2011 14th International IEEE Conference on
Intelligent Transportation Systems
Washington, DC, USA. October 5-7, 2011

A New "Smart Parking" System Based on Optimal Resource Allocation and Reservations

Yanfeng Geng and **Christos G. Cassandras**
Division of Systems Engineering
and Center for Information and Systems Engineering
Boston University
Brookline, MA 02446
gengyf@bu.edu, cgc@bu.edu

Abstract— **We propose a new "smart parking" system for an urban environment. The system assigns and reserves an optimal resource (parking space) for a user (driver) based on the user's objective function that combines proximity to destination and parking cost, while also ensuring that the overall parking capacity is efficiently utilized. Our approach solves a Mixed Integer Linear Program (MILP) problem at each decision point in a time-driven sequence. The solution of each MILP is an optimal allocation based on current state information and subject to random events such as new user requests or parking spaces becoming available. The allocation is updated at the next decision point ensuring that there is no resource reservation conflict and that no user is ever assigned a resource with higher than the current cost function value. Simulated case studies are included based on parking at part of the Boston University campus showing that we can achieve significant improvement over uncontrolled parking processes or state-of-the-art guidance-based systems. We also describe a laboratory setting where this system has been tested in real time.**

I. INTRODUCTION

On a daily basis, it is estimated that 30% of vehicles on the road in the downtown area of major cities are cruising for a parking spot and it takes an average of 7.8 minutes to find one [5]. This causes not only a waste of time and fuel for drivers looking for parking, but it also contributes to additional waste of time and fuel for other drivers as a result of traffic congestion. For example, it has been reported [14] that over one year in a small Los Angeles business district, cars cruising for parking created the equivalent of 38 trips around the world, burning $47,000$ gallons of gasoline and producing 730 tons of carbon dioxide.

Over the past two decades, traffic authorities in many cites are building 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 [1], [9], [15]. PGI systems are based on the development of autonomous vehicle detection and parking spot monitoring, typically through the use of sensors placed in the vicinity of parking spaces for vehicle detection and surveillance [2], [3], [7], [12]. However, it has been found that using PGI systems, system-wide reductions in travel time and vehicle benefits may be relatively small [16], [17]. Building upon the objectives of PGI systems, e-parking is an innovative platform which allows drivers to obtain parking information before or during a trip, and reserve a parking spot [13]. Drivers access the central system via cellular phone or Internet. Bluetooth technology recognizes each car at entry points, and triggers automatic reservation checking and parking payment [10]. Researchers also find that traffic congestion can be alleviated by controlling the parking price [15]. For example, in San Francisco there are already time-dependent or demand-dependent parking fees to achieve the right level of parking availability in different areas [4].

As pointed out in [8], current guidance-based systems have several shortcomings. For example, drivers may not find vacant parking spots by merely following a VMS; drivers may miss a better parking spot; parking space utilization becomes imbalanced; it causes new traffic congestion around the area with good parking spaces. 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 the user's objective function that combines proximity to destination and parking cost, while also ensuring that the overall parking capacity is efficiently utilized. Building on the results in [8], in this paper we refine the allocation algorithm to incorporate features making it suitable for real-world parking problems identified in carrying out a case study based on parking at part of the Boston University campus . We include extensive numerical results from simulations under this case study that support the feasibility of the "smart parking" idea and we compare several performance metrics with PGI systems.

The rest of the 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 and formulate the MILP problem solved at every

The authors' work is supported in part by NSF under Grant EFRI-0735974, by AFOSR under grant FA9550-09-1-0095, by DOE under grant DE-FG52-06NA27490, and by ONR under grant N00014-09-1-1051.

decision point. Simulations based on the case study at Boston University are given in Section IV. An indoor implementation platform is described in Section V. Finally, we conclude and discuss future work in Section VI.

II. System Framework

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

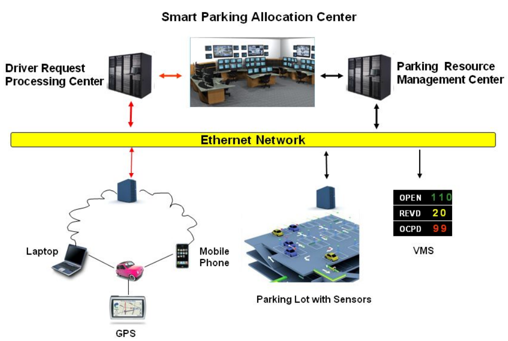

Fig. 1. Smart Parking Framework Overview

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 DPRC 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 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 one) 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 his own overly restrictive parking requirements) or if he fails to accept it for any other reason, he 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).

The realization of such a "smart parking" system relies on three main requirements. First, the allocation center has to know the status of all parking spots, the location of all vehicles issuing requests and traffic situations. As already mentioned, current sensing technologies make monitoring parking spots implementable. Moreover, standard GPS technology provides accurate localization and speed estimates of vehicles [12]. The second requirement involves effective wireless communication between vehicles and an allocation center. This is also achievable through existing wireless networks that may be proprietary or part of cellular telephone service providers. Finally, the center must be able to implement a reservation that guarantees a specific parking spot to a driver. This is achievable through wireless technology interfacing a vehicle with hardware that makes a spot 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. A "softer" scheme is to use a red/green light system placed at each parking spot, where red indicates that the spot is reserved and only the vehicle assigned to it may switch it back to green (a vehicle parked when the light is red is fined.) In what follows, we will not deal with technical details for meeting these three implementation requirements and concentrate instead on the methodology that enables us to make optimal parking space allocations and reservations.

III. Dynamic Resource Allocation Model

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 spots. We adopt a queueing model for the problem as shown in Fig. 2 (introduced in [8]), 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. At the kth decision point, the system makes allocations for all users in both the waiting queue and the queue (labeled RESERVE) of users 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 joins the RESERVE queue, otherwise he remains in WAIT. A user in RESERVE may be assigned a different resource after a decision point and returns to the same queue until he 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 the kth decision point we define the state of the allocation system, $X(k)$, and the state of the ith user, $S_i(k)$

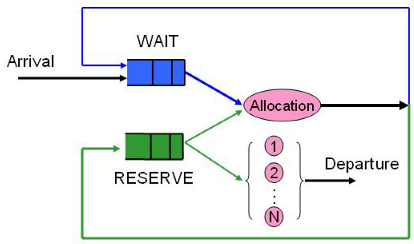

Fig. 2. Queueing Model for Dynamic Resource Allocation

as explained next. First, we define

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

where $W(k) = \{i$: user i is in the WAIT queue$\}$, $R(k) = \{i$: user i is in the RESERVE queue$\}$, and $P(k) = \{p_1(k), ..., p_N(k)\}$ is a set describing the state of the jth resource with $p_j(k)$ denoting the number of free parking spaces at resource j, $j = 1, \ldots, N$.

We assume that each resource j has a known location associated to it denoted by $y_j \in Z \subset \mathbb{R}^2$ in a two-dimensional Euclidean space, and its capacity is n_j. 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 kth decision point ($r_i(k) = 0$ if $i \in W(k)$), and $q_i(k)$ is the reservation status of user i:

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

Finally, $\Omega_i(k)$ is a feasible resource set for user i, i.e., $\Omega_i(k) \subseteq \{1, \ldots, N\}$ depending on the requirements set forth by this user regarding the resource it requests. We will define $\Omega_i(k)$ in terms of two *attributes* associated with user i. The first, denoted by D_i, is an upper bound on the distance 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 a 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, denoted by M_i, is an upper bound on the cost 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 adopted by the allocation system and may include 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 function of the total reservation time $r_i(k)$ and the traveling time from the user location at the kth decision time, $z_i(k)$, to a resource location y_j. Let $s_{ij}(k) = \|z_i(k) - y_j\|$ be this distance, and define the traveling time $t_{ij}(k) = f(s_{ij}(k), \omega)$, where ω denotes all random traffic conditions. We use $M_{ij}(r_i(k), t_{ij}(k))$ to denote the total expected cost for using resource j, evaluated at the kth decision time. Comparing $M_{ij}(r_i(k), t_{ij}(k))$ to M_i, leads to the constraint

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

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). In order to fully specify $\Omega_i(k)$, we further define

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

to be the set of free and reserved resources at the kth decision time and set

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

where, for simplicity, we have written $M_{ij}(k)$ instead of $M_{ij}(r_i(k), t_{ij}(k))$. 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 (i.e., if $p_j(k) = m \neq i$). If user i only provides final destination without requirements, $\Omega_i(k) = \Gamma(k)$.

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, $J_{ij}(k)$, if he 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 $J_{ij}(k)$. Define binary control variables:

$$x_{ij} = \begin{cases} 0 & \text{if user } i \text{ is not assigned to resource } j \\ 1 & \text{if user } i \text{ is assigned to resource } j \end{cases} \tag{8}$$

We can now define the allocation problem **(P)** at the kth decision point as follows:

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

s.t.

$$\sum_{j \in \Omega_i(k)} x_{ij} \leq 1, \qquad \forall i \in W(k) \tag{10}$$

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

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

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

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

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 total revenue without affecting the essence of our approach. 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) $J_{ij}(k) \le 1$, 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. The 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) still guarantees that each user in the RESERVE queue maintains a resource assignment. The capacity constraints (12) ensure that every resource is occupied by no more than $p_j(k)$ users. The constraints (13) add a unique feature to our problem by guaranteeing that every user in the RESERVE queue is assigned a resource which is no worse than the one most recently reserved, i.e., $q_i(k-1)$.

Problem **(P)** is a Mixed-Integer Linear Programming (MILP) problem [6] that can be solved using any of several commercially available software packages (we use ILOG CPLEX in this paper [11]). In this formulation, we can easily prove that the problem is always feasible. Indeed, letting the matrix $X \equiv [x_{ij}]$ denote a solution of (9), then the set

$$\{X : \sum_{j \in \Omega_i(k)} x_{ij} = 0, \; x_{mq_m(k)} = 1, \; i \in W(k), \; m \in R(k)\}$$

is always a feasible solution, since it implies that all users in $W(k)$ are not allocated and all users in $R(k)$ simply maintain their previous reservation (assuming that $R(k) \neq \varnothing$).

IV. Case Study Simulation Results

In this section, we describe a simulation environment based on parking at part of the Boston University (BU) main campus within the city of Boston, as shown in Fig. 3. There are totally 679 on-street parking spots and 1932 off-street parking spots in this part of the campus. We assume that all these spots are monitored and can be used by any drivers (student, faculty or visitor) without any time limit.

Preprocess. Even though we are facing an allocation problem in a small district, the problem scale is still very large. Thus, as a first step, we reduce the decision variables and constraints of problem (**P**) by "grouping" parking spots. Intuitively, all spots in the same garage or parking lot can be treated as one resource. We also group the on-street parking spots in the same street block. With this grouping method, we aggregate 679 on-street parking spots to 27 groups and 1932 off-street parking spaces to 14 groups. If a driver reserves a parking spot in one group, the system simply selects any available spot in that group for him when he arrives.

Following the same strategy, we also aggregate driver destinations by their locations. Buildings in the same block are treated as one destination, and we have totally 12 destinations. Fig. 3 shows the parking configuration after grouping, where red triangles represent destinations, blue squares represent parking lots and darkblue bars are on-street parking spaces.

Fig. 3. Case Study Environment

Settings. In all simulations, we assume that user arrivals to each destination i are Poisson distributed with rate λ_i. User travel times to reach their destination are exponentially distributed with rate γ. The resource occupancy time is also exponentially distributed with rate μ. The user cost parameter M_i is uniformly distributed in the interval $[0, M_{max}]$, and the walking-distance parameter D_i is also uniformly distributed in $[0, D_{max}]$.

We adopt a pricing scheme based on which the expected cost incurred by user i when assigned resource j at the kth decision point is

$$M_{ij}(k) = C_j \cdot (r_i(k) + t_{ij}(k) + T_i) \tag{15}$$

where C_j is the price of resource j, $r_i(k)$ is the time already spent at the RESERVE queue, $t_{ij}(k)$ is an estimate of the driving time for i to reach j, and T_i is the expected parking time of user i. Thus, $M_{ij}(k)$ combines a reservation cost $C_j(r_i(k) + t_{ij}(k))$ and actual parking cost $C_j T_i$.

The walking-distance cost is defined as $D_{ij} = w_{jd_i}$ where w_{jd_i} measures the walking distance from resource j to user i's destination d_i. We obtain w_{jd_i} by walking-time estimates using Google Maps.

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 kth decision time.

Performance Metrics. In order to assess the overall system performance over some time interval $[0, T]$, we define several appropriate metrics evaluated over a total number of users N_T served over this interval (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).

$1/\lambda_i$	Heavy Traffic			Normal Traffic		
	SP	G	NG	SP	G	NG
$u_p(T)$-ons	0.690	0.874	0.795	0.546	0.589	0.552
$u_r(T)$-ons	0.294			0.193		
$u_p(T)$-offs	0.406	0.491	0.487	0.328	0.318	0.410
$u_r(T)$-offs	0.188			0.122		
$w(T)$	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 I

PERFORMANCE METRICS UNDER DIFFERENT TRAFFIC INTENSITY

t_0	2	10	20	30	50	∞
$u_p(T)$-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

TABLE II

PERFORMANCE METRICS WITH DIFFERENT t_0

From the 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 $[0, T]$, and $q_i^* \in \{1, \ldots, N\}$ be the resource ultimately assigned to user $i \in P(T)$. Then $\bar{J}(T) = \frac{1}{|P(T)|}\sum_{i \in P(T)} J_{iq_i^*}$ measures the *average cost* of users served. Another metric we will use is the *wandering ratio* $w(T)$ defined as follows. Let $A_W(k) = \{i : i \in W(k),\ \|z_i(k) - d_i\| \leq \epsilon\}$ be the set of users who reach the vicinity of their destination, measured by $\epsilon \geq 0$, but are still in the WAIT queue at the kth decision point. Letting k_T denote the last decision point within the time interval of length T, we then define $w(T) = \frac{|A_W(k_T)|}{N_T}$.

Finally, we consider the *average time-to-park* $t_p(T)$, which is the time from the instant a user "arrives" (i.e., issues a parking request) to the instant he physically occupies a parking resource. $(t_p(T) - 1/\gamma)$ is the average wandering time. Notice that $t_p(T)$ is different from the traditional waiting time (from "arrival" to "allocation"), which is not of particular interest in the "smart parking" problem because the waiting time is partially due to a driver's requirements.

Results. 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 parking with guidance to free parking spaces (G). In both cases, we assume users start to look for parking when they reach regions defined by their walking distance. If there is guidance, users know exactly the location of free resources; otherwise, they search for free resources by themselves. We assume drivers always pick the nearest and cheapest available spot as their first choice.

In all simulations, we set $1/\gamma = 30$ min, $1/\mu = 60$ min, $M_{max} = \$8$, $D_{max} = 8$ min, $\tau = 1$ min and $\epsilon = 0$. On-street parking price is 0.25 \$ per 12 minutes, while off-street parking price is 2 \$ per 30 minutes. Every result is generated by the average of 5 simulations, with each lasting for $T = 3000$ min. We will examine all performance metrics we have defined under different traffic intensities by changing the interarrival times $1/\lambda_i$

In Table I, 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 $(u_r + u_p)$ increases compared to both G and NG approaches. On-street parking utilization generally exceeds off-street parking, which indicates that the system allocates first resources with low cost to users. From a user's point of view, we see decreases in both $w(T)$ and $\bar{J}(T)$, while average time-to-park is reduced by as much as half (from 70.85 to 35.34) compared to 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, $w(T)$ basically shows the fraction of users who are simply wandering around in search of parking, while $(t_p(T) - 1/\gamma)$ indicates the average searching time. We can see that SP not only dramatically decreases the number of wandering drivers, but it also decreases their searching time. At the same time, the smaller $\bar{J}(T)$ shows that users who ultimately parked obtained better-quality spots, either cheaper or nearer to their destination. Notice that $w(T)$ can be further decreased by using "immediate allocation" policy for users who are approaching their destination with no resource assigned, as detailed in [8].

We notice that in Table I the actual utilization by occupancy, $u_p(T)$, is smaller than G (however, $u_p(T) + u_r(T)$ is still higher under SP.) This is because a considerable fraction of resources is utilized by user reservations, no matter how far they are from the destinations. This benefits users since they can receive a quick response from the system and have guaranteed reservations. However, it also has two shortcomings. First, users who are close to their destination may fail to obtain an assignment because available resources may have been reserved by users still far away, whereas such users might agree with later assignments. Second, there is a high fraction of resources left physically vacant because of reservations, which may cause user discontent. 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 a threshold t_0: users within t_0 minutes away from their destination are considered for assignment, otherwise they are kept in the waiting queue. Therefore, the waiting set $W(k)$ in (**P**) is replaced with $\tilde{W}(k)$, defined as

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

Table II shows all performance metrics with different t_0 values under heavy traffic. As we can see, $u_p(T)$ indeed increases as t_0 decreases, while $w(T)$ and $t_p(T)$ generally become smaller compared to allocations without a 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, so that they have a smaller chance to be wandering and $t_p(T)$ approaches $1/\gamma$. However, since only a smaller group of users is now considered 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 the threshold t_0 requires careful consideration. For the 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 under different traffic intensities. We find that as the traffic intensity increases, the improvement offered by the SP approach becomes more significant.

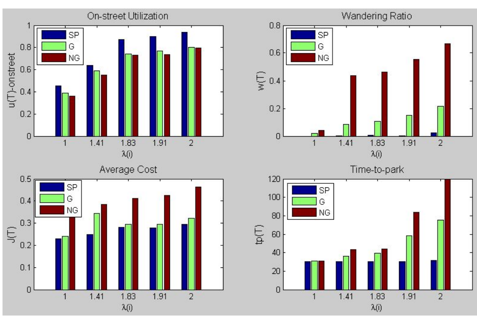

Fig. 4. Simulation Results Under $t_0 = 10$

V. Laboratory Implementation

To check the implementability of the "smart parking" approach, we have built an urban-like testbed as shown in Fig. 5. This contains the basic elements of an urban traffic environment, such as roads, vehicles, traffic lights, urban blocks, etc. To meet the three main requirements of "smart parking," we include the following features. First, vehicle locations and parking spot status are detected by overhead cameras, which serve as a GPS. Second, vehicles communicate with the central allocation system (a computer) by Wifi. We use KheperaIII robots as vehicles, which have a wireless communication capability. Third, we use wireless controllable parking barriers to guarantee a reservation. The computer periodically executes the optimal allocation algorithm and makes assignments, sends the assignment results to vehicles, and controls the opening and closing of each parking barrier.

Due to space limitations, we only have 5 parking spots and 4 vehicles. However, the "smart parking" concept is clearly illustrated in the implementation, including allocations dynamically updated when random events occur, e.g., a vehicle joining the system or delays due to traffic lights. One can also see how vehicles access reserved parking spots that meet their requirements without blindly competing.

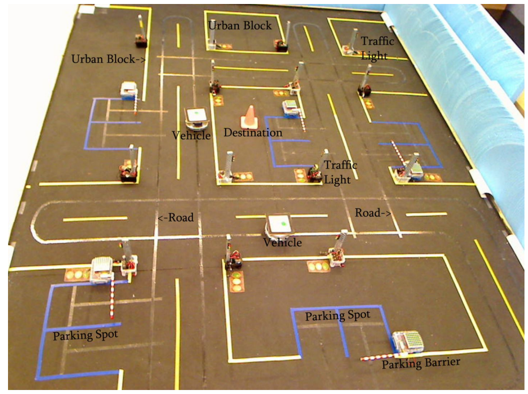

Fig. 5. Urban-like Laboratory Testbed

VI. Conclusions and Future work

We have proposed a "smart parking" system that exploits technologies for parking space availability detection and for driver localization and which optimally allocates and reserves parking spots to drivers instead of only supplying guidance to them. Based on a general dynamic resource allocation framework, we have focused on efficiently determining an optimal allocation strategy for both users and the system. Our simulation results on a case study of parking at Boston University show significant performance improvements over existing parking behavior, including guidance-based systems. Ongoing research focuses on selecting proper decision intervals and threshold parameters, and on the use of dynamic pricing control.

References

[1] http://www.streetlinenetworks.com/.
[2] http://www.intelligentparking.com/.
[3] http://www.parkhelp.com/.
[4] http://sfpark.org.
[5] R. Arnott, T. Rave, and R. Schob. *Alleviating Urban Traffic Congestion*. MIT Press, 2005.
[6] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
[7] K. Cheung and P. Varaiya. Traffic surveillance by wireless sensor network: Final report. Technical report, 2007. Technical Report UCB-ITS-PRR-2007-4, University of California, Berkeley.
[8] Y. Geng and C. G. Cassandras. Dynamic resource allocation in urban settings: A "smart parking" approach. *IEEE Multi-Conference on Systems and Control*, 2011. To appear.
[9] E. Griffith. Pointing the way. *ITS International*, page 72, 2000.
[10] T. B. Hodel and S. Cong. Parking space optimization services, a uniformed web application architecture. *Intelligent Transport Systems and Services, ITS World Congress Proceedings*, pages 16–20, 2003.
[11] ILOG Inc. *CPLEX 6.5 User's Manual*. Incline Village, NV, 1999.
[12] L. Mimbela and L. Klein. *A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems*. The Vehicle Detector Clearinghouse, 2000.
[13] C. J. Rodier and S. A. Shaheen. Transit-based smart parking: An evaluation of the San Francisco bay area field test. *Transportation Research Part C*, (18):225–233, 2010.
[14] D. Shoup. *The high cost of free parking*. APA Planner Press, 2005.
[15] D. Teodorovic and P. Lucic. Intelligent parking systems. *European Journal of Operational Research*, 175:1666–1681, 2006.
[16] R. G. Thompson and P. Bonsall. Driver's response to parking guidance and information systems. *Transport Review*, 17(2):89–104, 1997.
[17] 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. *The Journal of the Operational Research Society*, 52(10):1067–1077, 2001.