On-The-Fly Curbside Parking Assignment

Abeer Hakeem
amh38@njit.edu

Narain Gehani
gehani@njit.edu

Xiaoning Ding
xiaoning.ding@njit.edu

Reza Curtmola
reza.curtmola@njit.edu

Cristian Borcea
borcea@njit.edu

Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

ABSTRACT

This paper presents Free Parking System (FPS), a system for assigning free curbside parking spaces to drivers in the cities. FPS has two components: a mobile app running on the drivers' smart phones that submits parking requests and guides drivers to their parking spaces, and a server that manages the parking assignment process. Unlike existing parking systems, FPS is cost-effective as it does not rely on any sensing infrastructure and reduces parking space contention because it provides individual space assignments to drivers. The main novelty of FPS consists of its parking assignment algorithm, FPA, which combines a system-wide social welfare objective with a modified compound laxity algorithm to minimize the total travel time for all drivers. Furthermore, FPS adapts on-the-fly to new parking requests and to parking spaces occupied by drivers who do not use our system. The simulation results demonstrate that FPA reduces the total travel time by factor of 4 when compared to a baseline that mimics the way people search for parking today. It also reduces the travel time by 42% when compared to a greedy parking assignment algorithm. Furthermore, FPA provides substantial improvements even when many parking spaces are occupied by drivers who do not use FPS.

CCS Concepts

•**Information systems** → **Location based services;**
•**Human-centered computing** → **Mobile computing;**

Keywords

Parking Assignment, Travel Time, Social Welfare, Mobile App

1. INTRODUCTION

Many cities rely on free curbside parking to supplement their limited paid parking options, and many people prefer curbside parking to save money. However, finding an available free space during peak hours is challenging. Drivers who cruise in search of free curbside parking pay with time instead of money. Their cruising congests traffic, pollutes the air, and wastes fuel. For example, studies have found that 30% of the drivers in congested traffic were searching for curbside parking [20]. In a 15-block survey area in New York, drivers cruised a total 945,000 extra miles per year as they searched for curbside parking [19]. This accounted for a waste of over 47,000 gallons of gasoline and produced around 728 tons of carbon dioxide.

The free curbside parking problem is exacerbated by the drivers' lack of knowledge regarding available parking around their destinations. For example, let us assume that a driver is going to a concert attended by many people. Naturally, she wants to find parking as close to the concert hall as possible. However, as she approaches the concert hall, the driver wonders if she should park as soon as she sees an empty space or should she try to look for a closer space and potentially lose the empty space she saw. Our research addresses this dilemma. The goal of this research is to build a mobile app that allows the driver to input her destination and assigns her the best empty parking space available with respect to her destination and a system-wide optimization objective, which aims to minimize the overall travel time for all drivers.

Academic research as well as public and private initiatives have made remarkable efforts in recent years to solve the free curbside parking problem. These efforts have focused on applications that rely on new infrastructure to enable mobile devices to find curbside parking spaces in urban environments. A prime example of this type of applications is SFPark [1]. It relies on 8,000 sensors embedded in the streets of San Francisco, which can tell whether a parking space is available or not. The application shows a map with the available parking spaces in the driver's search area. The sensors cover about 25% of the available curbside parking in the city and cost USD $23M.

There are two problems with this type of solution. The first is the cost involved in deploying and maintaining the sensor infrastructure. The second is that all drivers see the same parking availability map at any given time, and many of them will compete for the same parking spaces. This will lead to congestion and driver frustration because the application does not attempt to provide individual guidance for drivers to specific parking spaces in order to minimize parking space contention.

This paper proposes a Free Parking System (FPS) that solves both problems. FPS does not need a sensing infrastructure, as it relies on driver cooperation to maintain the parking availability map. Furthermore, FPS not only guides

and assigns drivers to available parking spaces close to their destinations, but also reduces the total travel time (i.e., the sum of the driving time from the moment the parking request is submitted to the moment the car is parked and the walking time from the parking space to the destination) for all drivers. Unlike previous solutions, FPS is more economical and is able to optimize the travel time as a system wide objective, which improves the overall "social welfare". It also operates on-the-fly as it handles new parking requests received over time and parking spaces that are found to be occupied by drivers who do not use FPS. It is important to note that FPS does not assume that all drivers use our system. It discovers the spaces occupied by unsubscribed drivers when the subscribed drivers report them. Then, it considers these spaces available after a time period based on the age of the observation reports.

FPS has two components: a mobile app running on drivers' smart phones and a server, which is responsible for assigning parking spaces to drivers and providing individual parking guidance. In addition to submitting parking requests and providing parking guidance to drivers, the app reports to the server when a car is parked and when it leaves a parking space using input either from the drivers or from an activity recognition algorithm based on phone sensors (e.g., accelerometers and GPS). The server manages information about available parking spaces and handles parking requests in such a way as to optimize the social welfare system objective.

FPS uses a novel free parking assignment (FPA) algorithm to achieve this goal. FPA uses the social welfare criterion to solve driver contention for the same parking spaces in such a way as to minimize the total travel time to destination. FPA delays the parking space assignment as long as possible in order to accumulate more parking requests and thus perform a more efficient assignment. We created a modified version of the compound laxity algorithm [21] to determine how long a request can be delayed before it must be assigned a space. Our algorithm minimizes the total driving time to the parking spaces. By combining social welfare and compound laxity assignments, FPA is able to minimize the total travel time for all drivers.

We have evaluated FPS using two baseline assignment algorithms and two versions of FPA: (i) a naive algorithm that assumes a breadth-first-search for parking spaces around the destinations; (ii) a greedy algorithm that assigns the closest available space to destination as soon as the driver enters a pre-determined parking space allocation area; (iii) a basic FPA version that considers spaces occupied by unsubscribed drivers to remain occupied forever; (iv) an enhanced FPA version, FPA-1, that re-considers the spaces occupied by unsubscribed drivers after a time period. We performed extensive simulations using SUMO [5], a real map of a part of New York City with 1024 parking spaces, and as many as 768 drivers looking for parking. The results demonstrate that FPA reduces the total travel time by more than 4 times when compared to the naive algorithm and by 42% when compared with greedy when all the drivers use our system. FPA also provides substantial improvements even when 25% of the spaces are occupied by unsubscribed drivers, and FPA-1 performs the best among all algorithms in this scenario. For example, FPA-1 reduces the travel time by 52% compared to greedy.

The rest of this paper is organized as follows. Section 2 presents an overview of FPS. The greedy assignment algorithm is introduced in Section 3 in order to emphasize the problems with this simple solution and motivate the need for a more complex assignment algorithm. Section 4 defines the assignment problem and the social welfare optimization criterion. FPA is described in Section 5, and the evaluation results are presented in Section 6. Section 7 discusses related work. The paper concludes in Section 8.

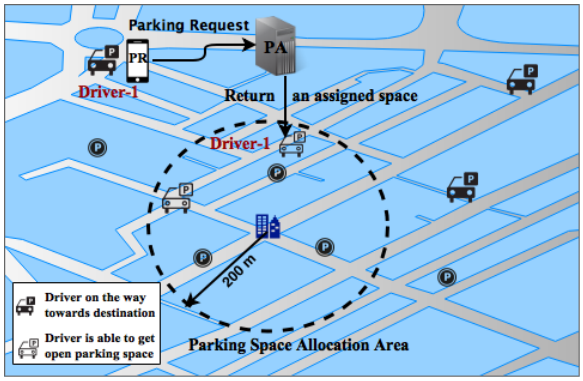

Figure 1: An FPS Example

2. FPS OVERVIEW

As shown in Figure 1, the FPS system consists of two components, namely parking requestor (PR) and parking allocator (PA). PR is a mobile app that runs on each driver's smart phone and is in charge of submitting parking requests, reporting parking status to PA, and guiding drivers to the assigned parking space. Each parking request contains the requesting driver's current location and the desired destination. The reporting of parking status relies on drivers manually registering their park and de-park status. Alternatively, the app can learn this status from an activity recognition service running on the phone [9]. The parking allocator (PA) runs on a central server, where it manages the incoming parking requests and aggregates the PR reports to determine the available parking spaces. For availability computation, PA assumes that not all drivers participate in our system, i.e., not all drivers are equipped with the PR component. This means that some parking spaces are occupied by drivers that are not part of FPS. The FPA algorithms (see Section 5) running at PA discover and iteratively monitor these parking spaces. In addition, PA could estimate the number of spaces that are occupied by non-participating drivers in order to reduce the number of unsuccessful assignments [13, 24].

The basic idea of the FPS parking assignment is described as follows. Drivers who are looking for parking spaces use PR to send requests to PA. All incoming requests are streamed into a queue and are processed first-come-first-serve. For each request, PA allocates the available parking space that best matches the driver's destination. PA does not assign parking spaces to drivers who are far from the destination in order to reduce the likelihood of assigned parking spaces not being available upon the driver's arrival. Such a situation could happen due to unsubscribed drivers, and the likelihood that a space is taken by an unsubscribed driver increases over time. Therefore, FPS just informs the

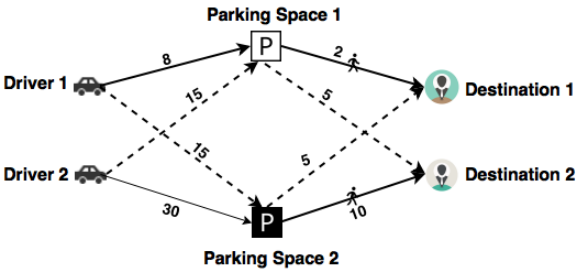

Figure 2: An Example of Parking Assignment

drivers that are far from their destinations that they will be assigned parking when they enter a *parking space allocation area* (see Figure 1). PR shows the drivers this area on the map, so they know when they should expect to receive a parking space as they approach the area. The*parking space allocation area* is defined as a circle with the destination at its center. We determined experimentally that the radius should be initially set to the average length of the roads within the whole region managed by a PA (e.g., a zip code). Then, the radius is adjusted periodically based on the parking occupancy rate in the area: the radius is increased when the occupancy becomes higher.

Once a driver enters the parking space allocation area, her parking request is scheduled for assignment and the assigned space is returned to the driver. FPS makes the assignment decision in such a way as to minimize the total travel time of the drivers.

3. STRAWMAN SOLUTION: GREEDY

A strawman solution for the FPS's parking space assignment algorithm is a greedy strategy that minimizes the travel time for each individual driver on a first-come-first-serve basis. Unfortunately, this strategy cannot guarantee that the total travel time for all drivers is minimized. On the contrary, the greedy strategy may lead to substantial increases in the total travel time.

For example, consider the parking problem shown in Figure 2, in which edge labels represent travel time in minutes. The travel time for each driver is the sum of the driving time to the parking space and the walking time between the parking space and the actual destination. Greedy yields the (driver, parking space) assignment $(Driver1, space1)$, $(Driver2, space2)$, and a total travel time of 50 minutes. On the other hand, there is another possible assignment $(Driver1, space2)$, $(Driver2, space1)$ with a total travel time of 40 minutes. This requires $Driver1$ to drive to a farther space, $space2$, rather than driving to $space1$ which is closer. An assignment that minimizes the overall total driving time is possible when a central authority can choose this parking assignment. We believe it is worth designing more advanced assignment algorithms that maximize the social welfare (e.g., minimize the total travel time over all drivers) because they will lead to less pollution, less wasted time in congestions, and overall better travel time for all the drivers. This, of course, is achived at the expense of slightly larger travel times for some drivers when compared to the greedy strategy.

4. PARKING ASSIGNMENT PROBLEM FORMULATION

In this paper, we consider a parking assignment problem defined as follows. Given a set of drivers, each of whom needs to reach a specific destination, and a set of curb-side parking spaces, we would like to assign the parking spaces to drivers in order to satisfy a system-wide objective. Let $S = \{s_1, s_2, ..., s_m\}$ be the fixed set of curb-side parking spaces distributed across a city region. Let $V = \{v_1, v_2, ..., v_n\}$ be the finite set of drivers that are trying to reach destinations in the considered city region. In this paper, we consider the number of drivers to be less or equal to the number of parking spaces. The drivers look for parking spaces close to their destinations, which include places such as banks, shops, houses, parks, hotels, and restaurants among others. Similar to the parking spaces, the destinations are geographically dispersed across a city region. We assume that drivers move according to the legal speeds and the congestion level on different road segments. We also assume that each driver v_i's smart phone can compute the approximate driving time to her destination, $D(O_i, des(i))$, simply based on the geographical distance between her original location O_i (i.e., the location from where the parking request has been submitted) and the destination $des(i)$. This information is attached to the parking request and is updated by driver's GPS as the driver approaches the destination.

The travel time for a driver v_i to reach her destination $des(i)$ includes two components:

- the driving time $D(O_i, s_j)$ from v_i's original location to a parking space s_j, and
- the walking time $W(s_j, des(i))$ from the parking space s_j to the destination $des(i)$.

Our goal is to determine an assignment Y of drivers to parking spaces that maximizes the system-wide social welfare. The social welfare is maximized by an assignment that minimizes the following objective:

$$\sum_{i=1}^{n} D(O_i, s_j) + W(s_j, des(i)), \quad n = |V|, s_j \in S \qquad (1)$$

In Y, the assignment of a driver v_i to a parking space s_j can be represented with a binary decision variable $y_{ij} : v_i \rightarrow s_j$:

$$y_{ij} = \begin{cases} 1, & \text{if } v_i \text{ is assigned to } s_j \\ 0, & \text{otherwise} \end{cases} \quad 1 \leq i \leq n, 1 \leq j \leq m \qquad (2)$$

$$\sum_{i=1}^{n} y_{ij} \leq 1, \quad 1 \leq j \leq m \ (i.e., s_j \in S) \qquad (3a)$$

$$\sum_{j=1}^{m} y_{ij} = 1, \quad 1 \leq i \leq n \ (i.e., v_i \in V) \qquad (3b)$$

A valid assignment must satisfy two constraints. One constraint requires that one driver can receive only one available parking space, as described in Equation 3a. The other constraint requires that an available parking space can be

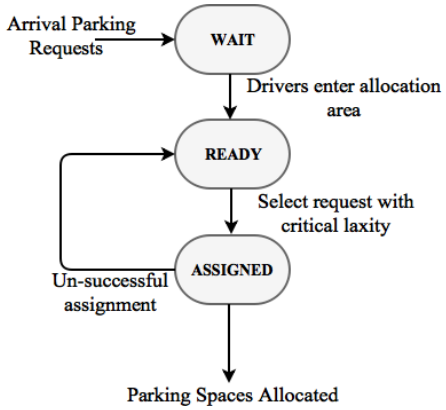

Figure 3: State Transition of Drivers in FPS

assigned to at most one driver, as described with Equation 3b. The violation of either constraint leads to invalid assignments, which are either wasteful (e.g., assigning multiple parking spaces to the same driver) or infeasible (e.g., multiple drivers sharing the same parking space).

5. FPA ALGORITHMS FOR PARKING SPACE ASSIGNMENT

This section presents two versions of FPA, a dynamic parking assignment algorithm used by the FPS system to manage driver requests over time subject to social welfare optimization. The algorithm handles a set of driver requests coming to the system independently by assigning available parking spaces to the drivers to satisfy their requests.

By reducing the problem of finding available parking spaces to an instance of the minimum-cost network flow problem on a directed bipartite graph, a strongly polynomial time can be achieved [3]. Although this method results in a minimum walking time and shows good computational properties, it can hardly meet our system-wide objective described in Equation 1 for two reasons. First, this method is designed for offline settings where the number of parking spaces and drivers are known and cannot be customized to a real-life, dynamic situation. Second, it only minimizes the total walking time.

Therefore, we propose a different algorithm to construct the parking assignment process dynamically over time and to maximize the social welfare described in Equation 1. The algorithm addresses two challenges. One is the selection of parking spaces, i.e., which parking space should be assigned to each driver to satisfy her request, and the other is when a parking space should be assigned to a driver. To address the first challenge, the algorithm tries to assign to each driver the parking space closest to her destination. Assigning parking spaces far away from the destinations increases driving distance and/or walking distance. To address the second challenge, the algorithm assigns a parking space to a driver when she approaches the destination and is about to look for a parking space. Assigning parking spaces too early reduces the utilization of parking spaces. Assigning parking spaces too late may results in increasing driving time and bad user experience.

Specifically, FPS periodically examines and updates the status of the drivers and their requests. The period can be determined as a function of the road network structure, parking spots distribution, and parking requests distribution. In our simulations, we experimentally determined that a period of 2s, which provides a good trade-off between performance and overhead. FPS moves the requests through the states shown in Figure 3. These states are described below:

- WAIT: When a driver request comes to the system, it is stored in a FIFO queue, waiting to be scheduled.
- READY: when the driver moves into the parking space allocation area, the request is marked as *READY*. FPS schedules READY requests using FPA, which will be described later in this section.
- ASSIGNED: The request enters this state when it is selected by FPA and assigned to a parking space. The request stays in the *ASSIGNED* state until the driver successfully park in her assigned parking space. If the assigned parking space is found to be occupied by an unsubscribed driver when this driver tries to park there, the request moves back to the *READY* state with a high priority assignment. The request is finally removed from the system when the driver leaves the parking space. The driver or its phone app will notify the system when the car leaves the parking space. To deal with the case in which the notification is not received (e.g., when the driver's phone is turned off or disconnected), each assignment has an expiration time, after which the request is also removed from the system and the parking space is deemed available again.

During each period, the main task of FPS is to select requests from the *READY* state and assign them. This is the job of the FPA algorithm, and its main steps are described in Algorithm 1.

Algorithm 1 FPA Pseudo-code

```
1: Given a destination des(v_i) and an estimated driving duration to destination D_i for each driver v_i ∈ V
2: Preallocation:
3: for each request v_i in READY state do
4:     Allocate to v_i the closest available parking space to des(v_i)
5: end for
6: Preallocation Adjustment:
7: for each request v_i in READY state do
8:     if v_i shares a parking space with another request then
9:         Find a new parking space for v_i that minimizes the total walking time
10:    end if
11: end for
12: Update the laxity of each driver v_i in READY state based on D_i and its currently allocated parking space
13: Search for a READY driver v with the minimum laxity value
14: Assignment:
15: Finalize the parking space assignment for v and change its state to ASSIGNED
16: Show the parking space on the smart phone of v.
```

For each request in the *READY* state, FPA first preallocates to the driver the closest available parking space to

her destination (lines 3-5). Then, it tests whether the pre-allocation can be a valid assignment for each request. The pre-allocation is valid if a parking space is not pre-allocated to more than one driver, as defined by 3a and 3b. If it is valid, FPA continues with line 12. If not, the system immediately adjusts the pre-allocation by re-allocating other parking spaces to some of the drivers to remove the duplicated assignments of parking spaces (lines 7-11). We use the solution to the flow problem described in [3] to select parking spaces since it can minimize the total walking time.

Note that the pre-allocation and the adjustment of pre-allocation do not actually assign the parking spaces. The actual assignments are delayed and take place only when the requests become urgent (lines 12-13). The urgency is measured by the *laxity* value $B(v_i)$ of each request v_i, which is defined as follows:

$$B(v_i) = min(D(C_i, s_j), D(C_i, des(i))) \quad (4)$$

where $D(C_i, s_j)$ is the estimated driving time of driver v_i from her current location C_i to the parking space s_j; and $D(C_i, des(i))$ is the driving time of the driver v_i from her current location to her destination. The intuition is that a parking space must be assigned to a driver before she reaches either her destination or an available parking space close to her destination (represented by the parking space pre-allocated to her). Thus, the smaller the laxity value is, the more urgently the request assignment must be finalized. When the laxity values are calculated, we round the values to whole seconds. FPA compares the laxity values of READY requests and selects the requests with the smallest laxity value to finalize their assignments.

The operations in lines 3-15 are repeated periodically to handle the remaining requests in the queues and the newly-arrived requests.

While we assume that FPS drivers are generally representative of the entire driving population, we do not assume that all or even a large fraction of drivers will use FPS. Therefore, FPS drivers may compete for parking spaces with non-FPS drivers, which we call unsubscribed drivers. Figure 4 illustrates how FPS manages parking spaces. Since not all the parking spaces are available to FPS drivers, FPS needs to maintain a list of spaces that may be potentially available (spaces 2, 5, and 6 in Figure 4). Free spaces may be detected in different ways. For example, the mobile app of the subscribed drivers can inform FPS when they leave a parking space (i.e., using algorithms based on analysis of GPS and accelerometer readings). Many vehicles are equipped with cameras, through which the availability of nearby parking spaces can be visually confirmed.

FPS also needs to keep track of occupied spaces and avoid assigning these spaces. While the spaces allocated by FPS itself can easily be maintained, there are spaces taken silently by unsubscribed drivers. For example, spaces 2, 3, 6, 8 and 10 are occupied by unsubscribed drivers in Figure 4. FPS relies on subscribed drivers to report these spaces to the system when they find that their parking spaces have already been occupied. When it receives such reports, FPS marks the spaces as "observed_occupied"; spaces 3, 8, and 10 are such examples in Figure 4. Then, FPS puts the requests of the drivers who reported these spaces back in the *READY* state. In our example, FPS does not yet know that spaces 2 and 6 are occupied because no subscribed driver has reported them. Such spaces are called "hidden spaces."

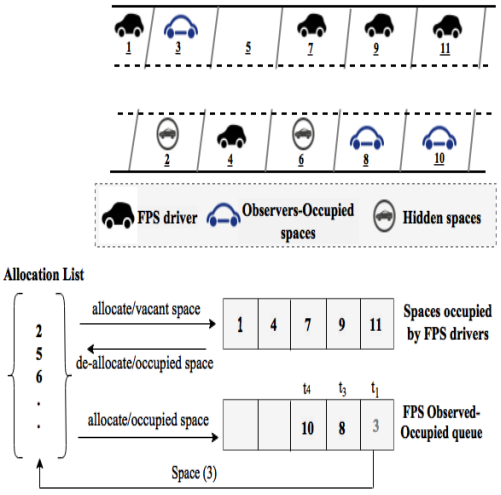

Figure 4: Illustration of Parking Spaces in Different States

Parking spaces marked as "observed_occupied" will not be assigned to other requests to avoid unsuccessful assignments. However, permanently marking parking spaces as "observed_occupied" inevitably reduces the utilization of parking spaces since "observed_occupied" spaces may become available later. To solve this problem, we propose FPA-1, an enhanced version of FPA to reclaim "observed_occupied" spaces. FPA-1 keeps track of how much time subscribed drivers occupy their parking spaces and maintains an average parking time value. Instead of this global average parking time, FPS could maintain per-street averages for higher accuracy. FPS assumes that "observed_occupied" spaces may also be occupied for similar amounts of time with the average parking time of subscribed drivers. When a space is reported to be taken by an unsubscribed driver, FPA-1 moves the space to a queue, named *observed_occupied queue*, and assigns a timer to this space, which expires after the average parking time. When the timer expires, the space is moved back to the allocation list (e.g., space 3 in Figure 4).

6. EVALUATION

This section evaluates the performance of FPA and FPA-1 when compared to Greedy and a Naive solution. Greedy assigns parking spaces to drivers as soon as they reach the initial parking allocation area in a first-come-first-serve manner. When selecting an available parking space for a driver, it always chooses the space closest to the driver's destination. The Naive strategy assumes the driver goes to the destination and, once there, she starts a breadth-first-search for parking spaces along the nearby road segments.

The evaluation is done via simulations over a real road network. The experiments simulate two different scenarios: *subscribed-driver-only scenario*, which assumes that all drivers in the system use FPS; *unsubscribed-driver-interference scenario*, which assumes there are a number of drivers who have not subscribed to FPS. In the second scenario, unsubscribed drivers may occupy, without notifi-

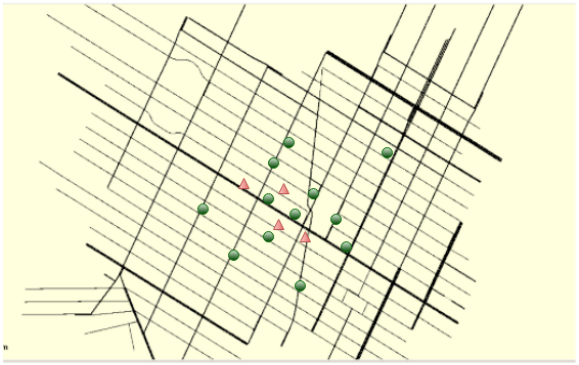

Figure 5: Road Network Used in Experiments with a Few Parking Spaces (green circles) and Destinations (red triangles) Illustrated Along the Roads

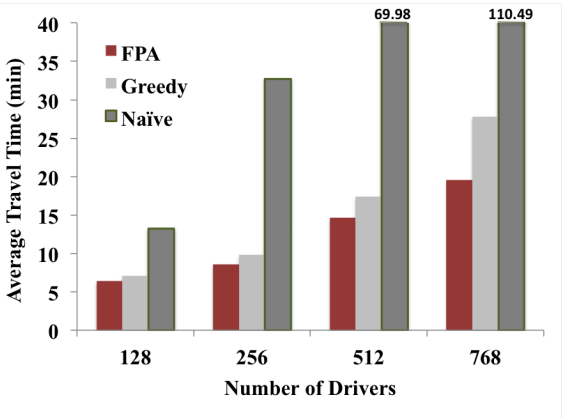

Figure 6: Average Travel Time with Different Number of Drivers and a Constant Number of Destinations (8)

cation, parking spaces known to the system as available.

We use *average travel time* metric to compare the performance of different assignment strategies. For each driver, it includes the time spent on driving to the parking space and walking from the parking space to the destination.

6.1 Simulation Setup

In our experiments, we use SUMO [5], to simulate vehicles going to their destinations in a business district in Manhattan, New York City. The road network and the locations of curbside parking places are imported into the simulator based on the real map of the district. The total number of parking spaces is 1024, and the total number of travel destinations is 400. Figure 5 shows the road network with 4 destinations (triangles) and 12 parking spaces (circles) around these destinations.

The starting locations and the destinations of the vehicles are randomly chosen. However, the destinations are chosen from a small region in the center of the map to ensure enough contention for parking spaces. Each vehicle moves along its route at the legal speed limit of each road on the route and the movement is restricted within the map. Every vehicle may adjust its speed for safety driving and to follow traffic laws. For example, it must keep a reasonable distance from the vehicle in front of it or it slows down when approaching an intersection or its parking space. Once a vehicle parks, we calculate the driving time and the walking time; For walking time, we consider an average speed of 1.4 m/s, which is reasonable for adults (men and women) [7, 15].

To simulate the scenarios with different parking densities and contention levels, we varied the number of vehicles, the number of parking spaces, and the number of destinations, which are as specified in each individual experiment. FPS starts each test with 1024 vacant parking spaces. The arrival rate of the requests falls within the range of 1 to 5 requests per second. The period for the parking assignment algorithm is set to 2s; this value was determined experimentally to provide a good trade-off between performance and overhead. For each experiment, we collected results from 5 runs and averaged them.

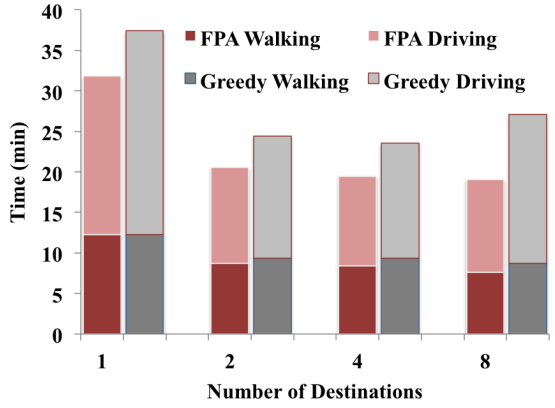

Figure 7: Walking and Driving Time of FPA and Greedy for Different Number of Destinations and a Fixed Number of Drivers (768)

6.2 Results for Subscribed-Drivers-Only Scenario

Figure 6 compares the performance of FPA, Greedy, and the Naive algorithm by varying the number of drivers from 128 to 768 with a fixed number of destinations (8) distributed in the centroid area of the map.

The results demonstrate that FPA outperforms the comparison algorithms. When the number of drivers increases, the average travel time grows quickly for the Naive algorithm. This is because the contention for the parking spaces close to the destinations leads to substantial traffic congestion, which is exactly what we observe in real life. FPA decreases the average travel time by a factor 4 compared with the Naive solution for 768 drivers (110.49 minutes). These results demonstrate the substantial impact FPS can have on driving and parking in the cities. As expected, the average travel time increases for FPA and Greedy with the number of drivers, but this increase is sub-linear. This is because these algorithms avoid having the drivers go to the destinations and then starting to search for parking.

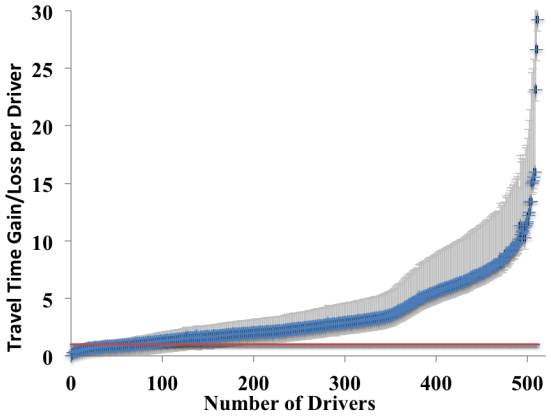

Figure 8: Distribution of Travel Time Gain/Loss for All Drivers in the System with a Fixed Number of Destinations (8). Gains Are Values Greater Than 1, and Losses Are Values Less Than 1. Error Bars Are Shown

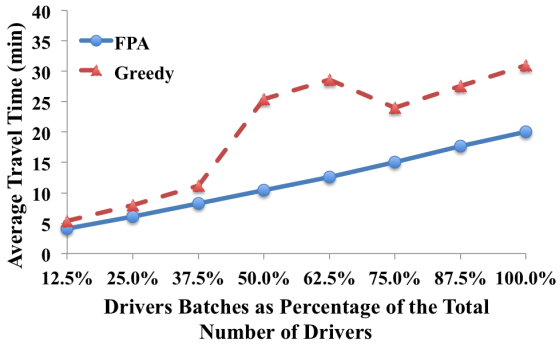

Figure 9: FPS Consistency over Time: 768 Drivers Divided into 8 Equal Batches as a Function of Their Arrival Time at Destination. The number of destinations is 8

Compared to Greedy, FPA is more effective as it reduces the average travel time by as much as 40%. These results can be explained by the design of FPA, which optimizes the system-wide travel time. As discussed, maximizing the social welfare leads to lower walking time, and our modified compound laxity algorithm leads to lower driving time.

Figure 7 shows the average travel time of 768 drivers when the number of destinations is varied from 1 to 8. The figure also plots the contribution of walking time and driving time in the total time. With more destinations, the advantage of FPA over Greedy becomes more prominent. Compared to Greedy, FPA reduces the average travel time by 18% in the one destination case and 42% in the 8 destination case. The reason is that, with more destinations, there is more space for FPA to perform optimization by balancing the driving and walking distances of the drivers with different destinations. Thus, both average driving time and average walking time can be reduced with FPA. As shown in Figure 7, FPA can reduce the average driving time by up to 61% and reduce the average walking time by up to 14% relative to Greedy. We observe that Greedy with two and four destinations performs better than with eight destinations. The reason is that some parking spaces could be allocated for more than one destination and Greedy is not able to allocate them effectively (i.e., similar to the example shown in Figure 2). This phenomenon becomes significant as the number of destinations increases to 8.

Since FPA minimizes the total travel time for all drivers, one may ask how is the performance of individual drivers impacted by our algorithm. To answer this question, we conduct an experiment to find out the travel time gains or losses for individual drivers. To measure the gains/losses, we calcuate the ratio between the travel time obtained by the Naive algorithm and the travel time obtained by FPA for each driver. If the ratio is higher than 1, the driver has benefited from FPA. Otherwise, the driver has not. Then for each run of the experiment, we sort the drivers in the ascending order of these ratios. We then, average the ratios for these sorted drivers as shown in Equation 5.

$$\frac{\sum_{j=1}^{N}(D_i^j(Naive)/D_i^j(FPS))}{N}, \quad i \in V \tag{5}$$

where N is the number of runs, and D_i^j is the total driving time for the driver in position i in the sorted driver list for experiment j for both Naive and FPS algorithms.

Figure 8 plots the distribution of individual travel time gains/losses for 512 drivers. The results show that 87.8% of the drivers obtain gains, and some of them have very large gains. Nevertheless, the number of drivers with losses is not negligible. From a practical point of view, a few bad experiences could impact the adoption rate of FPS. Therefore, we plan to investigate methods to limit the number of drivers who experience losses and bound the loss ratio to low values.

In the next experiment, we analyze the behavior of FPA and Greedy over time. We divide 768 drivers in 8 equal batches based on the time they arrive at their destinations.

Figure 9 shows that FPS performs consistently better than Greedy during the whole parking assignment process as new drivers enter the system over time. As expected, the average travel time for earlier batches is lower as there are more parking spaces available at locations closer to destinations when they arrive. Also, the difference between the two algorithms is not large because Greedy can perform a good assignment under these conditions. However, we notice that FPA performs substantially better than Greedy (up to 1.5 times) for the middle batches. Since Greedy simply moves each vehicle toward the closest parking space available, the total driving time and therefore congestion are higher, especially when the number of assigned vehicles increases and the number of available spaces decreases. Therefore, in Greedy, drivers waiting to be assigned are congested with drivers heading to their assigned spaces. The average travel time in the last three batches decreases because the assigned drivers park in their spaces and most of the vehicles on the road are waiting to be assigned. For the later batches, FPA is still clearly better, but it does not have as much room for optimization as it has for the middle batches. This is because fewer parking spaces are available.

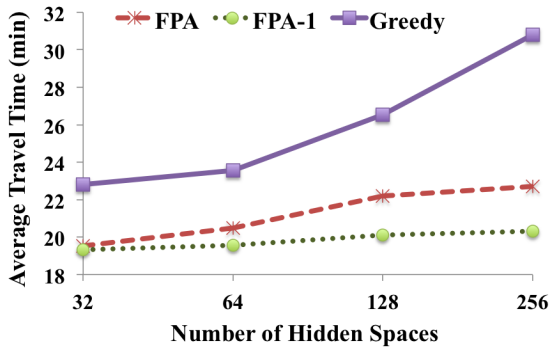

Figure 10: Average Travel Time when Varying the Number of Hidden Spaces: 768 Drivers and 8 Destinations

6.3 Results for Unsubscribed-Driver-Interference Scenario

To test the capability of FPS to tolerate interference from unsubscribed drivers, we randomly selected a number of spaces located in the parking space allocation area and marked them as "hidden" spaces, indicating that they are currently occupied by unsubscribed drivers (see Figure 4). FPS is not aware of a "hidden" space until a vehicle is assigned to that space and finds that the space is taken (i.e., an "observed_occupied" space). During the experiment, the "observed_occupied" spaces become available over time to simulate unsubscribed drivers leaving their parking spaces. The times for the "hidden" spaces to become available are assumed independent and exponentially distributed, but the average parking time for unsubscribed drivers is same to the average parking time for subscribed drivers. As the number of "hidden" spaces increases, we increase the radius of the parking space allocation area proportionally, such that there are still enough parking spaces available to the subscribed drivers.

Figure 10 compares the performance of FPA, FPA-1, and Greedy when the number of hidden spaces is varied from 32 to 256. We observe that both FPA and FPA-1 outperform Greedy, and their relative performance when compared with Greedy increases with the number of hidden spaces. We also notice that FPA-1 achieves lower average travel time than FPA, and its performance is almost constant. FPA-1 reduces the average travel time by 10% relative to FPA and 33% relative to Greedy on average. These results demonstrate that FPA-1 adapts very well to the interference caused by unsubscribed drivers.

7. RELATED WORK

Prior work has studied the problem of managing parking spaces from several angles. For example, there are proposals to install additional infrastructure to detect and monitor the availability of curbside parking. Ultrasonic sensor technology on top of each parking space or on a vehicle door can be used to determine the spatial dimensions of open parking spaces [17, 12]. Wireless sensors are also used to determine open spaces in a parking facility [16]. However, these proposals are expensive to deploy and maintain. For example, ParkNet [12] costs USD $500 for each parking space, and SFpark [1] costs USD $400 for each vehicle. The solution we propose in this paper does not rely on expensive infrastructure; instead, we rely on mobile phones, which is a cheaper, more convenient, and more flexible alternative.

There are several proposals focused on detecting and estimating the availability of parking spaces by considering only mobile phones [24, 13, 14]. In [24], a software solution for detecting and predicting the availability of curbside parking spaces is presented. The solution uses GPS and/or accelerometer sensors to automatically detect when drivers park and remove their cars and where the parking spaces are. Nawaz et. al. propose a smart phone based sensing system that leverages the ubiquity of WiFi beacons to monitor the availability of street parking spaces [14]. PocketParker is a system that predicts parking space availability in a parking lot. It detects arrival and departure events by leveraging existing activity recognition algorithms [13]. Currently, FPS learns the available parking spaces from the apps running on the drivers' smart phones. In the future, it could use existing work on predicting the parking space availability in order to minimize the number of unsuccessful parking assignment attempts. The novelty of FPS, however, is not focused on detecting available parking spaces. Its novelty is to assign optimal parking spaces to drivers to maximize the social welfare of the system.

Other prior works have focused on dissemination of reports of available parking spaces [23, 22, 8]. The focus of [23] was on peer-to-peer dissemination of parking reports. Vehicular ad hoc networking (VANET) is used in [22] to search for open parking spaces. As the vehicle navigates, it continuously receives reports about available spaces close to the area the driver intends to park from oncoming traffic. In [8], the proposed protocol does not disseminate the same information to all drivers in order to avoid competition among them for the same parking spaces. This type of solution is not ideal for the curbside parking problem or a situation when there are only a few parking spaces available because the availability of parking spaces in a crowded area can change quickly. Therefore, the lack of a reservation system makes it possible that drivers arrive to a fully occupied area and could also result in unnecessary overcrowding of certain areas.

There is also prior work on reservation systems for parking spaces. Boehle [6] presents a centralized reservation system. A parking service constantly gathers traffic data from participating vehicles. This data is then used to determine time-optimal routes from the vehicles' current position to the parking spaces. Delot et al. [8] propose a peer-to-peer reservation system in VANET. Parking spaces are reserved by requesting spaces to a specific peer called the coordinator for each space. However, these proposals do not seek to optimize any system-wide objectives. In contrast, FPS optimizes parking space allocation to maximize the social welfare objective.

Solutions based on differential pricing for the parking space assignment problem have been proposed as well [11, 4]. Mackowski et al. [11] developed a demand-based real-time pricing model to optimally allocate parking spaces in busy urban centers. FPS, on the other hand, does not require any pricing data as it deals with free spaces. Ayala et al. [4] developed a pricing model to minimize the system-wide driving

distance. However, the proposed pricing approach is offline in nature, as the number of vehicles and resources are known in advance and do not dynamically change. In contrast, our system handles dynamic vehicle and parking space data.

The work in [3] views the problem of finding parking as a competition for resources and presents a game-theoretic framework to model it. It considers both centralized and distributed models. For the centralized model, it shows an optimal solution can be computed in (strongly) polynomial time, but it does not provide a constructive algorithm on how to find the solution; furthermore, the model only considers an offline setting in which all parking requests are known in advance.

A survey conducted to understand the needs of drivers from the perspective of parking infrastructure and smart services [18] outlined the fragmentation of public and private parking providers, each one adopting their own technology. FPS can help avoid this problem, as it does not depend on any infrastructure and works on regular smart phones.

Parkarr is a mobile app that uses a check-in system to connect people who are looking for parking spaces with those leaving their spaces [2]. With Parkarr, drivers publicize her space and departure time to other drivers in need of parking spaces. However, it cannot address the problem of high demand, when multiple drivers chase the same space. FPS, on the other hand, was designed to solve this problem.

Recently, Geng and Cassandras [10] developed a model showing the promise of real-time parking assignment in order to achieve system optimal objectives (resource utilization). However, this model is limited to the closed confines of parking garages, and it does not consider the driving time cost which is an essential factor to minimize the total travel time as well as to reduce congestion. Unlike this work, FPS focuses on these factors in its parking assignment algorithms.

8. CONCLUSION AND FUTURE WORK

This paper considered the problem faced by a driver when trying to find a free parking space in an urban environment. We proposed a cost-effective and adaptive parking system, called FPS. Unlike existing approaches, FPS assigns parking spaces to drivers in a way that optimizes the social welfare. FPS uses a parking assignment algorithm, FPA, that minimizes the total travel time among all drivers and incorporates the effect of unsubscribed drivers that compete with FPS drivers for parking spaces. Our system was tested on a real road network and compared to Greedy and Naive parking assignment algorithms. The results show significant performance improvement over the other systems. Our ongoing research focuses on a distributed model where each driver looking for parking makes her own parking choice, and each parked driver acts as an independent agent to manage part of the assignment process.

9. ACKNOWLEDGMENT

This research was supported by the National Science Foundation (NSF) under Grants No. CNS 1409523, CNS 1054754, DGE 1565478, DUE 1241976, and SHF 1617749; the National Security Agency (NSA) under Grant H98230-15-1-0274; and by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under Contract No. A8650-15-C-7521. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, NSA, DARPA, and AFRL. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notice herein.

10. REFERENCES

[1] http://sfpark.org/.

[2] https://Parkarr.com/.

[3] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin. Parking Slot Assignment Games. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 299–308, 2011.

[4] D. Ayala, O. Wolfson, B. Xu, B. DasGupta, and J. Lin. Pricing of Parking for Congestion Reduction. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 43–51, 2012.

[5] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An overview. In *Proceedings of the Third International Conference on Advances in System Simulation (SIMUL)*, pages 63–68, 2011.

[6] J. Boehle. *City-based Parking and Routing System.* PhD thesis, Erasmus University Rotterdam, 2007.

[7] R. W. Bohannon, A. W. Andrews, and M. W. Thomas. Walking Speed: Reference Values and Correlates for Older Adults. *Journal of Orthopaedic & Sports Physical Therapy*, 24(2):86–90, 1996.

[8] T. Delot, N. Cenerario, S. Ilarri, and S. Lecomte. A Cooperative Reservation Protocol for Parking Spaces in Vehicular Ad Hoc Networks. In *Proceedings of the 6th International Conference on Mobile Technology, Application and Systems*, pages 30:1–30:8, 2009.

[9] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook. Simple and Complex Activity Recognition through Smart Phones. In *Proceedings of the 8th International Conference on Intelligent Environments (IE)*, pages 214–221, June 2012.

[10] Y. Geng and C. G. Cassandras. Dynamic Resource Allocation in Urban Settings: A "Smart Parking" Approach. In *Proceedings of the 2011 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, pages 1–6, Sept 2011.

[11] D. Mackowski, Y. Bai, and Y. Ouyang. Parking Space Management via Dynamic Performance-Based Pricing. *CoRR*, abs/1501.00638, 2015.

[12] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. ParkNet: Drive-by Sensing of Road-side Parking Statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 123–136, 2010.

[13] A. Nandugudi, T. Ki, C. Nuessle, and G. Challen. PocketParker: Pocketsourcing Parking Lot Availability. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 963–973, 2014.

[14] S. Nawaz, C. Efstratiou, and C. Mascolo. ParkSense: A Smartphone Based Sensing System for On-street Parking. In *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 75–86, 2013.

[15] T. Öberg, A. Karsznia, and K. Öberg. Basic Gait Parameters: Reference Data for Normal Subjects, 10-79 Years of Age. *Journal of rehabilitation research and development*, 30:210–210, 1993.

[16] B. Panja, B. Schneider, and P. Meharia. Wirelessly Sensing Open Parking Spaces: Accounting and Management of Parking Facility. In *Proceedings of the Americas Conference on Information Systems (AMCIS)*, 2011.

[17] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee. Parking Space Detection Using Ultrasonic Sensor in Parking Assistance System. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium (IV)*, pages 1039–1044, June 2008.

[18] E. Polycarpou, L. Lambrinos, and E. Protopapadakis. Smart Parking Solutions for Urban Areas. In *Proceedings of the IEEE 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2013.

[19] D. C. Shoup. Cruising for Parking. *Transport Policy*, 13(6):479, 2006.

[20] D. C. Shoup, A. P. Association, et al. *The high cost of free parking*. Planners Press, American Planning Association Washington, DC, USA:, 2005.

[21] P. Uthaisombut. Generalization of EDF and LLF: Identifying All Optimal Online Algorithms for Minimizing Maximum Lateness. *Algorithmica*, 50(3):312–328, Jan. 2008.

[22] V. Verroios, V. Efstathiou, and A. Delis. Reaching Available Public Parking Spaces in Urban Environments Using Ad Hoc Networking. In *Proceedings of the IEEE 12th International Conference on Mobile Data Management*, pages 141–151, June 2011.

[23] O. Wolfson, B. Xu, and H. Yin. Dissemination of Spatial-temporal Information in Mobile Networks with Hotspots. In *Proceedings of the Second International Conference on Databases, Information Systems, and Peer-to-Peer Computing*, pages 185–199, 2005.

[24] B. Xu, O. Wolfson, J. Yang, L. Stenneth, P. S. Yu, and P. C. Nelson. Real-Time Street Parking Availability Estimation. In *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management*, pages 16–25, 2013.