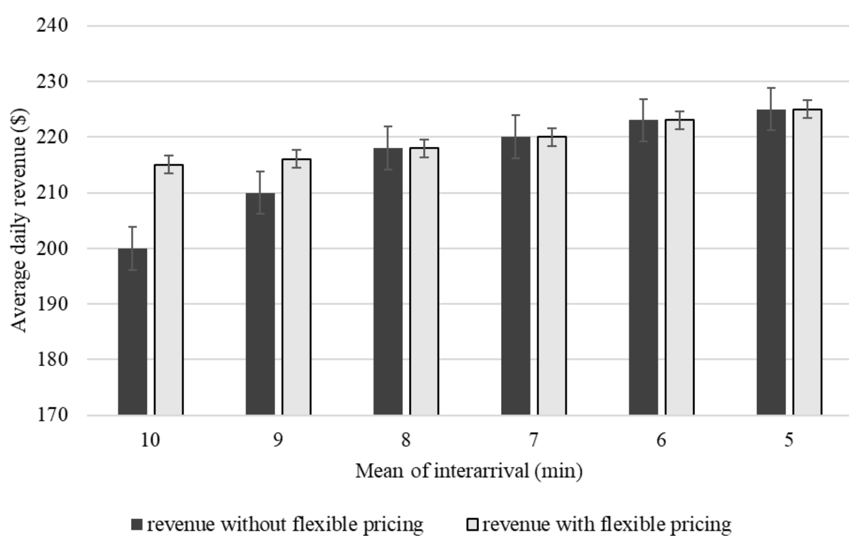



Figure 5-8 The evaluation of flexible pricing policy

### 5.2.3 The evaluation of the special parking spots

In the modern parking facilities, special types of parking spots, such as shared parking spots, electrical-charging parking spots, woman-priority parking spots, attract attention of traffic planners and policymakers. In the UWM campus, the electrical-charging parking spots are of low utilization, because the electrical vehicles are of low ownership. In the interview to the customers in the neighbor community of Bayshore Mall, Glendale, WI, the customers complain about the electrical-charging spots because the other spots are occupied while the electrical charging spots are empty in the most time. The reduction in performance is also the side effect of the special spot. The electrical charging spot is not only for the charging demand of the electrical car owner but also granting the priority of parking for the electrical vehicles and encouraging the potential of the electrical-powered vehicles. In the US, the tax on purchasing electrical vehicles is greater due to the lack of oil tax. The reserved parking spots for electrical vehicles are one of the limited ways to promote these vehicles with a new power source. According to the regional regulation in Beijing, China, at least 10% of the parking spots in the parking facilities shall be available for electrical charging and at least one of the parking spots shall be reserved parking spots.

To address these design concerns, this study provides a comprehensive tool to evaluate the potential outputs of the proposed design. To investigate the impact of the number of the electrical-charging only spots, it is assumed that the electrical-charging only spots do not have a time limit and these spots are located in the spots with the lowest utilization. Figure 5-9 illustrates the number of electrical-charging only spots impacts the average utilization, where the x-axis is the number of electrical-charging, the primary y-axis is for the average utilization of all spots, and the secondary y-axis is for the average daily revenue. Figure 5-9 shows the number of electrical-charging only spots has a negative correlation with the average utilization and the average daily revenue. This result is consistent with the fact that the ratio of the electrical vehicle is much lower than that of the gas vehicles according to the field survey. For the operational concern, the parking duration limitation of the electrical charging spots should be optimized. If the duration limitation is too low, the current customers cannot get served. If the duration is too high, the new customers may get rejected to the service. Figure 5-10 illustrates the impact of the duration limitation on the average utilization and the average daily revenue, where the number of electrical charging only spot equals one. Figure 5-10 shows the desired duration limitation is 240 min and 30 min with regards to the average utilization, and when the duration limitation is 30 min, the average daily revenue is optimized. However, the customers would feel worried about the penalty of the duration limit. Thus, the current policy of the campus parking facility is the 4-hour parking duration limit.

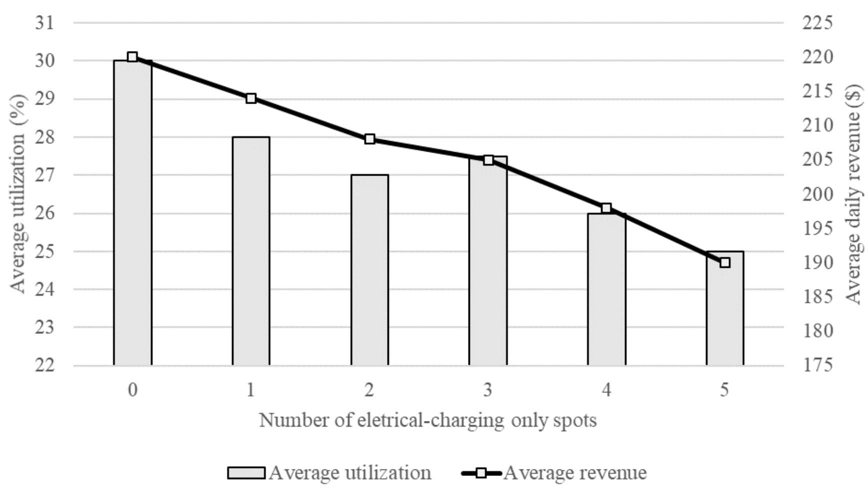



Figure 5-9 Comparison of average utilization when setting the various number of electrical-charging only spots

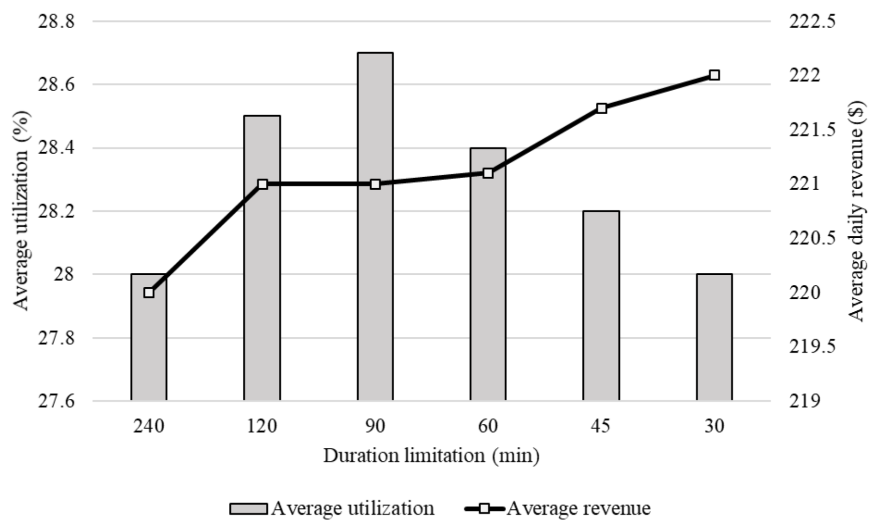



Figure 5-10 Comparison of various time limitation of electrical charging spots

New types of special spots are created since parking facilities have great externalities in land use, traffic, and business. Women priority spots are established for the door of vehicles can be fully opened and letting the strollers getting on and off, where the priority means the users of these spots are not legally enforced by the law. The objective of the establishment of these spots is out of the business decision. The setting of these special spots depends on the neighboring community of the parking facility. For example, the university parking facility doesn’t prefer the setting of women priority spots. However, such spots are preferred by the shopping centers and the hospitals because the settings of spots show the parking facility and the business managers care about the experience of woman customers, which promotes the business from the parking resource supply. Note that the business concern is one of the design criteria of the parking lot, and the proposed system does not capture the external business impact of the parking service.

For commercial sites, the shopping center manager and the operator may like to provide free parking for customers or parking hour extension for free. In other cases, the reserved spots are free for customers within a limited time (usually 30 minutes). For example, the reserved parking spot for online pickup customers is established in Bayshore Mall, Glendale, WI.

From the view of the urban traffic management, the manager of public parking lot plans to establish reserved parking spots for the car-sharing and carpooling vehicles. If the customers in the parking facilities have similar destinations, establishing the car-pooling spots benefit the mobility of the travelers. The objective is to promote the sharing of vehicles for serving more passengers in congested areas since the parking resource takes considerable lane use in the urban area and is of low efficiency.

The special spots may take more spaces and not the economy in the land use and impact the experience of the normal customers. However, the design concept is based on the external effects of the parking service instead of the profit and the performance of the facility. The setting of the special spots depends on the case-specific concern of the owner and manager. The proposed system is able to reduce the side effect of the setting of the special spot to the minimum.

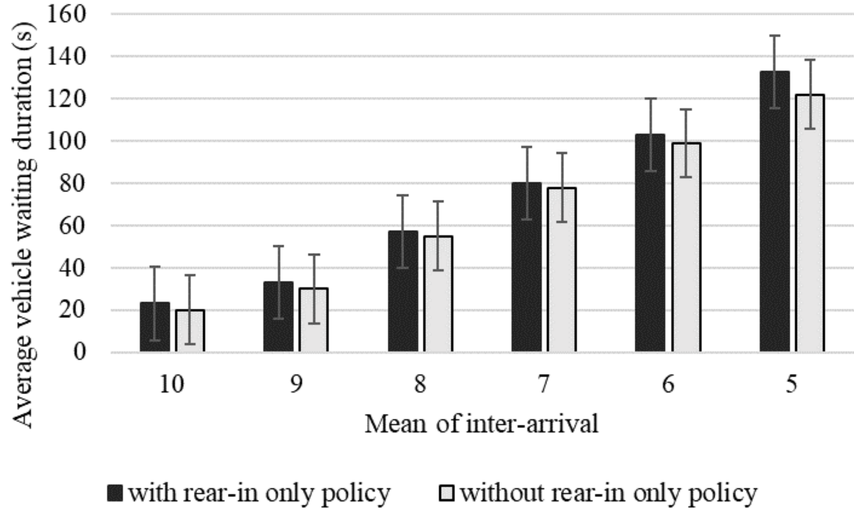
### 5.2.4 The evaluation of the reverse parking policy

In Japan, the reserve or rear-in parking is encouraged and widely accepted as the parking etiquette. In the US, it still raises the debating of which is safer, rear-in or head-in. And it is reported that reverse parking is illegal for some stalls but encouraged in others. There're two safety reasons to use the reverse parking (Huey et al., 1997): (a) The rear-in parking ensures reversing errors occur only within the confines of the car spot, and not in the open areas where moving vehicles and pedestrians. (b) When leaving the parking space, the vision of the reverse parking driver is clearer than the head-in parking driver.

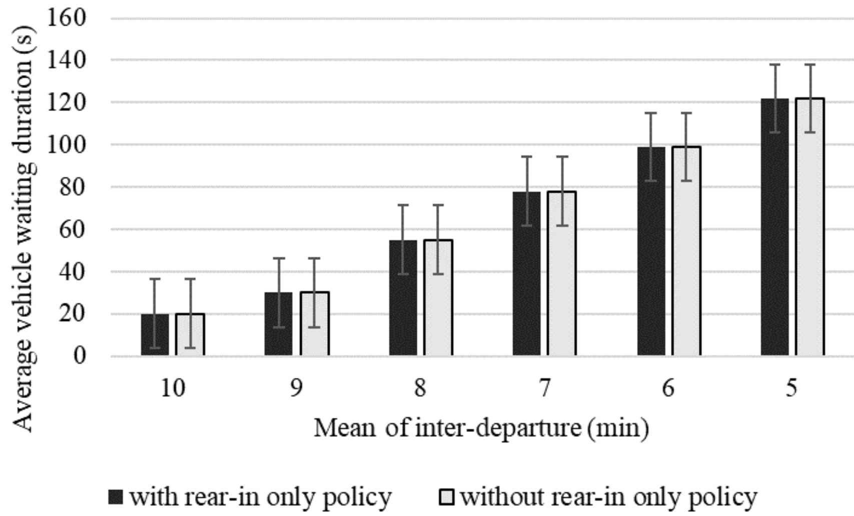
National Highway Traffic Safety Administration (NHTSA) estimated that "267 people are killed and 15,000 injured each year by drivers who back into them usually in driveways or parking lots” in 2012. NHTSA has ruled that all new vehicles under 10,000 pounds (including passenger vehicles, buses, and trucks) must be equipped with rear visibility technology by May 2018.

There lacks the evaluation of the policy of encouraging rear-in parking. The proposed system can not only measure the safety of the maneuvers, but also the efficiency of the parking facility with compliance with the policy. In the simulation model, the assumptions are made to replicate the simplified scenarios. The reverse-only parking policy is assumed for testing purpose. The non-restrictive parking policy is used as the benchmark. According to the experimental results, the rear-in parking policy can reduce the number of weaving between the pedestrians and the reversing vehicles, which means the rear-in parking is safer than the head-in parking. In addition, the efficiency of the reverse-only case is greater than the non-restrictive case. When another vehicle is waiting to use the same spot, the pulling-out maneuver of reverse parking does not occupy the conflicting right-of-way.

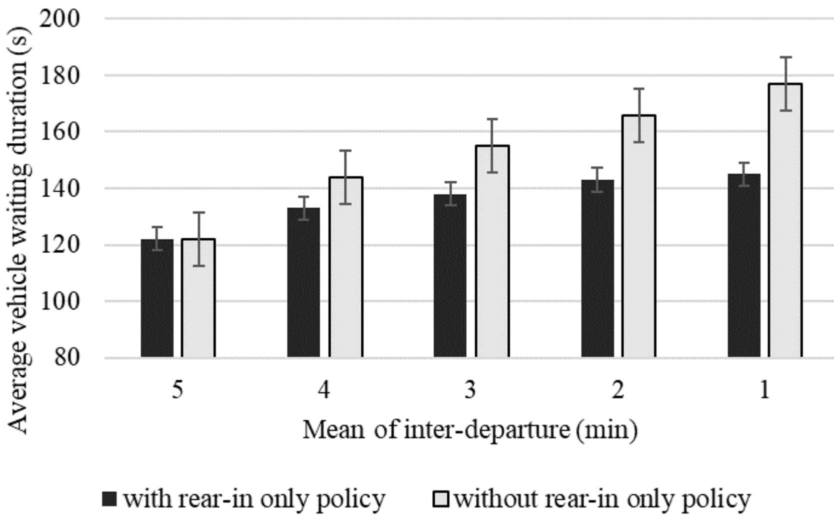
Note that the restrictive policy is not made as the law in the real world. To investigate the impact of the application of this policy, the simulation is used to derive the outcome. Figure 5-11 shows the simulated results show the rear-in policy may help to relieve the congestion in the parking system during vehicle departure rush hour. However, Figure 5-11 (a) shows the rear-in only policy may increase the duration of the parking maneuvers and reduce the efficiency during arrival rush hour. The range of mean inter-departure time in Figure 5-11 (b) is derived from the field survey data while the mean inter-departure time in Figure 5-11 (c) is assumed. Figure 5-11 (b) and (c) show the efficiency improvement depends on the departure pattern of the parking facility. In this case, the university does not have a significant p.m. rush hour, thus, the efficiency benefit of the policy is marginal.




(a) with a various mean of the inter-arrival time




(b) with a various mean of the inter-departure time




(c) with a various mean of the inter-departure time

Figure 5-11 Comparison of performances with and without rear-in only policy under various scenarios

# 6 Summary and conclusion

This research proposes an agent-based simulation framework for parking choice modeling to capture parking behaviors. The elements of the parking system, such as drivers, pedestrians, aisles, spots, entrances, etc., are modeled as agents. The agents are classified based on the measures and behaviors into two categories: entities (e.g. drivers and pedestrians) and locations (e.g. aisles and spots). The processes transition and the movement in the in-lot network of drivers and pedestrians are modeled as state machines. This study originally proposed to formulate the state machines of the processes transition and the movement in the first-order logic framework. The logic-based rules are presented in the pseudocode, which costs trivial efforts to be justified and solved by the logic programming language Prolog. The consequent choice behaviors of the entities are modeled to replicate the spot choice and route choice within the parking facility. The drivers are classified in the informed and uninformed. The informed drivers have the global vision of the parking lot and make choice based on the conventional multiple classification models. The uninformed parking spot choice model assumes the driver should make the decision one-by-one without aftereffect. In the parking spot choice model, the critical attribute of the potential spot is the walking distance from the parking spot to the pedestrian exit or the walking destination in the parking lot. In the route choice model, the critical attribute is the length of the potential path. If the parking guidance system is applicable, the occupancy of the path should be considered. The proposed model is extendable for modern special types of parking spots and intelligent parking management system and parking guidance information system. The historical record and statistics, such as utilization, turnover, occupied duration, etc., are collected to further analyze the potential outcome of the design and operational decision. The parking simulation engine is implemented in Java to justify the state machine and behavior models.

To investigate the performance of the proposed simulator, this study designs a Software as a Service (SaaS) Graphic User Interface (GUI) to visualize the movement of drivers and pedestrians within a parking lot and implements the simulation engine in Java and the web-based GUI in HTML/JavaScript/CSS. A methodology for data collection, processing, and extraction of user behaviors in the parking system is also developed. The application of the developed simulation system using a real-world case study demonstrates its capability of retrieving quantified measures and key performance indicators to support decisions in designing, maintaining, operating parking facilities.

To justify the proposed methodology, real-world data are collected, and the parameters of the proposed model are calibrated in the case of a surface parking lot on campus. The results of the goodness of fitting test show the inter-arrival is distributed exponentially with the mean 48.01 seconds. The experiments show the critical active capacity of this parking lot is 10 vehicles when the inter-arrival mean is 18 second. A Long-Short Term Memory (LSTM) neural network is used to predict the dynamic arrival and departure of the vehicles. The LSTM shows the prediction accuracy is 91% in the studied case. The measures of the simulation results may help to select the best parking lot layout. The heatmap for the utilization of the locations shows the spots and aisles closer to the pedestrian exits are more frequently used, which is consistent with the choice behavior model. The critical finding of this study is that the active capacity of the parking system. The definition of active capacity is the largest number of actively moving vehicles in the parking system. The parking simulator finds the value of the active capacity of the existing facility and the planning layout.

A discussion is conducted to compare the proposed parking simulator with the state-of-practice simulators. In comparison to the existing simulators, the proposed model can facilitate the specific traffic dynamic and choice models for the parking simulation.

A numerical study is conducted to justify the application of the proposed system and provide a simulator aided design method. The numerical tests show: (a) the smart check-in device has marginal benefits in reducing the vehicle waiting time. (b) the flexible pricing policy may increase the average daily revenue if the elasticity of the price is not involved. (c) The number of electrical charging only spots have a negative impact on the performance of the parking facility. (d) The rear-in only policy may increase the duration of the parking maneuvers and reduce the efficiency during arrival rush hour.

The proposed system can provide sufficient information to aid the design of smart parking facilities with multiuse, automated spots, shared spots, electrical charging, etc. Real-world cases are investigated to illustrate the simulator-aided parking facility design and management. Note that the objective of design and management is to improve the customer experience, however, the customer experience cannot be simply modeled with simulated quantitative measures due to the external impact factors.

Future research may include: (a) the calibration and integration with the parking guidance system and the sensor network within the parking facility; (b) the optimal strategy of the temporal permit-only policy if the permit user has the flexible reserved parking spot during the peak hour; and (c) the case study in multilevel and automated parking facilities.

## References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of OSDI 2016, 16, 265-283.

Abdelfatah, A.S. & Taha, M. (2014). Parking capacity optimization using linear programming. Journal of Traffic and Logistics Engineering, 2(3), 176-181.

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., ... & Bengio, Y. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.

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

Balas, E. (1975). Disjunctive programming: cutting planes from logical conditions. In O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Eds.), Nonlinear Programming, New York: Academic Press, 2, 279-312.

Balas, E. (1977). A note on duality in disjunctive programming. Journal of Optimization Theory Application, 21, 523-527.

Balas, E. (1979). Disjunctive programming. Annals of Discrete Mathematics, 5, 3-51.

Barth, P. (1995). Logic-Based 0-1 Constraint Programming. Boston: Kluwer Academic Publishers.

Barone, R.E., Giuffrè, T., Siniscalchi, S.M., Morgano, M.A. & Tesoriere, G. (2013). Architecture for parking management in smart cities. IET Intelligent Transport Systems, 8(5), 445-452.

Beheshti, R. & Sukthankar, G. (2015). A hybrid modeling approach for parking and traffic prediction in urban simulations. AI & Society, 30(3), 333-344.

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

Ben-Joseph, E. (2012). ReThinking a lot: the design and culture of parking, Vol. 7. Cambridge, MA: MIT Press.

Bergstra, J., Desjardins, G., Lamblin, P., & Bengio, Y. (2009). Quadratic polynomials learn better image features. Technical Report 1337. Département d'Informatique et de Recherche Opérationnelle, Université de Montréal.

Bonsall, P., Palmer, I. (2004). Modelling drivers' car parking behaviour using data from a travel choice simulator. Transportation Research Part C: Emerging Technologies, 12(5), 321-347.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Bureau of Justice Statistics (BJS), US. (2016). National Crime Victimization Survey. Retrieved from https://www.bjs.gov/index.cfm?ty=dcdetail&iid=245 on July 30th, 2018.

Caicedo, F., Blazquez, C., & Miranda, P. (2012). Prediction of parking space availability in real time. Expert Systems with Applications, 39(8), 7281-7290.

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, 1-26.

Capital City Development Corp (CCDC). (2012). Downtown Boise parking strategic plan appendix F1: Parking as an economic development strategy. Retrieved from http://www.ccdcboise.com/wp-content/uploads/2016/02/Document-F1-Parking-as-a-Economic-Development-Strategy-White-Paper.pdf on August 15, 2018.

Capital City Development Corp (CCDC). (2015). Downtown Boise parking supply/demand update. Retrieved from http://www.ccdcboise.com/wp-content/uploads/2016/02/Document-O1-Downtown-Boise-Parking-Supply-Demand-Update-2015.pdf.

Cassady, C. R., & Kobza, J. E. (1998). A probabilistic approach to evaluate strategies for selecting a parking space. Transportation Science, 32(1), 30-42.

Ceballos, G., & Curtis, O. (2004). Queue analysis at toll and parking exit plazas: a comparison between multi-server queuing models and traffic simulation. In ITE 2004 Annual Meeting and Exhibit. Retrieved from https://pdfs.semanticscholar.org/5528/1f0118277b2c4e557f63e79212ecccdce24d.pdf on July 30th, 2018.

Chaniotakis, E., & Pel, A. J. (2015). Drivers' parking location choice under uncertain parking availability and search times: A stated preference experiment. Transportation Research Part A: Policy and Practice, 82, 228-239.

Chollet, F. & others. (2015). Keras. Retrieved from https://keras.io on July 30th, 2018.

Chen, M., Hu, C., & Chang, T. (2011). The research on optimal parking space choice model in parking lots. In Proceedings of 3rd International Conference on Computer Research and Development (ICCRD). IEEE. 2, 93-97.

Chen, Y. Y., Lv, Y., Li, Z. & Wang, F. Y. (2016). Long short-term memory model for traffic congestion prediction with online open data. In Proceedings of 19th International Conference on Intelligent Transportation Systems, IEEE, 132-137.

Chrest, A. P., Smith, M. S., Bhuyan, S., Iqbal, M. & Monahan, D. R. (2012). Parking structures: planning, design, construction, maintenance and repair. 3rd ed. New York: Springer Science + Business Media.

Clocksin, W. F., & Mellish, C. S. (2012). Programming in Prolog: Using the ISO standard. New York: Springer Science + Business Media.

Colmerauer A., Kanouia H., Pasero R., & Roussel P. (1973). Un systeme de communication homme-machine enfrancais, Technical Report, Universite d'Aix-Marseilles II, Groupe intelligence articielle. (in French)

Colmerauer, A. (1986). Theoretical model of Prolog II. Logic programming and its applications, 3-31.

Colmerauer, A. (1990). An introduction to Prolog-III. Communications of the ACM, 33 (7). 52-68.

Colmerauer A. (1996). Specifications of Prolog IV. Draft.

Colmerauer, A, et al. (1996). Manual of Prolog-VI. Retrieved from http://prolog-heritage.org/en/m4.html. Accessed on July 30th, 2018.

Davidsson, P. (2000). Multi agent based simulation: beyond social simulation. In International Workshop on Multi-Agent Systems and Agent-Based Simulation. Berlin, Heidelberg: Springer. 97-107.

Design Guidelines for Off-Street Parking - City of Philadelphia. (2010). Retrieved from https://www.phila.gov/CityPlanning/resources/Publications/parkingdesign.pdf on July 30th, 2018.

Dieussaert, K., Aerts, K., Steenberghen, T., Maerivoet, S. & Spitaels, K. (2009). SUSTAPARK: an agent-based model for simulating parking search. In Proceeding of 12th AGILE International Conference on Geographic Information Science, 1-11.

Duan, Y., Lv, Y. & Wang, F.Y. (2016). Travel time prediction with LSTM neural network. In Proceedings of 19th International Conference on Intelligent Transportation Systems. IEEE. 1053-1058.

Dymchenko, S., & Mykhailova, M. (2015). Declaratively solving Google code jam problems with Picat. In International Symposium on Practical Aspects of Declarative Languages, Cham: Springer, 50-57.

Feitelson, E., & Rotem, O. (2004). The case for taxing surface parking. Transportation Research Part D: Transport and Environment, 9(4), 319-333.

Fellendorf, M. (1994). VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In 64th Institute of Transportation Engineers Annual Meeting Springer, 1-9.

Frank, E., Hall, M. A. & Witten, I. H. (2016). The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Morgan Kaufmann.

Fu, R., Zhang, Z. & Li, L. (2016), November. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of 2016 Youth Academic Annual Conference of Chinese Association of Automation. IEEE. 324-328.

Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In Proceeding of 30th Conference on Neural Information Processing Systems. 1019-1027.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. Cambridge: MIT Press.

Granot, F., & Hammer, P. L. (1971). On the use of boolean functions in 0-1 linear programming, Methods of Operation Research. 154-184.

Graves, A. (2012). Supervised sequence labelling with recurrent neural networks. Retrieved from http://books. google. com/books on July 30th, 2018.

Guo, L., Huang, S. & Sadek, A.W. (2013). A novel agent-based transportation model of a university campus with application to quantifying the environmental cost of parking search. Transportation Research Part A: Policy and Practice, 50, 86-104.

Hall, M., Frank, E., Holmes, E., Pfahringer, B., Reutemann, P., & Witten, I.H. (2009). The WEKA Data Mining Software: An Update. SIGKDD Explorations, Volume 11, Issue 1.

Hamilton, J. D. (1994). Time series analysis. Princeton, NJ: Princeton University Press.

Hammer, P. L., & Rudeanu, S. (1968). Boolean Methods in Operations Research and Related Areas, Springer, Berlin.

Harrel, C. R., Ghosh, B. K., & Bowden, R. (2004). Simulation using promodel. Simulation, 14, 3.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. 2nd ed. Springer, New York, NY.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Hooker, J. N., & Osorio, M. A. (1999). Mixed logical-linear programming. Discrete Applied Mathematics, 96, 395-442.

Hooker, J. N., Yan, H., Grossmann, I., & Raman, R. (1994). Logic cuts for processing networks with fixed charges. Computer Operation Research, 21, 265-279.

Horni, A., Montini, L., Waraich, R. A., & Axhausen, K. W. (2013). An agent-based cellular automaton cruising-for-parking simulation. Transportation Letters, 5(4), 167-175.

Huey, R., Harpster, J., & Lemer, N. (1997). Field measurement of naturalistic backing behavior. National Highway Traffic Safety Administration of United States. Technical Report, DOT-HS-808-532.

Iranpour, R. & Tung, D. (1989). Methodology for optimal design of a parking lot. Journal of transportation engineering, 115(2), 139-160.

Jaffar, J., & Lassez, J. L. (1987a). Constraint logic programming, In Proceedings of the 14th ACM Symposium on Principles of Programming Languages, 111-119.

Jaffar, J., & Lassez, J. L. (1987b). From unification to constraints, Logic programming 87, Proceedings of the Sixth Conference, Springer, Berlin, 1-18.

Jeroslow, R. E. (1987). Representability in mixed integer programming, I: characterization results, Discrete Applied Mathematics, 17, 223-243.

Jeroslow, R.E. (1989). Logic-Based Decision Support: Mixed Integer Model Formulation. Annals of Discrete Mathematics, 40.

Jin, Z. (2003). A Study of the Planning Method and Its Application for the Urban Parking. Urban Transport of China, 1(1), 23-27.

Kaelbing, L.P. (1986). An architecture for intelligent and reactive systems. In: Georgeff, M.P., and A.L. Lansky (Eds.). Reasoning About Actions and Plans - Proceedings of the 1986 Workshop. Los Altos, CA.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kowalski, R.A., (1974). Predicate logic as programming language, Proceedings of the IFIP Congress, North-Holland, Amsterdam. 569-574.

Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent development and applications of SUMO-Simulation of Urban MObility. International Journal on Advances in Systems and Measurements, 5(3&4), 128-138.

Levy, S. (1992). Artificial Life. New York: Vintage Books.

Levy, N., Martens, K., & Benenson, I. (2013). Exploring cruising using agent-based and analytical models of parking. Transportmetrica A: Transport Science, 9(9), 773-797.

Levy, N., Render, M. & Benenson, I. (2015). Spatially explicit modeling of parking search as a tool for urban parking facilities and policy assessment. Transport Policy, 39, 9-20.

Li, Q. (2014). Evaluation of Parking Guidance Information System with Multi-agent Based Simulation. Ph.D. dissertation. Kyoto University.

Ma, X., Tao, Z., Wang, Y., Yu, H. & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies, 54, 187-197.

Martens, K., Benenson, I., & Levy, N. (2010). The dilemma of on-street parking policy: exploring cruising parking using an agent-based model. In: Jiang, Bin, Yao, Xiaobai (Eds.), Geospatial Analysis and Modelling of Urban Structure and Dynamics, Geo Journal Library, 99, 121-138.

McAloon, K., & Tretko, C. (1995). 2LP: linear programming and logic programming. In: P. van Hentenryck, V. Saraswat (Eds.), Principles and Practice of Constraint Programming, MIT Press, Cambridge, MA, 99-114.

Tretkoff, C. (1996). Optimization and computational logic. John Wiley & Sons.

McDonald, A.I.A. & Sanders, S. (2007). The Parking Garage. Design and Evolution of A Modern Urban Form. Urban Land Institute.

Nagel, K., & Rasmussen, S. (1994). Traffic at the edge of chaos. In: Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, 222-225.

Nagel, K. (1996). Particle hopping models and traffic flow theory. Physical Review E. 53, 4655-4672.

Nagel, K. (1998). From particle hopping models to traffic flow theory. Transportation Research Record: Journal of the Transportation Research Board, 1644, 1-9.

National Development and Reform Commission of P. R. China. (2015). Instructions for enhancing urban parking facility construction. Document No. 1788. Retrieved from http://www.sdpc.gov.cn/fzgggz/nyjt/fzgh/201508/t20150811_744967.html on April 17, 2017.

National Development and Reform Commission of P. R. China (NDRC of P. R. China). (2016). Notice of Recent Tasks and Assignment for Promoting Construction of Parking Lots. Document No. 159. Retrieved from http://www.ndrc.gov.cn/gzdt/201603/t20160329_796694.html on April 17, 2017.

National Safety Council (NSC). (2016). Injury facts, 2015 Edition. Itasca, IL.

Obdeijn, C. (2011). Modelling parking guidance systems in S-Paramics: development of an S-Paramics tool to simulate parking guidance systems. Master thesis, University of Twente, Netherlands.

Off-Street Parking Design Manual - City of Solana Beach. (2012). Retrieved from https://www.ci.solana-beach.ca.us/vertical/sites/%7B840804C2-F869-4904-9AE3-720581350CE7%7D/uploads/SB_Off_Street.pdf on July 30th, 2018.

Paczuski, M., & Nagel, K. (1996). Self-organized criticality and 1/f noise in traffic. In Traffic and granular flow. Los Alamos Unclassified Report, 95(4108), 41.

Prevost, F. (1985). Curb parking: theory and model. Technical Report, 5-1. Institute of Transportation Studies, University of California, Davis.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Raman R., & Grossmann I. E. (1991). Relation between MILP modeling and logical inference for chemical process synthesis. Computers & Chemical Engineering, 15, 73-84.

Raman, R., & Grossmann, I.E. (1993a). Symbolic integration of logic in MILP branch and bound methods for the synthesis of process networks. Annals of Operation Research. 42, 169-191.

Raman, R., & Grossmann, I.E. (1993b). Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. Computers & Chemical Engineering, 17, 909-927.

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of adam and beyond. In International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=ryQu7f-RZ on July 30, 2018.

Rushton, B. T. (2001). Low-impact parking lot design reduces runoff and pollutant loads. Journal of Water Resources Planning and Management, 127(3), 172-179.

Santé, I., García, A. M., Miranda, D., & Crecente, R., 2010. Cellular automata models for the simulation of real-world urban processes: A review and analysis. Landscape and Urban Planning. 96(2), 108-122.

Seide, F., & Agarwal, A. (2016). Cntk: Microsoft's open-source deep-learning toolkit. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2135-2135.

Shaaban, K. & Pande, A. (2016). Classification tree analysis of factors affecting parking choices in Qatar. Case Studies on Transport Policy, 4(2), 88-95.

Shao, C. F., Zhang, X., et al. (eds.) (2016). Urban parking design and management. In Urban Transportation Design. Beijing: Beijing Jiaotong University Press.

Shi, S., Wang, Q., Xu, P., & Chu, X. (2016), November. Benchmarking state-of-the-art deep learning software tools. IEEE 7th International Conference on Cloud Computing and Big Data (CCBD), 99-104.

Siebers, P. (2007). Worker performance modeling in manufacturing systems simulation. In Handbook of Research on Nature-Inspired Computing for Economics and Management. IGI Global. 661-678.

Siebers, P. O., & Aickelin, U. (2008). Introduction to multi-agent simulation. In Encyclopedia of decision making and decision support technologies. IGI Global. 554-564.

Smith, M., Carwile, R. (2016). Parking. In Pande, A., Wolshon, B. Traffic Engineering Handbook: Institute of Transportation Engineers, 7th edition, 437-500.

Sterling, L., & Shapiro, E. Y. (1994). The art of Prolog: advanced programming techniques. MIT press.

Sun, Y., Fan, W., & Schonfeld, P. (2016). Static Parking Choice Model with Consideration of Parking Duration. Transportation Research Record: Journal of the Transportation Research Board, 2543, 134-142.

Thompson, R. G. & Richardson, A. J. (1998). A parking search model. Transportation Research Part A: Policy and Practice, 32(3), 159-170.

Tian, Y. & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. In IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). IEEE. 153-158.

Tisue, S., & Wilensky, U. (2004). NetLogo: Design and implementation of a multi-agent modeling environment. In Proceedings of agent. 7-9.

Turkay, M., Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. Computers & Chemical Engineering, 20, 959-978.

Waraich, R. & Axhausen, K. (2012). Agent-based parking choice model. Transportation Research Record: Journal of the Transportation Research Board, 2319, 39-46.

Waraich, R. A., Dobler, C., & Axhausen, K. W. (2012). Modelling parking search behaviour with an agent-based approach. In 13th International Conference on Travel Behaviour Research. International Association for Travel Behavior Research (IATBR), 1-12.

Wang, W. (2016). Theory and method of designing pedestrian aisle within parking lots. Highways & Automotive Applications, 175, 29-31.

Weant, R. A. (1978). Parking garage planning and operation. Westport, Connecticut: Eno Foundation for Transportation.

Wekerle, G.R. & Whitzman, C. (1995). Safe cities: Guidelines for planning, design, and management. Van Nostrand Reinhold Company.

Wikihow contributors. (2018). User parking lot etiquette. Retrieved from https://www.wikihow.com/Use-Parking-Lot-Etiquette on Aug 30, 2018

Wolfram, S. (1994). Cellular Automata and Complexity. Reading, MA: Addison-Wesley.

van der Waerden, P.J.H.J. & Timmermans, H. (1997). Parking simulation using a geographical information system. Decision Support Systems in Urban Planning, 310-322.

van der Waerden, P., Borgers, A. & Timmermans, H. (2003), January. Travelers micro-behavior at parking lots: a model of parking choice behavior. In Processing of the 82nd Annual Meeting of the Transportation Research Board, 1212.

van der Waerden, P., Timmermans, H. & da Silva, A. N. R. (2015). The influence of personal and trip characteristics on habitual parking behavior. Case Studies on Transport Policy, 3(1), 33-36.

Vinayakumar, R., Soman, K. P. & Poornachandran, P. (2017). Applying deep learning approaches for network traffic prediction. In Proceedings of 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE. 2353-2358.

Vo, T.T.A., van Der Waerden, P. & Wets, G. (2016). Micro-simulation of Car Drivers' Movements at Parking Lots. Procedia Engineering, 142, 100-107.

Ukpong, I., Dissanayake, D., & Zhang, B. (2007). Development of a micro-simulation traffic model to investigate on-street parking and its impacts on sustainable environments: A case study in the city of Newcastle upon Tyne. In 11th World Conference on Transport Research Society.

Xinhua News Agency (XNA). (2016). Guangzhou will raise 17.5 billion yuan to build parking spaces to relief parking difficulty. Retrieved on April 17, 2017 from http://news.xinhuanet.com/fortune/2016-03/06/c_1118247775.htm.

Yanjie, J., Wei, W., & Wei, D. (2009). Micro-level parking behaviors within parking facilities and optimal parking space choice model. Journal of Southeast University (Natural Science Edition), 39(12), 399-403.

Young, W. (1986). PARKSIM/1: a network model for parking facility design. Traffic Engineering & Control, 27(10), 606-613.

Young, W. & Thompson, R. (1987a). PARKSIM/1: a computer graphics approach for parking-lot layouts. Traffic Engineering & Control, 28(3), 120-123.

Young, W. & Thompson, R. (1987b). PARKSIM/1: data presentation and evaluation. Traffic Engineering & Control, 28(5), 294-297.

Young, W. (2000). Modeling Parking: in D. Hensher and K.J. Button (eds.), Handbook of Transport Modelling, 1, Oxford: Elsevier Science, 409-420.

Young, W. & Taylor, M. (1991). A parking model hierarchy. Transportation, 18(1), 37-58.

Young, W. & Weng, T.Y. (2005). Data and parking simulation models. In Simulation approaches in transportation analysis. Springer, Boston, MA. 235-267.

Yu, R., Li, Y., Shahabi, C., Demiryurek, U. & Liu, Y. (2017). Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM International

Conference on Data Mining. Society for Industrial and Applied Mathematics. 777-785.

Yuan, Y. & Liu, K. (2014). The 3D-simulation implementation of parking guidance system. In Proceedings of, 2014 International Conference on Audio, Language and Image Processing (ICALIP). IEEE. 635-639.

Yue, W. L. & Young, W. (1997a). Simulation in selecting the best parking lot layout. In Modsim97 International Congress on Modelling and Simulation Proceedings. Modeling and Simulation Society of Australia and New Zealand. Canberra ACT, Australia. 1484-1489.

Yue, W. L., & Young, W. (1997b). An introduction of a parking design and simulation model. Eastern Asia Society for Transportation Studies. 4, 595-613.

Zhao, Z., Chen, W., Wu, X., Chen, P. C. & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 11(2), 68-75.

Zhou, N. F. (2016). Programming in Picat. In International Symposium on Rules and Rule Markup Languages for the Semantic Web. Cham: Springer, 3-18.

Zhou, N. F., & Fruhman, J. (2017). Canonicalizing High-Level Constructs in Picat. In International Symposium on Practical Aspects of Declarative Languages. Cham: Springer, 19-33.

Zhou, N. F., & Kjellerstrand, H. (2014). Solving several planning problems with Picat. In Proceeding of the 26th Chinese Control and Decision Conference, IEEE, 346-350.

Zhao, C., Li, S., Wang, W., Li, X., & Du, Y. (2018). Advanced Parking Space Management Strategy Design: An Agent-Based Simulation Optimization Approach. Transportation Research Record. https://doi.org/10.1177/0361198118758671.

Zhuo, Q., Li, Q., Yan, H. & Qi, Y. (2017). Long short-term memory neural network for network traffic prediction. In Proceedings of 12th International Conference on Intelligent Systems and Knowledge Engineering. IEEE. 1-6.

# APPENDICES

## APPENDIX A A sample case for concurrent computing

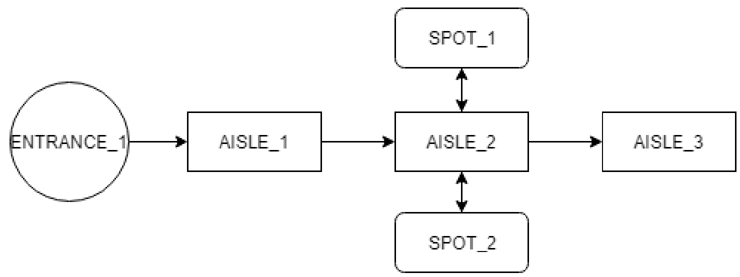



Figure A-1 A toy inner parking lot network

The concurrent version of the proposed simulation engine is implemented in Java. The example is encoded in Java, and the prompt output of this sample case is listed as follows:

[INFO] [04/25/2018 15:54:46.122] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] add location AISLE_1

[INFO] [04/25/2018 15:54:46.122] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] create clock

[INFO] [04/25/2018 15:54:46.122] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] add location AISLE_2

[INFO] [04/25/2018 15:54:46.122] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] add location AISLE_3

[INFO] [04/25/2018 15:54:46.122] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] add location SPOT_1

[INFO] [04/25/2018 15:54:46.123] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] add location SPOT_2

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] add location ENTRANCE_1

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] adding edge AISLE_1_AISLE_2

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] adding edge AISLE_2_AISLE_3

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] adding edge SPOT_1_AISLE_2

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] edge out akka://simulator/user/lot/AISLE_1 added

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] edge in akka://simulator/user/lot/AISLE_2 added

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] edge out akka://simulator/user/lot/AISLE_2 added

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] edge in akka://simulator/user/lot/AISLE_3 added

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] edge in akka://simulator/user/lot/AISLE_2 added

[INFO] [04/25/2018 15:54:46.152] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] edge in akka://simulator/user/lot/AISLE_2 added

[INFO] [04/25/2018 15:54:46.153] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] edge in akka://simulator/user/lot/AISLE_2 added

[INFO] [04/25/2018 15:54:46.153] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] edge in akka://simulator/user/lot/AISLE_2 added

[INFO] [04/25/2018 15:54:46.153] [simulator-akka.actor.default-dispatcher-5] [akka://simulator/user/printer] adding edge SPOT_2_AISLE_2

[INFO] [04/25/2018 15:54:50.108] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] now running

[INFO] [04/25/2018 15:54:50.112] [simulator-akka.actor.default-dispatcher-10] [akka://simulator/user/printer] VEHICLE_0 is created

[INFO] [04/25/2018 15:54:50.114] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] create a new vehicle

[INFO] [04/25/2018 15:54:50.119] [simulator-akka.actor.default-dispatcher-2] [akka://simulator/user/printer] move VEHICLE_1 to ENTRANCE_1

[INFO] [04/25/2018 15:54:50.121] [simulator-akka.actor.default-dispatcher-13] [akka://simulator/user/printer] entrance is occupied

/user/lot/AISLE_2

[INFO] [04/25/2018 15:54:50.130] [simulator-akka.actor.default-dispatcher-10] [akka://simulator/user/printer] move VEHICLE_1 out of /user/lot/AISLE_1

[INFO] [04/25/2018 15:54:50.131] [simulator-akka.actor.default-dispatcher-10] [akka://simulator/user/printer] move VEHICLE_1 to /user/lot/AISLE_2

[INFO] [04/25/2018 15:54:50.131] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] VEHICLE_1 is created

/user/lot/AISLE_3

[INFO] [04/25/2018 15:54:50.132] [simulator-akka.actor.default-dispatcher-8] [akka://simulator/user/printer] create a new vehicle

[INFO] [04/25/2018 15:54:50.132] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] move VEHICLE_1 out of /user/lot/AISLE_2

[INFO] [04/25/2018 15:54:50.132] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] move VEHICLE_2 to ENTRANCE_1

[INFO] [04/25/2018 15:54:50.132] [simulator-akka.actor.default-dispatcher-10] [akka://simulator/user/printer] move VEHICLE_1 to /user/lot/AISLE_3

[INFO] [04/25/2018 15:54:50.133] [simulator-akka.actor.default-dispatcher-9] [akka://simulator/user/printer] entrance is occupied

/user/lot/AISLE_2

[INFO] [04/25/2018 15:54:50.133] [simulator-akka.actor.default-dispatcher-2] [akka://simulator/user/printer] VEHICLE_1: next location is occupied

[INFO] [04/25/2018 15:54:50.133] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] move VEHICLE_2 out of /user/lot/AISLE_1

[INFO] [04/25/2018 15:54:50.134] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] move VEHICLE_2 to /user/lot/AISLE_2

[INFO] [04/25/2018 15:54:50.134] [simulator-akka.actor.default-dispatcher-14] [akka://simulator/user/printer] VEHICLE_1: next location is occupied

[INFO] [04/25/2018 15:54:50.134] [simulator-akka.actor.default-dispatcher-14] [akka://simulator/user/printer] VEHICLE_2 is created

[INFO] [04/25/2018 15:54:50.135] [simulator-akka.actor.default-dispatcher-12] [akka://simulator/user/printer] VEHICLE_2: next location is occupied

[INFO] [04/25/2018 15:54:50.135] [simulator-akka.actor.default-dispatcher-12] [akka://simulator/user/printer] create a new vehicle

[INFO] [04/25/2018 15:54:50.135] [simulator-akka.actor.default-dispatcher-6] [akka://simulator/user/printer] move VEHICLE_3 to ENTRANCE_1

[INFO] [04/25/2018 15:54:50.136] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] entrance is occupied

[INFO] [04/25/2018 15:54:50.136] [simulator-akka.actor.default-dispatcher-11] [akka://simulator/user/printer] VEHICLE_2: next location is occupied

[INFO] [04/25/2018 15:54:50.136] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] VEHICLE_1: next location is occupied

[INFO] [04/25/2018 15:54:50.136] [simulator-akka.actor.default-dispatcher-3] [akka://simulator/user/printer] VEHICLE_3: next location is occupied

[INFO] [04/25/2018 15:54:50.137] [simulator-akka.actor.default-dispatcher-5] [akka://simulator/user/printer] entrance is occupied

[INFO] [04/25/2018 15:54:50.138] [simulator-akka.actor.default-dispatcher-14] [akka://simulator/user/printer] VEHICLE_2: next location is occupied

[INFO] [04/25/2018 15:54:50.138] [simulator-akka.actor.default-dispatcher-14] [akka://simulator/user/printer] VEHICLE_1: next location is occupied

[INFO] [04/25/2018 15:54:50.138] [simulator-akka.actor.default-dispatcher-7] [akka://simulator/user/printer] VEHICLE_3: next location is occupied

[INFO] [04/25/2018 15:54:50.138] [simulator-akka.actor.default-dispatcher-7] [akka://simulator/user/printer] entrance is occupied

[INFO] [04/25/2018 15:54:50.139] [simulator-akka.actor.default-dispatcher-10] [akka://simulator/user/printer] VEHICLE_1: next location is occupied

[INFO] [04/25/2018 15:54:50.139] [simulator-akka.actor.default-dispatcher-7] [akka://simulator/user/printer] VEHICLE_2: next location is occupied

[INFO] [04/25/2018 15:54:50.139] [simulator-akka.actor.default-dispatcher-10] [akka://simulator/user/printer] VEHICLE_3: next location is occupied

[INFO] [04/25/2018 15:54:50.140] [simulator-akka.actor.default-dispatcher-4] [akka://simulator/user/printer] entrance is occupied

## APPENDIX B A sample code of PROLOG

```
:- dynamic
in/2.

/* report the position of C */
report(car(X)) :- in(car(X), P), write(car(X)), write(': '), write(P), write(' ,').
reportall :- report(car(1)), report(car(2)), report(car(3)).

/* don't move if it is already parked in a spot */
move(C) :- in(C, spot(_)),
   write(C), write(' parked'),nl.
/* make a move if possible */
move(C) :- in(C, P1), connect(P1, A, P2), not(in(_, P2)),
   processmachine(C,P2),
   write(C), write(' moved '), write(A), write(' -> '),
   retract(in(C, P1)), assert(in(C, P2)),
   reportall, nl.
/* wait without considering time */
move(C) :- in(C, P1), connect(P1, _, P2), in(_, P2),
   write(C), write(' waited'), nl.

/* set up parking lot */
connect(path(3), left, spot(3)).
connect(path(3), right, spot(4)).
connect(path(2), left, spot(1)).
connect(path(2), right, spot(2)).
connect(path(-2), forward, path(-1)).
connect(path(-1), forward, path(0)).
```

```
connect(path(0), forward, path(1)).

connect(path(1), forward, path(2)).

connect(path(2), forward, path(3)).

connect(path(3), forward, path(4)).

connect(path(5), backward, path(4)).

connect(path(4), backward, path(3)).

connect(path(-1), backward, path(-2)).

connect(path(0), backward, path(-1)).

connect(path(1), backward, path(0)).

connect(path(2), backward, path(1)).

connect(path(3), backward, path(2)).



/* solved if all three cars are parked */

solve(C1,C2,C3) :- in(C1, spot(_)), in(C2, spot(_)), in(C3, spot(_)).

/* otherwise, try to make some moves */

solve(C1,C2,C3) :- tick, move(C1), move(C2), move(C3), nl, solve(C1,C2,C3).


process/2.

/*change process cruising-> parked if moving into a spot*/

processmachine(C,P2):-
P2=spot(_),retract(process(C,cruising)),assert(process(C,parked)),write(C),write("
cruising->parked"),nl.

/*no change cruising if moving into an aisle*/

processmachine(C,P2):-P2\=spot(_),write(C),write(" cruising"),nl.


/*time step*/

tick:-write("next step"),nl.
```

```
/*randomly move*/
next_move(C,A,X):-in(C,Y),connect(Y,A,X),not(in(_, X)).
random_move(C):-findall(X,next_move(C,_,X),LST),random_member(Y, LST),
    processmachine(C,Y),
    write(C), write(' moved '), write(' -> '),
    retract(in(C, _)), assert(in(C, Y)),
    reportall, nl.

go :- retractall(in(_,_)),
     /* set up initial car position */
      assert(in(car(1), path(4))),
      assert(in(car(2), path(-1))),
      assert(in(car(3), path(0))),
      assert(process(car(1),cruising)),
      assert(process(car(2),cruising)),
      assert(process(car(3),cruising)),
      write('current cars are at: '),
      reportall, nl, nl,
      solve(car(1), car(2), car(3)).
```

# APPENDIX C Transaction sample data

Payment Type Server Time Terminal Date Meter Code Amount Total Duration Paid Duration Total Duration in mins Paid Duration in mins System ID Printed ID Space # Plate # Card Type Card #: Zone Desc Circuit Desc Park Code Park Meter Description Address Type User Type: End Date Free Duration: Free Duration in mins Currency

Credit Card 2017/4/30 12:21 2017/4/30 12:20 13200030 2 14 h 39 m 14 h 39 m 879 879 181356463 109546 474ZZJ BANK_ONLINE_EPSUM EMS Elevator Lobby 132 Unive of wisconsin *13200030 (Neops) Parking 1 2017/5/1 3:00 - 0 USD

Credit Card 2017/4/30 11:05 2017/4/30 11:04 13200030 2 15 h 55 m 15 h 55 m 955 955 181352475 109545 990VYZ BANK_ONLINE_EPSUM EMS Elevator Lobby 132 Unive of wisconsin *13200030 (Neops) Parking 1 2017/5/1 3:00 - 0 USD

Credit Card 2017/4/30 10:05 2017/4/30 10:05 13200031 2 16 h 54 m 16 h 54 m 1014 1014 181350831 61100 462DHP BANK_ONLINE_EPSUM EMS SE corner - LL 132 Unive of wisconsin *13200031 (Neops) Parking 1 2017/5/1 3:00 - 0 USD

Credit Card 2017/4/30 10:00 2017/4/30 9:59 13200031 2 17 h 17 h 1020 1020 181350706 61099 391TGP BANK_ONLINE_EPSUM EMS SE corner - LL 132 Unive of wisconsin *13200031 (Neops) Parking 1 2017/5/1 3:00 - 0 USD

Credit Card 2017/4/30 9:43 2017/4/30 9:43 13200031 2 17 h 16 m 17 h 16 m 1036 1036 181350358 61098 114TNJ BANK_ONLINE_EPSUM EMS SE corner - LL 132 Unive of wisconsin *13200031 (Neops) Parking 1 2017/5/1 3:00 - 0 USD

Credit Card 2017/4/30 6:01 2017/4/30 6:00 13200031 2 20 h 59 m 20 h 59 m 1259 1259 181346245 61097 954PEL BANK_ONLINE_EPSUM EMS SE corner - LL 132 Unive of wisconsin *13200031 (Neops) Parking 1 2017/5/1 3:00 - 0 USD

Credit Card 2017/4/29 15:36 2017/4/29 15:35 13200030 5 11 h 24 m 11 h 24 m 684 684 181324044 109544 793XAL BANK_ONLINE_EPSUM EMS Elevator Lobby 132 Unive of

wisconsin *13200030 (Neops) Parking 1 2017/4/30 3:00 -
0 USD

Credit Card 2017/4/29 14:44 2017/4/29 14:44 13200030 2.25 1 h 30 m
1 h 30 m 90 90 181322168 109543 136LVY
BANK_ONLINE_EPSUM EMS Elevator Lobby 132 Unive of
wisconsin *13200030 (Neops) Parking 1 2017/4/29 16:14 -
0 USD

Credit Card 2017/4/29 10:58 2017/4/29 10:57 13200031 0.5 18 h 56 m
18 h 56 m 1136 1136 181311954 61096 391TGP
BANK_ONLINE_EPSUM EMS SE corner - LL 132 Unive of
wisconsin *13200031 (Neops) Parking 1 2017/4/30 3:00 -
0 USD

Bills 2017/4/29 10:20 2017/4/29 10:20 13200031 2 1 h 20 m 1 h 20
m 80 80 181310818 61095 462DHP EMS SE
corner - LL 132 Unive of wisconsin *13200031 (Neops) Parking 1
2017/4/29 11:40 - 0 USD

Bills 2017/4/29 9:50 2017/4/29 9:39 13200030 1 40 m 40 m 40
40 181309952 109542 494BDB EMS
Elevator Lobby 132 Unive of wisconsin *13200030 (Neops)
Parking 1 2017/4/29 10:19 - 0 USD

## APPENDIX D Field survey worksheet

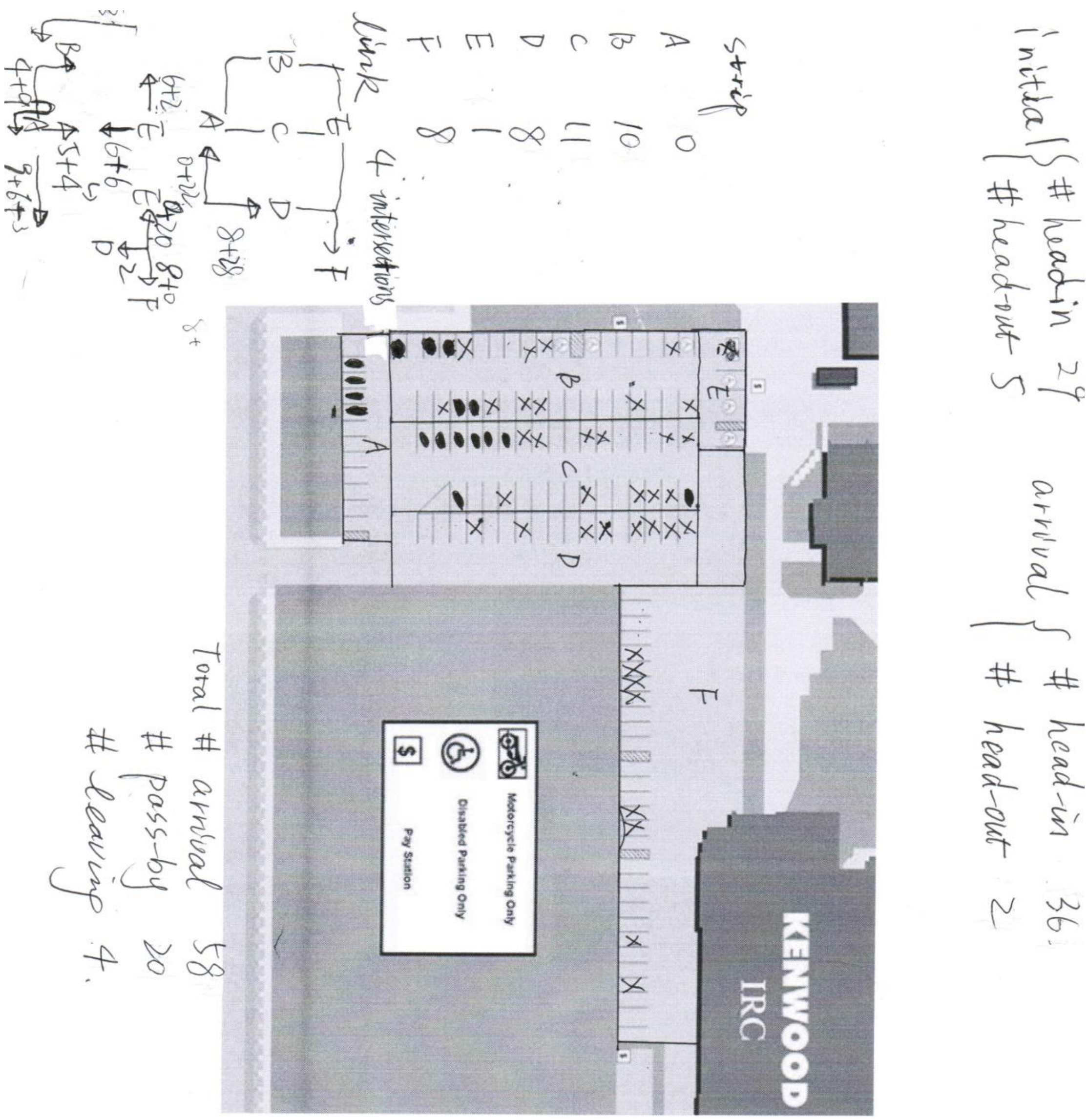



Figure A-2 A sample worksheet for the parking lot survey

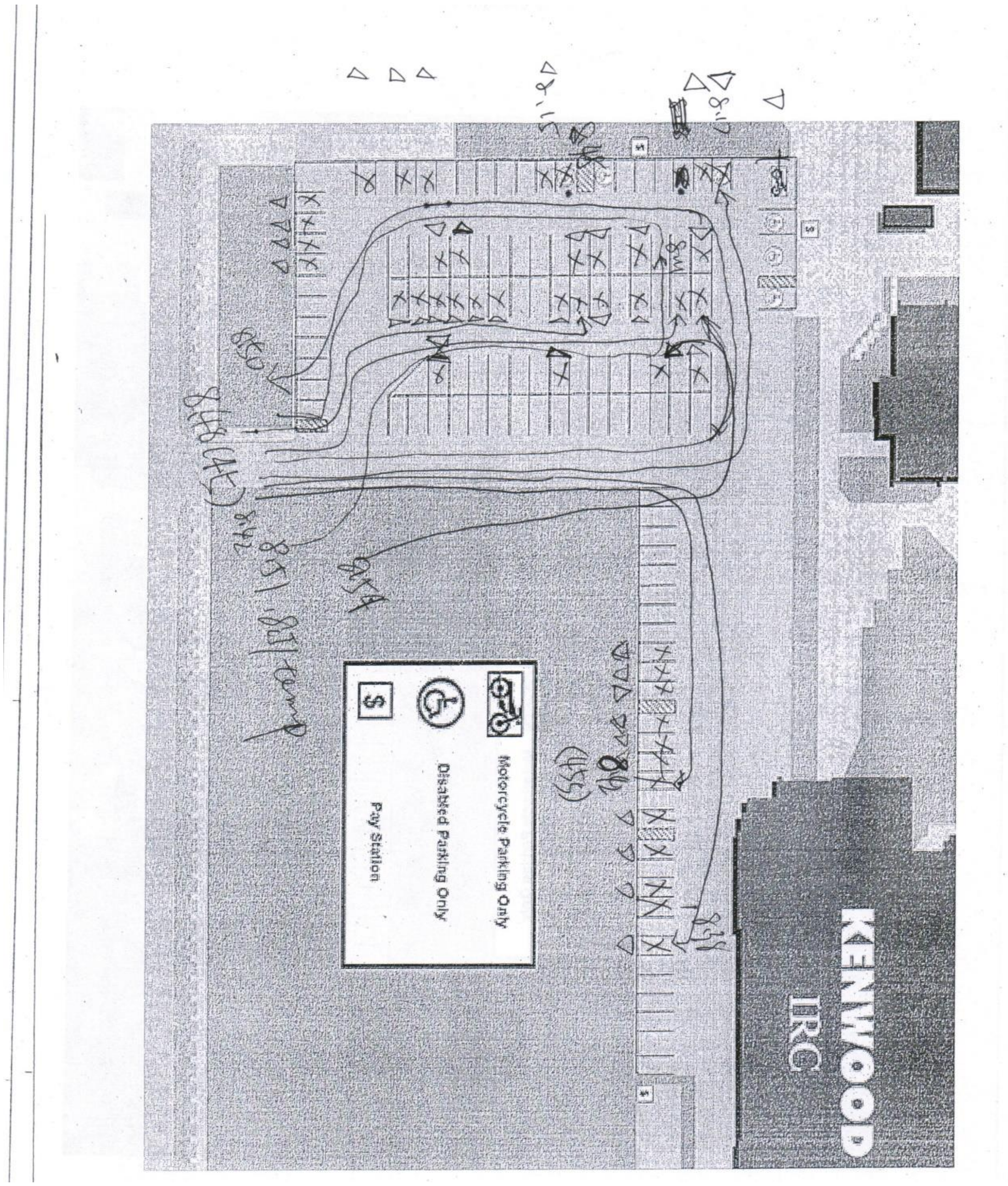



Figure A-3 A sample worksheet for the parking arrival and routing survey

The inter-arrival and parking duration file are in the ASC-II/TXT format which includes each line represents a value of parking duration in seconds.

An inter-arrival file example is shown as follows:

Table A-1 An inter-arrival data example

| Data entry |
| --- |

| |
|---|
| 8 |
| 13 |
| 3 |
| 293 |
| 3 |
| 3 |
| 54 |
| 50 |
| 36 |
| 56 |

A parking duration file example is shown as follows.

Table A-2 A parking duration example

| Data entry |
|---|
| 20 |
| 291 |
| 1440 |
| 448 |
| 487 |
| 60 |
| 327 |
| 1440 |
| 60 |
| 376 |

A choice behavior file example is shown as follows.

Table A-3 A sample of the spot choice behavior data

| walkingdistance | traveldistance | lanestatus | spotstatus | 'Class' |
|---|---|---|---|---|
| 10.5 | 35 | UNOCCUPIED | CLEAR | 1 |
| 21.5 | 22 | OCCUPIED | CLEAR | 0 |
| 35 | 12 | UNOCCUPIED | CLEAR | 1 |
| 10 | 56 | OCCUPIED | CLEAR | 0 |
| 35 | 67 | OCCUPIED | LEFT | 0 |
| 17 | 37 | OCCUPIED | RIGHT | 0 |
| 23.5 | 20.5 | OCCUPIED | BOTH | 0 |
| 14.2 | 10 | UNOCCUPIED | CLEAR | 1 |
| 20 | 20 | OCCUPIED | CLEAR | 0 |

# APPENDIX E Collected Vehicle Arrival Data from Video

Table A-4 The interarrival data from the field survey

| Index | Arrival Time (min:seconds) | Inter-arrival (seconds) |
|---|---|---|
| 1 | 00:33 | |
| 2 | 00:41 | 8 |
| 3 | 00:54 | 13 |
| 4 | 00:57 | 3 |
| 5 | 05:50 | 293 |
| 6 | 05:53 | 3 |
| 7 | 05:56 | 3 |
| 8 | 06:50 | 54 |
| 9 | 07:40 | 50 |
| 10 | 08:16 | 36 |
| 11 | 09:12 | 56 |
| 12 | 11:46 | 154 |
| 13 | 12:42 | 56 |
| 14 | 13:24 | 42 |
| 15 | 13:52 | 28 |
| 16 | 14:17 | 25 |
| 17 | 14:25 | 8 |
| 18 | 14:40 | 15 |
| 19 | 18:01 | 201 |
| 20 | 19:35 | 94 |
| 21 | 19:55 | 20 |
| 22 | 20:26 | 31 |
| 23 | 20:45 | 19 |
| 24 | 20:52 | 7 |
| 25 | 21:24 | 32 |
| 26 | 22:12 | 48 |
| 27 | 22:18 | 6 |
| 28 | 22:21 | 3 |
| 29 | 22:43 | 22 |
| 30 | 23:16 | 33 |
| 31 | 25:38 | 142 |
| 32 | 25:44 | 6 |
| 33 | 26:44 | 60 |
| 34 | 27:29 | 45 |
| 35 | 28:50 | 81 |
| 36 | 29:16 | 26 |
| 37 | 29:49 | 33 |
| 38 | 29:54 | 5 |
| 39 | 32:45 | 171 |
| 40 | 34:14 | 89 |
| 41 | 34:47 | 33 |
| 42 | 35:46 | 59 |
| 43 | 36:25 | 39 |
| 44 | 37:12 | 47 |

| | | |
|---|---|---|
| 45 | 38:21 | 69 |
| 46 | 39:00 | 39 |
| 47 | 39:51 | 51 |
| 48 | 40:32 | 41 |
| 49 | 41:00 | 28 |
| 50 | 41:24 | 24 |
| 51 | 43:47 | 143 |
| 52 | 44:38 | 51 |
| 53 | 44:50 | 12 |
| 54 | 45:00 | 10 |
| 55 | 45:08 | 8 |
| 56 | 45:20 | 12 |
| 57 | 45:49 | 29 |
| 58 | 45:55 | 6 |
| 59 | 47:08 | 73 |
| 60 | 47:13 | 5 |
| 61 | 48:30 | 77 |
| 62 | 50:45 | 135 |
| 63 | 50:53 | 8 |
| 64 | 51:48 | 55 |
| 65 | 51:57 | 9 |
| 66 | 52:09 | 12 |
| 67 | 53:09 | 60 |
| 68 | 53:59 | 50 |
| 69 | 54:11 | 12 |
| 70 | 54:26 | 15 |

## APPENDIX F A Sample code of LSTM predictor training and testing

```
# -*- coding: utf-8 -*-
"""
Created on Fri Aug 4 00:11:19 2017

@author: Yun Yuan
"""


# input
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import scipy as sp
import timeit
import time
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.callbacks import EarlyStopping
from sklearn.preprocessing import MinMaxScaler



def input_data():
    # read data from csv file
    df = pd.read_csv('LTR.csv')
    df.apply(lambda x: pd.to_numeric(x, errors='ignore'))
    df[['Entr_Time', 'Exit_Time']] = df[['Entr_Time', 'Exit_Time']].apply(pd.to_datetime)
    df_entr = df[['Rate', 'Entr_Time']].set_index('Entr_Time')
    df_exit = df[['Rate', 'Exit_Time']].set_index('Exit_Time')
```

```
    df2=pd.read_csv('LPA.csv')
    df2[['Date and Time']] = df2[['Date and Time']].apply(pd.to_datetime)
    df2_entr = df2[['Date and Time', 'Lot']][(df2.Direction == 'In') & (df2.Allowed == 'Yes')]
    df2_exit = df2[['Date and Time', 'Lot']][(df2.Direction == 'Out') & (df2.Allowed == 'Yes')]
    df2_entr = df2_entr[['Date and Time', 'Lot']].set_index('Date and Time')
    df2_exit = df2_exit[['Date and Time', 'Lot']].set_index('Date and Time')

    # aggregating to intervals
    intervals = '15'
    entr_count = np.array(df_entr.resample(intervals + 'T').count(), dtype=int)
    exit_count = np.array(df_exit.resample(intervals + 'T').count(), dtype=int)
    entr_count2 = np.array(df2_entr.resample(intervals + 'T').count(), dtype=int)
    exit_count2 = np.array(df2_exit.resample(intervals + 'T').count(), dtype=int)

    # preprocessing
    length = np.min((entr_count.shape[0], exit_count.shape[0], entr_count2.shape[0],
exit_count2.shape[0]))
    dataX = np.hstack((entr_count[len(entr_count) - length:len(entr_count)],
                   exit_count[len(exit_count) - length:len(exit_count)],
                   entr_count2[len(entr_count2) - length:len(entr_count2)],
                   exit_count2[len(exit_count2) - length:len(exit_count2)]))
    dataY = np.array(np.vstack((dataX[1:len(dataX)], np.zeros(4))), dtype=float)
    return dataX, dataY



# create LSTM model
def createModel(shape1, shape2, shape3):
    model = Sequential()
    # model.add(Embedding(input_dim=3,output_dim=3))
```

```
    model.add(LSTM(256, input_shape=(shape1, shape2), return_sequences=True,
activation='softsign'))
    model.add(Bidirectional(LSTM(256, return_sequences=True, activation='softsign')))
    model.add(Bidirectional(LSTM(256, return_sequences=True, activation='softsign')))
    #    model.add(LSTM(256,return_sequences=True, activation='softsign'))
    model.add(Bidirectional(LSTM(256, activation='softsign')))
    model.add(Dense(shape3, activation='softsign'))
    # model compiling
    model.compile(loss='mse', optimizer='adam', metrics=['acc'])
    return model


# error measures
def rmse(y_test, y):
    return sp.sqrt(sp.mean((y_test - y) * (y_test - y)))


def R2(y_test, y_true):
    return 1 - ((y_test - y_true) * (y_test - y_true)).sum() / (
            (y_true - y_true.mean()) * (y_true - y_true.mean())).sum()


def R22(y_test, y_true):
    y_mean = np.array(y_true)
    y_mean[:] = y_mean.mean()
    return 1 - rmse(y_test, y_true) / rmse(y_mean, y_true)


# data set
```

```
def create_dataset(X, Y, loop_back=3):
    dataX, dataY = [], []
    for i in range(len(X) - loop_back):
        dataX.append(X[i:(i + loop_back)])
        dataY.append(Y[i + loop_back])
    return np.array(dataX), np.array(dataY)



# training LSTM
def training(train_dataX, train_dataY):
    # normalize the dataset
    scaler = MinMaxScaler(feature_range=(0, 1))
    train_dataX = scaler.fit_transform(train_dataX)
    train_dataY = scaler.fit_transform(train_dataY)

    # create training data with lookback
    trainX, trainY = create_dataset(train_dataX, train_dataY)

    # model definition
    model = createModel(trainX.shape[1], trainX.shape[2], train_dataY.shape[1])

    # early stopping
    early_stopping = EarlyStopping(monitor='loss', patience=3)

    # model training
    history = model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2,
callbacks=[early_stopping])
    evals = model.evaluate(trainX, trainY)
    print('loss:' + str(evals[0]) + '\n' + 'acc:' + str(evals[1]))
```

```
    # make predictions
    trainPredict = model.predict(trainX)
    print('train rmse:' + str(rmse(trainPredict, trainY)))
    print('train R2:' + str(R2(trainPredict, trainY)))
    print('train R22:' + str(R22(trainPredict, trainY)))
    trainY = scaler.inverse_transform(trainY)
    trainPredict = scaler.inverse_transform(trainPredict)
    # plotting
    for i in range(train_dataY.shape[1]):
        plt.figure(figsize=(8, 4))
        plt.plot(trainY[:, i], label='trainY')
        plt.plot(trainPredict[:, i], label='predicted trainY')
        plt.title(i)
        plt.legend(bbox_to_anchor=(1, 1))
        plt.savefig(time.strftime('%Y-%m-%d %H%M%S') + ' 4-11_train ' + str(i), dpi=90)
        plt.show()
    return model


# testing LSTM
def testing(model, test_dataX, test_dataY):
    scaler = MinMaxScaler(feature_range=(0, 1))
    test_dataX = scaler.fit_transform(test_dataX)
    test_dataY = scaler.fit_transform(test_dataY)
    testX, testY = create_dataset(test_dataX, test_dataY)
    testPredict = model.predict(testX)
    print('test R2:' + str(R2(testPredict, testY)))
```

```
    print('test R22:' + str(R22(testPredict, testY)))
    testY = scaler.inverse_transform(testY)
    testPredict = scaler.inverse_transform(testPredict)
    for i in range(test_dataY.shape[1]):
        plt.figure(figsize=(8, 4))
        plt.plot(testY[:, i], label='testY')
        plt.plot(testPredict[:, i], label='predicted testY')
        plt.title(i)
        plt.legend(bbox_to_anchor=(1, 1))
        plt.savefig(time.strftime('%Y-%m-%dT%H%M%S') + ' 4-11_test ' + str(i), dpi=90)
        plt.show()


# main routine
def main():
    start = timeit.default_timer()

    # input a csv file
    dataX, dataY = input_data('LTR.csv')

    # split data for cross validation
    train_size = np.int(np.round(len(dataX) * 0.7))

    # training
    train_dataX = dataX[0:train_size]
    train_dataY = dataY[0:train_size]
    model = training(train_dataX, train_dataY)
```

```
    # testing
    test_dataX = dataX[train_size:]
    test_dataY = dataY[train_size:]
    testing(model, test_dataX, test_dataY)

    # save model to file
    model.save("model.json")


if __name__ == '__main__':
    main()
```

# CURRICULUM VITAE

Yun Yuan

Place of birth: Weifang, China

Education

B.S., Nanjing Agricultural University, P.R.C, June 2012

Major: Transportation Engineering

M.S., Nanjing Agricultural University, P.R.C, June 2015

Major: Vehicle Engineering

Dissertation Title: A Microscopic Simulation Laboratory for Evaluation of Off-Street Parking Systems

Publications

Yuan, Y., Liu, Y. Dynamic lane-based signal merge control for freeway work zone operations. Journal of Transportation Engineering, in press, 2018.

Yuan, Y., Liu, Y. Traffic management in a dynamic evacuation network: trade-off between signal and cross-elimination strategies. Transportation Research Part C: Emerging Technologies, in press, 2018.

Yuan, Y., Yu, J., Han, D, Tan, W, Jin, Z. Strategic planning of evacuation in urban network with balanced signal control and cross elimination strategies. Journal of Transportation Engineering, Part A: Systems, in press, 2018.

Yuan, Y., Yu, J. Locating transit hubs in a multi-modal transportation network: A cluster-based optimization approach. Transportation Research Part E: Logistics and Review, 2018. 113(1), 85-103.

Li, X., Wei, M., Hu, J., Yuan, Y., Jiang, H. An agent-based model for dispatching real-time demand-responsive feeder bus. Mathematical Problems in Engineering, 2018, in press.

Zhao, J, Yu, J., Xia, X, Ye, J., Yuan, Y. Exclusive bus lane network design: a perspective from intersection operational dynamics. Networks and Spatial Economics, 2018, in press.

Li, X., Shaw, J. W., Liu, D., Yuan, Y. Acceptability of Beijing congestion charging from a business perspective. Transportation, 2017, 1(1), 1-24.