Parking in Competitive Settings: A Gravitational Approach

Daniel Ayala, Ouri Wolfson, Bo Xu, Bhaskar DasGupta
University of Illinois at Chicago
Department of Computer Science
dayala@uic.edu, {wolfson, boxu, dasgupta}@cs.uic.edu

Jie Lin
University of Illinois at Chicago
Department of Civil and Materials Engineering
janelin@uic.edu

***Abstract*—With the proliferation of location-based services, mobile devices, and embedded wireless sensors, more and more applications are being developed to improve the efficiency of the transportation system. In particular, new applications are arising to help vehicles locate open parking slots. Nevertheless, while engaged in driving, travelers are better suited being guided to an ideal parking slot, than looking at a map and choosing which slot to go to. Then the question of how an application should choose this ideal parking slot becomes relevant.**

Vehicular parking can be viewed as vehicles (players) competing for parking slots (resources with different costs). Based on this competition, we present a game-theoretic framework to analyze parking situations. We introduce and analyze parking slot assignment games and present algorithms that choose parking slots ideally in competitive parking simulations. We also present algorithms for incomplete information contexts and show how these algorithms outperform even algorithms with complete information in some cases.

I. Introduction

Finding parking can be a major hassle for drivers in some urban environments. For example in [1], studies conducted in 11 major cities revealed that the average time to search for curbside parking was 8.1 minutes and cruising for these parking slots accounted for 30% of the traffic congestion in those cities on average. Even if the average time to find parking was smaller, it would still account for a large amount of traffic. Suppose that the average time to find parking were 3 minutes (as opposed to 8.1), each parking slot would still generate 1,825 vehicle miles traveled (VMT) per year [2]. That number would of course be multiplied by the number of parking slots in the city. For example, in a city like Chicago with over 35,000 curbside parking slots [3], the total number of VMT becomes 63 million VMT per year due to cruising while searching for parking. Furthermore, this would account for waste of over 3.1 million gallons of gasoline and over 48,000 tons of CO_2 emissions.

The advent of wireless sensors that can be embedded on parking slots has enabled the development of applications that help mobile device users find available parking slots around their locations. A prime example of this type of application is SFPark [4]. It uses sensors embedded in the streets of the city of San Francisco, that can tell if a slot is available. When a user wants to find a parking slot in some area of the city, the application shows a map with marked locations of the open parking slots in the area.

While this type of application is useful for finding the open parking slots around you, it does raise some safety concerns for travelers. The drivers have to shift their focus from the road, to the mobile device they are using. Then they have to look at the map and make a choice about which parking slot to choose from all the available slots that are shown on the map. It would be better (safer) if the app just guided the driver to an exact location where they are most likely to find an open parking slot. Then the question arises, which algorithm should the mobile app use to choose such an *ideal* parking location?

Our main concern in this work is to answer the preceding question. Parking can be viewed as a continuous query submitted by mobile devices to obtain information about spatial resources (parking slots). A mobile user wants to know which is the parking slot to visit in order to minimize various possible utilities like: distance traveled, walking distance to their destination, or monetary price of the parking slot. However, parking is also competitive in nature because after making a choice to visit a particular slot, the success in obtaining that slot will depend on if any other vehicles closer to that slot also made the same choice. This competition for resources (slots) lends itself for modeling this situation in a game-theoretic framework. We then present parking slot assignment games (PSAG) for studying competitive parking situations.

Two categories of PSAG will be considered in this work, complete and incomplete information PSAG. For the complete information PSAG, we relate the problem of finding the Nash equilibrium to the Stable Marriage problem [5]. We show the equivalency of Nash equilibria and Stable Marriage assignments for instances of PSAG.

For the incomplete information PSAG, the model that is most realistic and directly applicable to real-life application of parking slot choice, we present a gravitational approach for choosing parking. The Gravity-based Parking algorithm (GPA) is presented for this model. We evaluate the merits of the algorithm through simulation by comparing it to an algorithm that uses complete information and in which users follow their Nash equilibrium strategies. So in essence, we are comparing an algorithm that uses incomplete information against one that uses complete information. Our results show that in many instances, the GPA actually outperformed the Nash equilibrium on average in terms of driving time to park. The results also held when considering more general costs that included both driving and walking times. This type of analysis tests the value of having complete information for the parking problem. In various cases, our algorithm showed improvement of over 30% compared to the Nash equilibrium algorithm. This improvement amounts to savings of up to 930,000 gallons of gasoline and over 14,000 tons of CO_2 emissions per year in a major city like Chicago.

II. Related Work

Approaches for monitoring and sensing open parking slots have been presented recently. In this paper, we have assumed that these works exist and that vehicles can receive information about open parking slots at any time. In [6], ultrasonic sensor technology is used to determine the spatial dimensions of open parking slots. Wireless sensors are used in [7] to track open parking slots in a parking facility. These works show how one can detect open slots. In [8], detection is coupled with sharing of the parking slot information in a mobile sensor network. These mobile sensors generate a map view of parking slot availability. The value of having this parking information is tested in a P2P environment in [9] where through simulations it is shown how vehicles with access to data about open parking slota have an advantage over vehicles that don't.

Work has been performed on dissemination of reports of open parking slots [10]. In [10], a parking choice algorithm is presented that chooses parking slots based on a relevance metric that includes the age of the open parking report. Their work assumes that a driver knows the expected time a slot will remain available from now, and it knows how long it will take to travel there. In our context, it is as if a driver d knows what is the probability that another driver will get there before d. This is a strong assumption that we do not make in our work. Furthermore, the focus of [10] was on peer-to-peer (P2P) dissemination of parking reports. Wireless Ad-hoc networking is also used in [11] to search for open parking slots. They present an algorithm based on the time-varying Traveling Salesman Problem to compute a tour of the open slots in order for each vehicle to search for parking in the order of the computed tour. Like in [10], their approach depends on knowing the probability that the parking slot will still be open after some time. Furthermore, in [12], the relevance of parking reports in a Vehicular Ad-hoc Network is studied.

In [13] and [14], reservation systems for parking slots are studied. These systems attempt to circumvent the competition for parking slots by using reservations. In our work we analyze parking competition by using game theory. Indeed existing parking systems are competitive rather than reservation-based.

In [15], parking slot assignment games were introduced. To our knowledge, it is the first treatment of vehicular parking modeled as a competitive game. The PSAG was introduced but only considering user costs to be travel distances. In this work we will consider general cost functions when analyzing PSAG (travel distances, walking distances, price, etc.). The Gravity-based Parking algorithm (GPA) was also introduced in [15] and was evaluated against a greedy scheme that always chose closest slots to park. In this work we will evaluate GPA against Nash equilibrium strategies that use complete information (knowledge of locations of all other vehicles).

III. General Setup and Notation

The general setup of the parking problem is as follows:

- There are two types of *objects* as follows.
  - A set of n vehicles $V = \{v_1, v_2, \dots, v_n\}$.
  - A set of m open parking slots $S = \{s_1, s_2, \dots, s_m\}$.
- $\mathsf{dist} : (V \cup S) \times S \to \mathbb{R}$ is a distance function. It denotes the distance between a vehicle and a slot, or the distance between two slots.
- $\mathsf{cost} : V \times S \to \mathbb{R}$ is a cost function. It denotes the *cost* of a slot $s_j \in S$ to a vehicle $v_i \in V$. This cost is a general cost. It could include the distance from the vehicle to the slot, $\mathsf{dist}(v_i, s_j)$, the walking distance from s_j to v_i's destination, and/or other utilities that v_i cares about when choosing a slot.
- Each vehicle is assumed to be moving independently of all other vehicles at a fixed velocity. Without loss of generality, we assume that the speeds of all vehicles are the same[1].
- A valid *assignment* of vehicles to slots is one where each vehicle is assigned to exactly one slot. It can be defined as a function $g : V \to S$, where $g(v)$ is the assigned slot for vehicle $v \in V$.[2]
- The *cost of an assignment* g for a vehicle $v \in V$, $C_g(v)$, is defined as $\mathsf{cost}(v, g(v))$ if of all players assigned to slot $g(v)$, v is the closest to it.
  If some other vehicle assigned to $g(v)$ is closer to it than v, then v's cost based on g is $C_g(v) = \mathsf{cost}(v, g(v)) + \alpha$, where α is a *penalty* for not obtaining a parking slot.
- The *total cost of an assignment* g, C_g, is defined as:

$$C_g = \sum_{v \in V} C_g(v) \tag{1}$$

IV. Parking Slot Assignment Games

One could define a model in which a centralized authority was in charge of assigning the vehicles to slots. This authority would be looking to minimize some system-wide objectives (optimizing social welfare). In the transportation literature this is usually called a system optimal assignment. In [15], we show how this system optimal assignment can be computed in polynomial time. Even though this centralized model shows good computational properties, it is difficult to justify in real life to distributed mobile users that make their own choices. This is because optimizing social welfare may imply that some travelers will incur a greater cost for the good of others.

We then model parking as a competitive game in which individual, selfish players are competing for the available slots. Any game has three essential components: a set of *players*, a

[1] Otherwise, we simply need to rescale the distances for each vehicle in our algorithmic strategies.

[2] Based on this definition, there is a difference between where a vehicle is assigned and where a vehicle parks. If more than one vehicle is assigned to the same slot, then the closest one to it will park there. The others are left without parking. This will always happen when $n > m$.

set of possible *strategies* for the players and a payoff function (cost function) [16]. The payoff function determines what is the cost to each player based on a given *strategy profile*. If there are n players in the game then a *strategy profile* is an n-tuple in which the ith coordinate represents the strategy choice of the ith player. It basically represents the choices made by the n players.

In our case for the parking problem, we can define the *parking slot assignment game* (PSAG) as follows:

- The set of *players* in PSAG is V (the vehicles) and the set of available *strategies* to each player is S (the slots).
- The *payoffs* (costs) for each player in this game can be defined by the C_g function introduced in section III. Let $\mathcal{A} = (s_{v_1}, s_{v_2}, \ldots, s_{v_n})$ be the strategy profile chosen by the players, *i.e.* slot s_{v_i} is the chosen slot by vehicle v_i, $1 \leq i \leq n$. Let $g(v_i) = s_{v_i}$, then the cost for any player v_i will be $C_g(v_i)$.
- For this game, the penalty of not finding a parking slot, α, will be defined as a large constant quantity.

V. Nash Equilibrium for PSAG

In this section we introduce the Nash equilibrium for PSAG and establish its relationship with the Stable Marriage problem. The Nash equilibrium [17] is the standard desired strategy that is used to model the individual choices of players in a game. It defines a situation in which no player can decrease its cost by changing strategy unilaterally. The standard definition of Nash equilibrium translates to the following definition for PSAG:

Definition 1 (Nash Equilibrium for PSAG*):* Let $\mathcal{A} = (s_{v_1}, s_{v_2}, \ldots, s_{v_n})$ be a strategy profile for the PSAG. Let $\mathcal{A}_i^* = (s_{v_1}, s_{v_2}, \ldots, s_{v_{i-1}}, s^*_{v_i}, s_{v_{i+1}}, \ldots, s_{v_{n-1}}, s_{v_n})$, for $s^*_{v_i} \neq s_{v_i}$. Let g be the assignment function obtained from strategy profile $\mathcal{A}$ and g_i^* be the assignment function obtained from strategy profile $\mathcal{A}_i^*$. Then strategy profile $\mathcal{A}$ is a Nash equilibrium strategy for the players if $C_g(v_i) \leq C_{g_i^*}(v_i)$ for all i and any $s^*_{v_i} \neq s_{v_i}$.

$\mathcal{A}_i^*$ is the strategy profile obtained by only player v_i changing strategy from s_{v_i} to any $s^*_{v_i} \neq s_{v_i}$ for any $1 \leq i \leq n$. If the condition in the definition holds then it means that no player can improve by him alone deviating from the Nash equilibrium strategy. For the remainder of the paper, *equilibrium* and *Nash equilibrium* will be used interchangeably.

A. Stable Marriage Problem

The stable marriage problem has been studied in many different contexts independent of parking [5], [18]. It is a classical matching problem between two sets of objects (or agents). One of the sets is called the *men* and the other set is called the *women*. Each man has a preference order on the women and each woman has a preference order on the men.

Definition 2 (Stable Marriage [5]): An assignment of men to women is called *unstable* if there are men m and m', assigned to women w and w' respectively, but m' prefers w over w' and w prefers m' over m. A *stable matching* is an assignment of men and women that is not unstable.

B. Stability of Marriage for Parking

We can translate the stable marriage problem to the parking problem and use it to compute the Nash equilibrium for PSAG. We can say that the men are the set of vehicles (V) and the women are the set of parking slots (S). The preference order of each vehicle will naturally be determined by the cost function since the objective of each vehicle is to minimize this cost. Then, a vehicle $v \in V$ will prefer a slot $s \in S$ over another $s' \in S$ if $\mathsf{cost}(v, s) < \mathsf{cost}(v, s')$. Analogously, we will say that the preference order for the slots will be determined by the dist function. This setup leads to the following theorem:

Theorem 3: Suppose that the vehicles' preference order is determined by the cost function and the slots' preference order is determined by the dist function. Then an assignment g is a Nash equilibrium if and only if g is a stable marriage between the vehicles and slots.

Proof: Omitted for space considerations. ■

By the equivalency obtained between the Nash equilibrium for PSAG and stable assignments in PSAG, one can compute an equilibrium by finding a stable assignment between the vehicles and slots. One can use the Gale-Shapley deferred acceptance algorithm [5] to compute the equilibrium. This algorithm is an iterative procedure that runs in $O(n^2)$ time.

It can be shown that for some cases, like when $\mathsf{cost} = \mathsf{dist}$ and all distances are distinct, the equilibrium will be unique. Nevertheless, in general, by using the Gale-Shapley algorithm one could compute a vehicle-optimal or a slot-optimal assignment. A property of the vehicle-optimal assignment is that the vehicles will prefer it over any other stable assignment.

VI. Gravitational Strategies for Incomplete Information Context

A. Incomplete Information PSAG

We've shown that one can compute the Nash Equilibrium for PSAG by computing a stable marriage assignment between the vehicles and the slots. But this equilibrium is applicable only in a complete information setting. This is one where the vehicles are aware of what their payoffs will be based on their decisions and the decisions of others.

This complete information model is hard to justify in practice because of privacy concerns. Not all vehicles will be willing to share their location information at all times. Furthermore, tracking the locations of vehicles at all times, and sharing the locations of all of them with all the users of a system so that they can have up-to-the-second location data on all other potential parking competitors seems infeasible.

Then we wish to analyze PSAG in an incomplete information context. In this context, the players have no knowledge about the locations of the other players. Since they do not have complete access to the distance function, dist, then they have no way of knowing the payoff function for this game; *i.e.* given a strategy profile, none of the players have a way of knowing what its payoff will be.

In the incomplete information PSAG, players make some prior probabilistic assumptions about the locations of the other vehicles in the game and the analysis is performed based on the expectations given by the prior distributions. One can compute the expected costs based on the distribution that is used to denote the location of a vehicle. Then a player will be looking to minimize its expected cost. In this context, the analysis will compute the Nash equilibrium strategies for the players but considering expected costs. This equilibrium is analogous to the Nash equilibrium for PSAG (Def. 1) but instead of using cost given by the cost functions (C_g), it uses expected cost.

For this work, each player will assume that other players are distributed uniformly across the map. Unfortunately, computing the equilibrium for this incomplete information context is difficult in general [15]. Then heuristics are needed to compute ideal strategies for players in this more realistic model.

B. Gravity for Parking

The heuristic we want to introduce pushes vehicles towards areas where they are most likely to find a parking slot. Since all other vehicles are assumed to be distributed uniformly across space, this will increase the probability of finding a parking slot upon arrival to the area with a larger amount of available slots. Also, we want the algorithm to take into account the vehicle's location and its proximity to the surrounding slots. In [15], we proposed the Gravity-based Parking Algorithm (GPA), which encompasses these desired properties.

In the GPA, slots are said to have a gravitational pull on the vehicles. At any point in time, each slot has a gravitational force on the vehicle that will depend on the distance from the vehicle (magnitude) and location of the slot (direction). So then for each slot, a force vector is generated around the vehicle. Then, all of these vectors are added and the vehicle moves in the direction of the resultant vector (total gravitational force) for a specified time step. Then the process is repeated at the beginning of each time interval.

The classical formula for gravitational force is $F = \frac{Gm_1m_2}{d^2}$ where G is the gravitational constant, m_1 and m_2 are the masses of the respective objects and d is the distance between the objects. But for our purposes we can assume that the masses of the objects are constant. We want to compute the vector that represents total gravitational force generated by all the available slots to a vehicle and use the direction of that vector to move the vehicle in that direction. Then we consider a more simplified formula for gravitational force, since all the masses are constant, represented by:

$$F(v, s) = 1/\mathsf{dist}(v, s)^2 \tag{2}$$

$F(v, s)$ is the gravitational force generated by slot s towards vehicle v. To consider general costs, this formula can be generalized to:

$$F(v, s) = 1/\mathsf{cost}(v, s)^2 \tag{3}$$

With formula (3), one will compute gravitational pull by considering the general cost as the distance between the vehicle and the slot.

C. Gravity-based Parking Algorithm (GPA)

Let z denote the velocity of each vehicle (in units/s), which is constant for all vehicles. Each time step for the algorithm will be 1 second. Each vehicle v will perform the following steps in order to move one time-step at a time towards a parking slot:

- Let S' be the set of available slots (updated at every time step). Then for each $s \in S'$ generate vector of magnitude $F(v, s)$ that starts at v's location in direction of s.
- Add the computed force vectors and the result will be the total gravitational force generated by all the available slots on v.
- Move z units (velocity) in the direction given by the total force vector. If the closest slot to v is at a distance less than z then move straight to the closest slot.

These steps define the proposed heuristic for vehicles to use in the incomplete information PSAG. The intuition behind the algorithm is that a vehicle is better served moving towards areas of higher density of parking slots when the force to closer slots (determined by distance to them) is not strong enough.

VII. Comparing Gravitational Strategies to Complete Information Nash Equilibrium

In this section we will evaluate the GPA against the Nash equilibrium with complete information. This will be done to assess if one is better off in this parking application with the locations of the other vehicles as opposed to not knowing the locations and using the GPA to find parking. This evaluation will be done through simulation.

A. Simulation Environment

The simulation tests the GPA with varying number of values of n and m for the 2-dimensional Euclidean space in the unit square. The unit square is first partitioned into 16 equal-sized square regions. A random permutation of the regions is generated (uniform distribution) and is used as the ranking of the popularity of each region for available slots. Then the number of parking slots per region is determined by using the Zipf distribution based on the ranking of the region and the skew parameter used for the Zipf distribution. Then for each of the m slots a Zipf number between 1 and 16 is generated to determine its region, then inside that region its position is determined using the uniform distribution. The n vehicles' positions are generated using the uniform distribution on the unit square. Destinations for each vehicle are generated uniformly around each vehicle at a distance no larger than 0.2 units. These destinations will be used to compute walking distances from the slots where they eventually park. The cost metric to be used for testing will be one that includes driving time and walking time.

After generating the vehicles and slots, the algorithms are tested. The GPA is tested against the Nash equilibrium algorithm with complete information. In this setting, the algorithm for computing the equilibrium strategy that is used is the Gale-Shapley deferred acceptance algorithm [5], [18]. This method guarantees that m players will find parking based on the current configuration of the slots. For purposes of the simulation, the remaining $n - m$ players will not move locations since they know that they will not park in any of the currently available slots. For the GPA, vehicles move in a step-by-step fashion as dictated by the steps delineated in sec. VI-C. The simulation is run a second at a time so that vehicles can recompute their total force vectors at each second.

When a vehicle reaches an open parking slot, the time it took for it to find that slot is saved. Also, the walking distance from that slot to the vehicle's destination is saved. Then a new slot is generated on a randomly chosen (Zipf distribution) region. Also a new vehicle is generated at a random location (uniform distribution). The simulation run stops when a given time horizon of 3,600 seconds is surpassed.

The parameters of the simulation are: the number of vehicles (n), number of slots (m), and the regional skew of the Zipf distribution (k). The values that were tested for each parameter are detailed in table I. For each configuration of the parameters, 1000 different simulation runs were tested.

Parameter	Symbol	Range
Vehicles	n	{40}
Slots	m	{20,30,40}
Zipf Skew	k	{0, 0.5, 1, 1.5, 2, 2.5, 3}

TABLE I
PARAMETERS TESTED ON SIMULATION

B. Simulation Results

1) Driving Cost results: Figure 1 shows some of the results of the simulation that was performed (for $n = 40$ and $m = 20$). These results only take into account the driving time to find a slot. They show that in cases where the regional skew of the distribution of the slots is small, having the complete information and computing the equilibrium strategy is beneficial on the average.

However, for all cases with skew of 1 or higher, the GPA actually outperformed the complete information equilibrium on average. This means that acting as if you do not know the locations of the other vehicles and using GPA to guide you is actually better for those cases. Furthermore, the difference in average time to park was more significant in the best case for GPA than in the worst case. That is, the best case for GPA was when the regional skew was 3 and the difference in average time to park was around 20 seconds. But, GPA's worst case was when the regional skew was 0 (uniform), and the difference was 8 seconds. So in the extreme cases, the improvement for GPA was larger than for the equilibrium.

Figure 2 shows the results for all test cases of $n = 40$. It shows the percent improvement of the GPA algorithm over the complete information Nash equilibrium algorithm. This figure shows that as the parking situation becomes more contentious, *i.e.* the ratio of vehicles to slots is larger, the results of the GPA improve. When the number of vehicles and slots are equal, the improvements if any where marginal. This is to be expected since in these situations, there is a slot for everyone and since they have all the information available to them they can choose one that is guaranteed to be there for them. Still,

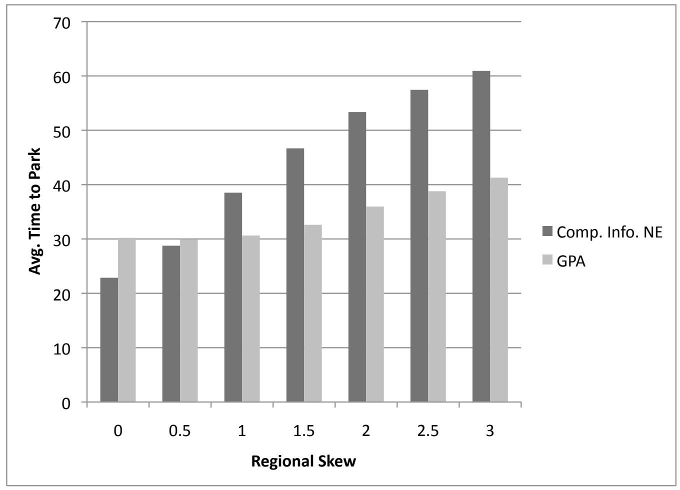

Fig. 1. Comparison between GPA and Compl. Info Nash Eq.

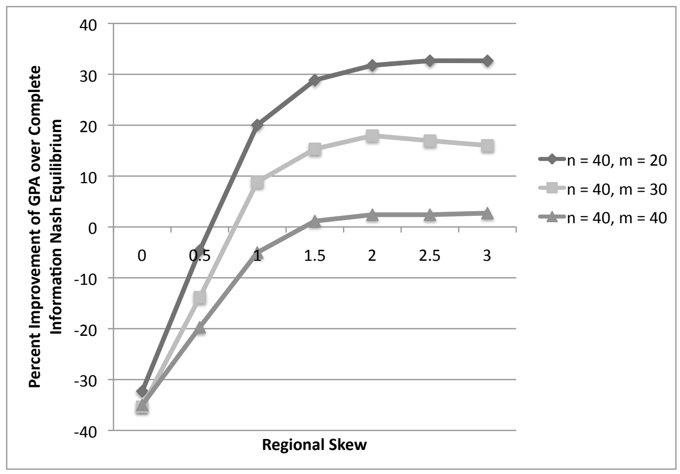

Fig. 2. Percent Improvement of GPA over Compl. Info Nash Eq.

in less contentious situations (like when $m = 30$ and $m = 40$), GPA showed good improvements over the equilibrium solution. Especially when the regional skew of the locations of the slots was high.

The improvements of the GPA algorithm, in terms of driving time, in some cases were as large as 30% improvement. A 30% improvement for the time taken to find a parking slot (proportional to the tested distance traveled because of constant velocity), would reduce vehicle miles traveled for a city like Chicago by 18.9 million VMT (according to analysis presented in sec. I). This gives a reduction of 930,000 gallons of gasoline and of over 14,000 tons of CO_2 emissions per year.

The GPA shows improvements over the equilibrium algorithm even though it doesn't have the luxury of complete information. It would seem unexpected that the GPA could outperform a complete information algorithm. This phenomenon can be explained by the fact that the Nash equilibrium does not make any assumptions about future events and about where will new slots most likely appear. These new slots arise in

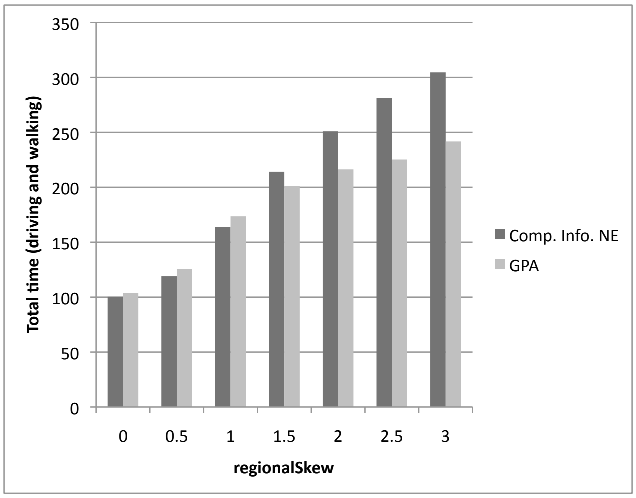

Fig. 3. Total Time Results - Driving and Walking Time ($n = 40$, $m = 20$)

the simulation to keep the ratio of vehicles to slots fixed. Moreover, new slots also arise in the real world in usually unpredictable locations that are distributed according to some spatial probability distribution like the one we use in this paper. The results of the simulation depend also on where future slots will appear, and the GPA does a better job of predicting future events on cases where the regional skew is high.

2) General Costs - Driving and Walking Time: Figure 3, shows results for all test cases of $n = 40$ and $m = 20$ when considering the total time (in seconds). Here total time means the driving time plus the walking time. In our simulation we saved the distances from the obtained parking slots to their destinations for each parked vehicle. We assume that a vehicle drives around an urban area at 20mph $\approx$ 30 ft/s. We also assume that the average person walks at a pace of 5 ft/s. Then, we compute the walking time by using a conversion factor of $30/5 = 6$. Figure 3 shows how for all cases studied, the GPA again has favorable results compared to the Nash equilibrium in complete information. For this more general cost, the GPA is comparable in performance to the Nash equilibrium even for cases with small skew. So then the results are improved when considering this new walking cost for the GPA since the results for low regional skews were not favorable when considering only driving distances.

VIII. Conclusion

In this paper our main goal was to analyze vehicular parking. We presented two models that can be used to study the parking problem in a game-theoretic framework. For the complete information model, in which vehicles are aware of the location and cost information of other players, we presented an algorithm for computing the Nash equilibrium for parking slot assignment games (PSAG). We established the relationship between the parking problem and the stable marriage problem. We also showed that the Nash equilibrium was actually equivalent to a stable marriage between vehicles and slots. For the incomplete information model, vehicles are not aware of the locations of the other mobile users that are also looking for parking. For this model we presented the Gravity-based Parking Algorithm (GPA). The merits of the GPA were tested using simulations. The simulations showed that for most competitive situations, using the GPA actually outperformed the Nash equilibrium algorithm that is used with complete information. This means that an algorithm that didn't have the luxury of using complete information actually outperformed one that did. The results also held when considering more general costs that included both driving and walking times.

Acknowledgment

This work was supported in part by the NSF IGERT program under grant DGE-0549489.

References

[1] D. Shoup, *The High Cost of Free Parking*. Chicago, IL: American Planning Association, 2005.

[2] ——, "Cruising for parking," *Transport Policy*, vol. 13, pp. 479–486, 2006.

[3] *Pricing the Curb: How San Francisco, Chicago and Washington D.C. are reducing traffic with innovative curbside parking policy*, Transportation Alternatives (www.transalt.org), New York, NY, July 2008.

[4] "http://sfpark.org/."

[5] D. Gale and L. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[6] W. Park, B. Kim, D. Seo, D. Kim, and K. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *IEEE Intelligent Vehicles Symposium*, 2008.

[7] B. Panja, B. Schneider, and P. Meharia, "Wirelessly sensing open parking spaces: Accounting and management of parking facility," in *AMCIS 2011 Proceedings*, 2011.

[8] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrashekharan, W. Xue, M. Gruteser, and W. Trappe, "Parknet: Drive-by sensing of road-side parking statistics," in *MobiSys*, San Francisco, CA, June 2010.

[9] E. Kokolaki, M. Karaliopoulos, and I. Stavrakakis, "Value of information exposed: wireless networking solutions to the parking search problem," in *Proc. of 8th Int. Conf. on Wireless On-Demand Network Systems and Services (WONS)*, Bardonecchia, Italy, January 2011.

[10] O. Wolfson, B. Xu, and H. Yin, "Dissemination of spatio-temporal information in mobile networks with hotspots," in *Proc. of the 2nd Intl. Workshop on Databases, Information Systems, and Peer-to-Peer Computing*, Toronto, Canada, 2004, pp. 185–199.

[11] V. Verroios, V. Efstathiou, and A. Delis, "Reaching available public parking spaces in urban environments using ad-hoc networking," in *IEEE Intl. Conf. on Mobile Data Management (MDM)*, 2011.

[12] P. Szczurek, B. Xu, J. Lin, and O. Wolfson, "Spatio-temporal information ranking in vanet applications," *Intl. J. of Next-Generation Computing*, vol. 1, no. 1, July 2010.

[13] J. Boehlé, L. Rothkrantz, and M. van Wezel, "Cbprs: A city based parking and routing system," *ERIM Report Series Reference No. ERS-2008-029-LIS*, May 2008.

[14] T. Delot, N. Cenerario, S. Ilarri, and S. Lecomte, "A cooperative reservation protocol for parking spaces in vehicular ad hoc networks," in *Proc. of the 6th Int. Conference on Mobile Techonology, Application and Systems*, Nice, France, September 2009.

[15] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin, "Parking slot assignment games," in *Proc. of the 19th Intl. Conf. on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2011)*, Chicago, IL, November 2011.

[16] E. Rasmusen, *Games and Information*, 4th ed. Blackwell Publishing, 2006.

[17] J. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[18] A. Roth and M. A. O. Sotomayor, *Two-sided Matching: A Study in Game-theoretic Modeling and Analysis*, ser. Econometric Society monographs. Cambridge, UK: Cambridge University Press, 1990.