IEEE Access Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SUSTAINABLE INFRASTRUCTURES, PROTOCOLS, AND RESEARCH CHALLENGES FOR FOG COMPUTING

Received October 12, 2018, accepted November 4, 2018, date of publication November 12, 2018, date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2880972

Towards Smart Parking Based on Fog Computing

CHAOGANG TANG[1], XIANGLIN WEI[2], CHUNSHENG ZHU[3], WEI CHEN[1], AND JOEL J. P. C. RODRIGUES[4,5,6], (Senior Member, IEEE)

[1]School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221000, China
[2]Nanjing Telecommunication Technology Research Institute, Nanjing 210007, China
[3]Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[4]National Institute of Telecommunications (Inatel), Santa Rita do Sapucaí 37540-000, Brazil
[5]Instituto de Telecomunicações, 1049-001 Lisbon, Portugal
[6]University of Fortaleza, Fortaleza 60811-905, Brazil

Corresponding author: Chunsheng Zhu (chunsheng.tom.zhu@gmail.com)

This work was supported in part by the Jiangsu Province Natural Science Foundation of China under Grant BK20150201, in part by the National Natural Science Foundation of China under Grant 51874300, in part by the National Natural Science Foundation of China and Shanxi Provincial People's Government Jointly Funded Project of China for Coal Base and Low Carbon under Grant U1510115, in part by the National Natural Science Foundation of China under Grant 51874299 and Grant 51104157, in part by the Qing Lan Project, the China Postdoctoral Science Foundation, under Grant 2013T60574, in part by the National Funding from the FCT—Fundação para a Ciência e a Tecnologia under Project UID/EEA/500008/2013, in part by Finep, with resources from Funttel under the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações—CRR) project of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações—Inatel), Brazil, under Grant 01.14.0231.00, and in part by the Brazilian National Council for Research and Development (CNPq) under Grant 309335/2017-5.

ABSTRACT An experience of finding a vacant parking slot can be very stressful in densely populated areas, especially in peak hours. Such parking process takes a long time, wastes significant gasoline, and emits extra vehicle exhaust that harms the environment. Smart parking, aiming to assist drivers in finding desirable parking slots more efficiently through information and communication technologies such as vehicle ad hoc networks (VANETs), has received extensive attention recently. Current VANETs-based parking slot allocations cannot provide a fully satisfactory solution, because vehicle communication devices—on-board units—and roadside units lack computational capabilities to perform humanized and accurate service provisioning, such as real-time parking slots information and probabilistic prediction on future parking slots. Therefore, we, in this paper, propose a fog computing-based smart parking architecture to improve smart parking in real time. Fog nodes deployed at parking lots, cooperating with each other, enable real-time parking slot information provisioning as well as parking requests processing. The cloud center can further enhance smart parking capability by enforcing global optimization on parking requests allocation. The experimental results of our approaches show higher efficiency compared with other parking strategies. The proposed fog computing-based smart parking can lower the average parking cost and minimize gasoline wastes and vehicle exhaust emission.

INDEX TERMS Parking slot, smart, VANETs, fog computing, architecture, real time.

I. INTRODUCTION

Parking problems have attracted more and more attention in the past few years, as the number of motor vehicles is explosively increasing. An experience of finding a vacant parking slot can be very stressful in densely populated area (e.g., Beijing). Often, drivers need to keep circling around the underground parking lot, or wait at the entrance to the ground parking lot, until a slot is available. The time consuming parking process results in a series of issues such as severe gasoline wastes and vehicle exhaust emissions that harms the environment.

Several countermeasures have been taken to overcome the parking problems. Smart parking systems, aiming to assist in finding vacant parking slots when vehicles enter the parking lot, have been developed and applied in many parking lots. For instance, they usually provide vehicles with real-time parking slots information and direction signs for vacant parking slots. To this end, large quantities of sensors are required for data collecting and accurate service provisioning, which, however, requires enormous expenditures. Nevertheless, few of these parking systems provide services for outside vehicles which are looking for parking spaces along the road. For instance, when there is no parking slot available and numbers of vehicles are waiting for parking outside, the parking lot does not provide efficient mechanism to notify the approaching vehicles in vicinity of the parking slot status. As a result, more and more vehicles are jammed at the parking lot, causing great inconvenience to drivers and severe gasoline wastes. However, from the economic viewpoint, the owners of the parking lots have little incentive to improve parking lot efficiency; they would prefer keeping vehicles waiting in queue rather than having the parking slots idle.

In this context, Vehicle Ad Hoc Networks (VANETs), which could interconnect vehicles and roadside units (RSUs) through IEEE 802.11p, are considered as a suitable choice for smart parking [1]. VANETs, usually consisting of RSUs and vehicles mounted communication devices, e.g., the on-board units (OBUs), can enable efficient information collecting and sharing among vehicles. On one hand, vehicles in VANETs can receive the parking lot information from other vehicles and RSUs; On the other hand, vehicles need to act as relay nodes disseminating the information to other vehicles.

However, a few challenges still exist in VANET-based smart parking systems. First, vehicles lack incentives to share parking information with other partners, especially for those with urgent parking demands. Secondly, energy consumption caused by serving as relay nodes is also a big concern when parking slot information is continuously disseminated among vehicles. Thirdly, ad hoc style information collecting and sharing cannot support timely information dissemination and update, which is very critical for smart parking systems. Last but not least, RSUs, responsible for caching and relaying the parking slots information to the vehicles nearby, are unlikely to achieve ubiquitous coverage due to costly installation, deployment and maintenance. As an immobile infrastructure located at a traffic dense area such as road side, intersections, and near parking lots for information dissemination, current RSUs do not have powerful computational capabilities. Thus, for humanized and accurate service provisioning, e.g., real-time parking slots information and probabilistic prediction on future parking slots, VANETs-based parking slot allocation strategies cannot provide a fully satisfactory solution.

The fog computing paradigm, also known as the edge computing and considered as one of key enablers of IoT and big data applications [2], brings computation and storage resources to the edge of network, enabling it to run the highly demanding applications while meeting strict latency requirements. Therefore, to overcome the aforementioned shortcomings of VANET-based smart parking systems, we propose a fog computing based smart parking strategy in this paper. Specifically, a few fog nodes with computation and storage resources are deployed near parking lots to collect and share information andhelp vehicles make parking decisions. Moreover, different fog nodes can collaborate with each other to derive a comprehensive parking information of the area to facilitate collaborative decision making. Besides, a remote cloud center with much more powerful processing capability can further optimize the efficiency of the smart parking system. Via fog computing, the traffic data and parking slots information can be processed at the edge of network in a real time fashion to alleviate the parking problems and even reduce vehicle exhaust emissions and environmental pollution.

The contribution of this paper is threefold, as follows:

1. We propose a parking slot allocation strategy and a four-layer architecture, applying fog computing into VANETs to provision real-time parking slot information. With fog computing, vehicles are receiving nodes, no longer responsible for information dissemination; information dissemination and multi-hop communication are accomplished by fog nodes.
2. The proposed strategy considers the comprehensive factors which can affect the decision marking, including walking costs, driving costs, waiting costs and more. Meanwhile, it gives drives options to choose their own preferences– some drivers may follow their own parking preferences while ignoring the suggestions from fog computing and cloud computing.
3. Extensive simulations are conducted to evaluate the efficiency and effectiveness of fog computing based smart parking strategies. The experimental results have proven that the smart parking strategies are outstanding compared to other parking strategies. To great extent, the fog computing based smart parking can efficiently reduce the parking costs for each vehicle with parking needs, and further reduce the gasoline wastes and vehicle exhaust emissions.

The remainder of the paper is organized as follows. Section II surveys some related works. Section III presents the application scenario and system architecture which combines fog computing and VANETs to provide customized and humanized parking services. Section IV formulates the parking problems based on fog computing. Section V proposes an efficient parking slot allocation scheme, followed by the experimental evaluation and analysis in Section VI. Finally, Section VII provides conclusions and directions of the future work.

II. RELATED WORKS

Smart parking aims to assist drivers in finding desirable parking slots more efficiently via ICTs, IoT, mobile internet, cloud computing and so on. Currently, there are several efforts which survey and investigate the smart parking solutions [3]–[8]. As a subarea of smart city [9]–[14], smart parking and smart transportation have also received extensive attention recently. Lin *et al.* [3] have presented a smart parking ecosystem and classified the current works by different functionalities and problematic focuses. For instance, they proposed three macro-themes based on the different parking solutions, i.e., information collection, system deployment, and service dissemination. VANETs enable the dissemination of parking lot information among vehicles. Vehicles with parking demands may choose to compete for only one parking space, resulting in an unsatisfying solution in the smart parking system. Delot *et al.* [7] have proposed a reservation protocol which can allocate parking spaces in VANETs and avoid the competition among vehicles. This approach forwards the parking spaces to specified vehicles to avoid the competition. However, how to select the vehicles to send the parking lot status is very crucial, this is because a fair parking space allocation should take into account several factors such as the parking time, the distance between vehicles and parking lot, and more. Faheem *et al.* [8] have introduced an agent into smart parking systems, which is responsible for collecting dynamic and complex traffic data. They also specified how to interact between vehicles and parking systems. A smart parking system named SPARK based on VANET is presented by Lu *et al.* [1] for large parking lots.

The data collected by IoT devices and sensors have increased explosively, which leads to response latency when parking requests are processed. Thus, the vehicle may miss a more appropriate parking lot when receiving the parking lot status. To support the computational demands and reduce the response time, fog computing has been introduced to smart transportation [2], [15]–[17]. The fog computing provides computing, networking and storage so that the advantages of cloud computing can be extended to the edge of networking, which is much closer to the IoT sensors.

To cope with the increasing demands for both communication and computation from vehicular applications, Hou *et al.* [18] conceived the idea of vehicular fog computing (VFC) for communication and computation. They utilize moving and parked vehicles as the infrastructure which leverages a collaborative multitude to perform communication and computation. They study the relationships among communication ability, connectivity, and mobility of vehicles. Xiao and Zhu [19] propose similar concept of vehicular fog computing. They turn the connected vehicles into mobile fog nodes and offer cost-effective and on-demand fog computing for vehicular applications.

To maintain the reliability of the real-time streaming and adapt to the change of mobile device behaviors and the computing resources, Huang and Xu [20] have illustrated a distributed scheme for real-time streaming via vehicular cloud-fog networks. They allocated the streaming from fogs and clouds in advance, and thus the time can be reduced. To relieve traffic congestion, reduce air pollution and improve driving comfort level, Kim et al. have considered the parking problem from the viewpoint of IoT [21], and have adopted fog computing and roadside to find vacant parking spaces. Matching theory is applied to solving the parking problem. A navigation and reservation based parking proposal system is developed in [22], with the purpose of relieve the parking problems. The involved method is to use the IoT technology to send data. They use genetic algorithms to find vacant parking spaces which is near to the current location.

Some researchers are dedicated to optimizing the parking space searching. For instance, Song and Mou [26] modeled the process of parking space searching as a game from the perspective of game theory and try to find a stable strategy. This approach however is of high computational complexity. To improve location accuracy in indoor parking, Balzano and Vitale [15] have proposed an alternative localization technique, which combines wireless radio signal strength-to-distance evaluation, vehicle communications, and DGP algorithm to find vacant parking spaces.

To process the explosively increasing data in the connected vehicles, Zhang *et al.* [27] leverage the notion of fog computing to ensure the quality of services such as the time latency, and they present a hierarchical model with intra-fog and inter-fog resource management. Besides some metrics such as energy efficiency and packet dropping rates are optimized. Combine fog computing and SDN in the connected vehicles, Park and Yoo [28] try to reduce control message overhead by adjusting the period of beacon messages and to support efficient failure recovery, and they present a real-time scheduling algorithm to recover the services.

The main difference between our proposal in this paper and the aforementioned researches is that we aim at providing humanized and accurate parking services with the help of fog computing, such as real-time parking information dissemination and parking slots predictions. A cloud center is adopted to achieve global optimization for parking request allocation. We also underline the cooperation among fogs to reduce the waiting time for vehicles in the parking lot.

FIGURE 1. An example of parking space prompts in smart parking lots.

III. SMART PARKING SCENARIO AND SYSTEM ARCHITECUTURE

The advance of Internet of Things (IoT) has enabled much richer means of event monitoring and information collecting by employing various types of sensors in specified monitoring area (e.g., signalized intersections, parking lots, and malls) [23]. Accordingly, we assume that the real-time parking information can be obtained in this paper. Actually, most smart parking lots have succeeded in providing accurate and real-time slot occupation information to drivers. Moreover, some smart devices are installed along the road to display the current parking spaces information. For instance, Figure 1 depicts an example of parking space prompts in smart lots to remind drivers with parking demand, where the location of parking lots and the number of vacant parking spaces are provided.

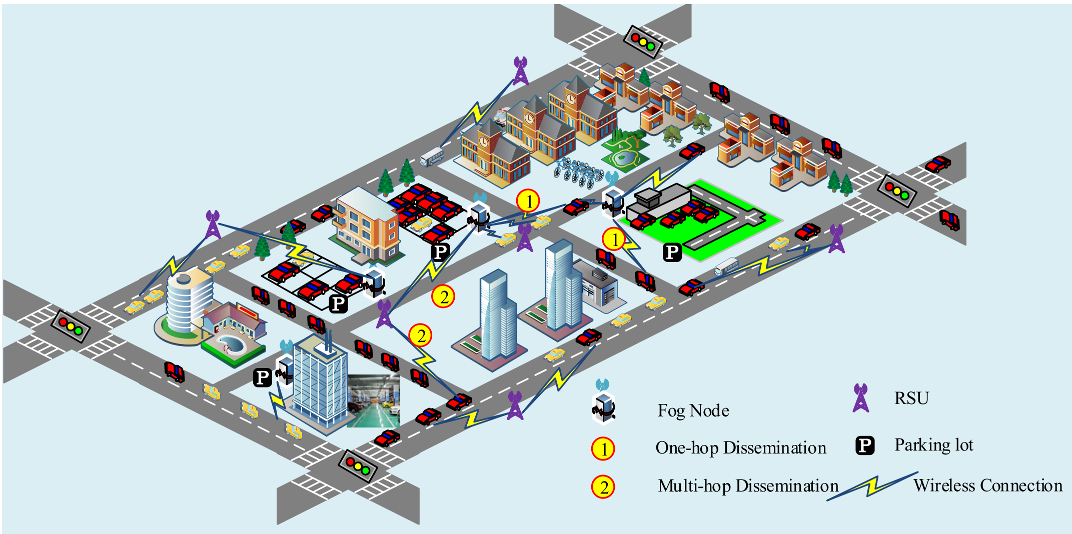


FIGURE 2. The application of fog computing in parking slots allocation.

However, to obtain these kinds of guide signs are not straightforward for the vehicles/drivers since there are no interactions between vehicles and parking lots. Further, checking guide signs may cause distracted driving behaviors and lead to potential traffic accidents. Besides, a situation may occur that multiple drivers may try their luck at the same parking lot for only one parking space due to the delay of the guide sign updates.

To tackle the parking problem, we face two main challenges here. The first one is how to enable the communications between vehicles and parking lots. The second one is how to determine the optimal or near-optimal parking lot for each individual vehicle for some particular time. In other words, who can provide the computation capability for the decision-making process. We can answer the first question from the existing solution, i.e. VANET. As for the second one, we think that a combination of fog computing and cloud computing may provide the solution [24]. This is due to the following two reasons. On one hand, a few fog nodes, which are deployed at different parking lots and intersections, could collect and disseminate both traffic and parking information and make decisions in a timely manner. On the other hand, a cloud data center with powerful processing capability may gather a global view of the parking area and provide some kind of global optimization for the decision making process.

Figure 2 depicts the smart parking scenario, where the fog computing is applied to parking slots allocation. Each parking lot is equipped with a fog node to achieve near real-time information collection, processing and decision-making. RSUs are placed along the traffic dense roads, and they connect to the fog nodes through wired network with low latency. In addition, vehicles and RSUs can communicate with each other as long as they are within the communication range. To enhance request processing abilities and seek global optimization on parking slot allocation, a remote cloud center is introduced to process the parking requests uploaded from the fog nodes in specified area.

A one-hop communication between fog nodes and RSUs as well as between fog nodes and vehicles is enabled, i.e., the traffic information or packets can be directly transmitted between them without the help of relay nodes. For vehicles with parking demand but beyond one-hop communication with fog nodes, RSUs can act as relay nodes to disseminate the parking slot information to vehicles in vicinity, and thereby multi-hop communications between vehicles and fog nodes are formed.

Enormous amount of parking lot information, including the number of vacant parking slots, the number of waiting vehicles at the entrance and the approaching vehicles from outside, can be collected and further periodically disseminated to vehicles with parking demand in real time. To avoid the situation that one parking lot has numbers of vehicles waiting at the entrance while another not far away has lots of vacant parking slots, fog nodes can collaborate with each other to share the parking space information. If no parking slots are available in all parking lots, fog nodes and the remote cloud can decide a potential optimal parking slots allocation scheme based on some evaluation metrics. As a consequence, the parking lots can make timely response to alleviate the parking problems e.g., by vacant slots dissemination and prediction, and further reduce driving time, gasoline wastes and vehicle exhaust emissions.

Considering the needs for the real-time decision making, fog computing is introduced as an intermediate layer in between the cloud data center and IoT devices to facilitate data collection, processing and analysis. Therefore, an architecture of fog computing based smart parking is then proposed in this paper, which consists of four layers, i.e., cyber physical layer, data management layer, data processing layer and application layer, respectively.

A. CYBER PHYSICAL LAYER

The cyber physical layer is composed of a densely distributed ecosystem which covers various sensors. This layer is in charge for collecting various types of data from both vehicles and parking lots. For instance, the vehicle related sensing data such as velocity, acceleration, vehicle heading can be collected by vehicle mounted sensor devices (e.g., accelerator, magnetometer, and inductive loop), while the data on parking lots including the parking spaces occupancy, the waiting vehicles can be collected by surveillance cameras Radio-frequency identification (RFID) tags, and so on. The core technology in the cyber physical layer is IoT, which enables direct interactions among various entities (e.g., sensors, routers, gateways).

B. DATA MANAGEMENT LAYER

To make data processing more efficient in the next stages, we add an intermediate layer called data management layer between the cyber physical layer and the data processing layer. Data management layer is responsible for data preprocessing such as data description, data fusion. The same event can be captured by sensors of different types while data from different sensors usually have a variety of representation. Thus, redundancy of data may exist when storing them. To manage data efficiently, it is necessary to process the sensed data in advance, such as redundancy removing. In addition, data fusion is also an essential part in data management layer, which integrates data from multiple sources to provide much more consistent, accurate and meaningful information than that provided by any individual data source. For instance, vehicle entering or exiting can be monitored by different techniques such as cameras, RFID, radar and so on. The data sensed by different techniques should be pre-processed before uploading to cloud center for further optimization.

C. DATA PROCESSING LAYER

Data processing layer is the core of the fog computing based smart parking; it integrates the application of fog computing and remote cloud center [25]. This layer is in charge of processing the sensed data via fog computing as well as cloud computing. Specifically, fog nodes record the periodically updated parking lot information, respond to parking queries, make predictions based on the monitored slots occupancy and the waiting vehicles, and upload parking queries to cloud center for further slots allocation decision. On the other hand, remote cloud center is responsible for global parking spaces allocation when the number of vehicles with parking demands is larger than that of parking spaces.

D. APPLICATION LAYER

The application layer offers specific intelligent applications and services to vehicles with parking demands, aiming to reduce driving time, gasoline wastes and vehicle exhaust emissions. For example, interested vehicles/drivers can issue query service to know the occupancy status of parking lots within the communication range. Regarding the parking lot selection, vehicles can be guided to suitable parking lot with the aid of positioning and navigation services.

IV. PROBLEM FORMULATION

In this section, we give an introduction to the involved entities (i.e., fog nodes and cloud nodes), with regards to the corresponding functionality as well as the roles played in the smart parking systems. Then we formulate the evaluation metrics on parking costs mathematically. Finally, the objective function is given which indicates how the parking request is allocated.

A. FOG NODES AND THE ROLES

Parking slots are spatial-temporal resources which can be monitored by various sensors (e.g., surveillance cameras, RFID tags and so on). In the proposed architecture, each parking slot is defined by a six-tuple, $ps = (slotID, OCC, vehID, timeStamp, DUR, SPCL)$,of which each element is detailed as follows:

- Parking slot ID ($slotID$): identification represented by an integer or the corresponding position (x_i, y_i) on the Euclidean plane. For instance, for parking slot localization based on surveillance cameras, each parking slot can be denoted by integers; otherwise, a corresponding position (x_i, y_i) might be a good alternative.
- Occupancy (OCC): This field denotes the occupancy status of the parking slot. If the parking slot is occupied, set OCC to 1 and 0, otherwise.
- Vehicle ID ($vehID$): If OCC equals 1, a unique identification should be used to identify the vehicle occupying the slot. Currently, either the license plate numbers or the OBUs can uniquely label the vehicles. If OCC equals zero, this field can be empty or assigned with a default value.
- Time stamp ($timeStamp$): This field records the timestamp when the vehicle starts to park at the parking space. If OCC equals zero, this field can be empty or assigned with a default value.
- Duration (DUR): During the interactions between vehicles and fog nodes, we assume that vehicles with parking demand will provide a rough parking time, so that fog nodes can predict the possible status of parking slots in the near future. Accordingly, we use DUR to approximately represent the parking time. If OCC equals zero, this field can be empty or assigned with a default value.
- Special use ($SPCL$): Some parking slots are provisioned exclusively for special purposes (e.g., police cars). We use $SPCL$ to denote this type of purposes.

Based on the descriptions above, each parking lot can be regarded as a set of ps, which are stored and managed by fog nodes. For instance, when a parking slot is occupied, the information about the parking lot can be updated in real time by fog nodes. Fog nodes disseminate the periodic beacon messages about the current parking lot information to RSUs. Specifically, Figure 3 depicts the general interactions among

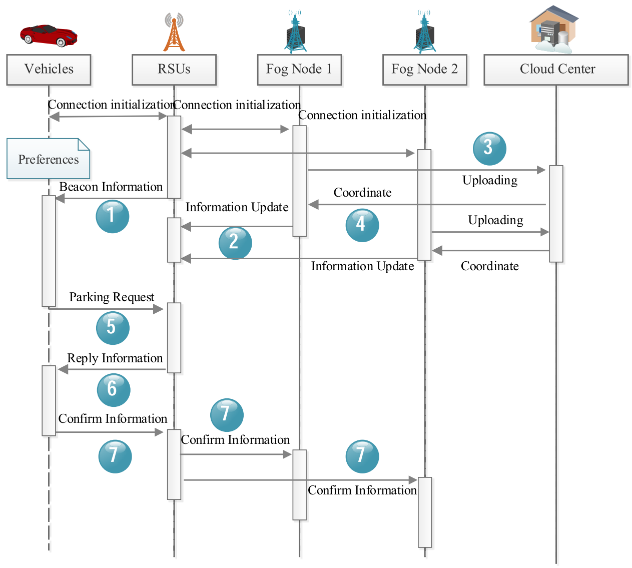


FIGURE 3. Interactions between vehicles and parking lot.

these entities (i.e., vehicles, fog nodes, RSUs, and cloud node). The procedure works as follows:

1. Fog nodes deployed at the parking lot disseminate to RSUs the periodic beacon messages $bcnMsg = (lotPos, timestamp, avlSlots, futSlots, curReq)$, where *lotPos* denotes the location of the parking lot, *timestamp* the time beacon information is sent, *avlSlots* the current number of vacant parking space, *futSlots* the number of new slots available in the near future, and *curReq* the number of vehicles approaching the parking lot. Note that the number of new vacant slots in the near future can be predicted by the duration *DUR* recorded in *ps*.

2. After receiving the beacon information, interested vehicles send to fog nodes the parking request $pReq = (vehPos, timestamp, OBU\_ID, pTime, Destination)$, where *vehPos* denotes the current position of vehicles, *OBU_ID* a unique identification of an OBU as well as the vehicle, *pTime* the rough parking time, *Destination* the place drivers want to leave for eventually.

3. After receiving the parking request via RSUs, the fog node copes with it as followings. Firstly, the fog node estimates the time by which the vehicle will arrive; Then the fog node predicts whether there is a vacant parking slot for the vehicle by combining the current slot information and new vacant slots at that time; If there is a vacant slot for the vehicle, the fog node send to it the reply information, denoted by $reInfo = (lotPos, OBU\_ID, timeStamp, avlSlots, futSlots, curReq, cost)$, where *OBU_ID* represents the receiving node (i.e., vehicles), *timestamp* the time the reply information is sent,*cost* the parking fees calculated by fog nodes. Note that *lotPos*, *avlSlots*, and *curReg* are the same meanings as defined in *bcnMsg*. If there is no vacant slot for the vehicle by the time it arrives, the fog node uploads *pReq* to the cloud center, where a globally optimal parking slot allocation scheme is determined based on the specified metrics such as parking fees. This is achievable since in the architecture fog nodes also disseminate *bcnMsg* to the cloud center besides RSUs. Thus, cloud center maintains the parking slot status of all the parking lots in the coordinated area.

4. After receiving the reply information, vehicles have two options. One is that the vehicle follows the suggestion and drive to the specified parking lot. The other is that drivers may ignore it, due to their own parking preferences. If the vehicle follows the instruction, it needs to send the fog node the confirm information which is very important for fog nodes to update the parking lot status.

In this proposed architecture, we do not take into account the parking slot reservation, for the reason that in some cases, e.g., when vehicles are very close to the parking lot and drivers are aware of its location, there is no need to turn on the communication devices (e.g., OBU). However, if parking slot reservation is enabled, these vehicles may be blocked at the entrance, even if they arrive much earlier than those with reservations. In addition, from the viewpoint of parking lot owners, keeping current parking spaces idle with vehicles waiting outside is not an efficient way to gain benefits. Even worse, traffic jam and car accidents will make owners suffer great losses, since the vehicles ordering the parking space may arrive too late or even do not appear at all.

B. CLOUD CENTRE AND THE ROLES

In traffic dense area with multiple parking lots in the vicinity, there may be no vacant parking spaces in all these parking lots in peak hours. Drivers may select the parking places based on their own preferences, which however can render a longer waiting time if other drivers with similar preferences behave the same way. To achieve an efficient parking slot allocation with regards to all parking places, the cloud computing is introduced to assist decision making. In our application scenario, each parking lot is equipped with a fog node which is responsible for service provisioning and parking slot prediction, while the cloud nodes located at the remote cloud center take charge of parking slot allocation among multiple parking lots in the coordinated area.

The interactions between cloud nodes and fog nodes have been illustrated in Figure 3. When no parking slot is available in these parking places, the parking requests sent by vehicles will be uploaded to cloud nodes, where the global optimization of parking slot allocation is performed with regards to the specified metrics.

C. TOTAL COSTS OF PARKING PRECEDURE

Several factors affect the decision making on which parking lot the vehicle should park, such as the drivers preferences, the destination, the parking fees, and so on. Specifically, we incorporate these factors and formulate the total costs taken in the parking procedure as follows:

$$\begin{aligned} cost &= C_{driving}(vehPos, lotPos) + C_{waiting}(Num_w, futSlots) \\ &\quad + C_{parking}(pTime) + C_{walking}(lotPos, Destination) \end{aligned} \tag{1}$$

where, $C_{driving}(vehPos, lotPos)$ represents the driving cost from the current position to the parking lot, $C_{waiting}(Num_w, futSlots)$ the waiting cost when vehicles wait at the entrance to the parking lot, $C_{walking}(lotPs, Destination)$ the parking fees, and $C_{walking}(lotPos, Destination)$ the walking cost from the parking lot to the destination. In the next, we will calculate these costs, respectively.

To estimate the driving cost and walking cost, we assume that distance between two positions, denoted by $Dis(pos_1, pos_2)$, can be obtained, e.g., by GPS in vehicles or smart phones. Thus the sum of driving and walking costs can be calculated as follows:

$$C_{driving} + C_{walking} = \delta \cdot Dis(vehPos, lotPos) + \gamma \cdot Dis(lotPos, Destination) \quad (2)$$

where, δ (resp.γ) denotes the driving cost (resp. walking cost) per distance unit.

The calculation of the waiting cost depends on the number of current vacant parking slots, the number of requests and the predicted vacant slots in the future. The number of vehicles waiting at the entrance can be monitored and counted by sensors such as surveillance cameras, the RFID tags, infrared ray and so on. The number of slots to be vacant can also be predicted based on ps. The number of approaching vehicles can be counted based on the confirm information sent by vehicles. As a result, the position of arbitrary vehicle (e.g., on the road or in the waiting queue) can be calculated. For example, there are ten vehicles approaching the parking lot and twenty vehicles already waiting at the entrance. For the vehicles on the road, according to the parking request $pReq$, the fog node can estimate the times by which each vehicle arrives and the arrival time can be denoted by a vector $T_{arr} = (t^1_{arr}, \ldots, t^{10}_{arr})$. Combing the periodic beacon message $bcnMsg$, the fog node can estimate the position of any vehicle which is waiting in queue.

Suppose that the information about ps is updated every fixed time interval. We denote by t_{invl} this fixed time interval. Let $Num_i(\geq 1)$ be the position of vehicle i waiting at the entrance, i.e., there are other $Num_i - 1$ vehicles in front of it. Let WT_i denote the total waiting time for vehicle i. In reality, the waiting time of arbitrary vehicle can be collected by various sensor devices. We in this paper estimate the waiting time with the help of ps and t_{invl}. To calculate the waiting time of vehicle i, we just need to find out how soon the number of parking slots to be available amounts to at least Num_i. In other words, we need to find the minimal number of updates of ps, denoted by k_{min},which satisfies:

$$k_{min} = \min\{k | \sum_{j=1}^{k} futSlots[j] \geq Num_i\} \quad (3)$$

where, $futSlots[j]$ represents the number of vacant slots in the j^{th} time interval. Thus, the waiting time can be expressed by:

$$WT_i = k_{min} \cdot t_{invl} + T_{ready} \quad (4)$$

Where T_{ready} represents the duration from the time vehicle begins to parking to the time the parking process is finished and we assume it is fixed, without consideration of drivers' preferences and skill differences. Accordingly, the waiting cost of vehicle i can be calculated as follows:

$$C_{waiting} = \lambda \cdot WT_i \quad (5)$$

where, λ denotes the waiting cost per time unit. Note that the precision of the waiting cost directly depend on the update frequency of ps. Higher update frequency of ps usually gives rise to preciser waiting cost, which however incurs additional computational and stored overheads.

The parking fees can be calculated as follows.

$$C_{parking}(pTime) = \begin{cases} \alpha, pTime \leq t_1 \\ \alpha + (pTime - t_1) \cdot \beta, \quad pTime > t_1 \end{cases} \quad (6)$$

where, t_1 is introduced as a time division point to cater for a real parking environment. For instance, when the parking time is smaller than t_1, the parking fees are unchangeable. When the parking time is longer than t_1, the parking fees increase linearly with the increase of parking time.

Thus the total cost can be calculated by fog nodes. However, considering the preferences of drivers, e.g., some drivers cares about parking fees more importantly than walking distance from parking lot to the destination while others are just the reverse, thus different weights can be assigned to the four parts based on drivers' preferences.

V. PARKING SLOT ALLOCATION ALGORITHM

A. PROBLEM STATEMENT

To facilitate our further discussion, Table I lists some key notations to be used through the paper. For the generalized parking problem, suppose there are m vehicles which send parking requests to fog nodes and n parking lots in the vicinity. We use an allocation indicator variable, denoted by $\varphi(i, j)$, to represent the parking decision. $\varphi(i, j) = 1$ if vehicle i parks at parking lot j and 0, otherwise. Many metrics can act as optimization objectives such as parking fees minimization, energy consumption minimization, and so on. Considering all aspects involved in the parking slot allocation problem, we in this paper regard the total costs as our objective function. Thus, the objective function about parking slot allocation problem P can be modeled as follows:

$$\text{Minimize}\,(P): \ f = \sum_{i=1}^{m}\sum_{j=1}^{n} \varphi(i, j) \cdot Cost(i, j)$$

$$\text{s.t. } Cost(i, j) = w_1 \cdot C_{driving}(i, j) + w_2 \cdot C_{waiting}(i, j) + w_3 \cdot C_{parking}(i, j) + w_4 \cdot C_{walking}(i, j) \quad \text{c1}$$

$$WT(i, j) \leq Deadline(i) \quad \text{c2}$$

$$C_{parking}(i, j) \leq Bgt(i) \quad \text{c3}$$

$$Dis(j, Destionation) \leq EXP_{dis}(i) \quad \text{c4}$$

TABLE 1. **Notation descriptions.**

Notation	Definition
m	Total number of vehicles requesting parking slots
n	Total number of parking lots
$Cost(i,j)$	Total cost when vehicle i parking at parking lot j
$C_{driving}(i,j)$	Driving cost to parking lot j for vehicle i
$C_{waiting}(i,j)$	Waiting cost at parking lot j for vehicle i
$C_{parking}(i,j)$	Parking cost at parking lot j for vehicle i
$C_{walking}(i,j)$	Walking cost from position of i to position of j
$\varphi(i,j)$	Allocation indicator variable
$WT(i,j)$	Waiting time at parking lot j for vehicle i
$Deadline(i)$	Deadline for vehicle i
$Bgt(i)$	Budget of parking fees for vehicle i
$Dis(i,j)$	Distance between the positions of i and j
$EXP_{dis}(i)$	Longest walking distance vehicle i can tolerate

$$\sum_{j=1}^{n}\varphi(i,j)=1 \qquad \text{c5}$$

$$\sum_{i=1}^{4}w_i=1,\quad 0\leq w_i\leq 1,\ i\in\{1,2,3,4\} \qquad \text{c6}$$

$$\varphi(i,j)\in\{0,1\} \qquad \text{c7}$$

Considering that different drivers may have different preferences, constraint c1 represents that the total costs are the weighted sum of four parts as defined above, where w_i represents drivers' preferences towards each part. Constraint c2 ensures that the waiting time of vehicle i at parking lot j should not exceed the deadline of i. Constraint c3 represents the parking fees should not go beyond the budget. Of the four parts which constitute the total costs, the cost on parking fees is the one on which people usually have the most direct and intuitive impression. People tend to choose a parking lot with an appropriate price. We denote this constraint on the budget by $Bgt(i)$. The distance from parking spot to the destination is usually an important factor in parking lot selection. We use constraint c4 to represent that the distance should not exceed the furthest distance drivers can tolerate. The allocation indicator variable $\varphi(i,j)$ denotes which vehicle should be allocated to which parking lot. However, one vehicle can be allocated to at most one parking lot, and thus we use constraints c5 and c7 to represent this constraint condition. Constraint c6 represents the preferences of different drivers towards the four parts.

The problem P is actually a special 0/1 multiple knapsack problem, where vehicles correspond to the items and parking lots correspond to the knapsacks without capacity limitation. Therefore, this combinatorial optimization problem is NP-hard. The most straightforward way to solve this problem is to enumerate each of n^m potential solutions, which however is not impractical in reality.

B. GREEDY ALLOCATION ALGORITHM

It is not appropriate to solve the parking slot allocation problem P by heuristic algorithms such as genetic algorithms and ant colony optimization, for the reason that these algorithms process the parking requests in batches instead of one by one in chronological order, which means that the parking slot is not allocated in real time fashion. It contradicts the principle of real time towards smart parking proposed in this paper. Besides, vehicles may miss the specified parking lot by the time the allocation result is received. Based on the above observations and analysis, we propose greedy parking slot allocation algorithms to obtain near optimum solution with regards to the objective function. Specifically, two algorithms which based on different types of parking requests are put forward in the next subsections.

1) SINGLE PARKING REQUEST BASED SLOT ALLOCATION ALGORITHM

In this paper, we assume that the fog nodes try to cope with the parking requests in the order they arrive, such that the requests can be responded in the real-time fashion. Specifically, the pseudo code of the greedy parking slot allocation algorithm is shown in Algorithm 1. We denote this single parking request based slot allocation algorithm by GPSA. By single parking request, we mean that most of parking requests arrive in sequence. In GPSA, the parking request is processed in chronological order. When receiving the parking request, the corresponding fog node will first check whether

Algorithm 1 Greedy Parking Slot Allocation Algorithm (GPSA)

Input: Fog Nodes (FN), Parking lots status, Parking request $pReq$.
Output: Parking allocation decision.
1: **If** FN_i receives the parking request $pReq$ **do**
2: **If** $avlSlots[i] > 0$ **do**
3: Response to the parking request by providing $lotPos[i]$.
4: **Else**
5: upload $pReq$ to cloud center.
6: **For** $j = 0: N$ **do**
7: **If** the constraints (c2-c4) are satisfied **do**
8: Calculate the total costs if the vehicle parks at $lotPos[j]$.
9: Variable C_{min} records the minimal cost calculated so far.
10: **Endif**
11: **Endfor**
12: Response to the parking request by providing $lotPos[j]$ with C_{min}.
13: **Endif**
14: **Endif**

the parking lot it is in charge of has vacant parking spaces or not. If there is a parking slot available, the fog node will respond to the vehicle which requests the parking slot by offering the vacant parking spaces. Otherwise, the request is uploaded to cloud center, and the cloud nodes are responsible for allocating the parking lot to it. It needs to traverse each parking lot and find the one with minimal parking costs, denoted in lines 6-11. A variable C_{min} is used to record the minimal parking cost which has been searched out so far. After that the corresponding parking lot can be sent to the vehicle which sending the parking request.

GPSA tries to seek global optimization with regards to the objective function by allocating the vehicle to the parking lot which has the minimal total costs for it. However, GPSA does not ensure that the global optimum value can be obtained for all the parking requests, since several factors may affect the decision making and thus lead to sub-optimal solution.

2) MULTIPLE PARKING REQUEST BASED SLOT ALLOCATION ALGORITHM

GPSA copes with the parking requests in chronological order. However, in peak hours, it is common that multiple parking requests arrive at the same time or the arrive intervals are negligible. For these concurrent parking requests, fog nodes store and process them in random order. In other words, GSPA does not have efficient mechanism to process these concurrent parking requests. In worst case, GPSA degrades to the random algorithm when the parking requests are all concurrent parking requests.

As one of these factors which affect the decision making and thus lead to sub-optimal solution, the order the concurrent parking requests are processed may have a direct impact on seeking the best solution. Different allocation plans may have different results.

Intuitively, there are several heuristic rules which can guide us to allocate these parking requests. One of them is to firstly calculate the parking costs of these parking requests, respectively. Then select the one with the minimum value of parking costs to allocate the parking lot. For the remaining parking requests, repeat the procedure iteratively, until all the parking requests are processed. However, this heuristic consumes more computational resources.

We notice that of all the four kinds of costs (i.e., $C_{driving}$, $C_{waiting}$, $C_{parking}$ and $C_{walking}$) which constitute the total parking costs, only the waiting costs are not independent of other vehicles, which means how long the vehicle waits depends on the number of vehicles in front of it as well as how often the vehicles which are parking depart from the parking lot. As for other three kinds of costs, they do not change any more, as long as the parking lot is determined. This observation inspires us to put forward another heuristic rule to allocate these concurrent parking requests. Namely, we can sort these requests by parking time and the request with minimum parking time is first processed. The remaining requests are processed iteratively, until all of them are processed. Consider a simplest case where there is one parking lot with only one parking slot available. Three parking requests, denoted by v_1, v_2, v_3 respectively, arrive at the same time. The corresponding parking times are 1, 2 and 3 hours respectively. Assume that another two parking slots to be available in 6 and 15 minutes later, respectively. There are two alternative ways to process the requests, as denoted in Fig. 4. One is that the parking request with shorter parking time is processed first; the other is that the parking request with longer parking time is processed first. It is obvious that when requests are processed in accordance with the former way, the parking slot to be available appear much earlier, i.e. 21 minutes in advance compared to the latter way.

The sooner the parking slot is available, the less time the vehicle needs to wait in queue. Therefore, we enhance the abilities of GPSA by leveraging this heuristic, denoted by EnGPSA. Firstly, fog nodes receive the parking requests by time slots. The number of parking requests received in time slot t_i is denoted by $Num(t_i)$. In this paper we assume that the requests received during the same time slot are considered to be concurrent requests. We process these requests as illustrated above. Specifically, the procedure is shown in Algorithm 2. The heuristic is activated when the number of parking requests in single time slot is larger than 1.

VI. SIMULATION RESULTS AND ANALYSIS

A. PARKING SCENARIO BASED ON REAL ENVIRONMENT

To evaluate the parking slot allocation algorithms proposed above, we have conducted extensive experiments in this section, which is based on a real environment shown in Fig.5. Fig. 5(a) shows a mall named Golden Eagle in Xuzhou City, surrounded by six parking lots labeled by colored numbers.

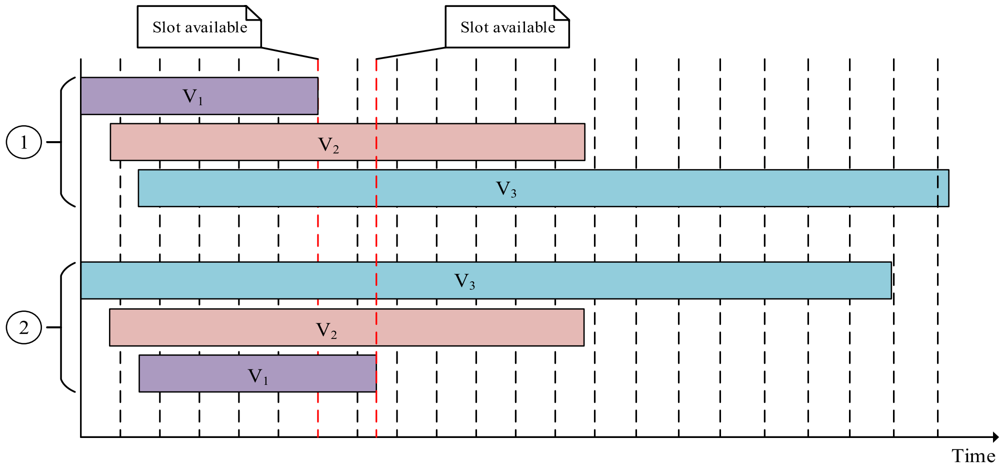


FIGURE 4. Two alternative ways to process the requests.

Algorithm 2 Enhanced Greedy Parking Slot Allocation Algorithm (EnGPSA)

Input: Fog Nodes (*FN*), Parking lots status, Parking request *pReq*.
Output: Parking allocation decision.

```
Count the parking requests Num(t_i) for each time slot t_i.
Foreach time slot t_i do
  If Num(t_i) = 1 do
    Call GPSA.
  Else If Num(t_i) > 1 do
    Sort the requests by parking time in an ascending order.
    For j = 0 :Num(t_i) do
      Call GPSA for request j.
    Endfor
  Endif
Endfor
```

The corresponding parking network is abstracted and presented by an undirected acyclic graph shown in Fig. 5(b), where vertexes denote the parking lots, and the edges denote the distances between two adjacent parking lots. In addition, the rectangular box around each vertex represents the parking lot information, with the numbers above representing the vacant parking spaces and the numbers below representing the number of vehicles waiting in queue at the entrance. Intuitively, the deployment of such dense parking lots with hundreds of parking spaces around one mall seems adequate to satisfy the parking demands.

However, based on our experiences and survey, several disadvantages which do not facilitate parking are observed. First of all, the number of parking spaces is far from enough, especially in the evening and at the weekends. Besides, the street, which is located on the north side of Golden Eagle and connects four parking lots (i.e., Parking lot 10, 1, 3, and 7), is one-way street with no U-turns and only permits to pass from east to west. The most straightforward way to select parking lot for vehicles is to try each one by one along the street. This attempt will explain based on our observation that the front parking lots (e.g., Parking lot 7 and 3) usually have a larger number of vehicles waiting for parking. Vehicles have to try to parking lot 1 or 10, if the front parking lots are all occupied and the number of vehicles waiting there goes beyond their tolerance. However, if no vacant parking spaces are available in p1 or p10, besides waiting, vehicles have to circle around the mall again to try other parking lots. It is a painful experience, considering the heavy traffic and multiple traffic lights from north to south. The inappropriate parking lot selection causes both time waste and vehicle exhaust emissions.

B. EXPERIMENTAL SETUPS

The initial experimental setups about the total parking spaces and vacant parking spaces are listed in Table 2. Note that parking lot P10 is a little far away from the mall, therefore the number of vacant parking slots is larger than that of other parking lots in the initial setups. Considering that the one-way road on the north side of Golden Eagle only permits vehicles passing from east to west, people tent to park at the first parking lot they meet based on our observations. Thus, the parking lot P7 has the least number of vacant parking lots at the beginning. For those vehicles which have parked at the corresponding lots, we assign them the random parking time duration which ranges from ten minutes to three hours. These parking times can be used to predict the waiting time of vehicles outside, as denoted in Eq. (3) and (4). Intuitively, people care more about waiting time than other evaluation metrics such as walking costs and driving costs, which can be achievable as long as the value of w_2 is higher than others. Specifically, if w_2 is set to 1, it means that waiting cost is the only evaluation metrics. We assume that the price per hour for parking are all the same for different parking lots to cater for the real parking environments.

GPSA and EnGPSA are investigated and compared to other two approaches in this paper. One is the the

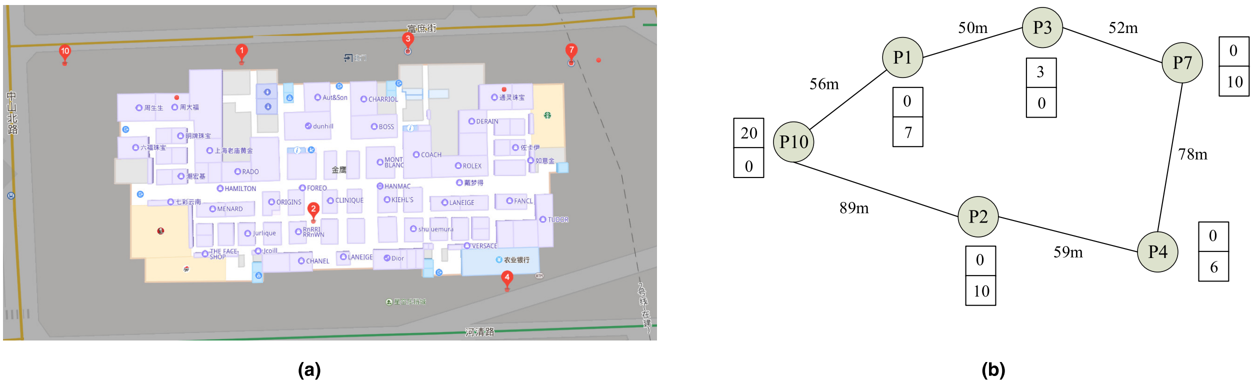


FIGURE 5. Parking scenario towards smart parking based on fog computing. (a) Real time scenario. (b) Parking lot network.

TABLE 2. Experimental setups on parking slots

Parking Lot	Vacant Slots	Total Parking lots
P1	17	30
P2	37	40
P3	25	30
P4	21	50
P7	10	115
P10	44	45

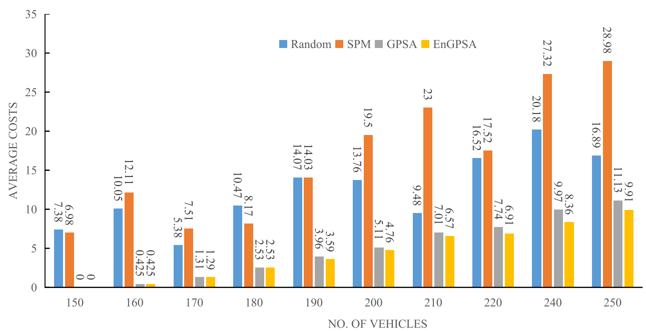


FIGURE 6. Average costs for each vehicle with three parking allocation approaches (small).

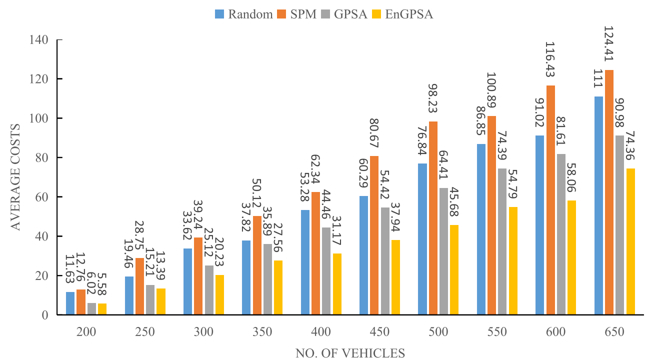


FIGURE 7. Average costs for each vehicle with three parking allocation approaches (large).

random approach; the other is the approach proposed in [21]. For simplicity, we denote this approach by SPM. The random approach actually assigns vehicles with parking demands to the parking lots in a random way, which does not take into account the total parking costs defined in Eq. 1. SPM, however, allocates parking requests based on uers's preferences towards the parking lot. As far as EnGPSA itself is concerned, the main factor which affects its performance is the arrival rate of the concurrent parking requests. We will also evaluate this kind of influence in the experiments.

C. EXPERIMENTAL RESULTS AND ANALYSIS

The total number of vacant parking slots at the beginning is 154, and we have conduct two sets of experiments to evaluate the performance of GPSA and EnGPSA, under different number of parking requests. The results are shown in Fig.6 and Fig.7, respectively.

In the first set of experiments, the number of parking requests ranges from 150 to 250 with a step of 10. From Fig.6 we can observe that the advantages of both GPSA and EnGPSA are obvious compared to the random approach, denoted by "Random", as well as SPM. The average performance improves almost by 64% and 69%, compared to the random approach and SPM, respectively. For the performance comparison between GPSA and EnGPSA, however, the average performance improvement for EnGPSA is not so obvious compared to GPSA. Two reasons may lead to this consequence. One is that the number of parking requests is not so large, and thus even for the vehicles in the longest waiting queue, they do not need to wait for very long time. Besides, in our experimental setups, we assign the waiting costs the largest weight in contrast to other three parking costs. The other is that the number of concurrent parking requests also has an influence on the performance of EnGPSA. If the number of arriving parking requests during one time slot is very large, EnGPSA may have a great advantage compared to GPSA. However, if there is only one parking request in each time slot, EnGPSA is actually the same as GPSA.

In the second set of experiments, the number of parking requests ranges from 200 to 650, with a step of 50. A few interesting results can be observed from Fig.7. First, both GPSA and EnGPSA still outperform the random approach and SPM. However, the perform improvement is not as much as the experimental results shown in Fig.6. This is because with the increasing number of parking requests, each parking lot has vehicles waiting in queue, to some extent, each vehicle needs to wait for parking, by either the random approach or the two fog computing based parking strategies. Second, for the performance comparison between GPSA and EnGPSA, the average performance improvement for EnGPSA is increasingly demonstrating its advantages compared to GPSA, which reflects the importance of our heuristic rule, i.e., the parking time of each vehicle to be parked does affect the performance of parking strategies. Vehicles with shorter parking time duration are assigned first, which will save a little more time for the vehicles waiting in queue outside. Last, when the number of parking requests are very large, the increase of the total parking by three approaches is approximately linear in the number of parking requests, which is different from the results shown in Fig.6. In Fig.6, the random approach shows more randomness compared to other two approaches, this is because there exist some cases that some parking lots have been fully occupied with long waiting queue outside, while some parking lots still have numbers of parking spaces available.

Note that no matter how the number of parking requests varies, GPSA and EnGPSA are still outperforming SPM. Several reasons which can lead to this kind of results are analyzed and listed as follows. First, in essence, SPM is not a real-time parking slot allocation algorithm, this is because the fog nodes cope with the parking requests in batch. Via receiving these parking requests, fog nodes do not process them immediately. They classify these parking requests based on users' preferences towards the parking lots. After that, fog nodes begin to process the parking requests based on the objective functions. Second, SPM aims to maximize the profits of parking lot owners. To this end, fog nodes always select the parking requests with the maximal parking time in the candidates, which on one hand contradicts our optimization objective as well as the heuristic rule, and on the other hand makes early requests with short parking time are delayed. Last but not least, fog nodes lack efficient mechanism to cope with the case that the response latency is very long due to the large number of parking requests with the same preferences.

In the third set of experiments, we evaluate the average parking costs of each parking lot. The random approach usually results in the uneven distribution of parking requests allocation. Some parking lot which is supposed to be assigned more parking requests are assigned few parking requests, while those which have long waiting queues still have parking requests coming. This will cause a longer waiting times for some vehicles with somethings urgent to do. The experimental results are shown in Fig.8. For the random approach, parking lot P3 has the maximum parking costs while P10 has the minimum parking costs. Intuitively, it is much fairer to reallocate the vehicles in waiting queue of P3 to P10, so as to reduce the parking costs of vehicles.

For GPSA and EnGPSA, the parking requests are assigned based on the parking costs computed by fog computing and

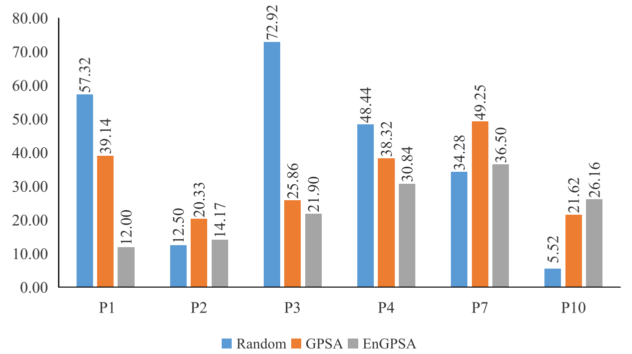


FIGURE 8. Average costs for each parking lot vs. three parking allocation approaches.

cloud computing. Each parking request will be allocated to the most appropriate parking lot with regards to the objective function. From Fig.8, we can see that GPSA and EnGPSA still outperforms the random approach for any of the parking lots. For the performance comparison between GPSA and EnGPSA, EnGPSA outperforms GPSA in most of parking lots except the parking lot P10. The only reason which can lead to this consequence is that to respect drivers' preferences towards parking lots, we permit that some vehicles which send the parking requests do not follow the suggestions and they finally choose their desired parking lot based on their own preferences. However, in the simulation, the chance that vehicles do not follow the instructions is very small. Except this consideration, EnGPSA is at least as good as GPSA.

Note that due to different optimization objectives compared to SPM, we only compare our algorithms with the random approach in the third set of experiments.

In the last set of experiments, we evaluate the influence of the number of concurrent parking requests on EnGPSA. EnGPSA degrades into GPSA when there is no concurrent parking requests at all. However, according to our observations in real environments, there do exist concurrent parking requests, especially in peak hours, which is the reason why EnGPSA is introduced to improve the performance of GPSA. To this end, in the experiments, we control the number of concurrent parking requests by a control factor, denoted by p. The control factor p varies from 0 to 1 with a step 0.2. Specifically, when p equals 0, EnGPSA degrades into GPSA. When p equals 1, all the parking requests are concurrent. We can decide the number of concurrent parking requests by $p * \text{Num}$. Here, Num is the total number of parking requests. Although it is impractical, we still use it in comparison with other values of p in the experiments. Then we decide which vehicles during which time slots are concurrent in a random way in the experiments.

The experimental results are shown in Fig.9. We observe that when the number of parking requests varies from 200 to 300, the average costs for vehicles are almost the same, no matter how p is varied. However, when the number of parking requests increases sharply from 300, it is obvious that the number of concurrent parking requests is improving the

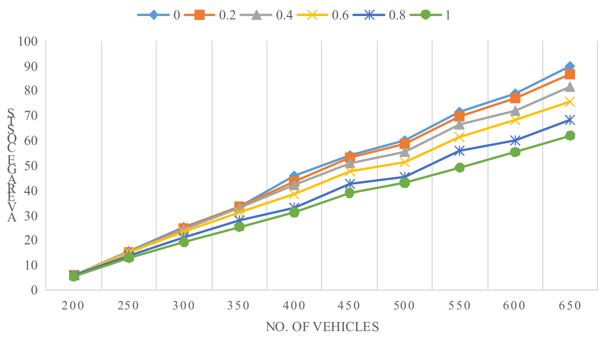


FIGURE 9. Average costs for each vehicle with different concurrent parking requests.

performance of EnGPSA compared to GPSA. From Fig.8 we can conclude that when the number of parking requests is very large, the number of concurrent parking requests has become crucial to the performance of EnGPSA. In other words, the more concurrent parking requests, the better the performance of EnGPSA in contrast to GPSA.

VII. CONCLUSION

Parking problems brings both severe gasoline wastes and vehicle exhaust emissions. To solve the parking problems, we have proposed a fog computing based smart parking strategy, which combines the advantages of VANETs and fog computing to provision parking services in real time fashion. To cope with the concurrent parking requests, a heuristic rule is incorporated into proposed smart parking approach, which assigns vehicles with shorter parking time duration a bigger priority among concurrent parking requests. The experimental results have proven that our fog computing based smart parking strategies can effectivey improve the parking problems.

For the future works, we plan to design more humanized parking strategies, which incorporates drivers' preferences to decision marking process.

REFERENCES

[1] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A new VANET-based smart parking scheme for large parking lots," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1413–1421, doi: 10.1109/INFCOM.2009.5062057R.

[2] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Things* (Technology, Communications and Computing). Singapore: Springer, 2018.

[3] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017, doi: 10.1109/TITS.2017.2685143.

[4] K. Hassoune, W. Dachry, F. Moutaouakkil, and H. Medromi, "Smart parking systems: A survey," in *Proc. 11th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, Mohammedia, Morocco, 2016, pp. 1–6, doi: 10.1109/SITA.2016.7772297.

[5] M. Y. I. Idris, Y. Y. Leng, E. M. Tamil, Z. Razak, and N. M. Noor, "Car park system: A review of smart parking system and its technology," *Inf. Technol. J.*, vol. 8, no. 2, pp. 101–103, 2009, doi: 10.3923/itj.2009.101.113.

[6] E. Polycarpou, L. Lambrinos, and E. Protopapadakis, "Smart parking solutions for urban areas," in *Proc. IEEE 14th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Madrid, Spain, Jun. 2013, pp. 1–6, doi: 10.1109/WoWMoM.2013.6583499.

[7] T. Delot, S. Ilarri, S. Lecomte, and N. Cenerario, "Sharing with caution: Managing parking spaces in vehicular networks," *Mobile Inf. Syst.*, vol. 9, no. 1, pp. 69–98, 2013, doi: 10.3233/MIS-2012-0149.

[8] Z. Faheem, S. A. Mahmud, G. M. Khan, M. Rahman, and H. Zafar, "A survey of intelligent car parking system," *J. Appl. Res. Technol.*, vol. 11, no. 5, pp. 714–726, 2013, doi: 10.1016/S1665-6423(13)71580-3.

[9] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surv.*, vol. 50, no. 3, p. 32, Oct. 2017, doi: 10.1145/3057266.

[10] C. Zhu, H. Zhou, V. C. M. Leung, K. Wang, Y. Zhang, and L. T. Yang, "Toward big data in green city," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 14–18, Nov. 2017.

[11] C. Zhu, X. Li, V. C. M. Leung, L. T. Yang, E. C. Ngai, and L. Shu, "Towards pricing for sensor-cloud," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2017.264952C.

[12] C. Zhu, V. C. M. Leung, K. Wang, L. T. Yang, and Y. Zhang, "Multi-method data delivery for green sensor-cloud," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 176–182, May 2017, doi: 10.1109/MCOM.2017.1600822.

[13] C. Zhu, L. Shu, V. C. M. Leung, S. Guo, Y. Zhang, and L. T. Yang, "Secure multimedia big data in trust-assisted sensor-cloud for smart city," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 24–30, Dec. 2017.

[14] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, and L. T. Yang, "Trust-based communication for the industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 16–22, Feb. 2018, doi: 10.1109/MCOM.2018.1700592.

[15] W. Balzano and F. Vitale, "DiG-Park: A smart parking availability searching method using V2V/V2I and DGP-class problem," *Proc. 31st Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Taipei, Taiwan, 2017, pp. 698–703, doi: 10.1109/WAINA.2017.104.

[16] T. Wang *et al.*, "Data collection from WSNs to the cloud based on mobile Fog elements," *Future Gener. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.07.031.

[17] T. Wang *et al.*, "Fog-based storage technology to fight with cyber threat," *Future Gener. Comput. Syst.*, vol. 83, pp. 208–218, Jun. 2018, doi: 10.1016/j.future.2017.12.036.

[18] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016, doi: 10.1109/TVT.2016.2532863.

[19] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Kona, HI, USA, Mar. 2017, p. 6-9, doi: 10.1109/PERCOMW.2017.7917508.

[20] C.-Y. Huang and K. Xu, "Reliable realtime streaming in vehicular cloud-fog computing networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Chengdu, China, Jul. 2016, pp. 1–6, doi: 10.1109/ICCChina.2016.7636838.

[21] O. T. T. Kim, N. D. Tri, V. D. Nguyen, N. H. Tran, and C. S. Hong, "A shared parking model in vehicular network using fog and cloud environment," in *Proc. 17th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Busan, South Korea, 2015, pp. 321–326, doi: 10.1109/APNOMS.2015.7275447.

[22] I. Aydin, M. Karakose, and E. Karakose, "A navigation and reservation based smart parking platform using genetic optimization for smart cities," in *Proc. 5th Int. Istanbul Smart Grid Cities Congr. Fair (ICSG)*, Istanbul, Turkey, 2017, pp. 120–124, doi: 10.1109/SGCF.2017.7947615.

[23] P. P. Ray, M. Mukherjee, and L. Shu, "Internet of Things for disaster management: State-of-the-art and prospects," *IEEE Access*, vol. 5, pp. 18818–18835, 2017, doi: 10.1109/ACCESS.2017.2752174.

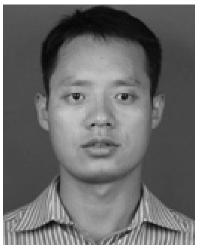
[24] M. Mukherjee *et al.*, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017, doi: 10.1109/ACCESS.2017.2749422.

[25] M. Mukherjee, L. Shu, D. Wang, K. Li, and Y. Chen, "A fog computing-based framework to reduce traffic overhead in large-scale industrial applications," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Atlanta, GA, USA, May 2017, pp. 1008–1009, doi: 10.1109/INFCOMW.2017.8116534.

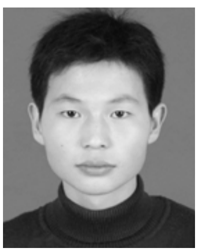
[26] C. Song and Z. Mou, "Noncooperative game theory based curb parking policies study," in *Proc. Int. Conf. Manage. Service Sci.*, Wuhan, China, 2011, pp. 1–3, doi: 10.1109/ICMSS.2011.5998256.

[27] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative fog computing for dealing with big data in the Internet of vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017, doi: 10.1109/MCOM.

[28] S. Park and Y. Yoo, "Network intelligence based on network state information for connected vehicles utilizing fog computing," *Mobile Inf. Syst.*, vol. 2017, 2017, Art. no. 7479267, doi: 10.1155/2017/7479267.

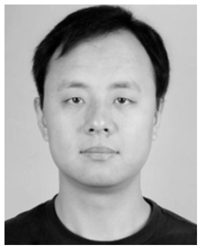


CHAOGANG TANG received the B.S. degree from the Nanjing University of Aeronautics and Astronautics in 2007 and the Ph.D. degree from the School of Information Science and Technology, University of Science and Technology of China, in 2012. He is currently an Assistant Professor with the China University of Mining and Technology. His research interests include mobile cloud computing, fog computing, Internet of Things, big data, and WSN.



XIANGLIN WEI was born in Suzhou, Anhui, China. He received the bachelor's degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, and the Ph.D. degree from the University of Science and Technology, Nanjing, in 2012. He is currently a Researcher with the Nanjing Telecommunication Technology Research Institute, Nanjing. His research interests include cloud computing, peer-to-peer networks, network anomaly detection, network measurement, and distributed system design and optimization. He has served as an editorial member for many international journals and a TPC member of several international conferences. He has also organized a few special issues for many reputable journals.

CHUNSHENG ZHU received the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Canada. He is currently a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, The University of British Columbia. He has authored over 100 publications published by refereed international journals (e.g., the IEEE Transactions on Industrial Electronics, the IEEE Transactions on Computers, the IEEE Transactions on Information Forensics and Security, the IEEE Transactions on Industrial Informatics, the IEEE Transactions on Vehicular Technology, the IEEE Transactions on Emerging Topics in Computing, the IEEE Transactions on Cloud Computing, the *ACM Transactions on Embedded Computing Systems*, and the *ACM Transactions on Cyber-Physical Systems*), magazines (e.g., the *IEEE Communications Magazine*, the *IEEE Wireless Communications Magazine*, and the *IEEE Network Magazine*), and conferences (e.g., IEEE INFOCOM, IEEE IECON, IEEE SECON, IEEE DCOSS, IEEE ICC, and IEEE GLOBECOM). His research interests mainly include Internet of Things, wireless sensor networks, cloud computing, big data, social networks, and security.



WEI CHEN received the Ph.D. degree in communications and information systems from the China University of Mining and Technology at Beijing, Beijing, China, in 2008. In 2008, he joined the School of Computer Science and Technology, China University of Mining and Technology at Xuzhou, Xuzhou, where he is currently a Professor. His research interests include machine learning, image processing, and wireless communications.

JOEL J. P. C. RODRIGUES (S'01–M'06–SM'06) is currently a Professor and a Senior Researcher at the National Institute of Telecommunications (Inatel), Brazil, and a Senior Researcher at the Instituto de Telecomunicações, Portugal. He is also the Leader of the Internet of Things Research Group (CNPq), the Director for Conference Development–IEEE ComSoc Board of Governors, an IEEE Distinguished Lecturer, the Technical Activities Committee Chair of the IEEE ComSoc Latin America Region Board, the President of the Scientific Council at ParkUrbis–Covilhã Science and Technology Park, the Past-Chair of the IEEE ComSoc Technical Committee on eHealth, the Past-Chair of the IEEE ComSoc Technical Committee on Communications Software, a Steering Committee Member of the IEEE Life Sciences Technical Community, and a Publications Co-Chair and a Member Representative of the IEEE Communications Society on the IEEE Biometrics Council. He is the Editor-in-Chief of two international journals and has been a General Chair and a TPC Chair of many international conferences, including IEEE ICC, IEEE GLOBECOM, IEEE HEALTHCOM, and IEEE LatinCom. He has authored or co-authored over 650 papers in refereed international journals and conferences, three books, and two patents. He had been awarded several outstanding leadership and outstanding service awards.