Parking Assignment: Minimizing Parking Expenses and Balancing Parking Demand Among Multiple Parking Lots

Oanh Tran Thi Kim, Nguyen H. Tran, *Senior Member, IEEE*, Chuan Pham, Tuan LeAnh, My T. Thai, *Member, IEEE*, and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*— Recently, a rapid growth in the number of vehicles on the road has led to an unexpected surge of parking demand. Consequently, finding a parking space has become increasingly difficult and expensive. One of the viable approaches is to utilize both public and private parking lots (PLs) to effectively share the parking spaces. However, when the parking demands are not balanced among PLs, a local congestion problem occurs where some PLs are overloaded, and others are underutilized. Therefore, in this article, we formulate the parking assignment problem with two objectives: 1) minimizing parking expenses and 2) balancing parking demand among multiple PLs. First, we derive a matching solution for minimizing parking expenses. Then, we extend our study by considering both parking expenses and balancing parking demand, formulating this as a mixed-integer linear programming problem. We solve that problem by using an alternating direction method of multipliers (ADMM)-based algorithm that can enable a distributed implementation. Finally, the simulation results show that the matching game approach outperforms the greedy approach by 8.5% in terms of parking utilization, whereas the ADMM-based algorithm produces performance gains up to 27.5% compared with the centralized matching game approach. Furthermore, the ADMM-based proposed algorithm can obtain a near-optimal solution with a fast convergence that does not exceed eight iterations for the network size with 1000 vehicles.**

***Note to Practitioners*—The efficiency of the parking assignment is critical to the parking management systems in order to provide the best parking guides. This article investigates the cost minimization problem for parking assignment while balancing parking demand among multiple parking lots (PLs). Previous parking assignment approaches do not jointly investigate the cost of parking and the cost of PL utilization. Therefore, they can fail to the local congestion problem caused by a large number of vehicles driving toward the same PL. In this article, a new method that considers both of minimizing parking expenses and balancing parking demand is proposed. It is obtained by using the alternating direction method of multipliers (ADMM)-based proposal that distributively solves a constrained optimization problem. Based on the experimental results, the ADMM-based algorithm outperforms the matching-based algorithm and the greedy algorithm in terms of the balancing parking demand and reducing parking expenses. The proposed method can be readily implemented in real-world industrial PLs. In the future work where parking assignments for electric vehicles are needed, our proposed mechanism can then be extended to solve the balanced electricity overload multiple charging stations.**

***Index Terms*— Alternating direction method of multipliers (ADMM), balancing parking demand, matching game, multiple parking lots (PLs), parking assignment.**

Manuscript received July 26, 2019; accepted October 11, 2019. Date of publication November 14, 2019; date of current version July 2, 2020. This article was recommended for publication by Associate Editor M. Dotoli and Editor S. A. Reveliotis upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant NRF-2017R1A2A2A05000995 and in part by the Basic Science Research Program through the NRF funded by the Ministry of Education under Grant NRF-2016R1D1A1B01015320. *(Corresponding author: Choong Seon Hong.)*

O. Tran Thi Kim, T. LeAnh, and C. S. Hong are with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 17104, South Korea (e-mail: ttkoanh@khu.ac.kr; latuan@khu.ac.kr; cshong@khu.ac.kr).

N. H. Tran is with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 17104, South Korea (e-mail: nguyen.tran@sydney.edu.au).

C. Pham was with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 17104, South Korea. He is now with the Synchromedia, École de Technologie Supérieure, Université du Québec, Quebec City, QC H3C1K3, Canada (e-mail: chuan.pham.1@ens.etsmtl.ca).

M. T. Thai is with the Computer and Information Science and Engineering Department, University of Florida, Gainesville, FL 32611-6120 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 17104, South Korea (e-mail: mythai@cise.ufl.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2019.2948200

I. Introduction

ACCORDING to the International Parking Institute (IPI), the number of vehicles on the road will reach 2.5 billion in 2050 [1]. Clearly, vehicle owners must circle around an overcrowding PL to search for an available parking space, thereby wasting money on fuels and time on searching the parking lot (PL). Parking is thus an emerging issue that affects not only people seeking for parking spaces but also city governments, as 30% of traffic in a typical downtown area is caused by the parking searching [2]. Hence, both academic and industrial studies have focused on parking assignment problems: how to assign vehicles to a parking space with a minimal cost. The purpose is to reduce traffic jams and emissions, thus enhancing the quality of life.

Most existing methods focus on minimizing parking costs for vehicles by trying to assign vehicles to available parking spaces that are closest to the vehicle's destination [3]–[8]. A majority of these works either address one PL [3]–[6]

or multiple PLs [7], [8]. For the case of one PL, these works ignored the potential cooperation with other PLs, which leads to inefficient parking space utilization. Since parking demands are dynamic, drivers may be searching extensively for a parking spot in one area, while other parking spots elsewhere remain mostly vacant. For instance, there are several free parking spaces at an office building in the evening, but a nearby restaurant has a high demand for parking. Thus, it is crucial to integrate available resources from PLs into one system and then allocate this resource to drivers efficiently. This approach will provide many benefits to society since it results in less congestion, fewer emissions, and lower rent. For multiple PLs, current works only consider to minimize parking assignment costs [3]–[8]. Without considering PL utilization, local congestion increases when some PLs have high demand, while others are underutilized. Therefore, balancing parking demand among multiple PLs should be considered to make the system be more efficient.

Most studies on finding parking spaces are generally classified into two types: sensor-based approaches and roadside units (RSUs)-based approaches. In sensor-based approaches [2], [9], [10], fixed sensors are embedded in the parking spots to monitor the parking spaces. The sensors detect the availability of spaces across some areas, and the locations of currently vacant parking slots are sent to the mobile devices so that the users can find a parking spot in the area. One of the biggest shortcomings of this scheme is that the drivers have to shift their focus from the road to their mobile devices. For the sake of safety, we need a well-guided system that can direct vehicles to an ideal open parking slot automatically.

Vehicular ad hoc networks (VANETs) have emerged as an optimal technology that can improve not only road safety but also provide a better driving experience. Approaches based on RSUs have therefore been considered as an effective method for parking management [11]–[13]. In such a method, RSUs have been used to manage PLs, and these RSUs can provide information about available parking spaces to the drivers. By using such an approach, one PL can be managed by multiple RSUs, and thus, a synchronization mechanism must be used [11], which is very complex and expensive. Both the aforementioned approaches only try to solve the problems of detecting open slots and guiding drivers to arrive at these open spaces. In addition, they only concentrate on available slots in dedicated PLs. The view is limited when there are abundant slots in many areas, as mentioned earlier. Therefore, the shared parking policy needs to be investigated to exhaust all idle parking resources. With this in mind, we consider fog computing and roadside cloud concepts for solving the parking problem in order to efficiently implement a shared parking policy.

In this article, the parking assignment problem (i.e., the assignment of vehicles to optimal parking spaces to minimize the total cost of the whole system) is addressed under sharing parking spaces among multiple PLs. Underlying these sharing scenarios, the parking assignment has to face a new problem: how to balance parking demand caused by dynamic and different parking demands among these particular shared PLs. The parking assignment problem in this article thus becomes a joint problem of minimizing the total parking cost and balancing the parking demand, which has not well explored in the literature. Our main contributions can be summarized as follows.

1) We analyze the parking assignment problem based on a matching approach. We model the parking assignment problem as a college admission matching game [14], [15], where one PL can accept many vehicles' parking requests, but each vehicle can only pick one PL. Our matching-based parking assignment is designed to minimize the parking expenses of the whole system by seeking a stable matching. This stable matching can guarantee that no vehicle can increase its profit by parking in a different parking space rather than the one assigned by our design.
2) While previous work has only focused on parking expenses, in this article, we deal with a new problem: balancing the parking demand among shared PLs. We thus formulate the parking assignment problem with two objectives: minimizing parking costs and balancing parking demand as a mixed-integer linear programming (MILP) problem.
3) For this MILP problem, we develop an algorithm based on the alternating direction method of multipliers (ADMM) method [16] to solve the parking assignment problem efficiently and achieve a near-optimal solution. This approach enables distributed implementation with affordable computational complexity.
4) Numerical results are analyzed to illustrate the performance of our proposed algorithms. The proposed ADMM-based algorithm outperforms the greedy by 36% in terms of parking utilization. This performance can attain up to 27.5% compared with matching game approaches.

The rest of this article is organized as follows. Section II describes our system model and related parameters. The optimal solution for minimizing parking expenses is derived in Section III. In Section IV, the solution is extended to the balancing parking demand among PLs, and an ADMM-based parking assignment problem is analyzed. Finally, we show the simulation results in Section V and draw conclusion in Section VI.

II. System Model and Problem Definition

A parking assignment system is observed in a shared multiple PLs system deployed in a smart city, as shown in Fig. 1. Hereafter, we describe in detail each component of the system.

A. System Components

1) PLs: In recent years, we have witnessed the rise of sharing resources in most network fields [17]–[19]. Therefore, in our model, sharing parking resources among multiple PLs is considered. A driver can park in a nearby expected destination if the initial desired parking place was not available or the expenditure is higher than expected. Beyond that, owners of private PLs will be incentivized to share the parking

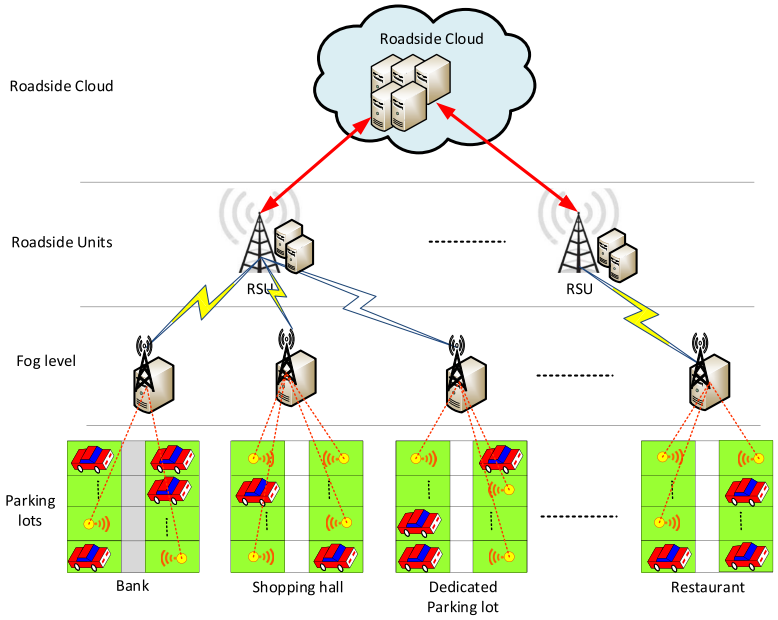

Fig. 1. Shared parking utilization in multiple PLs.

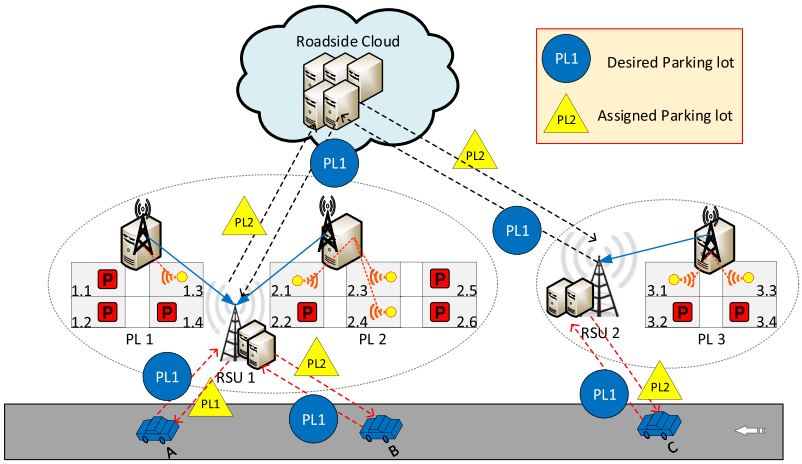

Fig. 2. Example of parking assignment in the fog and roadside environment.

spaces. Hence, both the drivers and PL owners gain economic benefits.

Large PLs, such as those at shopping malls and airports, are embedded with sensors, while private PLs, such as those at restaurants, stores, and bookstores, are equipped with surveillance cameras in order to forward parking statuses (parked, idle) to fog servers installed at these areas. For example, consider a scenario where there are three PLs labeled PLs 1–3, as shown in Fig. 2. Each PL sends its available spaces to its fog server: one free space from PL1, three free spaces from PL 2, and two free spaces from PL 3.

2) Fog Level: The term "fog computing" was first proposed by Cisco in 2012 [20], [21]. Fog computing is an extension of the cloud-based Internet obtained by introducing an intermediate layer between mobile devices and the cloud, aiming at smooth, low-latency service delivery from the cloud to mobile devices. The intermediate fog layer is composed of geodistributed fog servers that are deployed at the edge of the networks, e.g., banks, bus terminals, and shopping malls. Each fog server is a highly virtualized computing system, similar to a lightweight cloud server, and is equipped with onboard large-volume data storage, computers, and wireless communication facilities [22].

In our model, fog servers play the role of bridges for the exchange of parking information between PLs and RSUs. On the one hand, fog servers directly communicate with the sensors through a single-hop wireless connection, such as Bluetooth, WiFi, and Zigbee to gather the status of parking spots, which are either available or unavailable. On the other hand, the fog servers can be connected to the roadside cloud in order to transfer data regarding open spots. As shown in Fig. 2, the fog servers of PLs 1–3 forward the gathered information from the three PLs to RSUs 1 and 2.

3) RSUs: RSUs broadcast necessary information to drivers. Vehicles connect to an RSU in their coverage range to find the best parking place. RSUs need not only to keep track

of the limited vacant spots in such a way that the number of accepted cars does not exceed the availability of each PL but also to minimize the total expenditure of all vehicles in the system. RSUs thus act as the controllers of our model. Parking assignments need to be designed such that RSUs can carry out their duties mentioned earlier. Considering RSU 1, as shown in Fig. 2, we suppose that vehicle A is closer to PL 1 than vehicle B. The desired PL for both A and B sent to RSU 1 is PL 2. However, the parking status sent to RSU 1 by the fog servers states that there is only one free spot at PL 1. Therefore, the controller, RSU 1, transmits this assignment parking information to A and B such that A should park at PL 1, while B should park at PL 2.

4) Roadside Clouds: In Fig. 1, the roadside cloud plays a role as an intermediate layer among the RSUs to exchange PL information. When an RSU receives a required destination that is not within its area, the initial RSU sends this request to the roadside cloud. The roadside cloud then finds another RSU in the area of the required destination, and the process of finding a parking space is performed at this destination RSU. Afterward, the destination RSU sends the parking result back to the initial RSU through the roadside cloud. The drivers now can make parking requests at any time, from anywhere, through the roadside cloud. Consider an example shown in Fig. 2, and vehicle C sends the parking request to the RSU that is within its communication range, i.e., RSU 2. The desired PL of C is PL 1. However, PL 1 is not within the range of RSU 2, so RSU 2 helps C by forwarding this parking request to RSU 1, managing the parking status of PL1 through the roadside cloud. After receiving C's request, RSU 1 assigns C to PL 2 because PL 1 has no available parking spaces.

B. Problem Statement

We denote a set of vehicles by $\mathcal{V} = \{1, 2, \ldots, |\mathcal{V}|\}$ and a set of PLs by $\mathcal{L} = \{1, 2, \ldots, |\mathcal{L}|\}$. For any PL $j \in \mathcal{L}$, the number of vehicles that each PL j can serve is given by its quota, denoted by q_j. On the other hand, each vehicle $i \in \mathcal{V}$ has to spend its money on finding a parking space. This cost comprises both moving and parking cost.

1) Moving Cost: Without loss of generality, we assume that the vehicle and PL locations are given. For each vehicle, there are two types of location considered: the current location and the desired destination location. Let us use d_i^c and d_i^t to express the current location and the desired destination of vehicle i, respectively. For each PL, we use d_j to express the location of PL j. It is noticed that both vehicle and PL locations are predefined in the 2-D Euclidean space, $\mathbb{R}^2$. To reach a destination, each vehicle i first needs to move from its current location d_i^c to PL j at location d_j, inducing a traveling cost. Next, the owner of a vehicle i continually moves from PL j to its desired target d_i^t, inducing a walking cost. Therefore, the moving cost m_{ij} is determined as an aggregate of the traveling and walking costs defined as follows:

$$m_{ij} = \alpha \|d_i^c - d_j\| + \beta \|d_j - d_i^t\|, \quad \forall i \in \mathcal{V},\ j \in \mathcal{L}. \tag{1}$$

Here, the values of α and β are the monetary weights that are used to convert different units of the traveling costs and walking costs in the moving cost calculation.

2) Parking Cost: In detail, the parking cost is a function based on two parameters: parking price per time unit and parking duration. Without loss of generality, we assume that the parking prices of all spots in one PL are the same. Parking duration is the length of the time that a vehicle is parked at a PL. Thus, parking cost p_{ij} is determined as follows:

$$p_{ij} = \gamma_j t_i, \quad \forall i \in \mathcal{V},\ j \in \mathcal{L} \tag{2}$$

where γ_j is the parking price of PL j and t_i is the estimated parking duration of a vehicle i.

Combining the two cost models mentioned earlier gives the following vehicle assignment cost c_{ij} for vehicle i assigned to PL j:

$$c_{ij} = \theta_i m_{ij} + (1 - \theta_i) p_{ij}, \quad \forall i \in \mathcal{V},\ j \in \mathcal{L}. \tag{3}$$

Here, $\theta_i \in [0, 1]$ is a tradeoff parameter between the parking and moving costs. The system can judge the value of θ based on its concern for each cost, parking cost, and moving cost. For instance, θ_i is assigned to 0.5 if the system considers that these costs have an equal influence on the vehicle assignment cost, θ_i is greater than 0.5 if the system focuses more on the moving cost when observing the vehicle assignment cost, and θ_i is smaller than 0.5 if the system wants to give higher priority to the parking cost in valuing the vehicle assignment cost. Let $\mathcal{C}$ be a $|\mathcal{V}| \times |\mathcal{L}|$ total cost matrix such that each element c_{ij} is the vehicle assignment cost defined by (3).

From now, we use parking expenses in order to refer to the sum of the moving cost and parking cost. The parking assignment problem is to assign vehicles to parking spaces in such a way that the total parking expenses of the whole system are minimized [4], [5], [7].

First, we focus on minimizing the parking expenses, i.e., the moving and parking costs, of all vehicles by assigning each vehicle to the most suitable PL. The parking expense of PL j for all parked vehicles is

$$\sum_{i \in \mathcal{V}} c_{ij} x_{ij}. \tag{4}$$

Here, the binary variable x_{ij} indicates whether a vehicle i is assigned to PL j or not

$$x_{ij} = \begin{cases} 1, & \text{if vehicle } i \text{ is assigned at PL } j \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Then, the minimizing parking expenses problem can be stated as follows:

$$\textbf{P1: } \min \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{L}} c_{ij} x_{ij} \tag{6a}$$

$$\text{s.t. } \sum_{j \in \mathcal{L}} x_{ij} = 1, \quad \forall i \in \mathcal{V} \tag{6b}$$

$$\sum_{i \in \mathcal{V}} x_{ij} \leqslant q_j, \quad \forall j \in \mathcal{L} \tag{6c}$$

$$x_{ij} = \{0, 1\}, \quad \forall i \in \mathcal{V},\ j \in \mathcal{L}. \tag{6d}$$

In **P1**, constraint (6b) ensures that each vehicle can be parked at most one PL; additionally, constraint 6c ensures that each PL provides parking service up to a predefined quota. Since the value of x_{ij} is equal to 0 or 1, problem **P1** is an MILP, which is difficult to solve for realistic settings with numerous vehicle owners and parking slots [16]. Therefore, we propose two novel and practical algorithms using the matching theory and the ADMM approach that are suitable for a large set of vehicles and parking slots.

III. Matching Theoretic Approach for Minimizing Parking Expenses

Typically, the parking assignment **P1** can be solved using the MILP solvers. However, the matching of vehicle owners with PLs found by MILP solvers is conducted by considering the overall system efficiency. However, in practice, each vehicle owner and PL may have their own preferences from their own perspectives. For example, a vehicle owner may want to park as close as possible to her/his destination. Similarly, a PL may prefer providing parking service to vehicle owners that will park for a long time to make more profit. Thus, a design without considering individual vehicle owner and PL preferences may not be attractive to users, and unstable conditions may appear (i.e., vehicle owner and PL pairs preferring each other over their assigned partners). Beside using MILP solvers, from a practical point of view (e.g., solving time, a large number of users), a relaxation approach is commonly utilized to solve MILPs similar to **P1** by relaxing the integrality constraint (6d). The results of this relaxation then need to be converted again to a binary value of 0 or 1. To address these concerns, we thus use the matching theory to solve **P1** rather than a heuristic or metaheuristic algorithm. The stable matching process is used in order to consider user preferences. Furthermore, our solutions based on the matching theory will tackle the objectives and constraints of **P1** directly.

A. Matching Theory Preliminaries

A matching game framework efficiently finds a match between two separate sets of players [14], [15]. In our problem, there are two sets of players: vehicles and PLs. Each vehicle can be parked at most at one PL illustrated in constraint (6b), and each PL can provide service to many vehicles but only up to its available resources shown in constraint (6c). These concepts are matched to the core concepts of matching theory. Theoretically, the matching game tries to discover a pairwise matching between two sets of players that satisfy their mutual satisfaction. First, each player ranks one another using well-defined preference relations. Based on these preferred partners, an algorithm derived from the matching game provides stable matching as a result. This stable matching is as a map presenting the matched and unmatched partners of each player, which is equivalent to assigning a binary value of 0 or 1 for x_{ij}, as shown in (6d) that is why we can solve **P1** without any relaxing process. The meaning of the stable matching is the best association between two sets of players, which guarantees that no player can increase his own satisfaction by changing to another matching that differs from the original stable matching.

We formulate our problem as a many-to-one matching game in which a set of vehicles $\mathcal{V}$ will be assigned to a set of PLs $\mathcal{L}$, where each vehicle will be assigned to at most one PL. The concept of preferences is used to model the common and conflicting interests. Each vehicle would prefer a PL that is nearby its destination as well as its current position while providing a suitable price. The PL, however, only concerns with its revenue. Therefore, a PL prefers vehicles that have long parking times and will sell spots with the highest price first. Without loss of generality, we assume that each vehicle sends a parking request with its estimated duration parking time. Furthermore, we assume that vehicles and PLs trust each other. Pertaining to the details of our solution, basic definitions and notations are defined next.

The assignment of vehicles in $\mathcal{V}$ to PLs in $\mathcal{L}$ can be considered as an outcome of a many-to-one matching.

Definition 1: A emphmatching μ is defined as a function from the set $\mathcal{V}\cup\mathcal{L}$ to the set $\mathcal{V}\cup\mathcal{L}$ such that the following holds.

1) $|\mu(i)| \leq 1$ for each vehicle and $\mu(i) \in \mathcal{L} \cup \emptyset$.
2) $|\mu(j)| \leq q_j$ for $\mathcal{L}$ and $|\mu(j)| \in \mathcal{V} \cup \emptyset$.
3) $i \in \mu(j)$ if only if $\mu(i) = j$.

The definition states that matching is a many-to-one relation in the sense that each vehicle is assigned to a PL. Before setting an assignment of vehicles to PLs, each player needs to specify its preferences regarding the opposite set. The preference profiles built by the vehicles and PLs are denoted by $\mathcal{P}(i)$ and $\mathcal{P}(j)$, respectively. Let the tuple $(\mathcal{V}, \mathcal{L}, \succ_{\mathcal{V}}, \succ_{\mathcal{L}})$ be a many-to-one matching design. Here, $\succ_{\mathcal{V}} \triangleq \{\succ_i\}_{i\in\mathcal{V}}$ and $\succ_{\mathcal{L}} \triangleq \{\succ_j\}_{j\in\mathcal{L}}$ represent the set of preference relations of the vehicles and PLs, respectively.

1) Vehicles' Preference List: Each vehicle $i \in \mathcal{V}$ has a strict, transitive, and complete preference relation $\mathcal{P}(i)$ over $\mathcal{L}$. Here, $j \succ_i j'$ means that vehicle i prefers PL j to PL j'.

2) PLs' Preference List: Similar to the vehicles, each PL has a preference list of overall vehicles. Each PL $j \in \mathcal{L}$ also has a strict, transitive, and complete preference relation $\mathcal{P}(j)$ over the vehicle set $\mathcal{V}$. Then, $i \succ_j i'$ means that the PL j prefers vehicle i to vehicle i'. Next, based on [14], we design a parking assignment mechanism with the following definitions.

Definition 2: A matching μ is blocked by a pair of agents $\{(i, j)|i \in \mathcal{V}, j \in \mathcal{L}\}$ if there exists a pair (i, j) with $i \notin \mu(j)$ and $j \notin \mu(i)$ such that $i \succ_j \mu(j)$ and $j \succ_i \mu(i)$. Generally, such a pair is called a blocking pair.

Definition 3: A matching μ is stable if only if there is no blocking pair.

B. Matching-Based Parking Assignment

As we stated earlier, the matching outcome is determined based on the preference of vehicles' owners and PLs. Therefore, preference construction is a critical step when designing a parking assignment mechanism based on a matching game.

1) Constructing Vehicles' Preference List: From the vehicles' side, each vehicle i seeks a parking space to minimize the spending price. It means that vehicle i prefers to match

Algorithm 1 Vehicles' Preference List Algorithm

Input: All PLs in $\mathcal{L}$.
Output: $\mathcal{P}(i), \forall i \in \mathcal{V}$.
for *All vehicles* $i \in \mathcal{V}$ **do**
    1) Rank all PLs in ascending order based on the spending cost of vehicle i that is calculated by (3).
    2) Add the ranking list to the preference list of vehicle i.
end

Algorithm 2 PL Preference List Algorithm

Input: All vehicles in $\mathcal{V}$.
Output: $\mathcal{P}(j), \forall j \in \mathcal{L}$.
for *All PLs* $j \in \mathcal{L}$ **do**
    1) Rank all vehicles in decreasing order based on the parking cost calculated by (2).
    2) **if** **len**(the ranking list) $\leq q_j$ **then**
        Add the ranking list into the preference list of PL j.
      else
        Add the first $|q_j|$ vehicles i in the ranking list into the preference list of j.
end

with a PL such that the spending cost calculated by (3) is the smallest. We propose Algorithm 1 to build the preference list for each vehicle.

2) Constructing PLs' Preference List: From the PLs' side, each PL j always likes serving vehicles, which provides the highest incentive. Thus, they only consider the parking duration of each vehicle i and the available parking space. The construction of the PL preference list is described in Algorithm 2. In detail, the controller first ranks all vehicles in decreasing order based on the parking duration in (2). Next, it selects a subset of vehicles up to the available quota of PL j. Finally, these chosen vehicles are added to the preference list of PL j following the ranking order in the previous step.

The matching-based parking assignment problem is described in Algorithm 3. As mentioned earlier, the parking assignment problem is formulated as a many-to-one matching problem. Therefore, our goal is to find a stable matching, which is an optimal result by using a matching game. Gale [15] seeks a stable matching by implementing a deferred-acceptance algorithm. First, each vehicle and PL ranks their preferred partners by following Algorithms 1 and 2. Second, each vehicle i proposes its highest ranked PL, as presented in line 5 of Algorithm 3. Next, each PL j rejects all proposals it receives except for the $|q_j|$ vehicles having the highest ranked among them as shown by lines 6–11 of Algorithm 3. However, PL j will reject the current proposal to accept a new proposal if it prefers this new proposal than the current one, as indicated by lines 21–26 of Algorithm 3. Finally, the algorithm is iterated until all vehicles have a satisfactory proposal, for which a stable set of the parking assignment problem is found. With regard to the time complexity, the deferred acceptance

Algorithm 3 Matching-Based Parking Assignment

1: **Input:** $\mathcal{P}(i), \mathcal{P}(j)$, $\mathcal{Q} = [q_j]_{1\text{x}\mathcal{L}}$, $\forall i, j$.
2: **Output:** a matching μ.
3: **initialize:** Acceptance matrix $\mathcal{X} = [0]_{\mathcal{V}\text{x}\mathcal{L}}$,
    Temporary rejected list $\mathcal{R} = \emptyset$.
4: **for** $i \in \mathcal{V}$ **do**
5: $j \leftarrow \mathcal{P}(i) \setminus j$, j is the most preferred in $\mathcal{P}(i)$.
6: **if** $q_j > 0$ **then**
7: $\mathcal{X}[i, j] = 1$.
8: $q_j = q_j - 1$.
9: **else**
10: $\mathcal{R} = \mathcal{R} \cup i$.
11: $\mathcal{P}(i) = \mathcal{P}(i) \setminus j$.
12: **While** $\mathcal{R} \neq \emptyset$ or $\mathcal{Q} \neq [0]_{\mathcal{V}\text{x}\mathcal{L}}$ **do**
13: $i \leftarrow \mathcal{R} \setminus i$.
14: $j \leftarrow \mathcal{P}(i) \setminus j$, j is the most preferred in $\mathcal{P}(i)$.
15: **if** $q_j > 0$ **then**
16: $\mathcal{X}[i, j] = 1$.
17: $q_j = q_j - 1$.
18: **else**
19: $i' \leftarrow i$.
20: $\mathcal{K}_j = \{k \mid \mathcal{X}[k, j] = 1$ and $i' \succ_j k\}$.
21: **for** $k \in \mathcal{K}_j$ **do**
22: $\mathcal{R} = \mathcal{R} \cup k$.
23: $\mathcal{X}[k, j] = 0$.
24: $q_j = q_j + 1$.
25: $\mathcal{P}(j) = \mathcal{P}(j) \setminus k$.
26: $\mathcal{P}(k) = \mathcal{P}(k) \setminus j$.
27: Return a matching $\mu \leftarrow \mathcal{X}$
28: **end**

procedure in the general many-to-one case has a reasonable complexity depending on several settings [24]. For our parking assignment algorithm, the complexity lies in the number of accepting/rejecting decisions required to obtain a stable matching λ. In each iteration of Algorithm 3, a vehicle proposes to the PL to use the most preferred PL in its current preference list, and the PL then accepts or rejects the proposal. The maximum size of a vehicle's preference list is $|\mathcal{L}|$. Therefore, in any instance of a matching problem, Algorithm 3 converges into a stable matching μ within $O(|\mathcal{V}|*|\mathcal{L}|)$ iterations, where $|\mathcal{V}|$ is the number of vehicles and $|\mathcal{L}|$ is the number of PLs [25].

IV. ADMM-Based Approach for Minimizing Parking Expenses and Balancing Parking Demand

Although the parking assignment problem **P1** can be solved efficiently using a matching theory as described in Algorithm 3, **P1** does not capture a balanced parking demand, which is a critical factor as discussed earlier. Therefore, we need to extend **P1** to account for balancing parking demand among the PLs. The term "balancing parking demand" refers to considering PL utilization to avoid local congestion when some PLs have a high demand, while others are underutilized. If each vehicle i only cares about its own parking problem

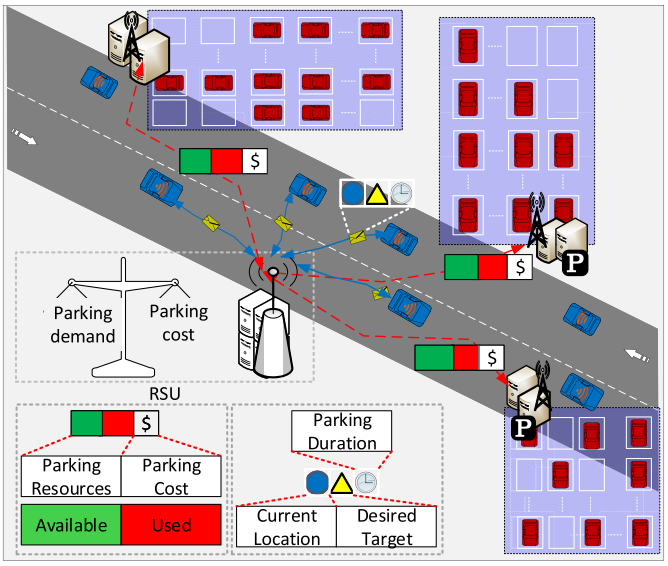

Fig. 3. Minimizing parking expenses and balancing parking demand among multiple PLs.

and considers only the minimum parking expenses c_{ij}, then this can lead to a vast number of vehicles making their way to the same single PL. Consequently, some PLs might have a long queue, while other PLs are underutilized, which causes localized congestion and pollution peaks. Therefore, we need to consider both parking expenses and balanced parking demand factor when assigning vehicles to one of the multiple PLs. The concept of minimizing parking expenses and balancing the parking demand for the parking assignment problem is shown in Fig. 3.

The total cost of the parking assignment of our model is now associated with two primary costs: 1) the parking expenses and 2) balancing the parking demand among multiple PLs. From now on, the first cost is referred to as the individual parking cost, while the second cost is the social cost. Thus, the social cost is defined as a value that captures the rate of parking utilization at each PL. We observe the second cost as the social cost because our goal is to use this cost to measure the influence of parking assignments on the city.

A. Problem Statement

To tackle the parking assignment problem by considering both individual parking costs and social costs, we introduce a current load, z_j, variable. This variable is used to determine how many vehicles are presently assigned to PL j. It is then calculated as follows:

$$z_j = \sum_{i \in \mathcal{V}} x_{ij}. \tag{7}$$

The utilization of PL j is defined based on the value of the current load z_j as follows:

$$\frac{z_j}{q_j}, \quad \forall j \in \mathcal{L}. \tag{8}$$

Here, q_j is kept track in a quota matrix $\mathcal{Q}_{1\times|\mathcal{L}|}$.

With the definitions from (7) and (8), the original parking assignment problem **P1** is refined as follows:

$$\textbf{P2: } \min \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{L}} \delta c_{ij} x_{ij} + \sum_{j \in \mathcal{L}} (1-\delta) \frac{z_j}{q_j} \tag{9a}$$

$$\text{s.t.} \sum_{j \in \mathcal{L}} x_{ij} = 1, \quad \forall i \in \mathcal{V} \tag{9b}$$

$$\sum_{i \in \mathcal{V}} x_{ij} = z_j, \quad \forall j \in \mathcal{L} \tag{9c}$$

$$0 \leqslant z_j \leqslant q_j, \quad \forall j \in \mathcal{L} \tag{9d}$$

$$x_{ij} = \{0, 1\}, \quad \forall i \in \mathcal{V}, \ j \in \mathcal{L}. \tag{9e}$$

Here, δ is a weight factor used to combine both the costs. The value of δ is $\in [0, 1]$.

Constraint (9b) expresses that no vehicle can be assigned to more than one PL. Without loss of generality, we assume that in the current time slot, the system has sufficient spaces to satisfy the parking demands. The parking capacity of PL j is given in (9d). Thus, the tolerance assignment metric cannot exceed its quota. Problem **P2** is an MILP, which is difficult to solve for a practical setting with large sets of vehicles and PLs [16]. To use the matching game approach presented in Section III, the preference lists of vehicles and PLs are bound to be predefined. Meanwhile, in problem **P2**, the value of the current load z_j dynamically changes based on the value of matching control variable x_{ij}. We thus cannot previously determine the preference list for vehicles and PLs. The matching approach is no longer suitable for solving the extend parking assignment **P2**. We thus need another approach to solve problem **P2** efficiently. In this section, we develop an effective algorithm to solve the problem using the ADMM. ADMM is known as an efficient method to solve convex optimization problems with convergence guarantee [26].

B. ADMM-Based Parking Assignment

To develop a practical distributed algorithm for the parking assignment problem, we first transform the original problem into a convex problem by relaxing the PL assignment indicators. In this case, we relax x_{ij} to be a continuous real variable in the range [0,1]. Now, problem (P1) can be converted into

$$\textbf{P3: } \min \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{L}} \delta c_{ij} x_{ij} + \sum_{j \in \mathcal{L}} (1-\delta) \frac{z_j}{q_j} \tag{10a}$$

$$\text{s.t.} \sum_{j \in \mathcal{L}} x_{ij} = 1, \quad \forall i \in \mathcal{V} \tag{10b}$$

$$\sum_{i \in \mathcal{V}} x_{ij} = z_j, \quad \forall j \in \mathcal{L} \tag{10c}$$

$$0 \leqslant z_j \leqslant q_j, \quad \forall j \in \mathcal{L} \tag{10d}$$

$$0 \leqslant x_{ij} \leqslant 1, \quad \forall i \in \mathcal{V}, \ j \in \mathcal{L}. \tag{10e}$$

Problem **P3** is a linear problem now; we, therefore, can effectively solve this problem by using commercial software packages. However, the matching operation of vehicles and PLs in **P3** is coupled together by (10c). Therefore, the individual parking cost depending on variable x_{ij} and the social

cost depending on variable z_j cannot be optimized separately. Meanwhile, our objective is to solve the parking assignment problem considering both individual parking cost and social cost in a distributed way. Since **P3** is a convex problem with a coupled linear equality constraint (10c), it fits into the ADMM framework [26]. Hence, we employ the ADMM method to tackle the problem **P3** in a way that not only minimizes both individual and social costs but also enables these costs to be optimized separately.

Toward this end, let us define the augmented Lagrangian associated with **P3** as follows:

$$L_p = \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{L}} \delta c_{ij}x_{ij} + \sum_{j\in\mathcal{L}}(1-\delta)\frac{z_j}{q_j} + \sum_{j\in\mathcal{L}}\lambda_j\left(\sum_{i\in\mathcal{V}} x_{ij} - z_j\right) + \sum_{j\in\mathcal{L}}\frac{\rho}{2}\left(\sum_{i\in\mathcal{V}} x_{ij} - z_j\right)^2 \quad (11)$$

where λ_j is the Lagrange multiplier corresponding to the coupled constraint $\sum_{i\in\mathcal{V}} x_{ij} = z_j$ and ρ is a scalar parameter. This augmented Lagrangian is then equivalently rewritten as

$$L_p = \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{L}} \delta c_{ij}x_{ij} + \sum_{j\in\mathcal{L}}\sum_{i\in\mathcal{V}}\lambda_j x_{ij} + \sum_{j\in\mathcal{L}}\frac{\rho}{2}\left(\sum_{i\in\mathcal{V}} x_{ij} - z_j\right)^2 + \sum_{j\in\mathcal{L}}(1-\delta)\frac{z_j}{q_j} - \sum_{j\in\mathcal{L}}\lambda_j z_j. \quad (12)$$

Obviously, there are three types of variables involved in procedure of our proposal, $x = [x_{ij}]_{|\mathcal{V}|\times|\mathcal{L}|}$, $z = [z_j]_{1\times|\mathcal{L}|}$, and $\lambda = [\lambda_j]_{1\times|\mathcal{L}|}$. By employing the ADMM method [26], the sequential minimization of x and z can be separately solved followed by the Lagrange multiplier λ update. The ADMM-based optimization problem **P3** can be obtained via the following iteration procedure.

x-minimization: For each iteration t, given the announced dual variable $\lambda^{(t)}$ and previously achieved $z^{(t)}$, a controller will solve the following problem to obtain the value of $x^{(t+1)}$:

$$\begin{aligned}\min_{x_{ij}}\ & \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{L}} \delta c_{ij}x_{ij} + \sum_{j\in\mathcal{L}}\sum_{i\in\mathcal{V}}\lambda_j^{(t)}x_{ij} + \sum_{j\in\mathcal{L}}\frac{\rho}{2}\left(\sum_{i\in\mathcal{V}} x_{ij} - z_j^{(t)}\right)^2 \\ \text{s.t.}\ & \sum_{j\in\mathcal{L}} x_{ij} = 1, \quad \forall i \in \mathcal{V} \\ & 0 \leqslant x_{ij} \leqslant 1, \quad \forall i \in \mathcal{V},\ j \in \mathcal{L}. \end{aligned} \quad (13)$$

z-minimization: Next, with the values $x^{(t+1)}$ and $\lambda^{(t)}$, each PL j can solve it owns problem based on the following equation:

$$\begin{aligned}\min_{z_j}\ & (1-\delta)\frac{z_j}{q_j} - \lambda_j^{(t)}z_j + \frac{\rho}{2}\left(\sum_{i\in\mathcal{V}} x_{ij}^{(t+1)} - z_j\right)^2 \\ \text{s.t.}\ & 0 \leqslant z_j \leqslant q_j, \forall j \in \mathcal{L}. \end{aligned} \quad (14)$$

Algorithm 4 ADMM Algorithm for Distributed Parking Assignment Problem (ADMM-PA)

1: **initialize:** $t = 0$, $\mathcal{C}$, $\mathcal{Q}$, $x^{(0)}$, $\lambda^{(0)}$, ρ.
2: **repeat**
3: $t \leftarrow t + 1$
4: *** At the controler:**
5: Collects $z^{(t)}$ from PLs.
6: Update $x^{(t+1)}$ according to (13);
7: Sends $x^{(t+1)}$, $\lambda_j^{(t)}$ to PLs.
8: *** At PL** j„ $\forall\ j \in \mathcal{L}$, **do:**
9: Collects $x^{(t+1)}$, $\lambda_j^{(t)}$ from the controller.
10: Update $z^{(t+1)}$:
11: $z_j^{(t+1)} = \sum_{i\in\mathcal{V}} x_{ij}^{(t+1)} + \frac{1}{\rho}(\lambda_j^{(t)} - \frac{1}{q_j})$;
12: Sends $z^{(t+1)}$ to the controler.
13: The controller updates Lagrange multiplier λ as follows:
14: Update $\lambda_j^{(t+1)} = \lambda_j^{(t)} + \rho\left(\sum_{i\in\mathcal{V}} x_{ij}^{(t+1)} - z_j^{(t+1)}\right)$.
15: **until** the stop criteria are met.

Dual update: During each iteration, the controller updates the dual variable λ as follows:

$$\lambda_j^{(t+1)} = \lambda_j^{(t)} + \rho\left(\sum_{i\in\mathcal{V}} x_{ij}^{(t+1)} - z_j^{(t+1)}\right). \quad (15)$$

Note that the x-minimization problem is convex and can thus be efficiently solved with standard convex optimization techniques. In the case of the z-minimization problem, each PL j can obtain the value of the variable z by directly determining $z's$ closed form in a distributed way. According to the derivation for z, the value of the variable z for PL j can be derived as $z_j^{(t+1)} = \sum_{i\in\mathcal{V}} x_{ij}^{(t+1)} + (1/\rho)(\lambda_j^{(t)} - (1/q_j))$.

C. Algorithm Design

Based on the abovementioned analysis, we propose Algorithm 4, which provides the procedures of the parking assignment in a decentralized way. At the beginning ($t = 0$), we initialize the value of matching control variable x and the Lagrange multiplier λ (line 1). For example, we can assign $x = [O]_{|V|\times|L|}$ and $\lambda = [O]_{|L|}$. These values are transmitted to the controller. After that, at each iteration t, the controller gathers the current load z sent by PLs (line 5). It then computes the next iteration value of the matching control variable x (line 6). The controller broadcasts the next iteration value of x and the current value of λ to PLs after that (line 7). After receiving this broadcast information, each PL updates its next iteration load metric z_j and sends it back to the controller (lines 8–12). As stated earlier, each PL updates the value of $z_j^{(t+1)}$ in a distributed way by computing the closed-form solution of problem (14) $z_j^{(t+1)} = \sum_{i\in\mathcal{V}} x_{ij}^{(t+1)} + (1/\rho)(\lambda_j^{(t)} - (1/q_j))$. Finally, the controller calculates the next iteration value of Lagrange multiplier λ (line 14). This process is repeated until the stop criteria are met (line 15). Generally, the stop criteria can be either exceeding the predefined number of iterations or obtaining the stable value of matching control variable x. It is noted that the value of x obtained from Algorithm 4 belongs to [0, 1]. We thus utilize a projection technique

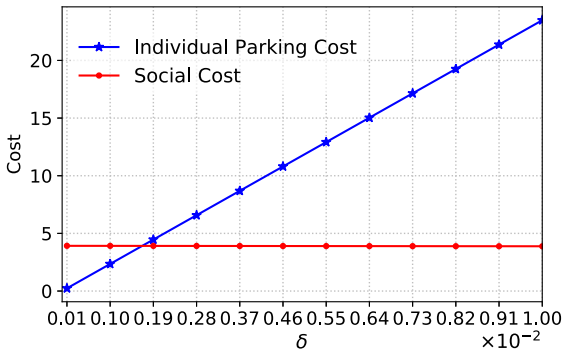

Fig. 4. Impact of weight parameter δ on the cost.

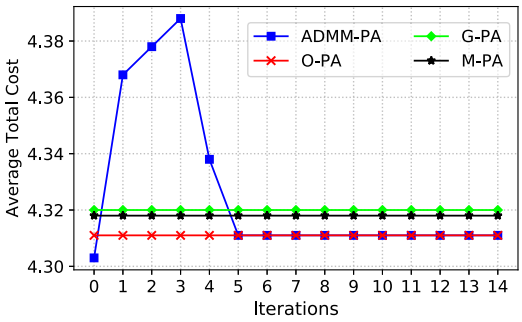

Fig. 5. Comparison of average total cost between ADMM-PA with M-PA, O-PA, and G-PA.

to recover the value of x to a value in $\{0, 1\}$, as defined in (5). Denote the set of binary numbers by $\mathbb{B} = \{0, 1\}$. The projection of the real variable x_{ij} onto the set $\mathbb{B}$ is given as $\Pi_{\mathbb{B}}(x_{ij}) = \arg\min_{b\in\mathbb{B}} \|x_{ij} - b\|$. The ADMM-based parking assignment problem is guaranteed to be convergent. The convergence can be proved using the gradient-based standard technique in [16], and the convergence rate is $\mathcal{O}(1/t)$ as analyzed in [27].

V. Numerical Results

We consider a system with 4 PLs, and 1000 vehicles distributed over a city of 2000 m × 2000 m in size. We design a gridlike road network [8]. Vehicles then stay parked for an exponentially distributed time with a mean of 20 min. All the algorithms in this article are implemented in Python 3.7 on an Intel Core i5-4670 TM 3.4-GHz CPU, 16-GB RAM PC.

For choosing the value of the weight factor δ, we compare the individual parking cost with social costs by varying the weight factor δ, as shown in Fig. 4. We observe that the individual parking cost is higher than social cost when $\delta > 1.66 \times 10^{-4}$ and vice versa. Therefore, choosing the value of δ depends on the objective priority. Specifically, when $\delta > 1.66 \times 10^{-4}$, we will have the individual parking cost that significantly dominates the social cost and vice versa, as shown in Fig. 4. Since our goal is to have a balance between individual and social costs, we thus assign the value of δ to 1.66×10^{-4}.

A. Performance on Total Cost

According to the settings mentioned earlier, we now evaluate the performance on total cost, the sum of individual parking cost and social cost, under different algorithms:

1) The first one is the optimal-based parking assignment, denoted as O-PA. The result of O-PA is found by implementing the problem (**P2**). As we mentioned earlier, (**P2**) is the MILP, and we thus need a solver to solve this problem. In this article, we use the GUROBI optimizer [28] to obtain the optimal solution. As using GUROBI solvers, the result of O-PA is conducted by considering the overall system efficiency.
2) The second one is the ADMM-based parking assignment demonstrated in Algorithm 4 and denoted as ADMM-PA. ADMM-PA takes both individual parking cost and social cost into consideration.
3) The third one is the matching-based parking assignment derived in Algorithm 3 and denoted as M-PA. M-PA is proposed to solve the original parking assignment defined in (**P1**). Unlike ADMM-PA, M-PA only focuses on the individual parking cost.
4) The last one is the greedy-based parking assignment denoted as G-PA. G-PA does not tackle either individual parking cost or social cost. G-PA sequentially assigns parking spaces to the nearest vehicles in each PL until the quota constraint is violated.

Performance on total cost under all algorithms is obtained by averaging over a large number of independent simulation runs. In each simulation run, we randomly distributed locations of PLs and vehicles, and quotas of each PL. We evaluate the performance of the proposed ADMM-PA algorithm with $|\mathcal{V}| = 1000$. Fig. 5 shows the average total cost under different algorithms. As shown in Fig. 5, the average total cost obtained by O-PA is the lowest because it is the optimal results. The average total cost obtained by ADMM-PA fluctuates at first iterations. However, when the ADMM-PA is convergent, ADMM-PA closely approaches the optimal solution. Since the ADMM-PA algorithm distributively chooses the number of parking slots to serve the parking requests in a way that both minimizes the total cost of the whole system and balances the parking occupancy rate among PLs, ADMM-PA outperforms M-PA and G-PA. Although M-PA cannot take both the individual parking cost and the social cost as ADMM-PA into account, M-PA still produces better performance than G-PA.

B. Performance on Convergence and Computational Time

As ADMM-PA has shown an improvement in minimizing both individual parking cost and social cost compared with M-PA and G-PA, our next concern is the convergence rate and computational time of ADMM-PA. We execute Algorithm 4 to evaluate the convergence of the ADMM-PA algorithm with different weight factors δ and network sizes $|V| = 1000$. It is observed from Fig. 6(a) that ADMM-PA converges to the optimal solution after around eight iterations with $\delta = 1.66 \times 10^{-4}$, which is efficient for solving the parking problem in large cities with intense traffic congestion. Meanwhile, comparing to $\delta = 1.66 \times 10^{-3}$, the total cost approximately approaches to the optimal value, but there is a

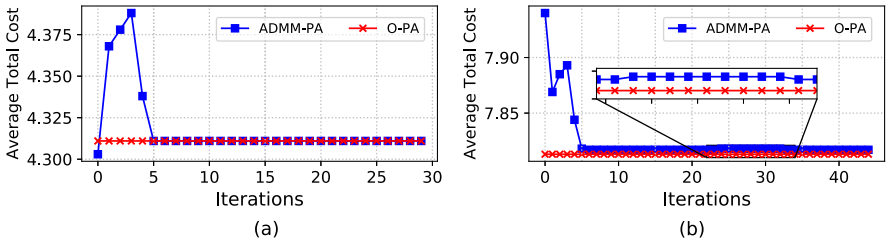

Fig. 6. Convergence of ADMM-PA algorithm with varying the weight factor δ and network size $|\mathcal{V}| = 1000$. (a) δ =1.66 $\times$ 10^{-4}. (b) δ =1.66 $\times$ 10^{-3}.

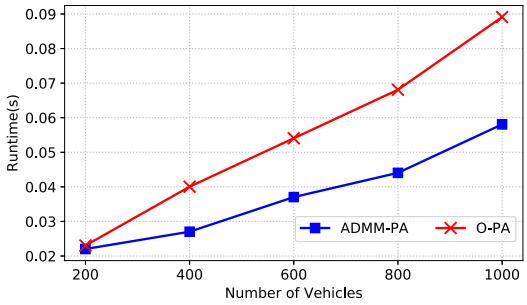

Fig. 7. Comparison of CPU time between ADMM-PA and O-PA.

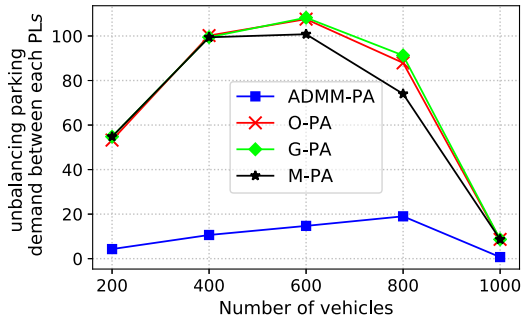

Fig. 8. Unbalancing parking cost among ADMM-PA, M-PA, O-PA, and G-PA.

gap between ADMM-PA and O-PA, around 10%, as shown in Fig. 6(b). The gap between ADMM-PA and O-PA is due to the recovery of the binary variable x. Let us recall that we have transformed the problem (P2) into the convex problem (P3) by relaxing the integrity of variable x. We thus need to recover the binary variable x after the convergence of the ADMM-PA process. However, as to be shown in Fig. 6, the gap is not significant. It means that the error of relaxing the integrity of variable x in ADMM-PA is negligible.

Next, we verify the computational efficiency of the ADMM-PA by measuring the run times according to the number of vehicles, as shown in Fig. 7, which shows that when $|\mathcal{V}| = 200$, the run times of O-PA and ADMM-PA are almost the same. However, ADMM-PA outperforms O-PA when $|\mathcal{V}| > 200$. In particular, when $|\mathcal{V}| = 1000$, ADMM-PA reduces the run times of O-PA by roughly 31%. In summary, when the number of vehicles increases, the system efficiency of ADMM-PA is higher than O-PA.

C. Performance on Parking Balancing

We further demonstrate the efficiency of the proposed ADMM-PA algorithm in balancing the parking demand among each PLs. The unbalanced parking cost is considered instead of the balanced parking demand parameter to illustrate the performance of ADMM-PA, as shown in Fig. 8. This value is calculated by $((1/4)\sum_{j=1}^{4}(z_j - (|\mathcal{V}|/4))^2)^{1/2}$ as derived from the concept of a population standard deviation. The low unbalanced parking cost illustrates that the total number of vehicles parked at each PL is very close to the average parking demand of the whole system. In Fig. 8, under all ranges of numbers of vehicles, ADMM-PA always outperforms the other algorithms. Let us recall that the result of O-PA is found by implementing the problem (**P2**). The matching operation of vehicles and PLs in O-PA are coupled together by (9c), and they, therefore, cannot be optimized separately. In other words, O-PA does not separately take the balancing parking demand into consideration, and it considers the overall system efficiency. On the contrary, ADMM-PA is proposed to solve the parking assignment problem by not only minimizing both individual and social costs but also enabling these costs to be optimized separately, which are the reasons why although O-PA achieves the best performance on total cost and convergence rate compared with other algorithms, as shown in Figs. 5 and 6, and O-PA reveals lower parking balancing efficiency than ADMM-PA. More specifically, it can be observed that ADMM-PA, M-PA, O-PA, and GA-PA achieve up to around 9.5%, 37%, 44%, and 45.5% of the unbalanced parking demand, respectively, for a network with 800 vehicles. Thus, ADMM-PA has the best performance regarding the parking balance. Additionally, O-PA attains a decrease in unbalanced parking demand of 1.5% compared with GA-PA, while ADMM-PA attains 27.5% and 36% higher parking utilization compared with M-PA and G-PA, respectively, for a network with 800 vehicles.

VI. Conclusion

In this article, the parking assignment problem to assign vehicles to the optimal PLs is studied. To utilize parking resources effectively, we investigate the parking assignment problem in which sharing parking spaces among multiple PLs is allowed. The parking assignment problem is formulated with two main objectives: minimizing individual parking costs and balancing parking demand. The matching-based parking assignment method is first derived to find a matching stable as the best assignment of vehicles to parking spaces with the smallest parking expenses. Next, we construct the mixed-integer optimization problem that solves both objectives. Since the constructed problem is an NP-hard problem, we study the ADMM method to solve the problem in a decentralized way. Numerical results have shown that our ADMM-based

algorithm can obtain fast convergence with nearly optimal results. Recently, we have witnessed an increasing number of electric vehicles driving on the road. Our balanced demand mechanism can then be extended to balance electricity overload among multiple charging stations, which we will consider in our future work.

REFERENCES

[1] *International Transport Forum—Transport Outlook 2012: Seamless Transport for Greener Growth*. Accessed: Feb. 17, 2019. [Online]. Available: https://www.itfoecd.org/transport-outlook-2012-seamless-transport-greener-growth

[2] S. Nawaz, C. Efstratiou, and C. Mascolo, "Parksense: A smartphone based sensing system for on-street parking," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 75–86.

[3] N. Taheri, J. Y. Yu, and R. Shorten, "A fair assignment of drivers to parking lots," Jul. 2015 *arXiv:1507.03504*. [Online]. Available: https://arxiv.org/abs/1507.03504

[4] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies," *IEEE Access*, vol. 3, pp. 1581–1591, 2015.

[5] Y. Xu, E. Alfonsetti, P. C. Weeraddana, and C. Fischione, "A semidistributed approach for the feasible min-max fair agent-assignment problem with privacy guarantees," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 333–344, Mar. 2018.

[6] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "iParker—A new smart car-parking system based on dynamic resource allocation and pricing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2637–2647, Sep. 2016.

[7] 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)*, Aug. 2015, pp. 321–326.

[8] A. Schlote, C. King, E. Crisostomi, and R. Shorten, "Delay-tolerant stochastic algorithms for parking space assignment," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1922–1935, Oct. 2014.

[9] L. Stenneth, O. Wolfson, B. Xu, and S. Y. Philip, "PhonePark: Street parking using mobile phones," in *Proc. IEEE 13th Int. Conf. Mobile Data Manage. (MDM)*, Jul . 2012, pp. 278–279.

[10] Y. Geng and C. G. Cassandras, "A new 'smart parking' system infrastructure and implementation," *Proc. Social Behav. Sci.*, vol. 54, pp. 1278–1287, Oct. 2012.

[11] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig, "Vanet-based approach for parking space availability," in *Proc. 14th Int. Workshop Veh. Ad Hoc Netw.*, Jan. 2007, pp. 75–76.

[12] R. Lu, X. Lin, H. Zhu, and X. Shen, "Spark: A new vanet-based smart parking scheme for large parking lots," in *Proc. INFOCOM*, Apr. 2009, pp. 1413–1421.

[13] H. Zhao, L. Lu, C. Song, and Y. Wu, "Ipark: Location-aware-based intelligent parking guidance over infrastructureless vanets," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 12, 2012, Art. no. 280515.

[14] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[15] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 120, no. 5, pp. 386–391, 2013.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[17] C. Yi, S. Huang, and J. Cai, "An incentive mechanism integrating joint power, channel and link management for social-aware D2D content sharing and proactive caching," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 789–802, Apr. 2018.

[18] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 912–925, Apr. 2018.

[19] X. T. R. Kong, S. X. Xu, M. Cheng, and G. Q. Huang, "IoT-enabled parking space sharing and allocation mechanisms," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1654–1664, Oct. 2018.

[20] "Fog computing, ecosystem, architecture and applications," Cisco, San Francisco, CA, USA, Cisco RFP-2013-078, 2013.

[21] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, Aug. 2012, pp. 13–16.

[22] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," Feb. 2015, *arXiv: 1502.01815*. [Online]. Available: https://arxiv.org/abs/1502.01815

[23] A. E. Roth, "Deferred acceptance algorithms: History, theory, practice, and open questions," *Int. J. Game Theory*, vol. 36, nos. 3–4, pp. 537–569, Mar. 2008.

[24] A. E. Roth and M. A. O. Sotomayor, *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis* (Econometric Society Monographs). Cambridge, U.K.: Cambridge Univ. Press, 1990.

[25] Y. Gao, Y. Wu, H. Hu, X. Chu, and J. Zhang, "Licensed and unlicensed bands allocation for cellular users: A matching-based approach," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 969–972, Jun. 2019.

[26] H. Choi, P. J. Seiler, and S. V. Dhople, "Propagating uncertainty in power flow with the alternating direction method of multipliers," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 4124–4133, Jul. 2018.

[27] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *J. Sci. Comput.*, vol. 66, no. 3, pp. 889–916, Mar. 2016.

[28] *Gurobi Optimization—The Best Mathematical Programming Solver*. Accessed: Feb. 17, 2019. [Online]. Available: http://www.gurobi.com/

Oanh Tran Thi Kim received the B.S. degree in pedagogy of information and the M.S. degree in computer science from Hue University, Hue, Vietnam, in 2008 and 2013, respectively. She is currently pursuing the Ph.D. degree with Kyung Hee University, Seoul, South Korea.

From 2008 to 2014, she was a Lecturer with the Hue Industrial College, Hue. Her research interest includes applying optimization techniques to cutting-edge transportation applications, such as smart parking lots, ride sharing, and smart electric vehicle charging.

Nguyen H. Tran (S'10–M'11–SM'18) received the B.S. degree from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2005, and the Ph.D. degree in electrical and computer engineering from Kyung Hee University, Seoul, South Korea, in 2011.

He was an Assistant Professor with the Department of Computer Science and Engineering, Kyung Hee University, from 2012 to 2017. Since 2018, he has been with the School of Computer Science, The University of Sydney, Sydney, NSW, USA, where he is currently a Senior Lecturer. His research interests include distributed computing, learning, and networks.

Dr. Tran received the Best KHU Thesis Award in Engineering in 2011 and several best paper awards, including the IEEE ICC 2016, Asia-Pacific Network Operations and Management Symposium (APNOMS) 2016, and IEEE ICCS 2016. He received the Korea NRF Funding for Basic Science and Research for the term 2016–2023. He has been the Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING since 2016.

Chuan Pham received the B.S. degree from the Ho Chi Minh City University of Transport, Ho Chi Minh City, Vietnam, in 2004, the master's degree from the Ho Chi Minh City University of Science, Ho Chi Minh City, in 2008, and the Ph.D. degree from Kyung Hee University, Seoul, South Korea, in 2017, all in electrical and computer engineering.

He has been a Post-Doctoral Fellow with the Department of Computer Science and Engineering, Kyung Hee University, since August 2017. Since January 2018, he has been a Post-Doctoral Fellow with Synchromedia, École de Technologie Supérieure, Université du Québec, Quebec City, QC, Canada. His research interests include applying analytic techniques of optimization and machine learning to network applications in terms of cloud and mobile-edge computing, datacenters, resource allocation for virtual networks, and the Internet of Things.

Tuan LeAnh received the B.Eng. and M.Eng. degrees in electronic and telecommunication engineering from the Hanoi University of Technology, Hanoi, Vietnam, in 2007 and 2010, respectively, and the Ph.D. degree from Kyung Hee University, Seoul, South Korea, in 2017.

He was a member of the Technical Staff of the Network Operations Center, Vietnam Telecom National Company, Vietnam Posts and Telecommunications Group (VNPT), Hanoi, from 2007 to 2011. Since March 2017, he has been a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include queuing, optimization, control, and game theory to design, analyze, and optimize the resource allocation in communication networks, including cognitive radio network, wireless network virtualization, ultra-reliable low-latency communication (URLLC), mobile edge computing, and vehicular communications.

My T. Thai (M'06) is currently a UF Research Foundation Professor with the Computer and Information Science and Engineering Department, University of Florida, Gainesville, FL, USA. The results of her work have led to six books and over 140 articles published in leading journals and conferences. Her current research interests focus on scalable algorithms, big data analysis, cybersecurity, and optimization in network science and engineering, including communication networks, smart grids, social networks, and their interdependency.

Dr. Thai received many research awards, including the UF Provosts Excellence Award for Assistant Professors, the UFRF Professorship Award, the Department of Defense (DoD) Young Investigator Award, and the National Science Foundation (NSF) CAREER Award. She received the IEEE MSN 2014 Best Paper Award and the 2017 IEEE ICDM Best Papers Award. She has been engaged in many professional activities. She has been a TPC Chair for many IEEE conferences. She has served as an Associate Editor for the *Journal of Combinatorial Optimization* (JOCO), the *Journal of Discrete Mathematics*, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING and a Series Editor for *Briefs in Optimization*. She is the Founding EiC of the *Computational Social Networks* journal.

Choong Seon Hong (S'95–M'07–SM'11) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Tokyo, Japan, in 1997.

In 1988, he joined KT, Gyeonggi-do, South Korea, where he was involved in broadband networks as a member of the Technical Staff. Since 1993, he has been with Keio University. He was with the Telecommunications Network Laboratory, KT, as a Senior Member of Technical Staff and as the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include future Internet, intelligent edge computing, network management, and network security.

Dr. Hong is a member of the Association for Computing Machinery (ACM), the Institute of Electronics, Information and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Korean Institute of Information Scientists and Engineers (KIISE), the Korean Institute of Communications and Information Sciences (KICS), the Korean Information Processing Society (KIPS), and the Open Standards and ICT Association (OSIA). He has served as the General Chair, the TPC Chair/Member, or an Organizing Committee Member of international conferences, such as the Network Operations and Management Symposium (NOMS), International Symposium on Integrated Network Management (IM), Asia-Pacific Network Operations and Management Symposium (APNOMS), End-to-End Monitoring Techniques and Services (E2EMON), IEEE Consumer Communications and Networking Conference (CCNC), Assurance in Distributed Systems and Networks (ADSN), International Conference on Parallel Processing (ICPP), Data Integration and Mining (DIM), World Conference on Information Security Applications (WISA), Broadband Convergence Network (BcN), Telecommunication Information Networking Architecture (TINA), International Symposium on Applications and the Internet (SAINT), and International Conference on Information Networking (ICOIN). He was an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and the IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He currently serves as an Associate Editor for the *International Journal of Network Management* and an Associate Technical Editor of the *IEEE Communications Magazine*.