A Probabilistic Approach to Parking
Benefits of Routing Instead of Spotting

Gabor Feher, Balazs Andras Lajtha and Akos Lovasz
Dept. of Telecommunication and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary

Keywords: Smart City, Parking, Parking Route, Parking Probability.

Abstract: Urban parking is an important issue in all modern countries. Technological advances, with in-car sensors and always connected smartphones have already paved the way to an ICT solution for this problem. However, every attempt - including that of such a big companies, as Google - has failed to provide a suitable solution. So far, the appeared solutions were centered around the notion of free parking spots. This approach does not take into account the dynamics of the traffic and the drivers outside of the system. Here we propose a fundamentally different approach based on parking probabilities and parking routes. Our solution can truly reduce the time, resource and environmental damage wasted on parking place hunting, while keeping the operational costs low and the users satisfied.

1 INTRODUCTION

In cities of the developed countries vehicle ownership per household is close to and may reach 1, and decreases only slowly (Goodwin, 2012). In the big cities, the cars cause urbanization problems, most significantly traffic jams and parking. Diseases connected to air pollution and the stress caused by the transportation have severe impact on all other area of human life. While shifting to the environment friendly, green transportation, traffic jams and related issues seem to persist or even worsen due to speed limits and reduction of road surface in favor of other uses like urban vegetation or bicycle lanes. However, the advances of the Information and Communication Technology (ICT) in smart cities can create a cure for the biggest problems.

One of the big problems of individual transportation is the problem of parking in frequented areas. In those places, a great share of the traffic is caused by drivers seeking a parking spot. A survey from 2005 (Arnott et al., 2005) claims that in big cities of the U.S., in every moment 30 percent of the drivers are looking for parking spots. Each single car spends 7.8 minutes to find a parking place in average. According to a more recent survey (Dohler et al., 2011) the average parking time is 15 minutes in cities over 1 million population in Europe. The average parking time is 13 minutes in Madrid, 15 minutes in Barcelona and 26 minutes in Granada. This also means that these cars searching for parking place, produce 2300 tons of CO_2 per day.

2 RELATED WORKS

In the topic of parking in smart cities, there are already numerous research papers and application published. We had a focus on Parking Guidance and Information (PGI) systems. These systems aim to solve the parking problem by providing information to users about the amount of available parking places. PGI systems are used for both parking lots and on-street parking, while there are some significant differences, the same basic concepts apply.

Every PGI system handles information in the form of available parking spots. The information about parking spots can be aggregated assigning a positive integer to a parking lot or a street segment, or it can be a flag associated to each individual parking spot, indicating it's current state. Parking spots are well defined in parking lots and in some cities where individual parking meters or sensors are deployed for each spot. However in many European cities parking spots are organic, only the parking style (parallel, perpendicular or angled) is regulated.

Feher G., Andras Lajtha B. and Lovasz A..
A Probabilistic Approach to Parking - Benefits of Routing Instead of Spotting.
DOI: 10.5220/0005495400950100
In *Proceedings of the 4th International Conference on Smart Cities and Green ICT Systems* (SMARTGREENS-2015), pages 95-100
ISBN: 978-989-758-105-2

2.1 Information Collection

There are two ways to collect information about parking spot availability. Either through infrastructure, or through crowd sourcing.

Existing infrastructure can be used such as parking meters (Caliskan et al., 2007) (Nawaz et al., 2013) However using parking meters does not provide accurate results, as in the case of the prepaid meters, usually the meters are overpaid and indicate occupancy even when the car has left. On the other hand a mobile phone based parking payment system can signal the start and the end of the parking really accurately, but lacks information about the position of the parking car. Furthermore local residents may be entitled to free parking, hence their parking events are not registered by any parking meters.

More accurate are the dedicated devices, like sensors built into the road (Kessler, 2011), (Dohler et al., 2011), or ultrasonic sensors in parking lots. These devices provide exact measurements at a high installation and maintenance cost, an investment that local authorities often choose not to make.

The other approach is to leverage user's resources and crowdsource the task of free spot detection. Crowdsourcing can either rely on user input, or smartphone sensors (Chen et al., 2013). When drivers and pedestrians are tasked to report parking spots, user motivation becomes an important aspect. Compensating users for their input may lead to falsely reported spots. On the other hand not providing external incentives results in freeriding that will reduce the performance of the system. (Lan and Wang, 2013)

Cameras and radars can detect parking spots passing by them, but require special mounts or external equipment. They also have a high error rate. Activity sensors are more accurate, but can only detect parking actions performed by the participating users (Lan and Shih, 2014). Many solutions are based on different types of sensors from gyroscopes through GPS positioning to WiFi signal strength analysis (Nawaz et al., 2013), but all suffer from the same problem: when application penetration is low, data about free parking spaces will be sparse, and usability will be limited.

2.2 Information Processing

Information about an individual parking spot's availability has a short expiration date. In frequented areas parking spots do not last a minute, systems can only work if future openings are predicted. The only accurate way of predicting the opening of a parking spot so far is through tracking the driver (Lan and Shih, 2014). When parking spot information is aggregated, predictions can be made with more certainty from historical measurements.

2.3 Information Dissemination

Information can be displayed to the users publicly or personally. Moreover, personalized information can be free or for sale.

In traditional Parking Information Systems (PIS) roadside boards display the number of free spaces in the parking lot or area. These systems have been extended with mobile applications that deliver the same information to every user.

Many research work have the conclusion that PIS are not effective, moreover in areas where there is a shortage in parking spots, they cause severe traffic jams, when drivers try to get to the last places. This realization focused research efforts on matching users to parking spots, displaying the information of only a single parking spot to each user. Wang, using simulation (Wang and He, 2011) proves that the time needed to park is shorter in the case of reserved parking than it is with PGI systems.

Besides the great amount of research work there are several smart phone applications aiming to solve the parking problem. Most of these applications failed and are already discontinued. Open Spot form Google (Kincaid, 2010) is an often mentioned example. While supported by Google, Open Spot ended up closing because the users were not collaborative enough and did not help others with signaling of the free parking spots. The TakeMySpot application followed the same path and suffered the same fate.

Despite the failures, newer and newer applications arise trying to establish themselves in the market. Many have simpler functionality, just displaying prices, zones, parking lots and not providing guidance or real time information. Such applications are Best-Parking or Parkopedia, covering many cities worldwide.

3 THE ParkingRoutes SOLUTION

Investigating the recent works in the field shows that the current solutions are all based on the concept of acquiring information of open spots. This information is than either displayed publicly, or delivered to the most suitable or highest bidding user. These works address many aspects of the problem, starting from the sensing of free spots to handling freeriding and predicting the effect of low application penetration. Even the definition of a parking spot raises research questions.

But this approach has many drawbacks. The collection of free and busy information about given parking spot is difficult or costly. The most reliable sensor based solutions requires large investments, while the maintenance could be expensive as well.

In crowded areas in rush hour parking spots open and get taken literally in seconds. A system signaling free parking spots is useless: the driver cannot get to the spot in time and ends up driving more than she would have without the application.

In highly frequented areas, aggregated parking information will be misleading. The amount of available parking spots will oscillate between zero and one, and the drivers would be discouraged, however he could have had good chance to take a freshly opened spot.

Instead of focusing on the quantity of discrete parking spots, we base our solution on a probabilistic nature of parking. We came up with an idea, to use parking routes. This novel approach creates a working solution without the previously identified drawbacks.

3.1 Parking Probabilities

Instead of focusing on the state of individual parking spot, we introduce the notion of parking probability. We assign parking probability to street segments: a part of the street delimited by two intersections. Segments have directions, meaning that parking probability might be different on the same street for cars traveling in the opposite direction. Parking probability is metric that indicates the probability of being able to find a parking spot on that segment. This metric is much closer to how we perceive parking availability, than a list of free parking spots.

The benefit of using parking probability is that while probability varies in time, it shows periodicity as opposed to the availability of a single parking spot, which is stochastic. Parking probability can be forecasted based on historical data supplemented by passive measurements. We will detail our proposed algorithm for determining parking probability after introducing our implemented service.

3.2 Collecting Parking Probability

As every other solution, ours is also based on the sensing of parking activity. While our implementation is based on crowdsensing, it can utilize the data from deployed sensory networks too.

3.3 Parking Route

We observed the way drivers search for parking spaces and conducted personal interviews on the subject to find patterns. People choose different strategies for different scenarios. When going home, the walking distance is very important: they are willing to trade a couple of minutes in the evening to have the car nearby when running late in the morning. They end up circling the block for half an hour. Should they have known it in advance they would have chosen differently, and still they perform the same search each day. When going to a regular event (play sports, go to work) people devise a parking plan. They scout areas that they expect to be less crowded, maybe further from the target, but with higher chance of an available parking spot. When people go to a target in an unknown area, they usually pass by their destination, and start looking for a parking spot afterwards. But instead of making tight loops as in the first scenario, they tend to map the area, widening the search until a suitable spot is found. Parking strategies involve a utility function that trades time spent cruising for a parking space for walking distance from the parking spot to the destination and some knowledge about parking availability.

While showing the parking probability might help the user to make better navigating decisions, evaluating the utility function for different possible routes is tricky, especially when users have to deal with probabilities. Our solution not only provides the map of estimated parking probabilities, but also advises a route for parking.

For each parking segment s, a walking cost c_s is calculated to the target. Than based on parking probabilities q_s, the location of the user and her target, our system recommends a parking route. A parking route is a continuous path p composed of parking segments each with an indication for parking or for travel. Travel is indicating a segment that the user should pass in order to reach areas more suitable for parking. When driving through a segment labeled for parking the user has to take advantage of the first parking opportunity. Only segments in $B_{walking}$ vicinity of the target can be marked for parking: $(c_s < B_{walking}$.

$$p = (s_1, s_2, \ldots, s_{l(p)}) \quad (1)$$

Each parking route guaranties that the probability of finding a parking spot on the route is greater than $1 - \varepsilon$, and parking route's total length is below $B_{driving}$.

$$\sum_{s\in p} h_s \leq B_{driving} \tag{2}$$

$$\prod_{s\in p} (1-q_s) \leq \varepsilon \tag{3}$$

It is important to note that the parking route is not the shortest path to a good parking space or area. While strange at first, it is common that a parking route does not reach the user's destination.

There might be no parking route for the given target and location, or there might be many. When multiple parking routes are present, they are ranked based on the cost function $w(p)$ provided by the user. For the sake of simplicity, we used the linear combination of walking distance and driving distance, with the user preference, a being the coefficient for walking.

$$w(p) = \sum_{i=1}^{l(p)} q_{s_i} \prod_{j=1}^{i-1} \left(1-q_{s_j}\right) \left(c_{s_i} + a \sum_{j=1}^{i} h_{s_j}\right) \tag{4}$$

When searching for the best route, we are minimizing this cost function.

$$\min_{p\,\in\,\mathcal{P}} \quad w(p) \tag{5}$$

When searching for the best parking route, we assume that we know the actual parking probability for every street segment of the target area.

The search for suitable routes consists of two phases. In the first phase maximal routes are enumerated. A maximal route is a route that's length is $B_{driving}$ and the parking probability along the route is more than $1-\varepsilon$. A maximal route can contain the same segments several times. To find all maximal routes, we use a breadth first search.

When a route is found a second search is performed to find the segments marked for parking and for travel. Each combination is considered and the parking probability is evaluated. If the parking probability for the combination is above $1-\varepsilon$, the cost function is evaluated.

It is easy to see that even the number of possible routes increases exponentially with $B_{driving}$, with an exponent between two and three depending on the topology of the given area, and the parking / travel segment determination has is exponential in length too.

As many path had similar cost, it is not crucial to find the optimal solution. For the second problem: finding the parking and traveling segments of a given route a greedy algorithm produced good results. Adding the segments in order of their evaluated cost until the desired overall probability is reached proved to a good approximation.

To reduce the number of routes to evaluate saving memory and runtime, we implemented a naive heuristic to restrict the search field. We assigned weights to each segment, depending on their distance to the destination and parking probability. The longer the route was the less likely the algorithm chose a segment with small weight. This reduced the exponent, keeping the problem space smaller.

3.4 Estimating Parking Probability

The practical interpretation of parking probability is the ratio of successful parking attempts on a segment, divided by the number of vehicles traveling through that segment with the intent of finding a parking spot. Hence to determine the parking probability on a given segment at a given time, these two numbers have to be measured or estimated.

When we propose a parking route to a driver, it is expected that she follows the route and attempts to park on the suggested segments. So a driver following the parking route can be considered a driver looking for a parking spot. If the driver parks on a segment, she will increase the number successful parking events. Hence we can measure parking probability by observing our users. Higher the application's penetration, the higher the accuracy of this measurement.

The paradox of this approach is that not the opening but the taking of a parking spot will increase parking probability on the segment.

It is also important to consider the temporal properties of parking probability. In different zones - residential, industrial, commercial - we observed different fluctuations of the probability, ranging from multiple vacant spots to full load with several cars circling for parking. On the other hand parking probability seems to be a periodic function, with daily and weekly cycles. Hence it is often better to estimate parking probability based on historical measurements than on recent ones. To handle this phenomenon we divided time to 15 minute long segments. We register measurements for these segments.

As the periodic nature for each segment may be different, the weight in the prediction of the recent measurement and the daily, weekly, monthly and yearly periods should be determined by examining the correlation between measured values. Also as current measurements accumulate, the effect of past measurements should be decreased.

3.5 Implementation

Our implementation consists of a database containing the parking probabilities, an event database, a predic-

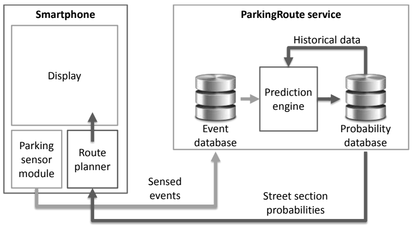

Figure 1: ParkingRoutes architecture.

tion service and a mobile client. Their connections are shown on Figure 1.

In order to get the required topology of the cities, we process the Open Street Map topology, and derive an oriented graph consisting of segments.

The mobile client is responsible for capturing parking events. Parking data is collected in the event database, and periodically processed by the prediction service to update the probability database. When the user wants to park, the mobile client queries the probability database for actual information. Than the route search is done on the mobile device, and the result is presented to the user.

Screenshots with navigations can be seen on Figure 2. The red car icon shows the actual position of the car, the orange target icon is the desired parking location and the dashed line is the parking route. The dashed line can have different sections with different colors. The dark blue section is a shortest route navigation to the parking zone. We do not plan any parking on this route. At the end of this section, the car enters to the area where the smartphone made the parking route from the entry point to the destination. It is also possible, that parking is not recommended on some sections of the parking route, as there are better sections for the stop. These street sections are colored to light blue.

Besides displaying the parking route, the parking probabilities for the street sections around the desired parking location are also displayed. The color key: green, yellow, orange and red shows the parking probability for the given section. Street sections in green color have high parking probabilities, while the red color means hardly any chance for parking. This gives the user the ability to evaluate the suggested route and in turn increase trust in the system.

The parking events, when the car is stopped at a place or when the car leaves the parking spot is signaled to the ParkingRoutes server together with the GPS position of the spot. On the server side the parking probabilities are adjusted based on these events.

3.6 Feasibility Studies

Since all the parking route computations are running on the smartphone, we investigated the resource demand of the routing algorithms, whether it is suitable for smartphones or not.

We made test in two different scenarios. The *downtown scenario* is a scenario where the streets are full of parking cars, and the parking probability is very low. In contrast, the other tested scenario is the *suburb scenario*, which has plenty of spaces on the streets, so the parking probability is generally high.

In the two scenarios the radius of the parking search area was set to 1000 meter, the parking route calculation worked with those street sections that were within this area. The maximum parking route length was set to 1000 meter.

Figure 2 displays the two scenarios, and the selected parking locations. The position of the car and the position of the desired parking place were fixed during the measurements.

Figure 2: The inner city and the suburb test scenarios.

We measured two metrics. The memory consumption and the running time of the algorithm. For the test we used a mid range Nexus 4 device with Android 4.1 operating system. The memory consumption was calculated by the algorithm itself. When the parking route search algorithm initiated a new recursive search from the given state, we allocated a memory block for that. The metric counted the maximum number of these blocks, which existed at the same time, and not the total number of allocations. With the runtime metric we measured the total execution time of the parking route planning algorithm, excluding the street section data downloading.

In order to be able to compare the results to other values, we measured a *shortest path* algorithm as well. This algorithm creates the shortest route from the car position to the target. This is not a parking route.

As neither the car and target positions, nor the street section probability database was changed during the tests, therefore subsequent runs on the given scenario produced always the same results.

Figure 3 displays the measurement results for the downtown and the suburb scenario. As it can be seen the optimal solution is slow and resource demanding compared to the other two algorithms. However, the worst case 273 ms running time and the 70k allocated memory blocks still makes this algorithm a feasible choice for the users. The simple heuristic algorithm created 62 memory blocks and ended within 50 ms in the challenging downtown scenario. As it was expected, the algorithms are faster and require less memory allocations in the suburb area, as the parking possibility is high enough on shorter routes. The optimal algorithm runs for 46 ms, and allocates 5537 memory blocks, while the simple heuristic algorithm takes 17 ms and 47 memory blocks.

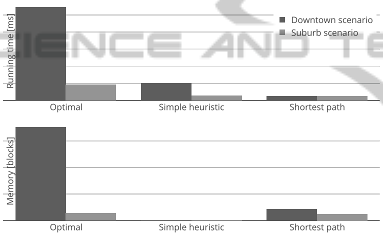

Figure 3: Resource allocation at the downtown and suburb scenario.

From the measurements we can see, that parking route searching in a reasonable sized areas, with a reasonable sized parking routes is feasible even with the slowest and most resource hungry *optimal* algorithm. The search is longer in areas where the parking possibility is low. Using the simple heuristic approach, the search is faster and requires less memory. For old devices with limited CPU and memory capabilities, this searching algorithm suits better than the optimal one.

4 CONCLUSION

The dawn of the smartphones brought new opportunities to solving the parking problem. Many proposed applications failed in early stages, and til this day there is no working solution for urban parking. The parking spot based approach taken by others cannot cope with low usage, as drivers outside of the system render the sparse available information obsolete by the time it is advertised.

We took a new approach that is not based on parking spot availability, but rather on the probability of parking on a given street segment. Parking probability is fundamentally different from individual parking spots, and as such we had to rethink and redesign every step of the parking guidance, from the definition of the information through data collection and data processing to the way information is presented to the user.

We answered all the design questions and implemented a system that demonstrates the feasibility of our idea. We run simulations to validate our solution. The initial results look promising, however more thorough traffic simulations are needed.

REFERENCES

Arnott, R., Rave, T., and Schb, R. (2005). *Alleviating Urban Traffic Congestion*, volume 1 of *MIT Press Books*. The MIT Press.

Caliskan, M., Barthels, A., Scheuermann, B., and Mauve, M. (2007). Predicting parking lot occupancy in vehicular ad hoc networks. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, page 277281.

Chen, X., Santos-Neto, E., and Ripeanu, M. (2013). Smart parking by the COIN. In *Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '13, page 109114, New York, NY, USA. ACM.

Dohler, M., Vilajosana, I., Vilajosana, X., and LLosa, J. (2011). Smart cities: An action plan. In *Proceedings of Barcelona Smart Cities Congress*.

Goodwin, P. (2012). Peak travel, peak car and the future of mobility.

Kessler, S. (2011). *How Smarter Parking Technology Will Reduce Traffic Congestion*.

Kincaid, J. (2010). Googles open spot makes parking a breeze, assuming everyone turns into a good samaritan.

Lan, K.-C. and Shih, W.-Y. (2014). An intelligent driver location system for smart parking. *Expert Systems with Applications*, 41(5):2443–2456.

Lan, K. C. and Wang, H. Y. (2013). On providing incentives to collect road traffic information. In *International wireless communications & mobile computing conference (IWCMC13)*.

Nawaz, S., Efstratiou, C., and Mascolo, C. (2013). ParkSense: a smartphone based sensing system for on-street parking. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, page 7586.

Wang, H. and He, W. (2011). A reservation-based smart parking system. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, page 690695.