Available online at www.sciencedirect.com
SciVerse ScienceDirect
ELSEVIER
Procedia - Social and Behavioral Sciences 54 (2012) 1278 – 1287
Procedia
Social and Behavioral Sciences

EWGT 2012

15th meeting of the EURO Working Group on Transportation

A new "Smart Parking" System Infrastructure and Implementation

Yanfeng Geng, Christos G. Cassandras

Division of Systems Engineering, and Center for Information and Systems Engineering
Boston Univeristy, Brookline, MA, 02446, USA

Abstract

We propose a new "smart parking" system for an urban environment. The system assigns and reserves an optimal parking space for a driver based on the user's requirements that combine 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. Implementation issues including parking detection, reservation guarantee and Vehicle-to-Infrastructure (V2I) or Infrastructure-to-Vehicle (I2V) communication are resolved in the paper. Our system can save driver time, fuel and expense, while reducing the traffic congestion and environment pollution. We also describe a deployment and testing pilot study of the system in a garage at Boston University

Keywords: Smart Parking, Dynamic Resource Allocation

---

1. 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 [1]. 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 [2] 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 cities 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

 doi:10.1016/j.sbspro.2012.09.842

be displayed on variable-message signs (VMS) at major roads, streets, and intersections or it may be disseminated through the Internet [3], [4]. 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 [5], [6]. However, it has been found that using PGI systems, system-wide reductions in travel time and vehicle benefits may be relatively small [7], [8]. 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 [9]. 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 [4]. 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 [11].

Although current parking guidance systems increase the probability of finding vacant parking spots, they have several shortcomings [13]. First, drivers may not actually find vacant parking spots by merely following the guidance. In 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 the *real-time* parking information using their mobile phone [12], there are also safety issues associated with drivers watching parking updates while driving. Second, even if a driver is successfully guided to a parking spot, such a system encourages increasing the probability of finding *any* parking spot at the expense of missing the opportunity for a *better* spot. 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, while 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 [13], [14], we propose a new concept for a "smart parking" system. Instead of providing parking information to a driver and letting him make a parking decision, the "smart parking" system assigns and reserves an optimal parking space based on the driver's requirements that combine proximity to destination and parking cost, while also ensuring that the overall parking capacity is efficiently utilized. We focus on solving the dynamic resource (parking spaces) allocation problem in [13], and present a simulated case study in [14] to illustrate the performance of the system. Simulation results show significant improvement over PGI systems. In this paper, we take a detailed look into the four basic requirements to construct a "smart parking" system, and describe our solution in a pilot implementation in a garage of Boston University.

The rest of the paper is organized as follows. In Section 2, we review the framework and allocation procedure of our "smart parking" system. In Section 3, we describe the four basic requirements in order to build such a system. A pilot implementation at a garage of Boston University is described in Section 4. Finally, we conclude and discuss future work in Section 5.

2. "Smart Parking" System Infrastructure

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.

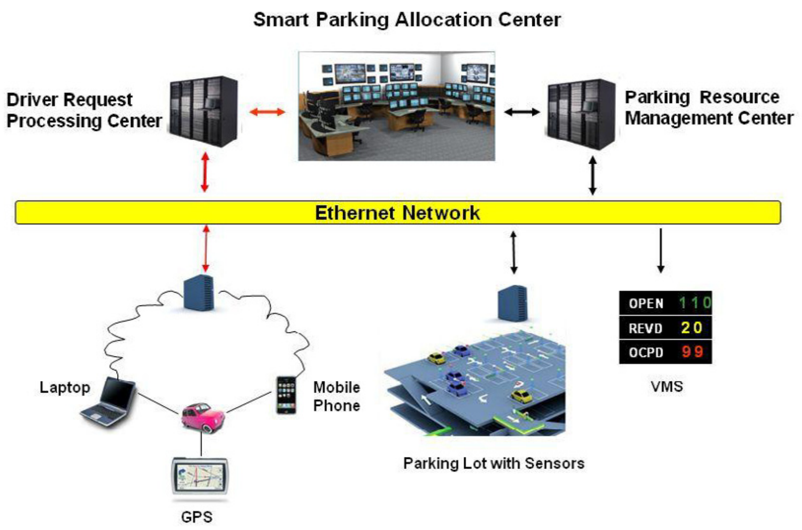

Figure 1. “Smart Parking” Infrastructure

The basic allocation process (see also [13, [14]) 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 whole procedure is described in Fig. 2.

The realization of such a “smart parking” system relies on four main requirements. First, the allocation center has to know the status of all parking spots and the location of all vehicles issuing requests. The second requirement involves effective wireless communication between vehicles and an allocation center. Third, the center must be able to implement a reservation that guarantees a specific parking spot to a driver. Finally, an effective parking resource allocation method needs to be implemented to ensure optimal allocations and reservations. In what follows, we will concentrate on the implementation details of these four requirements.

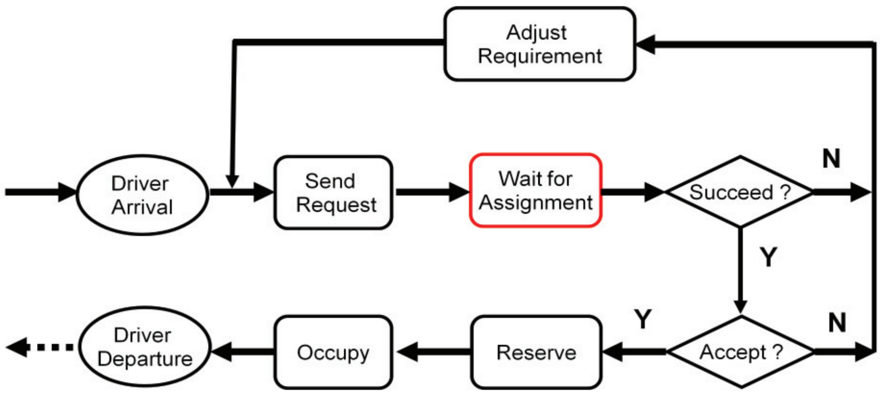

Figure 2. Basic "Smart Parking" procedure

3. "Smart Parking" System Implementation Requirements

3.1. Parking Detection and Driver Localization

First of all, the system relies on the knowledge of real-time parking information, based on which it makes and upgrades allocations for drivers. As already mentioned earlier, current sensing technologies provide several options to monitor parking spots. In our pilot project, we use ultrasonic sensors, which are mounted in MEMSIC IRIS wireless sensor motes. All sensors in one level of a garage or on one street construct an XMesh wireless network, which connects all sensors together and transmits sensing data to the upper tier via a gateway. The XMesh network supports multi-hop routing through which data packets can be relayed from one mote to another. Thus, sensors far away can still transmit data to the gateway via this technology. It also has the capability of quick reconfiguration when adding or removing a sensor to/from the system. These features make it very convenient to deploy the parking surveillance system in either off-street or on-street parking spaces.

Moreover, whenever the system starts to make an allocation, it requires the location information of all cars pending for allocation. Based on this information, it estimates the travelling time to the spot to be allocated, and provides driving directions after the allocation. Current vehicle tracking devices/systems provide the 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 more. This technology has been widely used in bus systems. However, it requires each driver to install a tracking device, which becomes the additional cost for drivers to use the system. In our project, we have built a smartphone application, which contains a function that constantly reads the GPS data in the phone and automatically reports to the system. The smartphone application also has other functions, which will be described later.

3.2. V2I and I2V Communication

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

In our project, 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 are 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 basic information. Drivers may also use the system without registering an ID, but they have to provide the general information each time they send a request. 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 conditions.

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

Notice that here both V2I and I2V communication are implemented through a smartphone application, and data are transmitted through the cellular network. Drivers may reserve a parking spot before a trip, and interact with the system by simply pushing buttons on the smartphone, which will not distract them from driving.

3.3. Reservation Guarantee

Parking reservations are a key feature of the "smart parking" system. In order to implement this function, when a parking spot is reserved by the driver, the system must guarantee that this parking space will not be taken by other cars. For off-street parking resources, it is relatively easy to prevent drivers from taking the spots which have been reserved by others. The system can perform ID checking (with RFID technology) at the gate of a garage or a parking lot. If the driver has a reservation, the gate opens and a spot number will be provided to him. Otherwise, he may be rejected, or allowed to park if there are vacant unreserved parking spaces.

For on-street parking resources, the scheme is more complicated because there is no ID checking capability for on-street parking spaces. Drivers may park in any spot if it is vacant. One method is 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. 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 spot, where different colors indicate different parking spot "states".

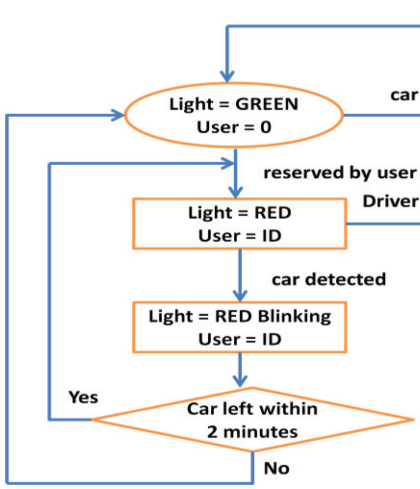

Figure 3. Light indicator state machine

In our project, we use such a light system to implement a reservation guarantee. We set a GREEN light to indicate that a vacant parking spot is available for any driver, a RED light to indicate the spot is reserved by other drivers, a YELLOW light to notify the driver who has reserved this space once he is in its vicinity, and a blinking RED light to signal the driver that he is parking at a spot reserved by others. The three LED lights are connected to and controlled by the IRIS mote. There is a light state machine in the sensor mote as described in Fig. 3. The light is GREEN when vacant and turns RED if it is reserved by a driver. When a driver is arriving at the parking spot, he should notify the system using the smartphone application. The system sends a command to the IRIS sensor mote, which switches the light at his reserved spot from RED to YELLOW. The driver then should be able to recognize his reserved spot and park there. After he parks, the light is turned off until the car leaves. If a driver violates the rule and parks in a space reserved by others, the RED light in that spot starts to blink and the driver should leave. If he does not leave within 2 minutes, the system knows which spot is taken and will either tow the car or issue a ticket with a fine; on the other hand, the system will start to make another assignment for the driver who had reserved that spot. If that driver cannot be reallocated but his reserved spot becomes vacant because the illegal car was towed or left before his arrival, that spot is still reserved for him; otherwise he will get a refund.

3.4. Optimal Allocation

One of the benefits of the "smart parking" system is that if finds the best parking space for each driver. This is done through an efficient allocation algorithm at the SPAC. This optimal allocation problem has been studied in [13] and [14]. In this paper, we focus on how to reduce the problem scale. We first review the formulation of the dynamic parking resource problem (details can be found in [13]).

We adopt a queueing model for the problem as shown in Fig. 4. At the *k*th decision point, we define the state of the allocation system $X(k)$ as follows,

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

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, $j = 1, 2, ..., N$. We assume that each resource j has a known location associated to it denoted by y_j in a two-dimensional Euclidean space, and its capacity is n_j.

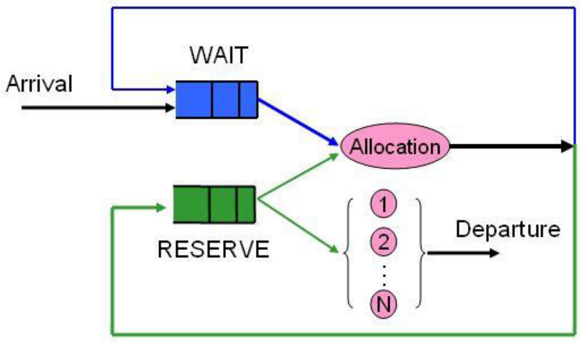

Fig. 4. Queueing Model for Dynamic Resource Allocation

We also define the state of the ith user $S_i(k)$ in the form:

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

where $z_i(k)$ is the location of driver i, $r_i(k) \in \mathbb{R}^2$ is the total time that user i has spent in the reserve queue up to the kth decision point, and $q_i(k)$ is the reservation status of user i:

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

Finally, $\Omega_i(k)$ is a feasible resource set for user i, i.e., $\Omega_i(k) \subseteq \{1, \dots, N\}$ 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 Di, is an upper bound on the distance between the resource that the user is assigned and his actual destination. If the user is assigned a resource j located at y_j, let $D_{ij} = \|d_i - y_j\|$. Then the constraint $D_{ij} \le D_i$ defines a requirement that contributes to the determination of $\Omega_i(k)$ by limiting the set of feasible resources to those satisfy it.

The second attribute for user i, denoted by Mi, is an upper bound on the cost this user is willing to tolerate for the benefit of reserving and subsequently using a parking spot. We assume that each user cost is a function of the total reservation time $r_i(k)$ and the travelling time $s_{ij}(k)$ from the user location at the kth decision time, $z_i(k)$, to a resource location y_j. We use $M_{ij}\left(r_i(k), s_{ij}(k)\right)$to denote the total expected cost for using resource j, evaluated at the kth decision time. Comparing $M_{ij}\left(r_i(k), s_{ij}(k)\right)$to Mi, leads to the constraints

$$M_{ij}\left(r_i(k), s_{ij}(k)\right) \le M_i \tag{4}$$

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 (4). In order to fully specify $\Omega_i(k)$, we further define $\Gamma(k)$ to be the set of free and reserved resources at the kth decision time and set

$$\Omega_i(k) = \left\{j: M_{ij}(k) \le M_i, D_{ij} \le D_{i,}, j \in \Gamma(k)\right\} \tag{5}$$

where, for simplicity, we have written $M_{ij}(k)$ instead of $M_{ij}\left(r_i(k), s_{ij}(k)\right)$. 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.

We 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:

$$J_{ij}(k) = \lambda_i \frac{M_{ij}(k)}{M_i} + (1 - \lambda_i) \frac{D_{ij}}{D_i} \tag{6}$$

where $\lambda_i \in [0,1]$ is a weight that reflects the relative important 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.

Define binary control variables:

$$x_{ij} = \begin{cases} 0, & \text{if user } i \text{ is not assigned to resource } j \\ 1, & \text{if user } i \text{ is assigned to resource } j \end{cases} \tag{7}$$

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

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

s.t.

$$\sum_{j \in \Omega_i(k)} x_{ij} \leq 1 \quad \forall i \in W(k) \tag{8}$$

$$\sum_{j \in \Omega_i(k)} x_{ij} = 1 \quad \forall i \in R(k) \tag{9}$$

$$\sum_{i \in W(k) \cup R(k)} x_{ij} \leq p_j(k) \quad \forall j \in \Gamma(k) \tag{10}$$

$$\sum_{j \in \Omega_i(k)} x_{ij} \cdot J_{ij}(k) \leq J_{iq_i(k-1)}(k) \quad \forall i \in R(k) \tag{11}$$

$$\left[\sum_{n \in \Omega_i(k)} x_{in}\right] - x_{mj} \geq 0, \quad \forall i,j,m \; s.t. \; j \in \Gamma(k), j \in \Omega_i(k), m \in W(k), t_{mj} > t_{ij} \tag{12}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in W(k) \cup R(k), j \in \Gamma(k) \tag{13}$$

This is Mixed-Integer Linear Programming (MILP) problem. Since the MILP problem is NP-hard, it becomes computationally intractable when applied to large urban areas. In such cases we can use the following steps to reduce problem complexity.

1) **Area partition.** We should observe that driver requests are independent if their destinations are far away from each other. Therefore, we can partition the whole area into several smaller districts. For each district, we solve problem (**P**) for all drivers whose destinations are located in that area. For those drivers whose destinations are in the border of two adjacent districts, they are considered for allocation at both districts, and assigned to the best between the two.

2) **Grouping resources**. Even at each district, the total number of parking spots may be large, especially at a business district or a downtown area. However, drivers normally do not ask for a specific spot, but only care which street or which garage to park in. Therefore, all spots in the same garage or parking lot can be treated as one resource. Similarly, we also group all on-street parking spots in the same street block as one resource. The problem scale is greatly reduced with this step.

3) **Discriminating users**. Drivers who are far away from their destination usually do not require a quick allocation result. Moreover, the system does not prefer long-time reservations. 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 is a waste of parking resources and may cause user discontent if they cannot park there. This can be resolved by restricting the number of users in the waiting queue who are assigned a resource. Thus, we introduce a threshold t_0: users only within t_0 minutes away from their destination are considered for assignment; otherwise they are kept in the waiting queue. The number of decision variables in problem (**P**) also decreases with this restriction.

If the problem scale is still large after the above three steps, we may consider relaxing the problem to find a near-optimal solution within a reasonable time [15], [16].

4. Off-street "Smart Parking" Implementation

We have deployed the system in one level of a garage at Boston University, which contains 27 parking spots. We have installed an ultrasonic sensor for each spot to detect its parking status. The ultrasonic sensor motes together with the light indicator are attached on the ceiling above each parking space. The ultrasonic beam faces down to the center of the spot so that a car can be detected if it parks there.

The 27 sensors form an XMesh wireless network. A gateway receives data from each sensor in the network and forwards it to an upper level database, which serves as the PRMC. The real-time parking information is published and updated on the web, and can therefore be obtained by users. Thus our system can still provide service as a normal PGI system. We have built a smartphone application, which drivers can use to send parking requests and make reservations. The application sends all user requests to a server, which operates as both DRPC and SPAC. The server maintains driver requests, solves the optimal allocation problem (**P**), updates the parking status database, and sends commands to control the parking light indicators. Fig. 5 shows the smartphone application and real-time parking information in the website.

Fig. 5 "Smart Parking" smartphone application

5. Conclusion

We have proposed a "smart parking" system that exploits technologies for parking space availability detection and for driver localization and which allocates optimal parking spots to drivers instead of only supplying guidance to them. We have described the system infrastructure and basic "smart parking" procedure. We studied the main requirements to implement such a system and provided the necessary solutions. A pilot implementation project has been carried out in a garage at Boston University.

References

[1] R. Arnott, T. Rave, and R. Schob. *Alleviating Urban Traffic Congestion*. MIT Press, 2005.
[2] D. Shoup. *The high cost of free parking*. APA Planner Press, 2005.
[3] E. Griffith. Pointing the way. *ITS International*, page 72, 2000.
[4] D. Teodorovic and P. Lucic. Intelligent parking systems. *European Journal of Operational Research*, 175:1666–1681, 2006.
[5] K. Cheung and P. Varaiya. Traffic surveillance by wireless sensornetwork: Final report. Technical report, 2007. *Technical Report UCBITS-PRR-2007-4*, University of California, Berkeley.
[6] L. Mimbela and L. Klein. A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems. *The Vehicle Detector Clearinghouse*, 2000.
[7] R. G. Thompson and P. Bonsall. Driver's response to parking guidance and information systems. *Transport Review*, 17(2):89–104, 1997.
[8] 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.
[9] 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.
[10] T. B. Hodel and S. Cong. Parking space optimization services, a uniformed web application architecture. *ITS World Congress Proceedings*, pages 16–20, 2003.
[11] http://sfpark.org.
[12] http://streetline.com
[13] Y. Geng and C. G. Cassandras. Dynamic resource allocation in urban settings: A "smart parking" approach. *IEEE Multi-Conference on Systems and Control*, pages 1-6, 2011
[14] Y. Geng and C. G. Cassandras. A new "smart parking" system based on optimal resource allocation and reservations. *14th Conf. on Intelligent Transportation Systems*, pages 979-984, 2011
[15] S. T. Hackman and L. K. Platzman. Near-Optimal Solution of Generalized Resource Allocation Problems With Large Capacities. *Operations Research*, Vol. 38 (5), 902-910, 1990.
[16] M. Fisher. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, Vol. 27, No. 1, 1981