# New "Smart Parking" System Based on Resource Allocation and Reservations

Yanfeng Geng, *Student Member, IEEE*, and Christos G. Cassandras, *Fellow, IEEE*

***Abstract*—We propose a novel "smart parking" system for an urban environment. The system assigns and reserves an optimal parking space based on the driver's cost function that combines proximity to destination and parking cost. Our approach solves a mixed-integer linear programming (MILP) problem at each decision point defined in a time-driven sequence. The solution of each MILP is an optimal allocation based on current state information and is updated at the next decision point with a guarantee that there is no resource reservation conflict and that no driver is ever assigned a resource with a cost function higher than this driver's current cost function value. Based on simulation results, compared with uncontrolled parking processes or state-of-the-art guidance-based systems, our system reduces the average time to find a parking space and the parking cost, whereas the overall parking capacity is more efficiently utilized. We also describe full implementation in a garage to test this system, where a new light system scheme is proposed to guarantee user reservations.**

***Index Terms*—Dynamic resource allocation, mixed-integer linear programming (MILP), parking guidance and information (PGI), reservation, smart parking.**

## I. Introduction

THE motivation for this paper is provided by the need to reduce traffic in urban settings caused by vehicles searching for parking. On a daily basis, it is estimated that 30% of traffic congestion in an urban downtown area is caused by vehicles cruising for parking space, and it takes the driver an average of 7.8 min to find a parking space [2]. This not only causes waste of time and fuel for drivers looking for parking but also contributes to additional waste of time and fuel for other drivers as a result of traffic congestion. For example, it has been reported [22] that, for over one year in a small Los Angeles business district, cars cruising for parking created the equivalent of 38 trips around the world, burning 47 000 gal of gasoline and producing 730 tons of carbon dioxide.

There has been considerable work in studying parking behaviors and improving parking efficiency. During the early stages of such research, a number of parking models were built to understand and replicate parking choice behavior, such as CLAMP [19], PARKSIM [33], PARKAGENT [4], multilayer [8], and others [26]. In most of these models, competitive alternatives are reasonably well known in advance to the decision maker (driver).

Over the past two decades, traffic authorities in many cities have developed so-called parking guidance and information (PGI) systems for better parking management. PGI systems present drivers with dynamic information on parking within controlled areas and direct them to vacant parking spots. Parking information may be displayed on variable-message signs (VMS) at major roads, streets, and intersections, or it may be disseminated through the Internet [12], [23], [24]. PGI systems are based on the development of autonomous vehicle detection and parking space monitoring, typically through the use of sensors placed in the vicinity of parking spaces for vehicle detection and surveillance [6]. These sensors can be classified as either "in-roadway" or "over-roadway" [18]. In-roadway sensors are either embedded in the pavement or taped to the surface of the roadway; examples include loop detectors, pneumatic road tubes, piezoelectric cables, etc. Over-roadway sensors are mounted above the surface of the roadway; examples include video, image, and acoustic signal processors [32]; microwave radar [30]; ultrasonic [17], magnetic, and passive infrared sensors [31]; and radio-frequency identification (RFID) readers [5]. However, it has been found that in using PGI systems, system-wide reductions in travel time and vehicle benefits may be relatively small [25], [29]. Building upon the objectives of PGI systems, e-parking is an innovative platform that allows drivers to obtain parking information before or during a trip and to reserve a parking spot [20]. Drivers access the central system via a cellular phone or the Internet. Bluetooth technology recognizes each car at entry points and triggers automatic reservation checking and parking payment [15]. More reservation-based parking systems are described in [27] and [28].

Researchers also find that traffic congestion can be alleviated by controlling the parking price [24]. For example, in San Francisco (SFPark), there are already time- or demand-dependent parking fees to achieve the right level of parking availability in different areas [21]. Dynamic parking negotiation is discussed in [7], where drivers may negotiate to find better and cheaper parking spaces.

Although current parking guidance systems increase the probability of finding vacant parking spaces, they have several shortcomings [10], [11]. First, drivers may not actually find vacant parking spots by merely following the guidance system. In

Manuscript received September 23, 2012; revised January 5, 2013; accepted March 6, 2013. Date of publication April 2, 2013; date of current version August 28, 2013. This work was supported in part by the National Science Foundation under Grant EFRI-0735974 and Grant CNS-1239021, by the Air Force Office of Scientific Research under Grant FA9550-09-1-0095 and Grant FA9550-12-1-0113, by the Department of Energy under Grant DE-FG52-06NA27490, by the Office of Naval Research under Grant N00014-09-1-1051, by the Army Research Office under Grant W911NF-11-1-0227, and by the Office of Technology Development, Boston University, Boston, MA, USA. The Associate Editor for this paper was M. C. Zhou.

The authors are with the Division of Systems Engineering and the Center for Information and Systems Engineering, Boston University, Boston, MA 02446 USA (e-mail: gengyf@bu.edu; cgc@bu.edu).

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

Digital Object Identifier 10.1109/TITS.2013.2252428

essence, such systems change driver behavior from searching to competing for parking: More drivers go toward the same available parking spots, and it is possible that none is free by the time some drivers arrive, thus forcing replanning and competition for other spots. Although there exist some smartphone applications for drivers to check real-time parking information using their mobile phone [23], there are also safety issues associated with drivers watching parking updates while driving. Second, even if a driver is successfully guided to a parking space, such a system increases the probability of finding *any* parking space at the expense of missing the opportunity for a *better* space. For example, a driver may pay to park at an off-street parking spot but miss the chance to obtain a nearby free on-street parking spot that may better serve him. Third, from the traffic authority point of view, parking space utilization becomes imbalanced: Parking spaces for which information is provided are highly utilized and cause higher traffic congestion nearby, whereas other parking spaces may be routinely left vacant. In general, guidance systems do not solve the basic parking problem. Even worse, they may cause new traffic congestion in areas where parking spaces are monitored.

In this paper, we propose a new concept for a "smart parking" system. This system explicitly *allocates and reserves optimal parking spaces* to drivers, as opposed to simply *guiding* them to a space that may not be available by the time it is reached. The allocation is based on each user's objective function that combines proximity to destination and parking cost while also ensuring that the overall parking capacity is efficiently utilized. The *reservation* in our "smart parking" system is different from that in the e-parking platform and others earlier mentioned. The latter only involves garage space reservations, and there is no attempt at any form of optimality, whereas in our "smart parking" system, drivers may reserve both off-street and on-street parking spaces, which are selected to be optimal based on a well-defined objective function structure.

In our problem, a key feature is that each driver has specific requirements and that only a subset of resources (parking spots) can satisfy them. This is similar to the skills-based routing (SBR) problem encountered in telephone call centers, where calls are routed based on the skills required for a server to respond to the call [1], [9]. Whereas, in SBR, a server remains assigned to a call until its completion, in "smart parking," we allow parking spaces to be *reallocated* so that a driver can continuously upgrade the resource assigned to him until it is physically occupied. Even without this complicating feature, however, dynamic routing problems in multiclass multipool call centers are outside the reach of exact analytical methods. Related research has focused on various forms of approximations to bypass the high dimensionality involved in determining optimal routing policies. For example, Koole and Pot [16] used approximate dynamic programming to solve a specific structured multiskill call center routing problem. Some work considers these problems in a heavy-traffic regime [14], where system utilization approaches one. Multiclass single-pool systems in this regime are analyzed in [3]. Gurvich and Whitt [13] also proposed a routing method for multiclass multipool systems based on a fixed-queue-ratio strategy. In this paper, we view the "smart parking" allocation process as a sequence of mixed-integer linear programming (MILP) problems solved over time at specific decision points.

The rest of this paper is organized as follows: In Section II, we introduce the framework of our "smart parking" system. In Section III, we describe the dynamic resource-allocation model we use and formulate the MILP problem solved at every decision point over time. Simulations based on a case study involving parking resources at Boston University, Boston, MA, USA, are given in Section IV. A garage implementation is described in Section V. Finally, we conclude and discuss future work in Section VI.

## II. "Smart Parking" System

Here, we describe the "smart parking" system framework and its operation.

### A. System Framework

Our proposed "smart parking" system takes the basic structure of PGI systems as one component. In addition, it includes a Driver Request Processing Center (DRPC) and a Smart Parking Allocation Center (SPAC). Fig. 1 shows this framework. The Parking Resource Management Center (PRMC) collects and updates all real-time parking information and disseminates it via VMS or the Internet (basic functions of PGI systems). The DRPC gathers driver parking requests and real-time information (i.e., car location), keeps track of the driver allocation status, and sends back the assignment results to drivers. Based on driver requests and parking resource states, the SPAC makes assignment decisions and *allocates and reserves* parking spaces for drivers.

As we can see, compared with PGI systems, the additional cost of these two new components is minimal. Only one or two servers are required to carry out computation and for user data storage.

The basic allocation process is described as follows. Drivers who are looking for parking spots send requests to the DRPC. A request is accompanied by two requirements: a constraint (upper bound) on parking cost and a constraint (upper bound) on the walking distance between a parking spot and the driver's actual destination. It also contains the driver's basic information, such as license number, current location, car size, etc. The SPAC collects all driver requests in the DRPC over a certain time window and makes an overall allocation at decision points in time, seeking to optimize a combination of driver-specific and system-wide objectives. An assigned parking space is sent back to each driver via the DRPC. If a driver is satisfied with the assignment, he/she has the choice to reserve that spot. Once a reservation is made, the driver still has opportunities to obtain a better parking spot (with a guarantee that it can never be worse than the current parking spot) before the current assigned spot is reached. The PRMC then updates the corresponding parking spot from *vacant* to *reserved* and provides the guarantee that other drivers have no permission to take that spot. If a driver is not satisfied with the assignment (either because of limited resources or because of his own overly restrictive parking requirements) or if he/she fails to accept it for any other reason,

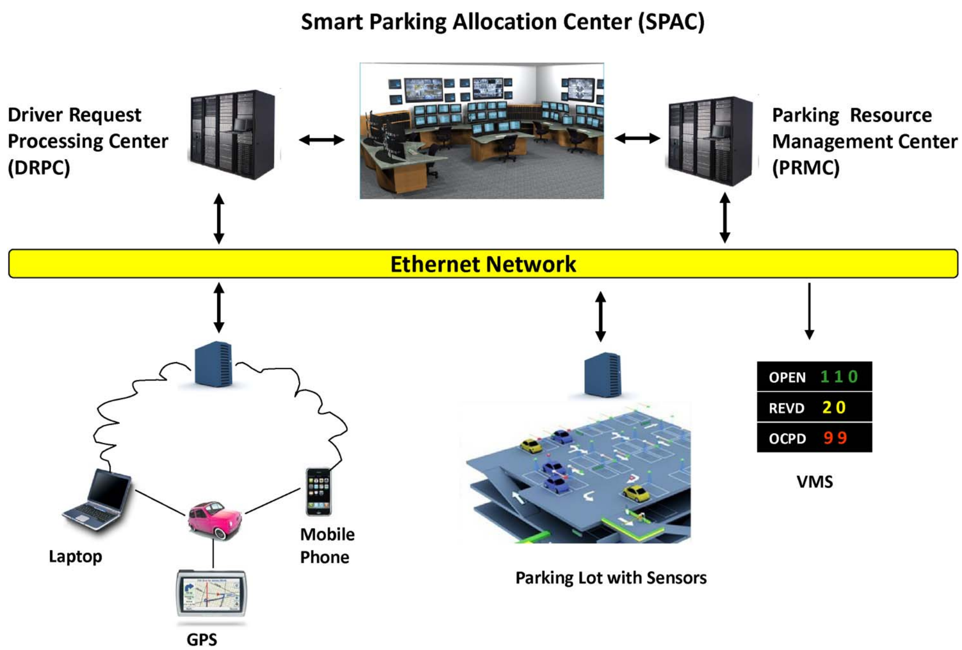



Fig. 1. "Smart parking" framework overview.

he/she has to wait until the next decision point. During intervals between allocation decisions made by the center, drivers with no parking assignment have the opportunity to change their cost or walking-distance requirements, possibly to increase the chance to be allocated if the parking system is highly utilized. (It is, of course, possible that no parking space is ever assigned to a driver.)

### *B. System Realization*

The realization of such a "smart parking" system relies on four main requirements.

*1) Parking Space Detection:* First of all, the system relies on the availability of real-time parking information, based on which it makes and upgrades allocations for drivers. As already mentioned, current sensing technologies provide several options to monitor parking spaces.

Moreover, whenever the system must make an allocation, it requires location information on all vehicles with pending requests. Based on this information, it estimates the traveling time to an allocated spot and provides driving directions to it. Current vehicle tracking devices/systems provide solutions to this problem. Vehicle tracking systems combine GPS tracking technology with flexible advanced mapping and reporting software. A vehicle tracking device is installed on a vehicle, which collects and transmits tracking data via a cellular or satellite network. The system receives real-time vehicle tracking updates, including location, direction, speed, idle time, start/stop, and so on. This technology has been widely used in bus systems.

*2) V2I and I2V Communication:* The second requirement involves an effective two-way communication between vehicles and the allocation center (infrastructure): vehicle-to-infrastructure (V2I) and infrastructure-to-vehicle (I2V). In our "smart parking" system, V2I communication involves drivers sending their parking requests, providing driver information, and confirming reservation to the system. I2V communication includes the DRPC sending allocation results, driving directions, and payment details back to vehicles. Cellular networks (CNs) are usually applied in V2I and I2V solutions, i.e., drivers interact with the system through their mobile phones.

In our implementation, we have developed a smartphone application through which drivers interact with the "smart parking" system. Using the application, drivers may log in the system with a unique ID, associated with which is a driver's general information, such as license number, credit card number, car size, etc. The ID is registered by the driver, and the DRPC maintains a database to store the driver's basic information. In the application, drivers also have the option to choose their destination, walking-distance preference, and parking cost tolerance. After the driver finishes all settings and sends out the request, the system will send back parking allocation results based on his parking preferences and the state of the system.

There are three kinds of allocation results as follows: 1) If the system fails to find a parking space for the driver, then a notification asks the driver to wait for the next allocation time. A detailed explanation is also provided regarding the failed allocation; for example, there are no vacant parking spaces, the driver's requirements are too strict, or the driver is too far away from his destination. The driver may then either release his parking request by changing his preferences to increase the chance to be allocated or simply do nothing but wait. 2) If a parking space is allocated to the driver but he/she is not satisfied with it, then he/she can reject the allocation and adjust his requirements. However, by doing this, he/she takes the risk that he/she may not be allocated a space at the next decision time. To prevent drivers from constantly rejecting successful allocations and adjusting requirements for better parking spots or to prevent drivers from always providing extremely strict conditions at the beginning and gradually relaxing them later, the system may charge an increasing fee if the number of requests exceeds a certain threshold. 3) If the driver is satisfied with the result, then the system reserves that space for him, and the application shows the driving directions to the reserved parking space. While he/she is driving, the system may notify him of a better parking spot based on his real-time position. The driver needs to respond and tell the system whether he/she accepts it or not. When the driver arrives at the parking spot, he/she needs to confirm parking at the allocated spot. All these driver responses are simply done by pushing a button in the application. When the car leaves the parking spot, a summary of charges is sent to him.

Notice that both V2I and I2V communication are implemented through a smartphone application, and data are transmitted through the CN. Drivers may reserve a parking space before a trip and interact with the system by simply pushing buttons on the smartphone, thus not causing distraction from driving.

*3) Reservation Guarantee:* Parking reservations are a key feature of the "smart parking" system. To implement this function, when a parking spot is reserved by the driver, the system must guarantee that this will not be taken by other vehicles. For off-street parking resources, it is relatively easy to prevent drivers with no reservation from taking the spot that has been reserved by someone else. The system can do ID checking (with RFID technology) at the gate of a garage or a parking lot. If the driver has made a reservation, then the gate opens, and a space number is provided to him. If the driver made no reservation, then he/she either may be allowed to park if there are empty unreserved spaces or is blocked from entering.

For on-street parking resources, the reservation scheme is more complicated because there is no central ID checking location so that drivers may park at any space if it is vacant. One method is through wireless technology interfacing a vehicle with hardware that makes a space accessible only to the driver who has reserved it. Examples include gates, "folding barriers," and obstacles that emerge from and retract to the ground under a parking spot; these are wirelessly activated by devices on-board vehicles, similar to mechanisms for electronic toll systems. However, this method is relatively expensive, and the hardware is not easy to install and maintain. A "softer" scheme is to use a light system placed at each parking space, where different colors indicate different parking space states. In our system, we use a GREEN light to indicate that a vacant parking spot is available for any driver, a RED light to indicate that the spot is reserved by other drivers, a YELLOW (or blinking YELLOW for increased visibility) light to attract a driver in the vicinity who has reserved that space, and a blinking RED light to notify a driver who is parking at a space reserved by someone else. An LED light with these three colors is connected to and controlled by an IRIS sensor node (also referred to as "mote") placed at each parking space. When a driver is approaching the parking space reserved for him, this is automatically detected by the GPS data sent from his smartphone through our "smart parking" application. (Alternatively, the driver can explicitly notify the system.) The system then sends a command to the IRIS mote, which switches the light at his reserved spot from RED to YELLOW (or blinking YELLOW). The driver should then be able to recognize his reserved spot and park there. After parking, the light goes off until the car leaves, and it returns to its GREEN or RED state if the parking space is reserved. If a driver violates the rule and parks at a space reserved by someone else, then the blinking RED provides a warning, and the driver should leave. If he/she does not leave, the system knows which space is occupied and will tow the car or issue a ticket; in the meantime, the system makes a new assignment for the driver who had actually reserved that spot. If the second assignment is worse than the previous assigned spot, then the driver receives some compensation, which may come from the violator's fine.

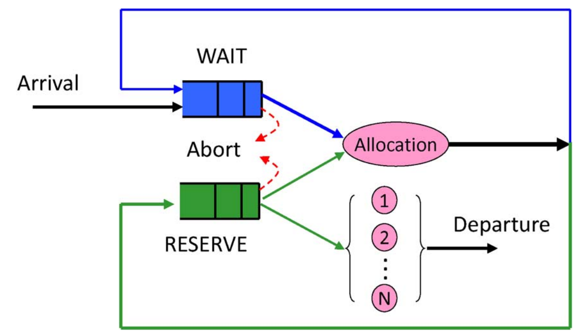



Fig. 2. Queueing model for dynamic resource allocation.

*4) Optimal Allocation:* One of the benefits of the "smart parking" system is that it determines the best parking space for each driver. This is done through an efficient allocation algorithm executed at the SPAC (see Fig. 1). In what follows, we will concentrate on the methodology that enables us to make optimal parking space allocations and reservations.

## III. Dynamic Resource Allocation

For the sake of generality, we will employ the term "user" when referring to drivers or vehicles and the term "resource" when referring to parking spaces. We adopt a queueing model for the problem, as shown in Fig. 2, where there are $N$ resources, and every user arrives randomly and independently to join an infinite-capacity queue (labeled WAIT) and waits to be assigned a resource if possible. At each decision point, the system makes allocations for all users in both the waiting queue and the queue of users (labeled RESERVE) who have already been assigned and have reserved a resource from a prior decision point. If a user in WAIT is successfully assigned a resource, he/she joins the RESERVE; otherwise, he/she remains in WAIT. A user in RESERVE may be assigned a different resource after a decision point and remains in this queue until he/she can physically reach the resource and occupy it. A user leaves the system after occupying a resource for some amount of time at which point the resource becomes free again.

At each decision point indexed by $k$, we define the state of the allocation system, i.e., $X(k)$, $k = 1, 2, \ldots$, and the state of the $i$th user, i.e., $S_i(k)$, $i = 1, 2, \ldots$, as explained next. First, we define

$$X(k) = \{W(k), R(k), P(k)\} \tag{1}$$

where $W(k) = \{i : \text{user } i \text{ is in the WAIT queue}\}$, $R(k) = \{i : \text{user } i \text{ is in the RESERVE queue}\}$, and $P(k) = \{p_1(k),$

$\dots, p_N(k)\}$ is a set describing the state of the $j$th resource with $p_j(k)$ denoting the number of free parking spaces at resource $j$, $j = 1, \dots, N$. (This is possibly $>1$ if a resource models a group of parking spaces, e.g., a parking garage, rather than an individual space.)

We assume that each resource has a known location associated to it, which is denoted by $y_j \in Z \subset \mathbb{R}^2$ in 2-D Euclidean space. We also define

$$S_i(k) = \{z_i(k), r_i(k), q_i(k), \Omega_i(k)\} \tag{2}$$

where $z_i(k) \in Z \subset \mathbb{R}^2$ is the location of user $i$, $r_i(k) \in \mathbb{R}^+$ is the total time that user $i$ has spent in the RESERVE queue up to the $k$th decision point ($r_i(k) = 0$ if $i \in W(k)$), and $q_i(k)$ is the reservation status of user $i$, i.e.,

$$q_i(k) = \begin{cases} 0, & \text{if } i \in W(k) \\ j, & \text{if user } i \text{ has reserved resource } j. \end{cases} \tag{3}$$

Finally, $\Omega_i(k)$ is a feasible resource set for user $i$, i.e., $\Omega_i(k) \subseteq \{1, \dots, N\}$, depending on the requirements set forth by this user regarding the resource requested. In general, $\Omega_i(k)$ may be a set that is specified by each user at each decision point; however, for the specific parking problem we are interested in, we will define $\Omega_i(k)$ in terms of *attributes* associated with user $i$ and defined as follows.

We associate two attributes to user $i$. The first, which is denoted by $D_i$, is an upper bound on the distance (walking distance or walking time) between the resource that the user is assigned and his actual destination $d_i \in Z \subset \mathbb{R}^2$. If the user is assigned resource $j$ located at $y_j$, let $D_{ij} = \|d_i - y_j\|$, where $\|\cdot\|$ is a suitable distance metric. Then, the constraint

$$D_{ij} \leq D_i \tag{4}$$

defines a requirement that contributes to the determination of $\Omega_i(k)$ by limiting the set of feasible resources to those that satisfy (4).

The second attribute for user $i$, which is denoted by $M_i$, is an upper bound on the cost that this user is willing to tolerate for the benefit of reserving and subsequently using a resource. The actual cost depends on the specific pricing scheme that is adopted by the allocation system and may include a flat fee for reserving a resource, a fee dependent on the total reservation time, and subsequently, a fee for occupying the resource. Our approach does not depend on the specific pricing scheme used, but we will assume that each user cost is a monotonically nondecreasing function of the total reservation time $r_i(k)$, user expected occupancy time $c_i$, and a function of the traveling time from the user location at the $k$th decision time, i.e., $z_i(k)$, to a resource location $y_j$. Let $s_{ij}(k) = \|z_i(k) - y_j\|$ be this distance, and define the traveling time

$$t_{ij}(k) = f\left(s_{ij}(k), \omega\right)$$

where $\omega$ is a random vector capturing all stochastic traffic conditions. We use $M_{ij}(r_i(k), t_{ij}(k), c_i)$ to denote the total expected monetary cost for using resource $j$, which is evaluated at the $k$th decision time. Note that $M_{ij}(r_i(k), t_{ij}(k), c_i)$ is an expectation since the actual cost is a random variable that depends on traffic conditions, which determine the time $t_{ij}(k)$ and on the resource occupancy time (e.g., the actual parking time) after the resource is reached. Once a pricing scheme is known, $M_{ij}(r_i(k), t_{ij}(k), c_i)$ can be evaluated if all random variables involved are characterized by known probability distributions. Alternatively, an estimate of $M_{ij}(r_i(k), t_{ij}(k), c_i)$ can be computed. Comparing $M_{ij}(r_i(k), t_{ij}(k), c_i)$ with $M_i$ leads with the constraint

$$M_{ij}\left(r_i(k), t_{ij}(k), c_i\right) \leq M_i. \tag{5}$$

This defines a second requirement that contributes to the determination of $\Omega_i(k)$ by limiting the set of feasible resources to those that satisfy (5). To fully specify $\Omega_i(k)$, we further define

$$\Gamma(k) = \{j : p_j(k) > 0, \quad j = 1, \dots, N\}$$

to be the set of free and reserved resources at the $k$th decision time and set

$$\Omega_i(k) = \{j : M_{ij}(k) \leq M_i, D_{ij} \leq D_i, \quad j \in \Gamma(k)\} \tag{6}$$

where, for simplicity, we have written $M_{ij}(k)$ instead of $M_{ij}(r_i(k), t_{ij}(k), c_i)$. Note that this set allows the system to allocate to user $i$ any resource $j \in \Omega_i(k)$, which satisfies the user's requirements even if it is currently reserved by another user. Thus, resource $j$ may be dynamically reallocated to different users at each decision point until $p_j(k) = 0$, signaling that there is no available resource.

*Remark:* Since $M_{ij}(k)$ is generally an estimate of the cost a user incurs, it is subject to noise contributed by random traffic events and, therefore, so is set $\Omega_i(k)$, as defined in (6). This implies that resource $j \in \Omega_i(k)$ may, in fact, be such that $j \notin \Omega_i(k + l)$ for some $l > 0$. Indeed, it is possible that $\Omega_i(k) \neq \varnothing$, whereas $\Omega_i(k + l) = \varnothing$. In such cases, a user may perceive as unfair the fact that he/she is assigned a feasible resource that ultimately becomes infeasible subject to his requirements. We will assume that this happens as a result of uncontrollable random events, in which case, the user must re-enter the allocation system with new $D_i$ and $M_i$ requirement parameters.

We can now concentrate on defining an objective function, which we will seek to minimize at each decision point by allocating resources to users. We use a weighted sum to define user $i$'s cost function, i.e., $J_{ij}(k)$, if he/she is assigned to resource $j$, as follows:

$$J_{ij}(k) = \lambda_i \frac{M_{ij}(k)}{M_i} + (1 - \lambda_i) \frac{D_{ij}}{D_i} \tag{7}$$

where $\lambda_i \in [0, 1]$ is a weight that reflects the relative importance assigned by the user between *cost* and *resource quality*. In the case of parking, resource quality is measured as the walking distance between the parking spot the user is assigned and his actual destination.

To capture the essence of “smart parking,” the objective of the system is to make allocations for as many users as possible and, at the same time, to achieve minimum user cost as measured by $J_{ij}(k)$. We introduce binary control variables

$$x_{ij} = \begin{cases} 1, & \text{if user } i \text{ is assigned to resource } j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

and define matrix $X = [x_{ij}]$. We can now formulate the allocation problem $(\mathbf{P})$ at the $k$th decision point as follows:

$$\min_X \sum_{i \in W(k) \cup R(k)} \sum_{j \in \Omega_i(k)} x_{ij} \cdot J_{ij}(k) + \sum_{i \in W(k)} \left( 1 - \sum_{j \in \Omega_i(k)} x_{ij} \right) \tag{9}$$

$$\text{s.t.} \quad \sum_{j \in \Omega_i(k)} x_{ij} \leq 1 \quad \forall i \in W(k) \tag{10}$$

$$\sum_{j \in \Omega_i(k)} x_{ij} = 1 \quad \forall i \in R(k) \tag{11}$$

$$\sum_{i \in W(k) \cup R(k)} x_{ij} \leq p_j(k) \quad \forall j \in \Gamma(k) \tag{12}$$

$$\sum_{j \in \Omega_i(k)} x_{ij} \cdot J_{ij}(k) \leq J_{iq_i(k-1)}(k) \quad \forall i \in R(k) \tag{13}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in W(k) \cup R(k), \quad j \in \Gamma(k). \tag{14}$$

In this problem, the objective function focuses on user satisfaction. One can formulate alternative versions that incorporate system-centric objectives, such as maximizing resource utilization or the total revenue without affecting the essence of our approach. The term $\min \sum_{i \in W(k) \cup R(k)} \sum_{j \in \Omega_i(k)} x_{ij} \cdot J_{ij}(k)$ in (9) aims to find the minimum cost over all users. If the system fails to allocate a resource to some user $i$, i.e., $\sum_{j \in \Omega_i(k)} x_{ij} = 0$, a cost of 1 is added to the objective function. Therefore, the added term $\sum_{i \in W(k)} (1 - \sum_{j \in \Omega_i(k)} x_{ij})$ in (9) is the total cost contributed by the number of "unsatisfied" users. Since by its definition in (7) $J_{ij}(k) \leq 1$, the added cost of value 1 is sufficiently large to ensure that a user is assigned to a resource if there are free qualified resources left. With this added term, the problem allocates as many users as possible.

Constraints (10) indicate that any user in the WAIT queue may be assigned, at most, one resource but may also fail to get an assignment. On the other hand, (11) guarantees that each user in the RESERVE queue maintains a resource assignment. Capacity constraints (12) ensure that every resource is occupied by no more than $p_j(k)$ users. Constraints (13) add a unique feature to our problem by guaranteeing that every user in the RESERVE queue is assigned a resource that is no worse than that most recently reserved, i.e., $q_i(k-1)$. Together, (11) and (13) ensure a reservation guarantee and improvement.

*Fairness:* As we can see from (10) and (11), a solution of $(\mathbf{P})$ gives a higher assignment priority to users in the RESERVE queue. This is because these users are already incurring a positive cost. (Recall that the pricing scheme we assume does not impose a fee to unassigned users, i.e., users still in the WAIT queue.) On the other hand, (10) makes no distinction among waiting users, regardless of where they are located. This introduces unfairness among waiting users. For example, a waiting user may be located right next to an available resource, which, however, is assigned to another waiting user at a considerably larger distance from it. To remove such unfairness, we add the following constraints:

$$\left[ \sum_{n \in \Omega_i(k)} x_{in} \right] - x_{mj} \geq 0 \quad \forall i, j, m \quad \text{s.t.} \quad j \in \Gamma(k)$$
$$j \in \Omega_i(k), \quad m \in W(k), \quad t_{mj} > t_{ij}. \tag{15}$$

These constraints are explained as follows. Consider resource $j$, which is available for assignment (i.e., $j \in \Gamma(k)$) and qualified for user $i$ (i.e., $j \in \Omega_i(k)$). If $i$ fails to be allocated any resource, we have $\sum_{n \in \Omega_i(k)} x_{in} = 0$, and (15) requires that $x_{mj} = 0$, i.e., any other waiting user $m$ located farther away from $j$ than user $i$ (i.e., $t_{mj} > t_{ij}$) is forbidden from being assigned to $j$. If, on the other hand, $\sum_{n \in \Omega_i(k)} x_{in} = 1$, i.e., user $i$ is assigned some resource, then $x_{mj} \leq 1$, i.e., there is no constraint on allocating resource $j$ to any user $m$ as long as condition (12) is satisfied. Thus, all subsequent references to $(\mathbf{P})$ refer to the original problem modified to include (15). We also note that there is no fairness issue related to users in the WAIT queue in terms of how long they have resided in it since this does not affect the cost objective unless a user is in the vicinity of his/her destination, which is a situation that we handle through the wandering ratio metric, as defined later in (17).

Moreover, in between any two decision points, users in the waiting queue who are close to their destination may reach it before having an opportunity to be assigned a parking space. To deal with this effect, we adopt the following *immediate allocation* (IA) policy: Whenever user $i$ is in the WAIT queue and reaches location $z_i$ such that $\|z_i - d_i\| \leq v_i \tau$, he/she is placed in an IA queue. Here, $\tau$ is the decision interval, and $v_i$ is the average driving speed. If this queue is not empty, then, as soon as user departure makes a resource available, the system immediately prioritizes user $i$ over other users in $W(k)$ and assigns him this resource if it is feasible. This IA problem is easy to solve. We define an "urgent" user set

$$I(k) = \{i : i \in W(k), \quad \|z_i - d_i\| \leq v_i \tau\}$$

and as soon as resource $j$ becomes free, we allocate it to user $i$ such that $J_{ij} = \min_{n \in I(k), j \in \Omega_n(k)} J_{nj}$, if such $i$ exists.

*Decision Points:* An important remaining issue concerns the choice of decision points over time or, equivalently, defining appropriate "decision intervals" $\tau(k)$, $k = 1, 2, \ldots$. In this paper, we pursue a time-driven strategy for decision making. After the $(k-1)$th decision point, the system waits for some duration $\tau(k)$ and then makes a new allocation over all users that arrived during $\tau(k)$ and all previous users residing in either the WAIT or the RESERVE queue. Clearly, there is a tradeoff: A large $\tau(k)$ may eventually yield a lower cost for all users involved, but it also forces a large number of users to remain in the WAIT queue with no assignment, until either it is too late because a user has reached his destination or it has lost patience and searches for resources by himself. In [10], we empirically explored the effect of varying $\tau(k)$ on the performance of the system.

*Scalability:* MILP problems are known to be NP-hard. If we deploy the system in a large urban area, problem $(\mathbf{P})$ becomes huge with a large number of variables and constraints. Obtaining a solution at each decision point becomes time consuming, and during this time, the system state changes, and the solution may no longer be optimal. In that case, we use the following steps to reduce the complexity of the problem.

1) *Area partitioning*: Observe that driver requests are independent if their destinations are far away from each other. Practically speaking, an allocation made for driver A has little or no influence on driver B if B's destination is several miles away from driver A. Thus, rather than aggregating all drivers and resources in one problem, we can instead partition the whole area into several small "districts." For each district, we solve problem $(\mathbf{P})$ for all drivers whose destinations are located in the district. Notice that drivers whose destinations are on the border of two adjacent districts are considered for allocation in both districts.
2) *Grouping resources*: Even in a single district, the total number of parking spaces may be large, particularly in a business district or downtown. However, drivers normally do not request a specific spot but only care for a street or a in which garage to park. Therefore, all spaces in the same garage or parking lot can be treated as a single resource, with $p_j$ denoting the number of vacant spaces. Similarly, we can group all on-street parking spots in the same street block as one resource. The system may randomly pick a vacant spot for the driver when he/she arrives. The problem size is thus greatly reduced.
3) *Discriminating users*: Drivers who are far away from their destination do not usually require an IA result. This is because once they make a reservation, they incur a cost that accumulates over time. Moreover, long-term reservations are detrimental to the system. First, users who are close to their destination may fail to obtain an assignment because available resources may have been reserved by users who are still far away and can be accommodated with later assignments. Second, there is a large fraction of resources left physically vacant because of reservations, which may cause user discontent when they cannot be allocated a resource. This points to a tradeoff in "smart parking" between a reasonable reservation scheme and parking space utilization. This can be achieved by restricting the number of users in the waiting queue who are assigned a resource. Thus, we introduce threshold $t_0$: Users within $t_0$ min away from their destination are considered for assignment; otherwise, they are kept in the waiting queue. Therefore, waiting set $W(k)$ in $(\mathbf{P})$ is replaced by $\tilde{W}(k)$, which is defined as

$$\tilde{W}(k) = \{i : i \in W(k), \quad t_{ij}(k) \leq t_0\}. \tag{16}$$

Of course, the number of decision variables in problem $(\mathbf{P})$ also decreases with this restriction.

*Performance Metrics:* In solving problem $(\mathbf{P})$, we aim to minimize user costs as defined by (7) at each decision point. To assess the overall system performance over some time interval $[0, T]$, we define several appropriate metrics that are evaluated over a total number of users $N_T$ served over this interval (e.g., a simulation run length).

From the system's point of view, we consider resource *utilization* as a performance metric and break it down into two parts: $u_r(T)$ is the utilization of resources by reservation (i.e., the fraction of resources that are reserved), and $u_p(T)$ is the utilization by occupancy (i.e., the fraction of resources that are physically occupied by a user).

From users' point of view, we first define a satisfaction metric for those users that actually occupy a resource. Let $P(T)$ be the set of such users over $[0, T]$. Moreover, returning to (7), let $q_i^* \in \{1, \ldots, N\}$ be the resource that is ultimately assigned to user $i \in P(T)$. We then define

$$J_{iq_i^*} = \lambda_i \frac{M_{iq_i^*}}{M_i} + (1 - \lambda_i) \frac{D_{iq_i^*}}{D_i}$$

$$\bar{J}(T) = \frac{1}{|P(T)|} \sum_{i \in P(T)} J_{iq_i^*}$$

measuring the *average cost* of users served. In addition, unlike traditional queueing problems, waiting times are not a measure of user satisfaction, since users do not actually need a resource until they have physically reached it. Instead, another metric that we will use is the *wandering ratio* $w(T)$, which is defined as follows: Let

$$A_W(k) = \{i : i \in W(k), \quad \|z_i(k) - d_i\| \leq \epsilon\}$$

be the set of users who reach their destination but are still in the WAIT queue at the $k$th decision point, where $\epsilon \geq 0$ is a small real number used to indicate that a user is in the immediate vicinity of his destination $d_i$. Letting $k_T$ denote the last decision point within the time interval of length $T$, we then define

$$w(T) = \frac{|A_W(k_T)|}{N_T}. \tag{17}$$

Finally, we consider the *average time-to-park* $t_p(T)$, which is the time from the instant a user sends a parking request to the instant he/she physically occupies a parking resource.

## IV. Simulation Results

In all simulations, we assume that user arrivals (requests) to each destination $i$ are Poisson distributed with rate $\lambda_i$. User travel times to reach their destination are exponentially distributed with rate $\gamma$. The resource occupancy time is also exponentially distributed with rate $\mu$. User cost parameter $M_i$ is uniformly distributed in interval $[M_{\min}, M_{\max}]$, and walking-distance parameter $D_i$ is also uniformly distributed in $[D_{\min}, D_{\max}]$. For simplicity, we adopt a constant decision interval $\tau(k) = \tau$, $k = 1, 2, \ldots$. Note that $\tau(k)$ can be made adjustable according to traffic conditions at the $k$th decision time.


Fig. 3. Case study environment.

We provide a simulation case study based on parking within the main campus of Boston University, as shown in Fig. 3. There are a total of 679 on-street parking spaces and 1932 off-street parking spaces in this part of the campus. We assume that all these spaces are monitored and can be used by any driver (student, faculty, or visitor) with no time limit.

Adopting the "grouping resources" method mentioned in the previous section, we aggregate 679 on-street parking spaces to 27 groups and 1932 off-street parking spaces to 14 groups. Following the same strategy, we also aggregate driver destinations: Buildings in the same block are treated as a single destination, and we consider a total of 12 destinations. Fig. 3 shows the parking configuration after grouping, where red triangles represent destinations, blue squares represent parking garages or lots, and dark-blue bars are on-street parking spaces.

We seek to quantify the improvement of the "smart parking" (SP) approach over an uncontrolled setting, where users park without any guidance (NG) and the case of guidance provided to available parking spaces (G). In both G and NG cases, we assume that users start searching for parking when they reach regions defined by their desired walking distance. For G, users know exactly the location of available spaces, and we assume that drivers always select the closest and least expensive available spot as their first choice. For NG, we assume that they search for vacant spaces with the following rules: On their way to the destinations, they check all nearby parking spaces within their desired walking distance; after they reach their destinations, they perform an increasing-radius search around them.

In all simulations, we set $1/\gamma = 30$ min, $1/\mu = 60$ min, $M_{\max} = \$8$, $D_{\max} = 8$ min, $\tau = 1$ min, and $\epsilon = 0$. The on-street parking price is \$0.25 per 12 min, whereas the off-street parking price is \$2 per 30 min. All results are generated by an average of five simulations, with each lasting for $T = 3000$ min. We will examine all performance metrics that we have defined under different traffic intensities by changing the interarrival times $1/\lambda_i$.

In Table I, we set high average arrival rate $\bar{\lambda}_i > 1.5$ to mimic heavy traffic and low arrival rate $\bar{\lambda}_i < 1$ to generate normal traffic. The performance metrics show that SP provides significant benefits over the G and NG approaches, where "-ons" indicates the on-street parking metrics, and "-offs" indicates the off-street parking metrics. From the system point of view, the total resource utilization $(u = u_r + u_p)$ increases compared with both G and NG approaches. On-street parking utilization generally exceeds off-street parking utilization, which indicates that the system first allocates resources with low cost to users.

TABLE I
PERFORMANCE METRICS UNDER TWO DIFFERENT TRAFFIC INTENSITIES

| $1/\lambda_i$ | Heavy Traffic | | | Normal Traffic | | |
|---|---|---|---|---|---|---|
| | $SP$ | $G$ | $NG$ | $SP$ | $G$ | $NG$ |
| $u(T)$-ons | 0.984 | 0.874 | 0.795 | 0.739 | 0.589 | 0.552 |
| $u_p(T)$-ons | 0.690 | 0.874 | 0.795 | 0.546 | 0.589 | 0.552 |
| $u_r(T)$-ons | 0.294 | NA | NA | 0.193 | NA | NA |
| $u(T)$-offs | 0.594 | 0.491 | 0.487 | 0.450 | 0.318 | 0.410 |
| $u_p(T)$-offs | 0.406 | 0.491 | 0.487 | 0.328 | 0.318 | 0.410 |
| $u_r(T)$-offs | 0.188 | NA | NA | 0.122 | NA | NA |
| $w(T)$ | 0.130 | 0.203 | 0.668 | 0.003 | 0.086 | 0.439 |
| $\bar{J}(T)$ | 0.292 | 0.319 | 0.462 | 0.255 | 0.344 | 0.385 |
| $t_p(T)$ | 35.34 | 70.85 | 119.77 | 31.35 | 36.16 | 43.20 |

TABLE II
PERFORMANCE METRICS WITH DIFFERENT $t_0$

| $t_0$ | 2 | 10 | 20 | 30 | 50 | $\infty$ |
|---|---|---|---|---|---|---|
| $u_p(T)$-ons | 0.839 | 0.789 | 0.771 | 0.742 | 0.708 | 0.690 |
| $u_r(T)$-ons | 0.014 | 0.110 | 0.198 | 0.239 | 0.277 | 0.294 |
| $u_p(T)$-offs | 0.495 | 0.461 | 0.443 | 0.439 | 0.428 | 0.406 |
| $u_r(T)$-offs | 0.008 | 0.065 | 0.116 | 0.147 | 0.174 | 0.188 |
| $w(T)$ | 0.012 | 0.070 | 0.066 | 0.068 | 0.095 | 0.131 |
| $\bar{J}(T)$ | 0.286 | 0.276 | 0.297 | 0.302 | 0.302 | 0.292 |
| $t_p(T)$ | 30.38 | 31.00 | 34.14 | 35.40 | 35.44 | 35.34 |

From a user's point of view, we see decreases in both $w(T)$ and $\bar{J}(T)$, whereas the average time-to-park is reduced by as much as half (from 70.85 to 35.34) compared with the G method under heavy traffic. For the G and NG methods, $w(T)$ is defined as the fraction of users who fail to obtain a parking spot on their first try. Thus, $w(T)$ shows the fraction of users who are simply wandering around in search of parking, whereas $(t_p(T) - 1/\gamma)$ indicates the average searching time. We can see that SP dramatically decreases not only the number of wandering drivers but their searching time as well. At the same time, the smaller $\bar{J}(T)$ shows that users who ultimately parked obtained better-quality spots, i.e., either cheaper or closer to their destination or both.

We notice that in Table I, the actual utilization by occupancy using SP, i.e., $u_p(T)$, is smaller than that of G under heavy traffic. (However, $u_p(T) + u_r(T)$ is still higher under SP.) This is because a considerable fraction of resources are utilized by user reservations, which prevents other users from occupying them. This is the reason why $w(T)$ under SP is not reduced as much as one might expect. An additional undesirable effect is the indignation that drivers may feel if they observe that a large fraction of parking spaces are empty but cannot be used due to reservations. To address this issue, we discriminate reservation requests by only allowing them when drivers are within an estimated travel time $t_0$ away from their destination. This has the added benefit of reducing the problem scale. Table II shows all performance metrics with different $t_0$ values under heavy traffic. Clearly, $u_p(T)$ indeed increases as $t_0$ decreases, whereas $w(T)$ and $t_p(T)$ generally become smaller compared with allocations without a $t_0$ time threshold. However, the total average user cost increases if we set $t_0$ too small. With this additional $t_0$ regulation, the system gives higher priority to users who are approaching their destinations; therefore, they have a lower chance of wandering and $t_p(T)$ approaches $1/\gamma$. However, since only a smaller group of users is now considered

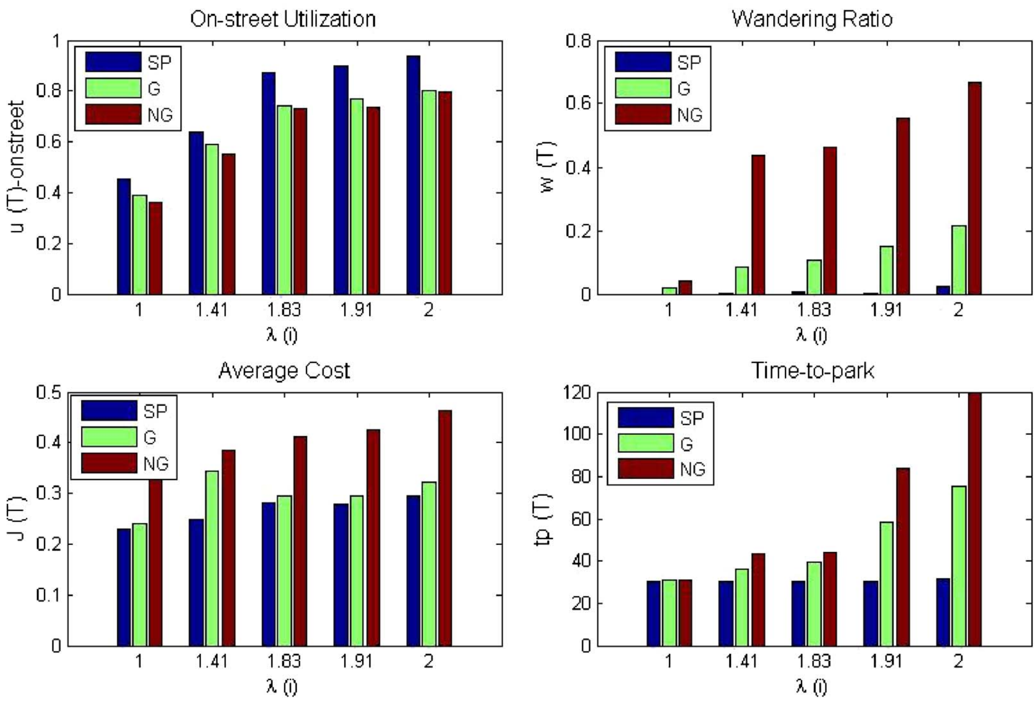



Fig. 4. Simulation results under $t_0 = 10$.

for allocation, the results are optimal for them but not for all users in the waiting queue; users farther away from their destinations generally end up with a parking spot of worse quality than closer users, and the overall average cost increases. Moreover, if we set $t_0$ too small, drivers have less time to adjust their requirements when they fail to be allocated. In short, the choice of threshold $t_0$ requires careful consideration. For example, in Table II, $t_0 = 10$ appears to be a good choice.

By setting $t_0 = 10$, we have obtained additional simulation results summarized in Fig. 4. All four performance metrics (off-street utilization is not shown here) are compared under different traffic intensities determined by $\lambda_i$. We find that in all scenarios, the SP approach improves parking resource utilization and decreases user cost and search time. As traffic intensity increases, the improvement offered by the SP approach becomes more significant. Moreover, $w(T)$ can be further decreased using the IA policy mentioned earlier. (Results are shown in [10].)

## V. Implementation

The "smart parking" system, as described in this paper, has been deployed in a garage at Boston University, which contains 27 parking spaces. At each parking space, we have installed a *Streetline* [23] parking detection sensor on the ground and an LED device for controlling our light system described in Section II-B. A *Streetline* gateway receives data from each sensor in the network and forwards it to an upper level database, which serves as the PRMC with the state (vacant or occupied) of each parking space. The real-time parking information is published and updated on the web and can be obtained by users. Thus, our system still provides the service of a normal PGI system. We have also installed cameras and use standard image processing algorithms, based on which the state of each parking space (vacant or occupied) is determined; the joint data from the ground sensors and the cameras are combined to increase the reliability of parking state estimates.


Fig. 5. "Smart parking" application and website.

We have also built a smartphone application (see http://smartpark.bu.edu/smartparking/home.php), through which users can send parking requests and obtain reservations. The application sends all user requests to a computer, which serves as both DRPC and SPAC (see Fig. 1). The computer maintains all driver requests, solves the optimal allocation problem $(\mathbf{P})$, updates the parking space state database, and sends commands to control the state of the light at each parking space device. Fig. 5 shows the smartphone application and real-time parking information website.

An important component of our implementation is a supervisory controller based on a finite-state automaton describing full system operations. Fig. 6 shows the automaton associated

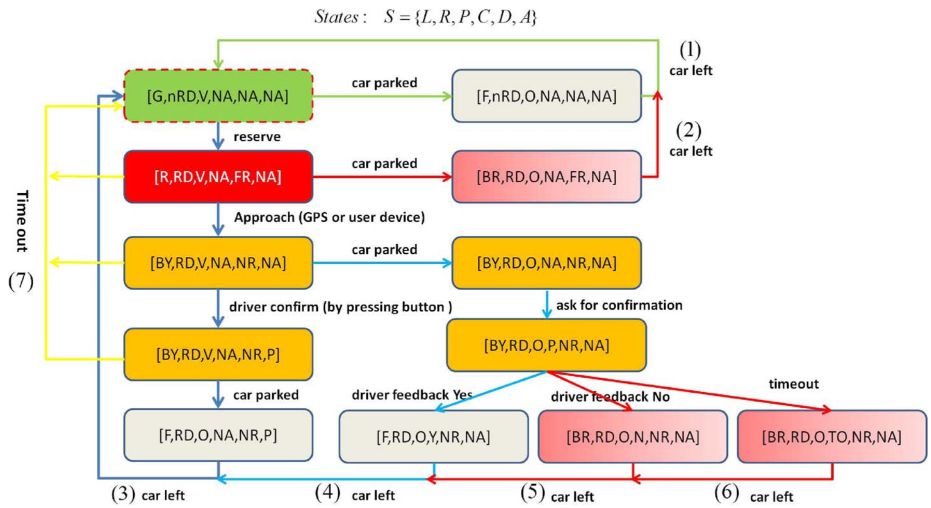



Fig. 6. Automaton of parking spot and light status.

with a single parking space. The state $S = \{L, R, P, C, D, A\}$ is defined as follows:

1) Light Status (L) = $\{GREEN(G), RED(R), YELLOW\_BLINK(BY), RED\_BLINK(BR), OFF(O)\}$;
2) Reservation Status (R) = $\{NOT\_RESERVED(nRD), RESERVED(RD)\}$;
3) Parking Status (P) = $\{VACANT(V), OCCUPIED(O)\}$;
4) Driver's Response Status (C) = $\{YES(Y), NO(N), PENDING(P), TIMEOUT(TO), NULL(NA)\}$;
5) Driver's Position Status (D) = $\{NEARBY(NR), FAR(FR), NULL(NA)\}$;
6) Driver's Self-Confirmation Status (A) = $\{CONFIRMED(P), NULL(NA)\}$.

In this automaton, *Driver's Self-Confirmation* indicates that the driver presses a button in the smartphone to confirm that he/she has parked before the sensor detects his vehicle in the parking spot. If the driver forgets to confirm parking but the sensor detects a car in his reserved spot, the system will ask whether he/she parked there or not. *Driver's Response Status* is then used to store the driver's response. The driver may answer YES or NO or not reply (TIMEOUT). The value *NULL(NA)* means that the status is not applicable. We also assume that all sensors have 100% accuracy. Thus, we assume that the *car parked* event in the automaton indicates that the sensor detects a car and that the car indeed parked there. However, if a sensor reports an erroneous parking status, we allow users to report the wrong situation using the smartphone, and the system will either adjust the sensor state or replace it with a new one.

The automaton describes the complete operational functionality of the system. Considering a vacant parking spot without reservation as the initial state, there are seven possible state transition flows (see Fig. 6) for a parking spot to return to the initial state. (The flows are color differentiated, and the flow number indicates the last step of the corresponding flow.)

Flow (1): A parking spot is occupied by a driver with no system allocation.

Flow (2): A parking spot is reserved by a driver, but it is occupied by a different driver. This is a violation.

Flow (3): A parking spot is reserved by a driver. The driver parks his vehicle and immediately confirms parking before the system sends any parking confirmation request.

Flow (4): A parking spot is reserved by a driver. The driver parks his vehicle but forgets to confirm. The system requests confirmation, and the driver says YES.

Flow (5): This is the same as Flow (4), except that the driver says NO. This indicates that the parking spot is occupied by a different driver. This is a violation.

Flow (6): This is the same as Flow (4), except that the driver did not respond to the parking confirmation request. In this case, the system conservatively assumes that the spot is occupied by someone else.

Flow (7): This flow involves all possible timeout events while a parking spot is reserved by a driver. For example, a parking spot is reserved by a driver, but the driver does not show up within 1 h.

The system is now being used by approximately 30 registered users. A pilot study is ongoing to analyze utilization and user cost data and collect feedback on system usability.

## VI. Conclusion and Future Work

We have proposed a "smart parking" system that exploits technologies for parking space availability detection and for driver localization and that allocates parking spots to drivers instead of only supplying guidance to them. We have focused on determining an efficient and optimal allocation strategy for both users and the system by solving a sequence of MILP problems, which are guaranteed to have a feasible solution and to satisfy some fairness constraints. Simulation results show significant performance improvements over existing parking behavior, including the use of guidance-based systems.

Current research focuses on selecting (possibly state-dependent) proper decision intervals and on the use of pricing control to adjust parking space prices for different classes of users or other bidding-type mechanisms that can enhance fairness. Moreover, through an ongoing collaboration with the City of Boston, we plan to expand deployment tests to on-street parking on several urban blocks.

## REFERENCES


[1] Z. Aksin, M. Armony, and V. Mehrotra, "The modern call center: A multi-disciplinary perspective on operations management research," *Prod. Oper. Manage.*, vol. 16, no. 6, pp. 665–688, Nov./Dec. 2007.

[2] R. Arnott, T. Rave, and R. Schob, *Alleviating Urban Traffic Congestion*. Cambridge, MA, USA: MIT Press, 2005.

[3] R. Atar, A. Mandelbaum, and M. Reiman, "Scheduling a multi-class queue with many exponential servers: Asymptotic optimality in heavy-traffic," *Ann. Appl. Probab.*, vol. 15, no. 4, pp. 2606–2650, 2004.

[4] I. Benenson, K. Martens, and S. Birr, "Parkagent: An agent-based model of parking in the city," *Comput. Environ. Urban Syst.*, vol. 32, no. 6, pp. 431–439, Nov. 2008.

[5] J. Benson, T. Donovan, P. Sullivan, U. Roedig, and C. Sreenan, "Car-park management using wireless sensor networks," in *Proc. 31st IEEE Conf. Local Comput. Netw.*, 2006, pp. 588–595.

[6] K. Cheung and P. Varaiya, "Traffic surveillance by wireless sensor network: Final report," Univ. California, Berkeley, CA, USA, Tech. Rep. UCB-ITS-PRR-2007-4, 2007.

[7] S. Chou, S. Lin, and C. Li, "Dynamic parking negotiation and guidance using an agent-based platform," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 805–817, Oct. 2008.

[8] M. Gallo, L. D'Acierno, and B. Montella, "A multilayer model to simulate cruising for parking in urban areas," *Transp. Policy*, vol. 18, no. 5, pp. 735–744, Sep. 2011.

[9] N. Gans, G. Koole, and A. Mandelbaum, "Telephone call centers: Tutorial, review and research prospects," *Manuf. Service Oper. Manage.*, vol. 5, no. 2, pp. 79–141, Apr. 2003.

[10] Y. Geng and C. G. Cassandras, "Dynamic resource allocation in urban settings: A "smart parking" approach," in *Proc. IEEE Multi-Conf. Syst. Control*, 2011, pp. 1–6.

[11] Y. Geng and C. G. Cassandras, "A new "smart parking" system based on optimal resource allocation and reservations," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 979–984.

[12] E. Griffith, "Pointing the way," ITS Int., Kent, U.K., 2000.

[13] I. Gurvich and W. Whitt, "Service-level differentiation in many server service systems: A solution based on fixed-queue-ratio routing," *Oper. Res.*, vol. 29, pp. 567–588, Mar./Apr. 2007.

[14] S. Halfin and W. Whitt, "Heavy-traffic limits for queues with many exponential servers," *Oper. Res.*, vol. 29, no. 3, pp. 567–588, May/Jun. 1981.

[15] T. B. Hodel and S. Cong, "Parking space optimization services, a uniformed web application architecture," in *Proc. ITS World Congr.*, 2003, pp. 16–20.

[16] G. Koole and A. Pot, "Approximate dynamic programming in multi-skill call centers," in *Proc. 37th Conf. Winter Simul.*, 2005, pp. 576–583.

[17] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *Proc. Int. Symp. Collab. Technol. Syst.*, 2008, pp. 48–57.

[18] L. Mimbela and L. Klein, *A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems*. Las Cruces, NM, USA: Vehicle Detector Clearinghouse, 2000.

[19] J. Polak and K. W. Axhausen, "Clamp: A macroscopic simulation model for parking policy analysis," presented at the 68th Annu. Meeting Transp. Res., 1989.

[20] C. J. Rodier and S. A. Shaheen, "Transit-based smart parking: An evaluation of the San Francisco bay area field test," *Transp. Res. C, Emerging Technol.*, vol. 18, no. 2, pp. 225–233, Apr. 2010.

[21] SFpark. [Online]. Available: http://sfpark.org

[22] D. Shoup, *The High Cost of Free Parking*. Chicago, IL, USA: APA Planner, 2005.

[23] Streetline. [Online]. Available: http://www.streetlinenetworks.com/

[24] D. Teodorovic and P. Lucic, "Intelligent parking systems," *Eur. J. Oper. Res.*, vol. 175, no. 3, pp. 1666–1681, 2006.

[25] R. G. Thompson and P. Bonsall, "Driver's response to parking guidance and information systems," *Transp. Rev.*, vol. 17, no. 2, pp. 89–104, 1997.

[26] R. G. Thompson and A. J. Richardson, "A parking search model," *Transp. Res. A, Policy Pract.*, vol. 32, no. 3, pp. 159–170, Apr. 1998.

[27] M. Tsai and C. Chu, "Evaluating parking reservation policy in urban areas: An environmental perspective," *Transp. Res. D, Transp. Environ.*, vol. 17, no. 2, pp. 145–148, Mar. 2012.

[28] H. Wang and W. He, "A reservation-based smart parking system," in *Proc. IEEE Comput. Commun. Workshops*, 2011, pp. 690–695.

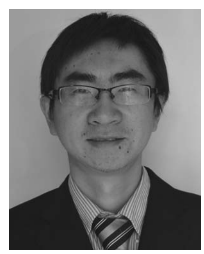
[29] B. J. Waterson, N. B. Hounsellm, and K. Chatterjee, "Quantifying the potential savings in travel time resulting from parking guidance systems—A simulation case study," *J. Oper. Res. Soc.*, vol. 52, no. 10, pp. 1067–1077, 2001.

[30] J. Wenger and U. Daimler-Benz, "Automotive mm-wave radar: Status and trends in system design and technology," in *Proc. Automot. Radar Navigat. Tech.*, 1998, vol. 230, pp. 1/1–1/7.

[31] J. Wolff, T. Heuer, M. H. Gao, S. Weinmann, and U. Voit, "Parking monitor system based on magnetic field sensor," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 1275–1279.

[32] K. Yamada and M. Mizuno, "A vehicle parking detection method using image segmentation," *Electron. Commun. Jpn. 3—Fundam. Electron. Sci.)*, vol. 84, no. 10, pp. 25–34, Oct. 2001.

[33] W. Young, *Parksim 1.1 Users Manual, Department of Civil Engineering*. Melbourne, VIC, Australia: Monash Univ., 1991.

**Yanfeng Geng** (S'11) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2005. He is currently working toward the Ph.D. degree with the Division of Systems Engineering, Boston University, Boston, MA, USA.

His research interests include intelligent transportation systems, optimal control, and stochastic optimization.

**Christos G. Cassandras** (F'96) received the B.S. degree from Yale University, New Haven, CT, USA, in 1977; the M.S.E.E. degree from Stanford University, Stanford, CA, USA, in 1978; and the S.M. and Ph.D. degrees from Harvard University, Cambridge, MA, USA, in 1979 and 1982, respectively.

From 1982 to 1984, he was with ITP Boston, Inc., Belmont, MA, where he worked on the design of automated manufacturing systems. From 1984 to 1996, he was a faculty member with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA. Currently, he is the Head of the Division of Systems Engineering and a Professor of electrical and computer engineering with Boston University, Boston, MA, and a Founding Member of the Center for Information and Systems Engineering. He has published over 300 papers and five books. He specializes in the areas of discrete-event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems.

Dr. Cassandras was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1998 to 2009 and has served on several editorial boards. In 2012, he was the President of the IEEE Control Systems Society and a recipient of several awards. He is a member of Phi Beta Kappa and Tau Beta Pi and a Fellow of the International Federation of Automatic Control.