

# Coordinated Autonomous Vehicle Parking for Vehicle-to-Grid Services

Albert Y.S. Lam, James J.Q. Yu, Yunhe Hou, and Victor O.K. Li

Department of Electrical and Electronic Engineering

The University of Hong Kong

Pokfulam Road, Hong Kong

Email: {ayslam, jqyu, yhou, vli}@eee.hku.hk

**Abstract**—Thanks to the various advantages over conventional cars, autonomous vehicles (AVs) will take a more important role in the future transportation system. Since AVs are typically electric vehicles (EVs), they can contribute to vehicle-to-grid (V2G) services. While it is generally not feasible to dictate EV routes, we can design AV travel plans to fulfill certain system-wide objectives. In this paper, we focus on the AVs looking for parking and study how they can be led to appropriate parking facilities to support V2G services. We formulate the problem as an integer linear program (ILP). It can be solved by a standard ILP solver but the required long computational time may not be acceptable in practical scenarios. To overcome this, we also develop a more efficient heuristic. We perform extensive simulations to study different perspectives on solving the problem. Simulation results reveal that the heuristic can significantly reduce the computational time with negligible degradation in objective function value. The quality of the optimal solution is insensitive to the number of available parking facilities. A coarser time scale can improve computational time but degrade the solution quality resulting in possible infeasible solution.

## I. INTRODUCTION

Due to various green initiatives, electric vehicles (EVs) are getting popular and it is not hard to foresee that they will take a substantial share in ground transport in the future. EVs are closely related to the power system as they largely power up their batteries from the grid. Moreover, they can be regarded as “movable energy storage” and constitute a significant energy repository. This repository is flexible in the sense that no fixed large battery needs to be installed and its capacity can be adjusted to suit the needs. This architecture is generally called vehicle-to-grid (V2G) and it can be utilized to support the grid with various demand response and auxiliary services. A V2G system is generally associated with a parking facility where a large number of EVs can stay and contribute their batteries to support various V2G services.

There are many related efforts studying the relationship between V2G and the supported services. [1] investigated how demand response helps reduce peak power demand and shape the V2G aggregated demand profile. [2] studied the impact of EV mobility on demand response for V2G and presented a dynamic complex network model of V2G mobile energy networks. In [3], an EV scheduling algorithm was developed to optimize bidding of V2G for various ancillary services, including frequency regulation and spinning reserve. It maximized the aggregator’s profit while providing peak

load shaving to the utility. [4] designed a V2G aggregator for frequency regulation and a dynamic programming algorithm was developed to control the optimal charging for the vehicles. [5] estimated the capacity of V2G for frequency regulation with a queueing network model and this can facilitate the the regulation contract between an aggregator and the grid operator. All these suggest that V2G may potentially be beneficial to the grid and one of the keys of success is to ensure the availability of EVs to participate in V2G.

Autonomous vehicles (AVs), also known as driverless cars and robotic cars, refer to those vehicles which can navigate without human intervention. They are equipped with numerous sensors to facilitate their interactions with the surrounding environments. An AV may be fully or partially driverless; a driver can guide the movement in the “normal” mode and it can implement self-navigation in the “autonomous” mode without the driver’s input. AVs enjoy many advantages over conventional cars, like avoiding collisions due to human errors, lessening traffic congestion, and reducing physical space for vehicle parking. There are numerous research projects related to AVs. For instance, [6] designed an obstacle avoidance motion control scheme for AVs operating in uncertain dynamic environments. [7] developed a hierarchical controller for AVs to track reference paths in uncertain conditions and with external disturbances. In [8], AVs co-operated in a public transportation system, in which AVs were scheduled with centralized control. Admission control of the system was also fully investigated. [9] focused on the pricing issue of the AV public transportation system and developed a combinatorial auction-based strategy-proof pricing scheme. The automotive industry is also developing AV technologies. Google launched the self-driving car project and built a fully functioning prototype without a steering wheel and pedals [10]. A Tesla car can enable its autonomous driving ability with a software update [11]. Thus AVs do not just engage in idle theorizing and they can have practical use sooner or later.

In general, AVs are a kind of EVs and they also contain batteries to store energy for propulsion. Hence, AVs can participate in V2G as most EVs do. Due to their self-driving ability and advanced vehicular communication technologies, AVs can be coordinated to orchestrate more co-operative exercises. AVs are advantageous over ordinary EVs in the sense that the intrinsic uncontrollable EV behaviors, with

respect to their appearance at V2G infrastructure, can now be overcome. Moreover, different V2G-supporting parking facilities have diverse V2G objectives and they have different “demands” of EVs anchoring at the facilities at different times. We can now deploy more effective V2G services by appropriately assigning AVs to the parking facilities to meet their EV demands. Therefore, in this paper, instead of studying how AVs contribute to V2G in parking facilities, we investigate how to coordinate AV parking to facilitate V2G services. To the best of our knowledge, this work is the first to study AVs in the smart grid context. The rest of the paper is organized as follows. In Section II, We develop models for the road network, AVs, and parking facilities and illustrate the system operation. Section III formulates the AV parking problem as an optimization problem and we propose an effective heuristic to solve the problem in Section IV. In Section V, we evaluate the system performance and conclude the paper in Section VI.

## II. SYSTEM MODEL

The system is composed of three types of components, namely, a road network, AVs, and parking facilities. This section serves to provide models of these system components and illustrates how the system operates.

### A. Road Network

We describe the accessibility of the AVs to and from the parking facilities with a road network. The road network is modeled by a complete directed graph  $G(\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N}$  is the set of all possible locations where the AVs and the parking facilities are located.  $\mathcal{E}$  represents the set of edges connecting the locations. Each  $(i, j) \in \mathcal{E}$  is associated with the distance  $d_{ij}$  indicating the expected travel distance from  $i$  to  $j$ .  $d_{ij}$  is in general not equal to  $d_{ji}$  and this accounts for the possible asymmetry of travel distances in different directions. Note that  $G(\mathcal{N}, \mathcal{E})$  is not a direct representation of the corresponding road network; in  $G(\mathcal{N}, \mathcal{E})$ , a node is always accessible by another node in one hop. We can construct  $G(\mathcal{N}, \mathcal{E})$  from the road system by specifying a route from  $i$  to  $j$  with the corresponding distance, for each  $(i, j)$  pair. For instance, we may employ Dijkstra’s algorithm [12] to suggest the shortest route to connect  $i$  to  $j$ . We assume  $d_{ij}$ ’s are static at the time of assignment.  $d_{ij}$ ’s can be revised to reflect the updated traffic conditions in any subsequent assignments.

### B. Autonomous Vehicles

We denote the set of AVs which need parking by  $\mathcal{K}$ . Each  $k \in \mathcal{K}$  is specified by the tuple  $(\underline{n}_k, \bar{n}_k, \underline{t}_k, \bar{t}_k, \underline{e}_k, \bar{e}_k, d_k^{\max}, \alpha_k)$ . The autonomous parking mode of  $k$  is turned on at  $\underline{n}_k \in \mathcal{N}$  at time  $\underline{t}_k$  with state of charge (SOC)  $\underline{e}_k$  and it is expected to return to  $\bar{n}_k \in \mathcal{N}$  by time  $\bar{t}_k$  ( $\bar{t}_k \geq \underline{t}_k$ ) with SOC  $\bar{e}_k$ , which represents the minimum SOC at which the battery of AV is expected to reach when the driver uses the car again after parking.  $\underline{n}_k$  is allowed to be different from  $\bar{n}_k$  for the convenience of the driver. As  $k$  is expected to park in one of the parking facilities, the driver may desire to confine the total distance that its AV travels during  $(\underline{t}_k, \bar{t}_k)$ . The maximum

distance that AV  $k$  is allowed to travel in the autonomous mode is indicated by  $d_k^{\max}$ . If the assigned parking facility  $f$  is known, the AV can estimate the amount of time and energy required to reach  $f$  from  $\underline{n}_k$  and those required to arrive at  $\bar{n}_k$  from  $f$  based on its details (including its locations, driving speed, and energy consumption rate). We define the function  $\alpha_k$  to accomplish such estimation as

$$[\underline{m}_k, \bar{m}_k, \underline{\epsilon}_k, \bar{\epsilon}_k] = \alpha_k(\underline{n}_k, \bar{n}_k, \underline{t}_k, \bar{t}_k, \hat{n}_f), \quad (1)$$

where  $\hat{n}_f$ ,  $\underline{m}_k$ , and  $\bar{m}_k$  refer to the location of  $f$ , the duration for  $k$  to reach  $\hat{n}_f$  from  $\underline{n}_k$  and the duration for  $k$  to return to  $\bar{n}_k$  from  $\hat{n}_f$ , respectively.  $\underline{\epsilon}_k$  and  $\bar{\epsilon}_k$  are the amounts of energy required to support the first and second legs of the parking journey, respectively.

### C. Parking Facilities

We consider a set of parking facilities  $\mathcal{F}$ , each of which represents a V2G system connected to the grid as in [5]. Each  $f \in \mathcal{F}$  is described by the tuple  $(\hat{n}_f, p_f, c_f, \beta_f)$ .  $p_f = [\rho_t^f]_{1 \leq t \leq D}$  denotes the demand profile of  $f$ , where  $\rho_t^f$  gives the number of AVs required to support the V2G services at  $f$  in the  $t$ th time slot and  $D$  is the latest time slot in the time horizon (The time slot operation will be explained in Section II-D). There is much work in the literature describing how to utilize EVs to facilitate different kinds of V2G services, e.g., frequency regulation [5]. The basic principle is that, for  $f$  to provide various V2G services, it needs to acquire a certain number of vehicles for charging and discharging. Here we model the demand on the vehicles for V2G for the given time horizon by  $p_f$ .  $c_f$  denotes the capacity of  $f$  dedicated to the current operation. In other words, it represents the number of AVs which  $f$  can accommodate in the time horizon. We assume that  $f$  is capable of determining how long AV  $k$  should park at  $f$ . In this parking duration,  $k$  will be charged up to a level that at least  $\bar{e}_k$  will be retained when reaching  $\bar{n}_k$ , with the consideration of an appropriate charging rate and the amount of energy charged or discharged to support V2G. Consider that  $f$  can facilitate the estimation with the function  $\beta_f$  based on the SOC specifications of AV  $k$  as

$$\hat{m}_k^f = \beta_f(e_k', e_k''), \quad (2)$$

where  $\hat{m}_k^f$  is the duration that  $k$  should stay at  $f$ .  $e_k' = \underline{e}_k - \underline{\epsilon}_k$  and  $e_k'' = \bar{e}_k + \bar{\epsilon}_k$  represent the SOCs of  $k$  when arriving at  $f$  and when leaving from  $f$ , respectively, where  $\underline{\epsilon}_k$  and  $\bar{\epsilon}_k$  are computed from (1).

### D. Operation

Suppose that there is a control center which ordnates the parking of AVs. This control center aims to serve a dedicated group of AVs, e.g., the AV Public Transportation System [8], or to provide a kind of parking service to its subscribed AVs.

Similar to many existing V2G implementations (e.g., [5]), the system is considered to operate in a time-slot basis. The time horizon is described by time slots  $\{t = 0, 1, \dots, D\}$ . As providing auxiliary services is one of the core functions

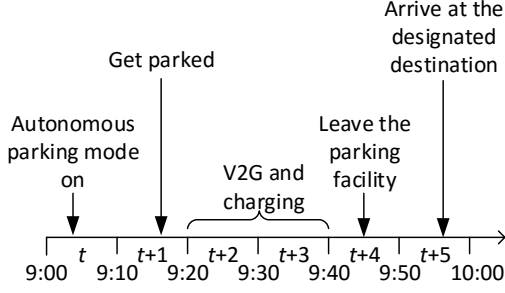


Figure 1. A time slot conversion example.

of V2G in which the extent of participation needs to be committed in advance in the corresponding auxiliary service markets, each parking facility  $f$  is supposed to be able to estimate its demand profile  $p_f = [\rho_t^f]_{1 \leq t \leq D}$  by  $t = 0$ . Moreover, with the advancement of vehicular communication technologies (e.g., vehicular ad-hoc networks [13]), the governed AVs are all connected and they can predict their travel plans for the near future. Thus it is possible for the system to determine the set of AVs with parking intention during the period  $\mathcal{T} = \{t = 1, \dots, D\}$  by  $t = 0$ .<sup>1</sup> Therefore, we assume that all necessary information, from both the AVs and parking facilities, is available at  $t = 0$  and we will assign the AVs of  $\mathcal{K}$  to appropriate parking facilities of  $\mathcal{F}$  for the period  $\mathcal{T}$ .

To fit into the time-slot implementation of V2G, without loss of generality, we convert all the time parameters introduced in Sections II-B and II-C, including  $\underline{t}_k, \bar{t}_k, \underline{m}_k, \bar{m}_k$ , and  $\hat{m}_k^f$ , to the time-slotted format. Consider the scenario given in Fig. 1 which depicts the schedule of a particular AV  $k$ . Each time slot lasts for 10 minutes and the time slots start at 9:00, 9:10, 9:20, and so on. The AV is ready to park in time slot  $t$  and it arrives at a parking facility in  $t + 1$ . It leaves the parking facility in  $t + 4$  and returns to its designated destination in  $t + 5$ .  $k$  is only available for V2G and charging in  $t + 2$  and  $t + 3$ . We can simply set  $\underline{t}_k = t + 1$ ,  $\bar{t}_k = t + 5$ , and  $\hat{m}_k^f = 2$ . It takes one slot for the first leg and one slot for the second leg of its journey, i.e.,  $\underline{m}_k = \bar{m}_k = 1$ . In this way, we have not only reserved sufficient time for  $k$  to travel, but  $k$  can also be made to fit into the V2G slotted operation.

After an assignment for the time horizon  $\{t = 0, 1, \dots, D\}$  has been done, another assignment can be performed after time  $\Delta t > 0$ , i.e., for  $\{t = 0 + \Delta t, 1 + \Delta t, \dots, D + \Delta t\}$ . If  $\Delta t$  is larger than  $D$ , it is like a fresh restart such that the two assignments have no correlation. If  $\Delta t$  is smaller than  $D$ , it is possible that some AVs are still undergoing the schedules settled in the first assignment. We can still consider these AVs in the later assignment such that their parameters are revised to reflect their updated statuses accordingly. For example, if AV  $k$  is parking at the parking facility  $f$  at  $t = \Delta t$ , we may simply set its starting location to  $\hat{n}_f$ , i.e.,  $\underline{n}_f = \hat{n}_f$ , for the later assignment.

<sup>1</sup>As AVs are more predictable, we assume that the availabilities of all AVs are known in advance. This is valid when it comes to confined transportation systems, e.g., the AV Public Transportation System [8]. Moreover, we may adjust  $D$  based on the available amount of information about the AVs and parking facilities.

### III. PROBLEM FORMULATION

To facilitate the formulation of the problem, we define two binary variables  $x_{kt}^f$  and  $y_k^f$  as follows:

$$x_{kt}^f = \begin{cases} 1 & \text{if AV } k \text{ is assigned to Parking Facility } f \\ & \text{in the time slot } t \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

and

$$y_k^f = \begin{cases} 1 & \text{if AV } k \text{ is parked at } f \\ 0 & \text{otherwise.} \end{cases}$$

Although  $x_{kt}^f$  implies  $y_k^f$ , the introduction of  $y_k^f$  can make the formulation simpler.

There are a number of requirements governing the assignment of the AVs to the parking facilities. First, each AV should be allocated to a parking facility for proper parking. In general, an AV  $k$  should not impose unnecessary burden to the traffic and should stay stationary in a parking facility most of the time from  $\underline{t}_k$  to  $\bar{t}_k$ . Hence, we consider that an AV will be assigned to one and only one parking facility during its off-duty period. This can be specified by

$$\sum_{f \in \mathcal{F}} y_k^f = 1, \forall k \in \mathcal{K}. \quad (4)$$

If AV  $k$  is assigned to Facility  $f$ , it will stay at  $f$  for a sufficient number of time slots for charging and supporting V2G services. Recall that the parked duration  $\hat{m}_k^f$  depends on its SOC specifications, the travel distances between its specific locations and  $f$ , and the expected utilization of  $k$  for V2G by  $f$ . When the details of  $k$  and  $f$  are given, by computing (1) and (2),  $\hat{m}_k^f$  is indeed a constant. We can represent such condition with the following inequality:

$$\hat{m}_k y_k^f \leq \sum_{t=1}^D x_{kt}^f \leq M y_k^f, \forall k \in \mathcal{K}, f \in \mathcal{F}, \quad (5)$$

where  $M$  is a sufficiently large positive number.

It takes time for an AV  $k$  to travel from its original position  $\underline{n}_k$  to a parking facility  $f$  and return to a designated location  $\bar{n}_k$  after parking. The time periods for these two legs of journey are specified by  $\underline{m}_k$  and  $\bar{m}_k$ , respectively (see Eq. (1)). If  $k$  is parked at  $f$  at time  $t$ , we should reserve at least  $\underline{m}_k$  time slots for  $k$  to reach  $f$ . In other words, if  $x_{kt}^f = 1$ , then there are at least  $\underline{m}_k$  time slots with  $x_{ks}^f = 0$ , where  $s < t$ . That is  $\sum_{s=\underline{t}_k}^{t-1} (1 - x_{ks}^f) \geq \underline{m}_k$ . This can be satisfied by imposing the following inequality:

$$\sum_{s=\underline{t}_k}^{t-1} (1 - x_{ks}^f) \geq \underline{m}_k x_{kt}^f, \forall k \in \mathcal{K}, f \in \mathcal{F}, t \in \mathcal{T}. \quad (6)$$

Similarly, if  $k$  is parked at  $f$  at time  $t$ , we should reserve at least  $\bar{m}_k$  time slots for  $k$  to get back to  $\bar{n}_k$  from  $f$  by  $\bar{t}_k$  for  $x_{kt}^f = 1$ . This is equivalent to:

$$\sum_{s=t+1}^{\bar{t}_k} (1 - x_{ks}^f) \geq \bar{m}_k x_{kt}^f, \forall k \in \{k | \bar{t}_k \leq D\}, f \in \mathcal{F}, t \in \mathcal{T}. \quad (7)$$

An AV  $k$  should be assigned to a facility  $f$  such that its total travel distance does not exceed  $d_k^{max}$ . In other words, if  $y_k^f = 1$ , then  $d_{n_k f} + d_{f \bar{n}_k} \leq d_k^{max}$ . This can be further described by:

$$(d_{n_k \hat{n}_f} + d_{\hat{n}_f \bar{n}_k}) y_k^f \leq d_k^{max}, \forall f \in \mathcal{F}, k \in \mathcal{K}. \quad (8)$$

Since AV  $k$  is available for parking from  $t_k$  to  $\bar{t}_k$  only, it should not be assigned to any parking facility any time before  $t_k$  and from  $\bar{t}_k$  onward. This can be specified with the following two equalities:

$$\sum_{t=1}^{t_k-1} x_{kt}^f = 0, \forall f \in \mathcal{F}, k \in \mathcal{K} \quad (9)$$

and

$$\sum_{t=\bar{t}_k}^D x_{kt}^f = 0, \forall f \in \mathcal{F}, k \in \{k | \bar{t}_k \leq D\}. \quad (10)$$

To meet the demand stemmed from the V2G services, we should secure enough AVs parked at  $f$  based on its demand profile  $p_f$ . It is not uncommon to summarize the grid requirements with a total amount of energy required at each aggregator, e.g., in [3]. We can also represent this amount of energy with a number of vehicles, each of which contributes equal portion, e.g., in [5]. Moreover, the number of AVs parked at  $f$  should not exceed its capacity  $c_f$ . These can be ensured with the following inequality:

$$\rho_t^f \leq \sum_{k \in \mathcal{K}} x_{kt}^f \leq c_f, \forall f \in \mathcal{F}, t \in \mathcal{T}. \quad (11)$$

AVs should be parked as long as possible. We can do this by maximizing the occupancy, i.e., assigning the AVs to the parking facilities in as many time slots as possible. This is equivalent to maximizing  $\sum_{k \in \mathcal{K}, t \in \mathcal{T}, f \in \mathcal{F}} x_{kt}^f$ .<sup>2</sup> Thus we can formulate the problem as:

$$\begin{aligned} & \text{maximize} && \sum_{k \in \mathcal{K}, t \in \mathcal{T}, f \in \mathcal{F}} x_{kt}^f \\ & \text{subject to} && (4)-(11). \end{aligned} \quad (12)$$

#### IV. HEURISTIC

Problem (12) is an integer linear program (ILP) and most corresponding solvers can determine the optimal solutions when the problem size is small. However, when the problem size increases, the required computational time is likely to grow exponentially (this will be verified in Section V). We cannot tolerate long computational time in some practical scenarios and hence we need a more effective method to address the problem. To do this, we proposed a heuristic, which comprises two phases. The first phase attempts to fulfill the energy profile requirements of the parking facilities, i.e., (11). In the second phase, we determine the best parking facility and time slots for those AVs which have not been allocated to any facilities in the first phase. The pseudo-code of the algorithm is presented in Algorithm 1.

<sup>2</sup>If economic cost of energy needs to be explicitly considered, we can simply replace the objective function with the related cost function.

---

#### Algorithm 1 Heuristic

---

- 1: Initialize solution matrix  $\mathbf{X}$ .
  - 2: **for**  $\forall t \in \mathcal{T}$  **do**
  - 3:   Compute  $\delta_t^f$  using (13) for all  $f$ .
  - 4:   **while**  $\sum_f \delta_t^f > 0$  **do**
  - 5:     Update  $a_{kt}^f, \forall k, f$  using (14).
  - 6:     Generate  $\{\lambda_{tk}^f | a_{kt}^f = 1, \forall k, f\}$ .
  - 7:     Find the best  $(k, f)$  combination by sorting  $\lambda_{tk}^f$ .
  - 8:     Decide the parking segment  $[\underline{t}, \bar{t}]$  and modify  $\mathbf{X}$  according to (15).
  - 9:     Update  $\delta_t^f$  for all  $f$ .
  - 10:  **while**  $\sum_{f \in \mathcal{F}, t \in \mathcal{T}} x_{kt}^f < 1, \exists k$  **do**
  - 11:   **for**  $\forall k \in \mathcal{K}, f \in \mathcal{F}$  **do**
  - 12:     Generate  $\omega_{k\pi}^f$  with (16) for each segment  $\pi$ .
  - 13:     Find the best  $(k, f, \pi)$  combination by sorting  $\omega_{k\pi}^f$ .
  - 14:     Modify  $\mathbf{X}$  according to (15) by assigning  $\pi$  to  $[\underline{t}, \bar{t}]$ .
- 

#### A. The First Phase

The algorithm starts by initializing a solution matrix  $\mathbf{X} = [x_{kt}^f]_{k \in \mathcal{K}, t \in \mathcal{T}, f \in \mathcal{F}}$  with all elements of zero value, where  $x_{kt}^f$  carries the same meaning as defined in (3). Then the algorithm works iteratively to fulfill the lower bound of the energy profile  $\rho_t^f$ , and each iteration corresponds to one time slot. In the  $t$ th iteration, we define

$$\delta_t^f \triangleq \rho_t^f - \sum_{k \in \mathcal{K}} x_{kt}^f \quad (13)$$

to represent the deficit for the energy profile for each  $f$ . While there exists a  $\delta_t^f > 0$ , we find the best AV among all to fulfill the deficit. To do this, we assign an availability index  $a_{kt}^f$  for each vehicle  $k$  with respect to each parking facility  $f$ . It is a Boolean index composed of five clauses, defined as

$$\begin{aligned} a_{kt}^f \triangleq & (t \geq \underline{t}_k + \underline{m}_k) \wedge (t \leq \bar{t}_k - \bar{m}_k) \wedge \left( \sum_{k \in \mathcal{K}} x_{kt}^f < c_f \right) \wedge \\ & \left[ \left( \sum_{f \in \mathcal{F}} \sum_{t'=1}^{t-1} x_{kt'}^f = 0 \right) \vee (x_{k,t-1}^f = 1) \right] \wedge \\ & (d_{n_k \hat{n}_f} + d_{\hat{n}_f \bar{n}_k}) \leq d_k^{max}. \end{aligned} \quad (14)$$

These clauses correspond to some of the constraints given in Problem (12). The first two clauses check if  $t$  is eligible for  $k$  to appear at  $f$  by considering the traveling times  $\underline{m}_k$  and  $\bar{m}_k$ . The third inspects whether there is room for further allocation based on the capacity  $c_f$ . The fourth checks if  $k$  has been assigned to a facility in any previous time slots. Note that  $k$  is still eligible to park at  $f$  if it is assigned to  $f$  in the immediate past time slot (i.e., continuous parking). The last clause confirms whether traveling to  $f$  will not exceed its allowed travel distance. For each  $k$  available for parking at  $f$ , i.e.,  $a_{kt}^f = 1$ , we define a 2-tuple performance score

$\lambda_{tk}^f \triangleq \langle \phi_{tk}^f, \psi_{tk}^f \rangle$ , where

$$\phi_{tk}^f = \begin{cases} \sum_{k \in \mathcal{K}} a_{kt}^f - \delta_t^f, & \delta_t^f > 0, \\ |\mathcal{K}|, & \text{otherwise.} \end{cases}$$

$\psi_{tk}^f = \sum_{t'=\underline{t}_k+\underline{m}_k}^{\bar{t}_k-\bar{m}_k} \delta_{t'}^f$ . This design follows a basic allocation rule that the parking facility with the least number of available AVs shall be handled first.  $\phi_{tk}^f$  measures how rich the ‘‘AV resource’’ is available to each  $f$  and  $\psi_{tk}^f$  is the sum of energy profile deficits during the available parking period of  $k$ . We arrange  $\lambda_{tk}^f$  in a list by sorting  $\phi_{tk}^f$  in the ascending order. For any tie,  $\psi_{tk}^f$  are sorted in the descending order. The best  $(k, f)$  combination is associated with the first  $\lambda_{tk}^f$  in the list.

Given  $k$  and  $f$ , the longest possible parking period is  $[\underline{t}_k + \underline{m}_k, \bar{t}_k - \bar{m}_k]$ . However, it is possible to have some  $\tilde{t} \in [\underline{t}_k + \underline{m}_k, \bar{t}_k - \bar{m}_k]$  with the number of assigned AVs reaching the capacity  $c_f$ . Hence  $[\underline{t}_k + \underline{m}_k, \bar{t}_k - \bar{m}_k]$  is divided into multiple segments and each of these segments has room for further AV allocation. Among these segments, we select the longest one, denoted by  $[\tilde{t}, \tilde{t} + 1]$ , and allocate  $k$  into it by assigning  $x_{k\tilde{t}}^f$ :

$$x_{k\tilde{t}'}^f = 1, \tilde{t}' = \tilde{t}, \tilde{t} + 1, \dots, \tilde{t}. \quad (15)$$

Next we update  $\delta_t^f$  according to (13) again for all  $f$ .

### B. The Second Phase

After the first phase, it is possible that some AVs are not allocated to any parking facility. In the second phase, we assign these AVs to appropriate parking facilities. Similarly as above, for each  $f$ , we can divide  $[\underline{t}_k + \underline{m}_k, \bar{t}_k - \bar{m}_k]$  into segments. For each segment  $\pi = [\underline{t}_\pi, \bar{t}_\pi]$ , we determine a 2-tuple performance score  $\omega_{k\pi}^f$  as

$$\omega_{k\pi}^f = \langle \bar{t}_\pi - \underline{t}_\pi, \sum_{t \in \pi} (c_f - \sum_{k \in \mathcal{K}} x_{kt}^f) \rangle. \quad (16)$$

The first value in  $\omega_{k\pi}^f$  stands for the length of  $\pi$  while the second specifies how much room is left for further assignment for  $f$  in  $\pi$ . We sort all  $\omega_{k\pi}^f$  in the descending order by the first value, and then by the second value for any tie. The best  $(k, f, \pi)$  tuple is associated with the first  $\omega_{k\pi}^f$  in the sorted list, and the selected  $k$  is assigned to  $f$  during  $\pi$  using (15). This process repeats until all AVs have been allocated.

## V. PERFORMANCE EVALUATION

There are two methods to solve the problem, i.e., by a standard ILP solver and by our proposed heuristic. In general, a standard ILP solver can give the optimal solution if the problem is solvable by the solver. In this section, we evaluate their performance on different scales of the problem. The problem size changes with different numbers of AVs and parking facilities and random cases are generated based on some realistic settings. Unless stated otherwise, we assume that there are 100 time slots (i.e.,  $D = 100$ ) evenly separated in a horizon of two hours. Consider a residential area of  $5 \times 5$  km<sup>2</sup>, within which we randomly place required numbers of AVs and parking facilities by specifying  $\underline{n}_k$ ,  $\bar{n}_k$ , and  $\hat{n}_f$  accordingly. Suppose that the AVs travel at

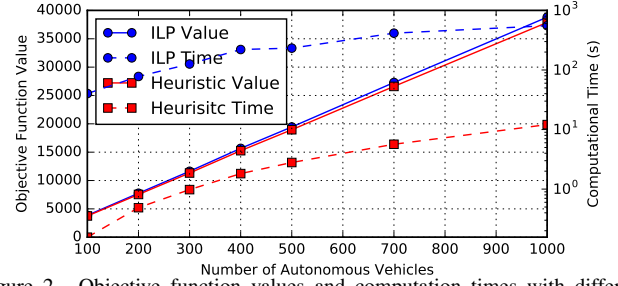


Figure 2. Objective function values and computation times with different numbers of AVs.

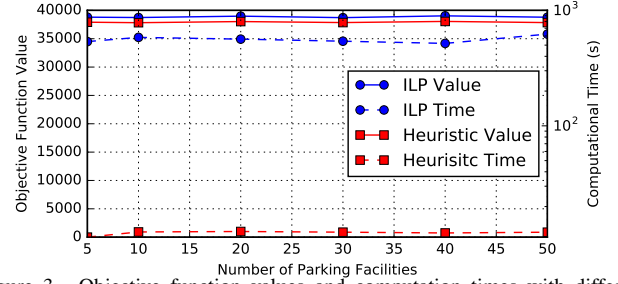


Figure 3. Objective function values and computation times with different numbers of parking facilities.

a constant speed of 30 km/h. For AV  $k$ , the travel times spent on the two legs for parking, i.e.,  $\underline{m}_k$  and  $\bar{m}_k$ , are assigned based on the corresponding distances. We also set  $\hat{m}_k^f = \text{rand}(1, \bar{t}_k - \bar{m}_k - (\underline{t}_k + \underline{m}_k))$ . These capture  $\alpha_k$  and  $\beta_f$ . We specify  $\underline{t}_k$  and  $\bar{t}_k$  by  $\underline{t}_k = \text{rand}(0, D - \underline{m}_k - \bar{m}_k)$  and  $\bar{t}_k = \text{rand}(0, D - \underline{m}_k - \bar{m}_k) + \underline{t}_k + \underline{m}_k + \bar{m}_k$ , where  $\text{rand}(\cdot, \cdot)$  produces an integer uniformly distributed between the two inputs inclusively. If  $\bar{t}_k > D$ , then AV  $k$  will not need to return to  $\bar{n}_k$  during the time horizon. Finally, the energy profile of Parking Facility  $f$  is set as  $\rho_t^f = \text{rand}(0, a_t^f / |\mathcal{F}|), \forall t \in \mathcal{T}$ , where  $a_t^f$  is the number of AVs that are available to park in  $f$  at  $t$ . The parking capacity  $c_f$  is set to  $|\mathcal{K}|/2$  for all  $f$ . This allows us to generate feasible instances more easily to inspect the computational abilities of the methods. All simulations are performed on a computer with Intel Core-i5 CPU at 2.90 GHz with 8 GB RAM. The simulations are coded with Python on Linux and we adopt Gurobi [14] as the ILP solver.

We perform three tests: (1) different numbers of AVs with a fixed number of parking facilities, (2) different numbers of parking facilities with a fixed number of AVs, and (3) changing time scales. In the first test, we consider a setting for a small neighborhood, where there are five parking facilities. We generate random cases of 100, 200, 300, 400, 500, 700, and 1000 AVs. Fig. 2 depicts the objective function values (i.e., occupancies) and computational times (in log scale) obtained by the two methods. Each point in the figure corresponds to the average results from 25 cases. It can be observed that the two methods produce similar objective function values and the heuristic shows just a minor inferiority when the number of AVs is large. Both methods need more computational time when the number of AVs grows. However, the ILP method takes much longer time for a larger problem instance and this may not be practical. As a whole, the heuristic performs well

Table I  
EFFECTS OF TIME SCALING ON % OPTIMALITY AND COMPUTATIONAL TIME.

Case	10	20	30	40	50	80	100
I	N/A	94.27	95.32	97.14	98.83	97.54	100.00
II	N/A	N/A	N/A	96.97	98.67	98.10	100.00
III	N/A	95.54	95.98	96.87	98.73	97.47	100.00
IV	N/A	N/A	96.72	98.01	99.79	97.74	100.00
V	N/A	N/A	96.04	97.26	99.00	97.85	100.00
VI	N/A	N/A	95.29	97.39	99.55	97.62	100.00
VII	N/A	N/A	96.73	96.75	98.66	98.21	100.00
VIII	N/A	94.72	95.28	96.19	98.93	97.32	100.00
IX	N/A	94.49	95.62	96.65	98.46	97.82	100.00
X	N/A	94.94	95.83	96.83	98.65	97.84	100.00
Avg-time (s)	N/A	2.28	4.26	6.90	10.04	22.96	40.23

in practical scenarios.

In the second test, we fix the number of AVs to 1000 and consider cases of 5, 10, 20, 30, 40, and 50 parking facilities. Fig. 3 gives the computed objective function values and computational times, where each data point represents the average of 25 cases. We can see that both the objective function value and computational time are not very sensitive to the number of parking facilities. Thanks to the fact that the occupancy of a vehicle at a parking facility in a time slot has no difference from any other in the objective function, as long as the parking facilities are sufficient to accommodate the AVs, more parking facilities available will not help improve the objective function value. We can understand the trend of computational time in a similar way.

In the third test, we investigate the impact of time scaling. Recall that a given time horizon is divided into slots and we can make the division finer with more time slots for the same period. We generate 10 random cases for the same horizon of two hours. For each case, we divide the horizon into 10, 20, 30, 40, 50, 80, and 100 time slots with the same settings of AVs and parking facilities. In other words, we are solving the same problem instances with different time scales only. For example, the 51<sup>th</sup> time slot in the 100-scale corresponds to the 26<sup>th</sup> and 5<sup>th</sup> in the 50- and 10-scale, respectively. Since time scaling is intrinsic to the problem, we demonstrate its effects on the optimality and thus we show the results here with the ILP approach only. Table I illustrates the percentage (%) optimality of the different scales with respect to the 100-scale for the 10 cases. The 100-scale is the finest and gives the best results in term of quality. When scaling down, the % optimality drops slightly because the flexibility of assignment decreases. However, too coarse scaling (e.g., 10-scale) can result in unfeasible solutions. Table I also shows the computational times averaged over the feasible cases. This suggests that scaling-down can improve the computational time significantly due to the reduced problem size. Therefore, there exists a tradeoff between solution quality and computational time.

## VI. CONCLUSION

AVs will occupy a substantial share of ground transport in the near future. When parked, AVs can participate in V2G exactly as EVs do. The difference is that AVs can be instructed to travel based on some system-wide objectives. In this paper, we study how to coordinate AVs intending to park, to reach parking facilities for supporting V2G services. We formulate the problem in the form of ILP. Besides standard ILP solvers, we propose an effective heuristic to solve the problem. Simulations reveal that the heuristic can significantly reduce the computational time with negligible degradation in objective function value. The quality of the optimal solution is insensitive to the number of available parking facilities. A coarser time scale can improve computational time but degrade the solution quality resulting in possible infeasible solution.

## ACKNOWLEDGMENT

This research is supported in part by the Theme-based Research Scheme of the Research Grants Council of Hong Kong, under Grant No. T23-701/14-N.

## REFERENCES

- [1] F. Rassaei, W.-S. Soh, and K.-C. Chua, "Demand response for residential electric vehicles with random usage patterns in smart grids," *IEEE Trans. Sustainable Energy*, vol. 6, no. 4, pp. 1367–1376, Oct. 2015.
- [2] R. Yu, W. Zhong, S. Xie, C. Yuen, S. Gjessing, and Y. Zhang, "Balancing power demand through EV mobility in vehicle-to-grid mobile energy networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 79–90, Feb. 2016.
- [3] E. Sortomme and M. A. El-Sharkawi, "Optimal scheduling of vehicle-to-grid energy and ancillary services," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 351–359, Mar. 2012.
- [4] S. Han, S. Han, and K. Sezaki, "Development of an optimal vehicle-to-grid aggregator for frequency regulation," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 65–72, Jun. 2010.
- [5] A. Y. S. Lam, K.-C. Leung, and V. O. K. Li, "Capacity estimation for vehicle-to-grid frequency regulation services with smart charging mechanism," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 156–166, Jan. 2016.
- [6] G. Franze and W. Lucia, "A receding horizon control strategy for autonomous vehicles in dynamic environments," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 695–702, Mar. 2016.
- [7] C. Chen, Y. Jia, M. Shu, and Y. Wang, "Hierarchical adaptive path-tracking control for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2900–2912, Oct. 2015.
- [8] A. Y. S. Lam, Y.-W. Leung, and X. Chu, "Autonomous vehicle public transportation system: Scheduling and admission control," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1210–1226, May 2016.
- [9] A. Y. S. Lam, "Combinatorial auction-based pricing for multi-tenant autonomous vehicle public transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 859–869, Mar. 2016.
- [10] L. Gannes. (2014, May) Google's new self-driving car ditches the steering wheel. [Online]. Available: <http://recode.net/2014/05/27/googles-new-self-driving-car-ditches-the-steering-wheel/>
- [11] R. Bradley. (2014, Oct.) The electric-vehicle maker sent its cars a software update that suddenly made autonomous driving a reality. [Online]. Available: <https://www.technologyreview.com/s/600772/10-breakthrough-technologies-2016-tesla-autopilot/>
- [12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.
- [13] D. N. Cottingham, "Vehicular wireless communication," University of Cambridge, Tech. Rep. UCAM-CL-TR-741, Jan. 2009.
- [14] Gurobi optimization. [Online]. Available: <http://www.gurobi.com/>