On-Street Parking Guidance with Real-Time Sensing Data for Smart Cities

Kin Sum Liu*, Jie Gao*, Xiaobing Wu†, Shan Lin‡
*Department of Computer Science, Stony Brook University, NY, USA
†Wireless Research Centre, University of Canterbury, New Zealand
‡Department of Electrical and Computer Engineering, Stony Brook University, NY, USA
Email: {kiliu, jgao}@cs.stonybrook.edu, barry.wu@canterbury.ac.nz, shan.x.lin@stonybrook.edu

***Abstract*—On-street parking is an essential component of parking infrastructure for smart cities, which allows users to park near their destinations for short term. However, due to limited capacity, saturated on-street parking becomes a serious and widespread problem for urban transportation systems. Greedily searching for an on-street parking spot in a saturated area is often a frustrating task for drivers, and cruising for vacant parking spots results in additional delays and impaired local circulation. With the recent development of networked smart parking meter, real-time city-wide on-street parking information becomes available for more efficient parking management. In this paper, we design an online parking guidance system that recommends parking spots in real-time based on the parking availability prediction. With a receding horizon optimization framework, our solution minimizes the user's driving and walking cost by adapting the spatiotemporally dynamic supply and demand in the local area, significantly reducing parking competitions in a timely manner. We implement and evaluate our solution with a dataset of 13,503,655 parking records collected from 5228 in-ground sensors distributed in the Australian city Melbourne. The evaluation results show that our approach achieves up to 63.8% delay reduction compared with existing solutions.**

***Keywords*—on-street parking; smart parking; smart city**

I. Introduction

As urban population continued to grow in the last few decades, increasing urban traffic imposes significant challenges on the transportation system infrastructure in smart cities, such as managing accidents, traffic congestion, and parking difficulties [1]. For example, people who drive to the city usually need to find parking spots close to their destinations. Since on-street parking is low-cost and available widely across the city, it is preferred over other central garage facilities. However, searching for available parking space on street is a very challenging task. To know the spot availability at a road segment, a driver needs to drive around that specific location. It is estimated that searching for a parking spot creates additional delays and impairs local circulation. One study reveals that an average of 30% of the traffic in busy areas is caused by vehicles cruising for vacant parking spots [2]. In central areas of large cities cruising may account for more than 10% of the local circulation as drivers can spend 20 minutes looking for a parking spot [3].

Smart parking meters have been deployed in metropolitans, such as San Francisco and Melbourne, they can sense the occupancy status of the spots and report real-time availability information via network infrastructure. However, uncertainty of the time-sensitive occupancy information grows significantly as the distance between a user and his/her destination increases. As a result, multiple vehicles may aggressively seek for a limited number of spots at the same location, which exacerbate the intrinsic competition among users. Such uncoordinated driving behavior escalates local congestion and introduces extra cruising time for those vehicles that arrive late. There are a large number of research works on different aspects of intelligent parking systems in the last few years, which include occupancy detection [4], [5], system development [6], and shared service design [7]. Some research works present parking assignment algorithms [8], [9], [10], [11], which rely on reservation-based solution designed for controlled off-street parking facilities. However, these reservation-based solutions do not work for on-street parking, because there is no mechanism to enforce reservation. Moreover, these works don't address the uncertain demand introduced by external users that competes with system users for parking spots.

In fact, historical parking meter dataset offers rich spatiotemporal information about parking demand in metropolitan areas, which allows us to learn availability patterns and predict probabilities of parking occupancy of each road segment at different times of the day. Researchers have been able to predict parking occupancy rate by different machine learning methods, such as regression tree and neural network [12], [13]. On the other hand, Global Positioning System (GPS) systems also provides real-time location and mobility of each modern vehicle. Such real-time information provides opportunities to improve the efficiency of parking coordination. In this paper, we consider the following problem: optimizing for drivers' anticipated future cruising time to search for spots and walking time required to reach their destinations, while fulfilling current, local parking demand. To address this problem, we take a receding horizon approach to adaptively adjust parking recommendations for users based on their current and predicted travel progress and the parking occupancy status. By predicting and adapting to parking demand changes, our solution effectively resolves user competition in a distributed manner.

There are a few papers on on-street parking, most of them employ a game-theoretical approach [14], [15]. These works provide valuable insights to the nature of the problem, but such solutions are difficult to apply in reality. Different from these

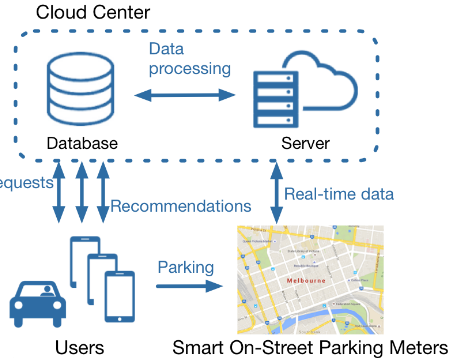


Fig. 1. The framework of the real-time on-street parking guidance system

designs, we employ a simple and practical framework based on prediction and real-time parking data and traffic data, which effectively coordinate incoming vehicles and reduce parking competition during rush hours. In [16], authors consider a probability of successful parking within a certain distance from the current location, and apply Traveling Salesman Tour (TSP) to search for parking spots. Our solution is different since we consider competition among multiple drivers and minimizing multiple objectives: both driving and walking time of the users.

The contributions of this work are as follows,

- To the best of our knowledge, this is the first study on on-street parking management using receding horizon optimization with real-time large-scale data. We design a generic framework that considers both current and future parking spot availability, minimizing users' driving and walking time under practical constraints.
- We formulate the multi-user on-street parking problem and show its NP-hardness.
- To deal with demand uncertainty caused by competition of external users, our framework incorporates demand prediction based on historical datasets into real-time coordination, which is essential for accurate and efficient parking coordination.
- We implement and evaluate our solution with a dataset in the Australian city Melbourne with 13,503,655 parking records collected from 5228 in-ground sensors distributed around 328 street segments in 27 areas across the central business district. The evaluation results show that our approach achieves up to 63.8% delay reduction compared with existing solutions.

II. Design Overview

Fig. 1 shows the framework of our on-street parking guidance system. A user uses the parking guidance mobile app to send a parking request with the current location and the destination to the cloud center. The cloud server maintains a list of all user requests and updates the real-time parking space status. The server also estimates the users' arrival time and runs the recommendation algorithm based on the parking availability prediction model. At each decision point, a parking recommendation sequence of road segments is sent back to the user. The user then follows the sequence to travel through corresponding road segments to search for parking.

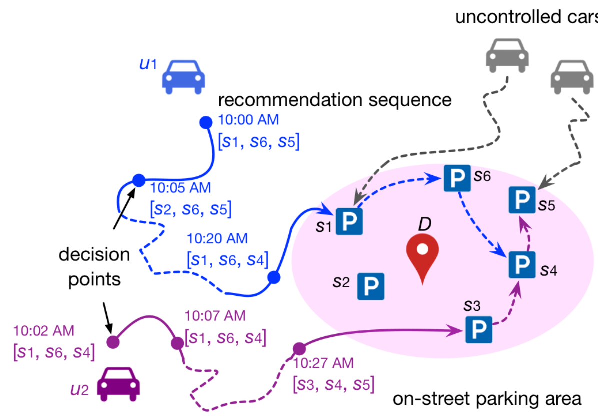


Fig. 2. Receding horizon optimization of multi-user on-street parking

A. Assumptions

We make two practical assumptions in this work: (i) not all drivers searching for on-street parking use our system; (ii) our system may change users' parking behavior.

According to the assumption (i), users will compete with some "external drivers" that search for parking but do not use our parking guidance system. And due to the assumption (ii), the parking availability prediction should depend on both the historical patterns and the real-time sensing data.

For simplicity, the parking cost is defined as the combination of two components: a) driving time cost: the cost generated by driving from current location to the parking location; b) walking time cost: the cost generated by walking from the parking location to the destination. Note that our cost function is not exclusive, other factors including price, time restrictions, etc can be included as objectives in the same framework.

Therefore, the principle of our on-street parking guidance system is to recommend a sequence of on-street parking spaces to the driver, which balances the driving cost and walking cost according to our prediction of parking availability. Intuitively, drivers always tend to search the street nearest to their destinations. The greedy algorithm are suboptimal especially when the destination is in crowded areas and it is difficult to find vacant parking spots. As a result, the driver has to cruise longer for a vacant parking spot, introducing extra traveling time and walking time.

The basic idea of the on-street parking guidance is illustrated in Figure 2. Specifically, there are six street segments for on-street parking near the destination, denoted by $s_1, s_2, \ldots, s_6$. After the user submits a query request, a recommendation sequence will be sent to him. During his trip to the destination, the recommendation sequence will be updated at each decision point (e.g., every 5 minutes). In the example of Figure 2, the latest update for user u_1 is $[s_1, s_6, s_4]$. If following the guidance, u_1 will search s_1 first, and then go to s_6 if s_1 is occupied. Unlike the greedy algorithm which retains proximity as a high priority, the parking guidance system reduces total cost which includes both driving cost and walking cost.

Suppose there is another user u_2 searching for parking in the same destination area as u_1, as shown in Fig. 2, then the competition between users will occur if the system serves users separately. To resolve conflict, we dynamically update the estimated arrival time of the users at the parking spots and recommend a spot to the user who arrives first. For a smart parking system, assigning a parking spot to the "first-arrived" user can avoid the conflict and reduce the waste of parking resources. For example, suppose s_1 is recommended to both u_1 and u_2 as the first option. If there is only one vacant parking spot in s_1, it will cause competition between u_1 and u_2. If u_2 is estimated to arrive earlier than u_1, then s_1 will still be recommended to u_2 while u_1 may receive a new recommendation at the next decision point. Suppose u_2 has a traffic congestion afterwards and is estimated to arrive later than u_1, then the recommendation will be updated correspondingly. In that way, the multi-user competition is resolved and the on-street parking spots are fully utilized.

III. Mathematical Model

A. Problem Statement

Suppose there are a set of users $U = \{u_i | i = 1, \ldots, N\}$ who are searching for on-street parking, and a set of street segments $S = \{s_j | j = 1, \ldots, M\}$ each has a fixed number of spots for parking. $l_i(k)$ denotes the location of u_i at the kth decision point, and $h_j(k)$ describes the number of vacant parking spots of s_j at the kth decision point.

At the kth decision point, we denote by $\Gamma(k)$ a set of streets with available parking spots.

$$\Gamma(k) = \{s_j : h_j(k) > 0, \quad j = 1, \ldots, M\}$$

Let W_i denote the upper bound of walking distance accepted by u_i from the parking location to the destination, and w_{ij} is the walking distance if user u_i parks at street s_j. Define $\Omega_i(k)$ as a set of feasible candidate streets for user u_i,

$$\Omega_i(k) = \{s_j : w_{ij} < W_i, \quad s_j \in \Gamma(k)\}$$

The binary control variable at kth decision point is defined as:

$$x_{ij}(k) = \begin{cases} 1, & \text{if street } s_j \text{ is assigned to user } u_i \\ 0, & \text{otherwise} \end{cases}$$

If street s_j is assigned to user u_i, define the cost at kth decision point as $c_{ij}(l_i(k))$ which depends on the current location of u_i.

The parking problem is thus formulated as a dynamic assignment problem which assigns streets to users over time. The reasons are: 1) The decisions will be changed as real-time traffic and occupancy status change, as well as other factors which influence the cost. 2) The next locations should be provided if recommended parking location has been occupied.

The objective function is to minimize the total cost over T time slots by allocating streets to users. The constraints are: (i) At most one street can be assigned to a user at any decision point; (ii) One street may be assigned to multiple users, while the number of users cannot exceed its capacity.

$$\begin{aligned} \min_x \quad & \sum_{k=1}^{T} \sum_{u_i \in U} \sum_{s_j \in \Omega_i(k)} c_{ij}(l_i(k)) \cdot x_{ij}(k) \\ \text{s. t.} \quad & \sum_{s_j \in \Omega_i(k)} x_{ij}(k) = 1 \quad \forall u_i \in U, k \\ & \sum_{u_i \in U} x_{ij}(k) \le h_j(k) \quad \forall s_j \in \Gamma(k), k \\ & f_i(l_i(k), x_i(k)) = l_i(k+1) \quad \forall u_i \in U, k \\ & x_{ij}(k) \in \{0, 1\} \quad \forall u_i \in U, s_j \in S, k \end{aligned} \tag{1}$$

The location of user u_i at the beginning of the $(k+1)$th time period is given by a function $f_i(l_i(k), x_i(k))$ which depends on the previous location and the assignment decision.

B. Computational Complexity

Even if the location (state) dynamics is constant (i.e. $l_{i+1}(k) = f_i(l_i(k), x_i(k)) = l_i(k)$, binary resource allocation is known to be difficult because of the intrinsic hardness of integer programming. While the dynamics is not constant in the parking problem clearly, we observe that the optimization (1) is equivalent to the dynamic resource allocation (DRA) problem [17] in the sense that our problem is to allocate M streets (resources) to N users (activities) over T time periods:

At the beginning of the kth time period, the state of each user is the current location. And for each user u_i, a task $a_i(k)$ is selected from the set $\mathcal{A} = \{1, 2, \ldots, M\}$. Let $a_i(k) = j$ represents street s_j is assigned to u_i in the kth time period. During the kth time period, it generates cost $c_i(l_i(k), a_i(k))$ according to the current state and the assignment, and consumes resource $r_{ij}(a_i(k))$.

$$r_{ij}(a_i(k)) = \begin{cases} 1 & \text{if } a_{i,k} = j \\ 0 & \text{otherwise} \end{cases}$$

The object is to minimize the total cost over T time periods.

$$\begin{aligned} \min_a \quad & \sum_{k=1}^{T} \sum_{u_i \in U} c_i(l_i(k), a_i(k)) \\ \text{s. t.} \quad & \sum_{u_i \in U} r_{ij}(a_i(k)) \le h_j(k) \qquad \forall j, k \\ & f_i(l_i(k), a_i(k)) = l_i(k+1) \qquad \forall i, k \end{aligned} \tag{2}$$

The decision variables are the actions $a_i(k)$ which represents the assignment solution of kth time period. It is demonstrated that problem (2) is equivalent to problem (1).

If we define

$$c_i'(l_i(k), a_i(k)) = \frac{1}{c_i(l_i(k), a_i(k))}$$

then problem (2) can be rewritten as

$$\begin{aligned} \max_a \quad & \sum_{k=1}^{T} \sum_{u_i \in U} c_i'(l_i(k), a_i(k)) \\ \text{s. t.} \quad & \text{constraints of problem (2).} \end{aligned} \tag{3}$$

which is a standard form of DRA problem.

Without any additional assumption on the structure of the problem, DRA is even NP-hard to approximate within any constant factor through the reduction from a package integer problem to a DRA problem [17].

C. Challenges

The stochastic nature of a real parking application is more difficult than the rather deterministic assignment problem

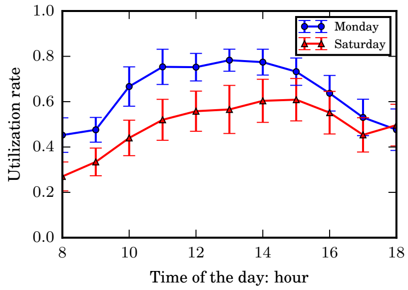


Fig. 3. Average on-street parking utilization on Weekday and Weekend

described above, because in the real scenario, the cost function depends not on the current parking availability but also the future parking information. Therefore, we will first discuss how we predict future parking availability and then will discuss the system design to give a parking recommendation sequence to a single user, and how the approach can be extended to avoid multi-user competition.

IV. Parking Availability Prediction

A. Data Description

The data used in this paper is collected from Melbourne Parking Events 2014 data set [18]. The City of Melbourne has installed around 5228 in-ground sensors distributed around 328 street segments in 27 areas across the central business district.

In the data set, a record for each parking event includes: arrival time, departure time, bay ID, area ID, street ID, between street 1, between street 2, which gives a duration of occupancy for one specific parking spot. In this paper we focus on parking street segments instead of individual parking spots, therefore parking availability data is usually aggregated for a street segment. The utilization rate of a street segment s_i at time t can be measured by

$$r_i(t) = \frac{o_i(t)}{N_i} \tag{4}$$

where $o_i(t)$ is the number of occupied parking spots of s_i at time t, N_i is the total number of parking spots of s_i.

For most of the parking spots, the sensors will be turned on at the start of the operational time (e.g., early morning at 7:30 AM) and turned off at the end of the operational time (e.g., night at 6:30 PM). In order to deal with the abnormality, we only use data from 8:00 AM to 6:00 PM to learn historical parking patterns and prediction model.

B. Feature Set

Fig. 3 compares the historical pattern of utilization rate on Monday and Saturday over all the street segments. It shows that the parking utilization characteristics are different according to the day of week, both in terms of magnitude and shape. Fig. 4 shows the correlations between the occupancy of the same street segment with different time lags (i.e., 10:00 AM and 9:50 AM, 9:40 AM ...). It suggests a strong temporal correlation for parking utilization for small time lags. The correlation coefficients decreases as the time lag increases.

The aim is to predict the utilization rate of one specific street segment given a specific date and time. Therefore, we consider the the input as $\{s_i, r_i[t_{prev} : t_{now}], t, dow\}$ and the

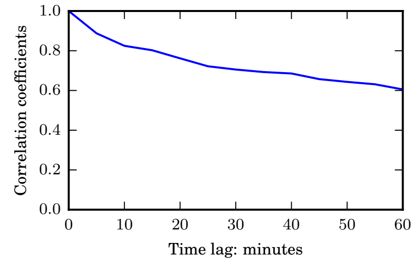


Fig. 4. Correlation between the occupancies of one street segment over time

output as $\hat{r}_i(t)$ where $\hat{r}_i(t)$ is the predicted utilization rate of street segment s_i at time t, $r_i[t_1 : t_2]$ is real-time occupancy status of s_i between time t_1 and t_2, t_{now} is the current time, t_{prev} is the previous time point ahead of the current time, t is the predicted time, dow is the day of week.

C. Prediction Model

A number of researches have focused on how to detect the occupancy state of parking spots in dedicated parking facilities, and thus proposed various parking availability prediction models [19], [20], [21]. However, the prediction of parking availability for on-street parking is more difficult than off-street parking since the variance of on-street parking is relatively higher. In addition, our guidance system may change the parking behavior of drivers. Thus, prediction only by using historical parking patterns is not accurate enough. The utilization rate data is time series and according to Fig. 4, there is linear association between lagged observations. This suggests past observations might predict current observations. And that motivates us to propose an autoregressive (AR) model for parking availability with real-time sensing data.

1) AR model w.r.t. utilization rate: For street segment s_i, an AR process that depends on k past observations can be written as

$$r_i(t) = c + \sum_{m=1}^{k} \varphi_m r_i(t-m) + \varepsilon(t) \tag{5}$$

where $\varphi_1, \ldots, \varphi_k$ are the parameters of the model, c is a constant, and $\varepsilon(t)$ is white noise.

Let $\hat{r}_i(t)$ be an unbiased estimator of $r_i(t)$:

$$\begin{aligned} \hat{r}_i(t) &= \mathbb{E}[r_i(t) | r_i(t-m), m = 1, \ldots, k] \\ &= \hat{c} + \sum_{m=1}^{k} \hat{\varphi}_m r_i(t-m) \end{aligned} \tag{6}$$

where $\hat{\varphi}_1, \ldots, \hat{\varphi}_k$, $\hat{c}$ denote the estimated parameters, which can be learned statistically from historical data. The past observations $r_i(t-1), \ldots, r_i(t-k)$ can be obtained from sensing datasets.

2) AR model w.r.t. utilization rate variation: To deal with the stochastic issue of on-street parking availability prediction, reference [20] employed the trending and detrending techniques, which are commonly used in time series analysis for financial applications. That approach is aimed to separate the deterministic part of data from the random parts. In [20], the authors focused on parking areas, which is a group of parking locations (i.e., street segments). Similarly, the approach can also be applied to one certain street segment, which is a group of parking spots.

We define the utilization rate variation of s_i as

$$v_i(t) = r_i(t) - r_i(t-1) \tag{7}$$

A trend is defined as the moving mean of the data. Thus, the trend of utilization rate variation is calculated by averaging the variation of a certain street segment over the past W weeks. The average variation $\bar{v}_i(t)$ for s_i at time t is obtained from

$$\bar{v}_i(t) = \frac{1}{T} \sum_{j=0}^{W} w_j \sum_{m=0}^{T-1} v_i(t - m - jT_w) \tag{8}$$

where T is the length of the moving window in minutes, T_w is the length of a week which depends on the choice of time step, W is the number of weeks in consideration, and $w_j(\sum_{j=0}^{W} w_j = 1)$ are positive weights to emphasize the data of most recent weeks.

Then, the utilization rate variation can be decomposed into two components

$$v_i(t) = \bar{v}_i(t) + \tilde{v}_i(t) \tag{9}$$

where $\bar{v}_i(t)$ is the trend component as defined in (8), and $\tilde{v}_i(t)$ is the stochastic component.

Therefore, we can detrend the utilization rate variation with respect to the variation trends of the corresponding time of day and day of week, and then extract the stochastic component of s_i at time t, which is defined as

$$\tilde{v}_i(t) \triangleq v_i(t) - \bar{v}_i(t) \tag{10}$$

where $v_i(t)$ is the real-time utilization rate variation defined in (7) and $\bar{v}_i(t)$ is calculated from (8).

Similarly, let $\hat{v}_i(t)$ be an unbiased estimator of $\tilde{v}_i(t)$, which can be obtained by the autoregressive model described in (6).

After predicting the utilization rate variation, the estimated utilization rate of s_i at time t (i.e., the occupancy probability) can be obtained as

$$\hat{r}_i(t) = r_i(t-1) + \bar{v}_i(t) + \hat{v}_i(t) \tag{11}$$

V. Single-User On-Street Parking Guidance

For a driver searching for an on-street parking spot, the request is represented by his current location denoted by $\alpha \in \mathbb{R}^{1\times 2}$ and destination denoted by $\delta \in \mathbb{R}^{1\times 2}$. First, some candidate street segments will be filtered according to the user's preferences, such as walking distance threshold, parking facility type, parking time and parking fee, *etc*. Suppose there are n candidate street segments for on-street parking which satisfies the user's preferences. The position of the i-th street segment is denoted by $s_i \in \mathbb{R}^{1\times 2}, i = 1, 2, \dots, n$, and the position matrix for all possible street segments is denoted by $S \in \mathbb{R}^{n\times 2}$.

We define the decision vector as $X^k = [x_1^k, \dots, x_n^k]$, with $x_i^k \in \{0, 1\}, i = 1, \dots, n$ as a binary variable vector, satisfying that $x_i^k = 1$ if and only if we recommend the street segment s_i to the driver during time slot k. Then the constraint $X^k \mathbf{1}_n = 1, k = 1, \dots, T$ must be satisfied, since the driver should be recommended to only one street segment at each time slot, where $\mathbf{1}_n$ is a length n column vector of all 1's.

A. Driving Cost

One design requirement is to reduce the driving cost.

TABLE I

PARAMETERS AND VARIABLES

Parameters	Description
n	the number of candidate street segments for parking
$S \in \mathbb{R}^{n\times 2}$	the position of n street segments
T	the length of the recommendation sequence
$\alpha \in \mathbb{R}^{1\times 2}$	the current position of the user
$\delta \in \mathbb{R}^{1\times 2}$	the destination of the user
$P \in \mathbb{R}^{n\times T}$	the occupancy probability of n street segments for T time slots
$\lambda \in [0, 1]$	the weight factor between the two costs
$X^k \in \{0, 1\}^{1\times n}$	recommendation decision at time slot k

Traveling from current location α to the recommended position will incur a driving time cost associated with the recommendation decision during the first time slot

$$d^1 = dist(\alpha, X^1 S) \tag{12}$$

where $dist(x, y)$ represents the cost from location x to y.

Remark 1. *Generally, $dist(x, y)$ is a distance function which can be defined as anything that represents cost. When real-time traffic information is available to the system, time duration can be used as $dist(x, y)$ to deal with extraordinary events such as road congestion, weather, law, and etc. In real application, such information can be obtained by open resources such as Google Maps APIs [22].*

To consider the future cruising in the decision, we calculate the driving cost during time slot k similarly

$$d^k = dist(X^{k-1}S, X^k S), k = 2, \dots, T \tag{13}$$

But even when the driver arrives at the recommended street segment, he may find that all parking spots there are occupied. Therefore, we need a probabilistic estimation for parking availability. We define the probability that the street segment s_i is occupied during time slot k as $p_i^k = \mathbb{P}(s_i \text{ is occupied at time slot } k)$ and the probability matrix as $P^k = [p_1^k, \dots, p_n^k]^T$.

Remark 2. *Previous works have developed multiple ways to learn parking availability prediction model [19], [20], [21], [23]. The accuracy of the prediction model will affect the results of the recommendation solutions. Predicted values of occupancy information p_i^k depend on the modeling method of the smart parking system. For instance, if the system only applies historical parking patterns to learn the prediction model, p_i^k will not be updated. If the system learns parking availability prediction by both historical data and real-time information such as [19], [23], p_i^k will be updated with real-time sensing data.*

The driving cost throughout the time horizon is defined as the expected total driving distance

$$\begin{aligned} c_d &= \sum_{k=1}^{T} \left(d^k \prod_{l=1}^{k-1} \mathbb{P}(X^l S \text{ is occupied at time slot } l) \right) \\ &= \sum_{k=1}^{T} \left(d^k \prod_{l=1}^{k-1} X^l P^l \right) \end{aligned} \tag{14}$$

B. Walking Cost

If the driver successfully parks at time slot k at parking space $X^k S$, walking distance from there to destination δ is

$$w^k = dist(X^k S, \delta), k = 1, \dots, T \tag{15}$$

The walking cost throughout the time horizon is the expected total walking distance

$$\begin{aligned} c_w = & \sum_{k=1}^{T} \Big(w^k \times \mathbb{P}(X^k S \text{ is not occupied at time slot } k) \\ & \times \prod_{l=1}^{k-1} \mathbb{P}(X^l S \text{ is occupied at time slot } l) \Big) \\ = & \sum_{k=1}^{T} \Big(w^k (1 - X^k P^k) \prod_{l=1}^{k-1} X^l P^l \Big) \end{aligned} \tag{16}$$

C. Problem Formulation

The objective function is the combination of both driving cost and walking cost. However, there exists a trade-off between those two objectives since the two costs mean different for the driver. For example, adding one more mile of walking distance is less acceptable than adding the same amount of driving distance. Thus, we add a weight parameter $\lambda \in [0, 1]$ when summing up the costs related to both objectives. All the parameters and variables are listed in Table I.

$$\begin{aligned} \min_{X} \quad & J = \lambda c_d + (1 - \lambda) c_w \\ \text{s.t.} \quad & d^1 = dist(\alpha, X^1 S) && (12) \\ & d^k = dist(X^{k-1} S, X^k S), k = 2, \dots, T && (13) \\ & w^k = dist(X^k S, \delta), k = 1, \dots, T && (15) \\ & X^k \mathbf{1}_n = 1, k = 1, \dots, T \\ & X^k \in \{0, 1\}^{1 \times n} \end{aligned} \tag{17}$$

The user will get a parking guidance from the beginning of the trip, and the solution will be updated at every decision point according to the prediction model. At each decision point, a sequence of parking locations is provided as: $X^1, \dots, X^T$. If the user cannot find a vacant parking spot with the recommendation X^1, he may follow the next decision X^2. The process of the on-street parking recommendation is summarized in Algorithm 1.

Remark 3. *When integer programming is not efficient enough for a large-scale user case regarding to the problem size, one relaxation method is replacing the constraint $x_i^k \in \{0, 1\}, \forall k, i$ by $0 \le x_i^k \le 1$ With this approximation, every element of X_k is not restricted to a binary variable. After getting an optimal solution X^k of the relaxed form of (17), set the largest value of X^k to 1, and the others to 0.*

VI. Multi-User Competition

So far, we have described a system where one single user is trying to find a vacant parking spot. However, competition between drivers occurs when multiple users simultaneously receive an indication of the same parking space.

Since we adopt a probability model to predict parking availability, the system can be improved by introducing multi-user information. For user j denoted by u_j, the probability that the i-th street segment s_i is occupied during time slot k is $p_i^k(u_j)$, which can be learned by both historical and real-time data with a feature set mentioned in Section IV.

$$\begin{aligned} p_i^k(u_j) &= \mathbb{P}(s_i \text{ is occupied at } t_{pred}) \\ &= g(s_i, r_i[t_{prev} : t_{now}], t_{pred}, dow) \\ &= g_i(t_{pred}) \end{aligned} \tag{18}$$

where $g(\cdot)$ represents the prediction model and the input is the feature set discussed in Section IV, t_{pred} is the predicted time that u_j arrives at s_i. We use $g_i(t_{pred})$ to represent the occupancy probability of s_i at time t_{pred} learned by prediction model $g(\cdot)$ for short.

Algorithm 1 On-street parking recommendation

Inputs: current location α, destination δ, length of sequence T, update cycle t_{update}, weight λ

Initialization: candidate street segments S

while user does not successfully park **do**
  if time is the beginning of t_{update} time slot **then**
    update current location α;
    update occupancy probability P;
    solve problem (17);
    push updated sequence to user if necessary
  end if
end while

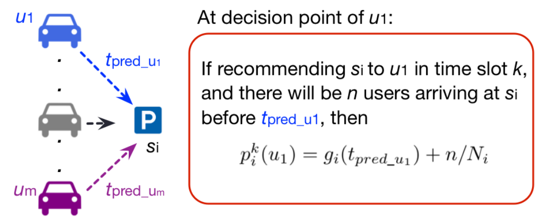


Fig. 5. Multi-user competition

In order to solve the multi-user competition problem, $p_i^k(u_j)$ should be modified with multi-user information when solving the optimization problem of formulation (1):

$$p_i^k(u_j) = g_i(t_{pred}) + \sigma(u_j) \tag{19}$$

where $\sigma(u_j)$ is the multi-user factor that other users will arrive at s_i earlier than u_j.

The principle of our approach is to serve the "early-arrived" user as a priority. At each decision point, the system makes recommendations to all users. For those who will arrive later, the recommendation solutions for them include the influence of those "early-arrived" users to avoid conflict.

For example, if users $u_1, \dots, u_m$ are competing for the same street segment s_i, as shown in Fig. 5, and they are estimated to arrive at s_i at time $t_{pred\_u_1}, \dots, t_{pred\_u_m}$ respectively. At the decision point of u_1, when solving problem (17), we need to get the occupancy probability of s_i at time $t_{pred\_u_1}$. If there will be n users arriving earlier than u_1, then $p_i^k(u_1)$ will be increased by adding the factor that n extra parking spots of s_i will be taken before time $t_{pred\_u_1}$ Thus, a new recommendation sequence with the lowest cost will be updated for u_1 based on the modified occupancy probability. s_i will be recommended to u_1 if it is still a good solution for him, otherwise it will be replaced by another street segment.

To realize such a probability-based approach, a tuple

Algorithm 2 Calculation of multi-user factor

Inputs: user u_j, street segment s_i, predicted time t_{pred}
Output: multi-user factor $\sigma(u_j)$

initialize $\sigma(u_j) \leftarrow 0$;
for each user u_q and $u_q \neq u_j$ **do**
  if $u_q.street == s_i$ and $u_q.t_{arrival} < t_{pred}$ **then**
    // if there is a user who arrives earlier than u_j,
    // increase the number of occupied spots by 1
    $\sigma(u_j) \leftarrow \sigma(u_j) + 1/N_i$;
  end if
end for

($street$, $t_{arrival}$) must be maintained for each user when there is an update, where $street$ is the first street segment in the recommendation sequence, $t_{arrival}$ is the estimated arrival time from the user's current location to $street$ which can be obtained by server according to real-time traffic.

When calculating the cost function, if s_i (with total N_i parking spots) is recommended to u_j during time slot k, the probability $p_i^k(u_j)$ will be added by the multi-user factor $\sigma(u_j)$, which is calculated by Algorithm 2.

VII. Data-driven Evaluations

A. Evaluation of Parking Prediction

Mean Square Error (MSE) is used commonly to measure the prediction accuracy. Consider time series $y(t)$ $(t = 1, \ldots, n)$ together with its predicted series $\hat{y}(t)$, the MSE is defined as

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{t=1}^{n} \big(y(t) - \hat{y}(t)\big)^2 \tag{20}$$

where $y(t)$ and $\hat{y(t)}$ are actual and predicted utilization rates respectively, and n is the total number of street segments.

We use the historical data collected in 20 successive Mondays, from 06/30/2014 to 11/10/2014. The data of the first 19 days are used for training, and the last day for testing.

Fig. 6 shows the MSE calculated for the predictions made at 10:30 AM using the data of 11/10/2014. We compare the performance of three prediction models: (i) historical patterns without real-time information; (ii) AR model I (AR model w.r.t. utilization rate); (iii) AR model II (AR model w.r.t. utilization rate variation).

Since the variance of parking patterns is relatively high even for the same street segment in different days, prediction only using historical data is not accurate enough. The prediction accuracy can be significantly improved by using time series prediction with real-time sensing data. In that case, the MSE increases as the prediction horizon grows because of the accumulated error over time. AR model II which uses trending and detrending techniques has a even higher prediction accuracy, especially in the short time horizon, which is essential to applications such as parking routing and last mile navigation.

B. Simulation Setup

We conduct a data-driven simulation based on the Melbourne Parking Events 2014 data set [18] in two areas. *Area A* is a business district of Melbourne city, including 13 street segments and 152 parking spots in total for on-street parking, while *Area B* is a market place which has 6 street segments

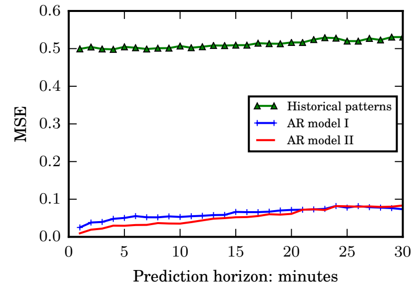


Fig. 6. Comparison of prediction errors with different prediction horizons

with 211 parking spots. Both areas have high parking demand during the day, we chose those areas to demonstrate the capacity of our solutions.

During the simulation, the real-time occupancy information is obtained from the data set on the specific date. The initial locations of users were generated arbitrarily in Melbourne city. The update period t_{update} is set as 5 minutes since the average time duration that the occupancy status of a street segment is unchanged is 4.34 minutes according to the analysis of the data set. The parking availability prediction model is learned from the dataset of the first 10 months of 2014, and the data of November is used for testing.

In the evaluation, the time duration is chosen to represent the distance function $dist(x, y)$, which can be obtained by Google Maps Distance Matrix API [24], in order to take the real-time traffic into consideration. Therefore, the total time delay, which starts from searching for a parking spot to arriving at the exact destination, should be measured to reflect the approach's performance.

Number of searched street segments (n_s): the number of street segments the user has searched before he successfully finds a vacant parking spot.

Extra Searching Delay (ESD): the searching cost caused by the failure of the recommendation. Since the user has searched n_s street segments before he finds a vacant parking spot (i.e., the parking space is in the n_sth street segment of the recommendation sequence), EST is calculated by

$$\text{ESD} = \begin{cases} 0 & \text{if } n_s = 1 \\ \sum_{k=2}^{n_s} t(X^{k-1}S, X^k S) & \text{if } n_s = 2, \ldots, T \end{cases}$$

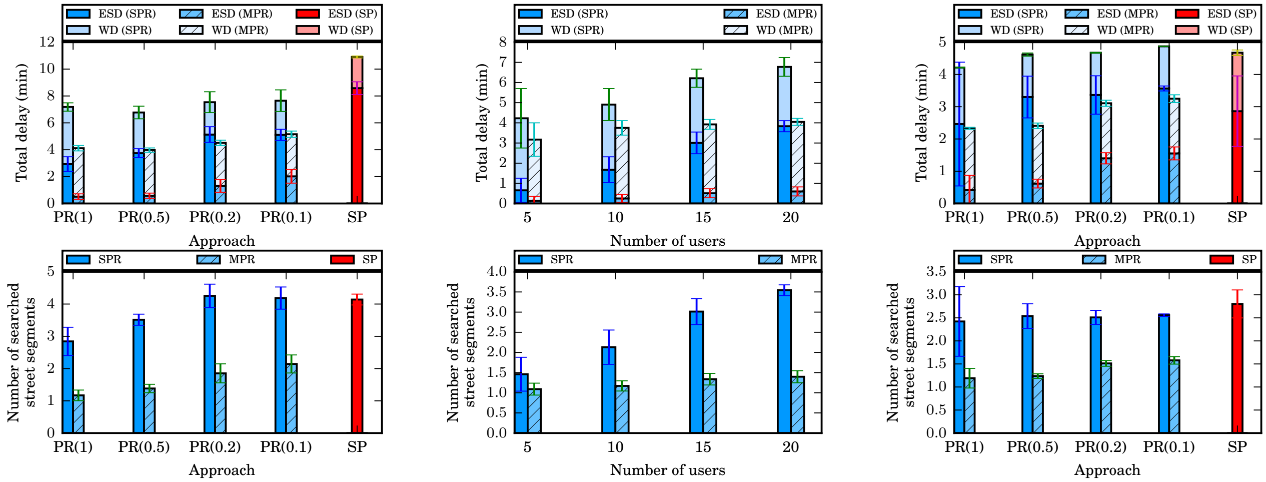
Walking Delay (WD): the degree of closeness from the parking location to the destination, which is obtained by

$$\text{WD} = dist(X^{n_s} S, \delta)$$

C. Experimental Results

We compare our approach with the state-of-the-art Smart Parking (SP) algorithm in [11], which is a dynamic resource allocation algorithm based on price and distance to destination. Here we assume the price for every parking spot is the same. For our parking recommendation (PR), we tested the approach with or without multi-user factor, as multi-user parking recommendation (MPR) and single-user parking recommendation (SPR), respectively. We also compare the performance with different values of parameter λ.

To test the performances of our solutions under high demand, we chose the center of business district (*Area A*) as the destination of 20 users. For each method, we did 600



(a) Performance comparison between our Parking Recommendation (PR) and the Smart Parking (SP) algorithm. Parameter λ in PR(λ) indicates the importance of driving cost.

(b) Performance comparison in *Area A* with different number of users. When there is high competition, MPR solution which considers multi-user factor performs better.

(c) Performance comparison in *Area B* under a disruptive event like a concert. Our MPR solution is robust to volatile disturbance.

Fig. 7. Evaluation Results

experiments (20 for each day) over 1 month (November, 2014) during the rush hours (between 11:00 AM and 2:00 PM). The hidden users are automatically introduced by running the data-driven simulations. Then we calculated their average cost and number of searched street segments. The experimental results are shown in Fig. 7(a), the standard deviations are also plotted.

SP vs PR: During non-rush hours, since the vacant parking spots are sufficient, the greedy algorithm may preform well when minimizing the walking delay. However, during rush hours, its performance deteriorates significantly, since those streets near the destination may be fully occupied. Thus it causes more extra searching delay and the driver has to search more street segments.

Influence of λ: The parking recommendation approach has a weight parameter $\lambda \in [0, 1]$ representing the weight of the driving cost. A larger λ will reduce the cost of searching because it tends to find the most possible available place to park. While the recommended parking location, however, may be farther to the destination. A smaller λ means the weight of the driving cost becomes less important, which returns a solution with a smaller walking delay, but a larger extra searching delay and more searched street segments. By properly choosing the parameter λ (e.g. $\lambda = 0.5$ in this case), although the walking delay is more than the greedy algorithm, the total delay can be significantly reduced by 63.8% and less streets will be searched using MPR.

SPR vs MPR: When there are multiple users using our parking guidance system for parking guidance in the same region, then those users may cause competition for the same parking spot. As demonstrated in Section VI, we improve our approach by modifying the occupancy probability of one specific street segment with the multi-user factor. SPR may recommend to multiple users with the same street segment that does not have enough vacant parking spots, and then competition will incur. After the recommended street segment becomes fully occupied, those who arrive later have to continue searching, which causes more extra searching distance and more searched street segments. With MPR approach, the multi-user information has been incorporated to make decisions, and therefore the influence of multi-user competition can be eliminated.

We tested the impact of the number of competing users: 5, 10, 15 and 20 users to compare their performances. We set $\lambda = 0.5$ for our algorithms. The experimental results are shown in Fig. 7(b). With a small number of users (e.g., 5 users), there will be sufficient vacant parking spots for them, so both approaches perform well. When the number of users increases, however, the performance of SPR deteriorates significantly, while the total delay of MPR increases slightly: total delay reduced by 40.9% with 20 users.

High Competition under Disruptive Events: in this experiment, we simulated a disruptive event, such as a concert, in *Area B*, we generated 100 users to drive to this area at similar time to test the performance of our approach under high competitions. The experimental results are shown in Fig. 7(c). We can see that MPR consistently achieves better total delay and searched street segments than SP: the total delay can be reduced by 47.2% compared with SPR, and 48.7% compared with SP (with $\lambda = 0.5$).

VIII. Related Work

This paper is related to both smart parking systems and transportation assignment algorithms. We firstly discuss research on smart parking systems. There are a large number of research works on different aspects of intelligent parking systems, which include occupancy detection [4], [5], system development [6], dynamic pricing [20], [25], and even shared service design [7]. Recently traffic authorities in many cities have developed Parking Guidance Information (PGI) systems, which typically employ sensor networks to detect parking occupancy and provide real-time parking service. However, most of the current PGI systems only publish the parking information to drivers directly [26], [9], [27]. A few systems are able to select an optimal parking space according to drivers' preferences and current state information [11], [28], [29], but drivers may not actually find vacant parking spots by merely following the one-time recommendation from the guidance system. Therefore, these existing PGI systems are not "smart" enough. Our solution provides sequences of parking recommendations to users dynamically until they find parking.

There are many research on transportation assignment algorithms that aim to allocate parking spots to reduce parking competition: a scenario when multiple users are looking for parking in a crowded area. Such parking competition leads to a phenomenon called "multiple-car-chasing-single-space", which may cause severe traffic congestion [25]. To address this problem, some researchers formulated dynamic resource allocation problems [25], [30], [11]. Specifically, Basu *et al.* [25] presented a travel distance based approach, which is to assign the parking spot to the nearest user. However, this work assumes that the nearer driver will arrive earlier, which ignores the real-time traffic information. Geng *et al.* [11] adopted a queueing model which allocates parking spaces to drivers with reservation. Mouskos *et al.* [30] formulated a max-min problem that considers parking rate of a specific parking lot and the travel time cost to the final destination. These reservation-based designs are suitable for off-street parking resources. For on-street parking resources, however, it requires extra hardware infrastructure and is thus costly to realize in the citywide traffic system.

There are a few papers on on-street parking employ a game-theoretical approach [14], [15]. These works provide valuable insights, but they are difficult to apply in reality. Different from these designs, by building a spatiotemporal model of parking distributions with large scale smart meter data, our receding horizon optimization approach can provide online parking recommendation to accommodate variations in traffic and requests with little overhead. In [16], the authors formulated a specific type of Traveling Salesman Problem. But this design does not consider the multi-user competition problem which is essential for on-street parking. Our work provides sequences of parking recommendation to users and also addressed the parking competition, matching dynamic parking demand and supply effectively.

IX. Conclusion

In this paper, we design a receding horizon control approach to coordinate on-street parking behavior of urban users based on occupancy prediction model and real-time travel and occupancy data. Our framework incorporates multiple objectives and balances the parking supply and demand in a dynamic and distributed manner. This design is not exclusive to different traffic models and searching algorithms. Trace driven simulation results show that our approach achieves up to 63.8% delay reduction compared with existing solutions.

Acknowledgment

The authors would like to thank Jie Zhang for the help in the implementation. J. Gao and K. S. Liu would like to acknowledge support through NSF CCF-1535900, CNS-1618391 and DMS-1737812. Shan acknowledges support from NSF awards CNS 1553273 and CNS 1463722.

References

[1] J. Belissent, "Getting clever about smart cities: new opportunities require new business models," 2010.
[2] D. C. Shoup, "Cruising for parking," *Transport Policy*, 2006.
[3] D. Shoup, "The high cost of free parking," 2011.
[4] J. Cherian, J. Luo, H. Guo, S. Ho, and R. Wisbrun, "Parkgauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics," in *IEEE MDM 2016*.
[5] G. Grassi, K. Jamieson, P. Bahl, and G. Pau, "Parkmaster: an in-vehicle, edge-based video analytics service for detecting open parking spaces in urban environments," in *SEC*, 2017.
[6] H. Wang and W. He, "A reservation-based smart parking system," in *INFOCOM WKSHPS*, 2011.
[7] W. Griggs, J. Y. Yu, F. Wirth, F. Husler, and R. Shorten, "On the design of campus parking systems with qos guarantees," *IEEE Transactions on Intelligent Transportation Systems*, 2016.
[8] Y. Zheng, J. Cao *et al.*, "An intelligent car park management system based on wireless sensor networks," in *First International Symposium on Pervasive Computing and Applications*. IEEE, 2006.
[9] R. Lu, X. Lin, H. Zhu, and X. S. Shen, "Spark: a new vanet-based smart parking scheme for large parking lots," in *INFOCOM 2009*.
[10] C. Jin, L. Wang, L. Shu, Y. Feng, and X. Xu, "A fairness-aware smart parking scheme aided by parking lots," in *2012 ICC*.
[11] Y. Geng and C. G. Cassandras, "New smart parking system based on resource allocation and reservations," *IEEE Transactions on Intelligent Transportation Systems*, 2013.
[12] Y. Zheng, S. Rajasegarar, and C. Leckie, "Parking availability prediction for sensor-enabled car parks in smart cities," in *2015 ISSNIP*, 2015.
[13] T. Rajabioun and P. A. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Transactions on Intelligent Transportation Systems*, 2015.
[14] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin, "Parking slot assignment games," in *GIS '11*.
[15] D. Mackowski, Y. Bai, and Y. Ouyang, "Parking Space Management via Dynamic Performance-Based Pricing," *ArXiv e-prints*, 2015.
[16] E. H.-K. Wu, J. Sahoo, C.-Y. Liu, M.-H. Jin, and S.-H. Lin, "Agile urban parking recommendation service for intelligent vehicular guiding system," *Intelligent Transportation Systems Magazine, IEEE*, 2014.
[17] V. F. Farias and B. Van Roy, "Approximation algorithms for dynamic resource allocation," *Operations research letters*, 2006.
[18] *Parking Events 2014*, https://data.melbourne.vic.gov.au/Transport-Movement/Parking-Events-2014/mq3i-cbxd.
[19] Y. Zheng, S. Rajasegarar, and C. Leckie, "Parking availability prediction for sensor-enabled car parks in smart cities," in *IEEE ISSNIP*, 2015.
[20] T. Rajabioun and P. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *Intelligent Transportation Systems, IEEE Transactions on*, 2015.
[21] M. Caliskan, A. Barthels, B. Scheuermann, and M. Mauve, "Predicting parking lot occupancy in vehicular ad hoc networks," in *IEEE VTC*, 2007.
[22] *Google Maps APIs*, https://developers.google.com/maps/.
[23] A. Klappenecker, H. Lee, and J. L. Welch, "Finding available parking spaces made easy," *Ad Hoc Networks*, 2014.
[24] *Google Maps Distance Matrix API*, https://developers.google.com/maps/documentation/distance-matrix/.
[25] P. Basu and T. D. Little, "Networked parking spaces: architecture and applications," in *IEEE VTC*, 2002.
[26] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," in *3rd international workshop on Vehicular ad hoc networks*, 2006.
[27] *BestParking*, http://www.bestparking.com/.
[28] T. Rajabioun, B. Foster, and P. Ioannou, "Intelligent parking assist," in *IEEE MED*, 2013.
[29] J. Fu, Z. Chen, R. Sun, and B. Yang, "Reservation based optimal parking lot recommendation model in internet of vehicle environment," *Communications, China*, 2014.
[30] K. Mouskos, J. Tvantzis, D. Bernstein, and A. Sansil, "Mathematical formulation of a deterministic parking reservation system (prs) with fixed costs," in *IEEE MELECON*, 2000.