A Simple Model of Cruising for Garage Parking

Oliver Ullrich[1], Benjamin Brandt[1], Daniel Lückerath[2], Naphtali Rishe[1]

[1]School of Computing and Information Sciences, Florida International University,
ECS 243C, 11200 SW 8th St, Miami FL-33199

[2]Institut für Informatik, Universität zu Köln, Weyertal 121, 50931 Köln, Germany

oullrich@cs.fiu.edu

Parking guidance systems have to be calibrated and evaluated before their application in the field. One step to accomplish this is to evaluate them in a simulated environment.

This paper presents a simple model of cruising for parking in a parking garage with focus on individual drivers' behavior. While the described model itself is too basic to completely depict a driver's decision making process, it is a step towards a more comprehensive representation. In the process of developing the model it became clear which pieces are still missing from the puzzle, and what data has to be collected to successfully represent a parking garage's operational day.

After a short introduction this paper shares some background on garage parking and related work. A simple garage parking model is presented, followed by some thoughts on necessary steps to expand and refine the model. The paper concludes with a summary of the lessons learned and an outlook on further research.

1 Introduction

With parking garages containing 2,000 or more individual spaces (see figure 1), computer based systems which provide guidance and recommendations to find available parking in these major structures are significantly beneficial to users, and also improve resource utilization for parking providers. Such parking guidance algorithms have to be calibrated and evaluated before their application in the field. One step to accomplish this is to evaluate them in a simulated environment.

This paper presents a simple simulation model of cruising for garage parking, which will, once extended and rigorously validated, serve as a virtual testbed for calibrating and evaluating garage parking guidance algorithms. The model applies a combination of two simulation paradigms: while the model's basic mechanics, e.g. the arrival of cars, is modeled in an event-based (see [4] and [6]) fashion, the agent-based paradigm (see [14]) is utilized for modeling the drivers' decision making.

After this introduction, the paper continues with sharing some background on garage parking and related research (see section 2); then a simple model of cruising for garage parking is presented (see section 3). Following on to this, the paper describes the future research steps necessary (see section 4) and closes with a short summary (see section 5).

2 Background

2.1 Garage parking

The term garage parking refers to the process of entering a building at least partially designated for car parking, finding and navigating to an available parking spot, leaving the car at that spot for a while, and then de-park by finding the shortest or most convenient path from the parking spot to a vehicular exit. As the intended application for the developed model is to test recommendation algorithms which are concerned with reducing the time spent cruising for available parking, the last part of the process, de-parking, is beyond the scope of this paper and will not be discussed further. The described buildings are often referred to as parking garages, but also as multistorey car parks, parkades, or parking structures.

Garage parking, together with parking lot parking, is often described by the more general term off-street parking. This contrasts with on-street parking with its diverse modes: parallel parking, angular parking, perpendicular parking.

The parking garage usually consists of a number of connected levels, which are themselves composed of a number of areas. Each area contains of a number of parking spots fit for individual cars. The readers will know this decomposition from their own experience: "I parked my car on a spot on level 3, in area C."

Vehicular access to the parking garage is granted, often at the ground floor, by entry and exit lanes, which are usually unidirectional. Pedestrians access the garage via elevators or stairways, or on the ground floor by doorways. Pedestrian access ways are usually bi-directional.

2.2 Related research

Corresponding to its importance in planning and design of public spaces, on-street parking has seen a lot of research attention, both in general modeling (see e.g. [2], [3], [13], [18], [19], [22], [23], [24]) and in simulation modeling (see [5], [7], [9], [8], [15], [16]). Most of the more recent simulation models being at least partially agent-based (see [5], [7], [9], [15]). Building upon this, Dieussaert et al. (see [7]) and Horni et al. (see [9]) combine agent-based modeling with the cellular automata paradigm, while Gallo et al. (see [8]) construct a multi-layer network supply model. Some authors (see [7], [15]) utilize the described models to evaluate pricing and other policy considerations, while others (see [5], [9], [8], [16]) apply them to analyze technical methods to reduce cruising time and thereby traffic in general.

Figure 1: Parking garage with approx. 2,000 parking spaces

Only a few models (see [1], [5], [7], [22]) consider off-street parking: Asakura and Kashiwadani (see [1]) apply a model to examine the effect of different types of on-street and off-street parking availability information on overall system performance, but do not examine the drivers' behavior inside of individual parking lots. Benenson et al. (see [5] and also [13]) develop a spatially explicit model of parking search and choice, with simulated drivers cruising through an artificial or real-life city center model, giving them both on-street and off-street parking options. Dieussaert et al. (see [7]) also are interested in the traffic patterns generated by cruising for parking. They model on-street parking as well as parking lots and garages, but consider parking lots and garages as simple sinks, not modeling their interior. Van der Waerden et al. (see [22]) develop a simple cellular automata based sub-model for choosing parking spaces inside a parking lot, but clearly set their focus on modeling traffic patterns resulting from the whole, city-wide process of traveling and parking.

Only a few of the described models consider off-street parking in any form. None of these is detailed enough for the evaluation of garage parking recommendation systems.

3 Modeling cruising behavior

An agent-based model usually includes two components (see [14]): the agents themselves, and the environment they interact with.

The agents are usually self-contained and autonomous; they have attributes whose values change over the course of a simulation run. Their behavior is determined by a set of rules, and they interact dynamically with other agents and the environment they exist in. In more complex models, agents are often goal-directed and adaptive, and may even be heterogeneous. Individual agents usually only interact with a local sub-set of the environment and other agents, i.e. in addition to their own memory, only local information is available to them.

In addition to their communication with their set of neighbors, agents interact with their environment. This information might provide only basic information, e.g. the agent's position in the environmental model. It may also provide more detailed information, e.g. the capacity and real-time rate of occupancy of parking garage areas. While in many cases the environment might be modeled as an attributed graph structure (see [20]), it sometimes is built as a complex simulation itself, e.g. based on cellular automata (see [12]).

In the described model, drivers and their cars are modeled as agents adhering to a set of rules and acting on local information, while the parking garage is modeled as an attributed neighborhood graph, and constitutes the agents' environment.

3.1 Modeling parking garages

The parking garage is modeled as an attributed graph $G(A,E)$ representing the garage's layout and the areas' neighborhood relations. A node $a \in A$ represents an area of the parking garage, an edge $e(a_i,a_j) \in E$ with $a_i, a_j \in A$ represents a direct connection of two areas a_i and a_j which is traversable by car. If all access lanes in the parking garage are two-way, the garage can be modeled as an undirected graph. If some or all segments only allow for one-way traffic, a directed graph can be established. As it is the garage planner's basic objective to ensure reachability of each parking area, the graph consists generally only of one connected component.

Each node $a \in A$ is attributed by its total number of parking spots z_a, the number of currently occupied spots $o_a(t)$ at time t, by extension also the number of free spots $f_a(t) = z_a - o_a(t)$ at time t, and the average time r_a a car needs to traverse the area. The recommendation method to be tested (see [21]) explicitly considers only these areas, and does not depict individual parking spots. Therefore, a spatially explicit modeling of these individual spots is not necessary in this context.

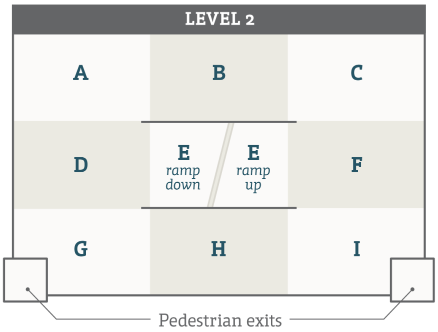

Figure 2: Simplified parking garage level with two exits, two bi-directional ramps, and nine areas

Each edge $e(a_i,a_j) \in E$ is attributed by a time r_e a car needs to traverse the edge from leaving a_i, and to reach the indicated area a_j. In most cases, if areas are directly adjoining, $r_e = 0$ can be assumed.

In this simplified model we assume an infinite traversal capacity for nodes and edges, therefore ignoring congestion resulting from multiple cars cruising the same area.

The garage's entry lanes are modeled as special nodes $a_e \in A_e \subset A$ with $z_{a_e} = 0$, which serve as sources for the transient car agents. In discrete modeling, interarrival times are usually approximated with an exponential distribution with an arrival rate of λ and average interarrival times of $\mu = 1/\lambda$ (see [4], pg. 248). In the described model the distribution parameter $\lambda_{a_e}(t)$ is established for each entry lane a_e by input data analysis (see section 3.3) and dependent on the time t of day. Technically, the agents are generated by the event-based framework at each entry node at appropriately distributed simulation times.

A parking garage's exit lanes are modeled as special nodes $a_x \in A_x \subset A$, again with $z_{a_x} = 0$, doubling as sinks for transient entities, collecting statistic data and removing the agents from the model.

Figure 2 shows a simplified layout of a parking garage level, while figure 3 shows the corresponding partial model graph.

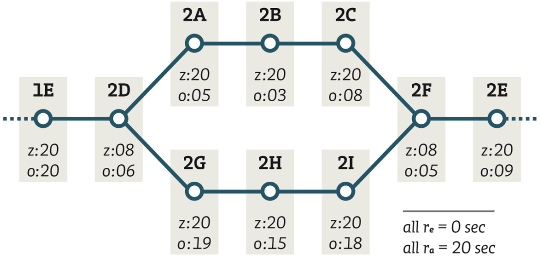

Figure 3: Partial model graph of a parking garage level

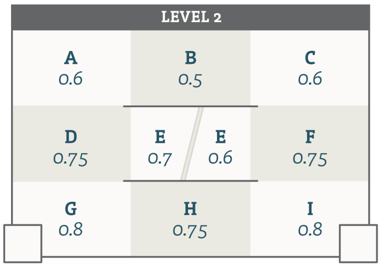

Figure 4: Parking garage level with attractiveness values

3.2 Towards modeling driver behavior

Agents enter the model from one of the entry lane nodes in A_e, and iteratively move from node a_i to node a_j along edge $e(a_i,a_j) \in E$. On any given node $a_i \in A$ the agent, after spending a time of r_{a_i} searching the area for available parking, has to take two decisions: It has to decide whether to park in the current area (parking decision), and, if not, where to go next (routing decision).

To enable the agents to take these decisions, the model considers a number of points:

Basic maneuvering: To avoid infinite loops, an agent administers a counter $v(a_i)$, which represents the number of times an area a_i has been visited by that agent. If an agent always chooses the routing option a_j with the lowest $v(a_j)$, every circle will eventually be broken. In addition to that, as cars are rarely seen to turn on the spot in a parking garage, agents can never move onto the area they just left.

Attractiveness: The model assumes that a driver prefers to park at a spot which is as attractive as possible. This attractiveness might correspond to the spot's distance to a pedestrian exit or vehicular entry. The model therefore assumes an order of attractiveness on a parking garage's areas: $1.0 \geq c(a_{i_1}) \geq ... \geq c(a_{i_n}) \geq 0.0$ (see figure 4). Agents prefer areas with greater values of $c(a_i)$ to areas with lower attractiveness.

Real-time availability: Drivers also consider real-time availability: if they observe that no spaces are available in an area ahead, they are not attracted to it. Obviously, without technical measures (which are not assumed at this point) the drivers cannot have total knowledge of the current state of the garage, but can look ahead only locally. How far drivers can look ahead is dependent on their individual position and architectural attributes of the garage. This simple model assumes a local look-ahead of only one area: an agent therefore knows the number $f_{a_i}(t)$ of available slots in the current area a_i, and in directly neighboring areas. Therefore we assign a look-ahead set $L_{a_i} \subseteq A$ for any current area a_i.

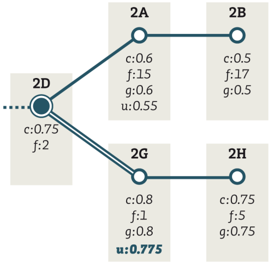

Figure 5: A simplified option tree

Based on these considerations, and starting out from the current position a_i as root, an option tree is constructed. This is accomplished by considering iteratively all neighboring areas a_j reachable from a_i via an edge $e(a_i,a_j) \in E$, and from there on succeeding neighbors with a maximum depth of m. In a simple implementation this value might be set to $m = 2$ (see figure 5). The branch starting with the area last visited is removed from the tree, adhering to the no-turn-around rule.

For each element a_j in that tree, a value $g(a_j)$ is calculated: if $a_j \in L_{a_i}$ and $f_{a_j}(t) = 0$ then $g(a_j) = 0$, else $g(a_j) = c(a_j)$. Thus, if the agent observes an area with currently zero available slots, it is not at all attracted to that area. Then for each immediate option a_j a value $u(a_j)$ is calculated by averaging the area's $g(a_j)$ value and its n successors' $a_{j1}...a_{jn}$ individual $g(a_{ji})$ values (again see figure 5):

$$u(a_j) = \frac{g(a_j) + g(a_{j1}) + ... + g(a_{jn})}{n+1} \qquad (1)$$

At each simulation step, the agent makes a parking decision, followed by a routing decision if necessary. To take a *parking decision*, it selects the a_g with the maximum $u(a_g)$ out of the current area a_i's immediate neighbors. If $f_{a_i}(t) > 0$ and $c(a_i) \geq u(a_g)$, the agent decides to park at the current area a_i, else it moves on with the routing decision.

To take the *routing decision*, the agent only considers the options with the least $v(a)$. From these, the agent selects the option with the greatest $u(a_j)$. It moves to that area via the edge $e(a_i, a_j)$, a move that will be completed after a time of $r_{e(a_i,a_j)}$.

If all areas have been visited, i.e. all $v(a) > 0$, and no available parking spot has been found, the agent concludes that the parking garage is full, and stops searching.

3.3 Input and output data

To successfully model a given parking garage's operations on a specific operational day a variety of *input data* has to be available. While part of that data is fundamental for a given parking garage, other elements are dependent on days of the week, special events, etc, which are to be modeled.

Fundamental to a specific parking garage are its layout and configuration, represented in the attributed graph $G(A, E)$. This data can be gathered by studying floor plans, or by endeavoring on a fact finding expedition.

To model a specific operational day the distribution of the number of entering cars over time t has to be available, especially the arrival rate $\lambda_{a_e}(t)$ on each entry lane $a_e \in A_e$. As many parking garages control access with technical means, e.g. license plate reader and inventory systems, this data is often available. It can be complemented, or even replaced, by own measurements.

Additional information is necessary to model local drivers' preferences. The attractiveness value $c(a)$ for each parking garage area $a \in A$ can be established by computational means, e.g. by combining the distances of a spot to the vehicular entry and exit lanes, as well as to the pedestrian exits, or by interviewing local experts.

There might be more than one of these orders, based on whether the parking garage is used by customers with distinct destinations, e.g. 40% might come to park as close to the supermarket as possible, while 60% might be staff, patients, or visitors of a hospital. These distribution could change over the time of day, e.g. when the hospital might be closed for the evening, but a cinema starts to attract parking visitors. As these classes of preferences are shared by many drivers, only a few different orders of attractiveness might be enough to represent all drivers' intentions for any given garage.

The distribution of parking duration again can be measured with the help of cameras and license plate reader and inventory systems, or by interviewing local experts. The average cruising speed can simply be measured.

The model's *output data* is utilized both as the primary result of the simulation, as well as for validation purposes. The main result necessary to evaluate the effect of parking recommendation methods is of course the distribution of the time spent to search for parking. For validation purposes a variety of other data is interesting, e.g. the number of cars in individual areas over the time of the operational day. These results can easily be compared to real-world observations.

4 A look ahead

Several important points are obviously missing from the described model: A comprehensive model should consider long-term experience and expectations, especially on the state of parking garages at any given time of day. It should also consider availability information and parking recommendation systems (ranging from simple red/green light arrangements at the entry lanes, via computer screens showing the number of free spots on each garage level, to more comprehensive, smart-phone based systems (see [21])). Furthermore, many parking providers offer different classes of parking decals, with some classes having more options then others: at a university campus, administrative and faculty/staff might be allowed to park at any given area, while students might only park at labeled student parking. Disabled drivers might have reserved spots available, likewise electric cars. A business park's parking provider might distinguish executive, employee, and visitor parking.

While in this simple model parking and routing decisions are seen as distinct steps, a more comprehensive representation would probably model them as one encompassing decision.

5 Summary and further research

This paper presented a simple agent-based model of cruising for parking in a parking garage. Beyond the parking structure's layout and attributes, the model considers basic navigation, an order of attractiveness on the garage's areas, and local information on current availability. In addition to this, the paper also presented some thoughts on necessary steps to expand and refine the model to make it applicable as a tool for recommendation method evaluation.

Though the model in its current state is still very basic, it has thus been confirmed that discrete modeling techniques, especially the agent-based paradigm, are suitable to model garage parking behavior.

As described, a lot of work still remains to be done: The parking behavior model has to be refined, both sub-models have to be implemented based on an inhouse simulation software framework, then calibrated and validated utilizing available data on specific parking garages. After the completion of these steps, the model will be applied to the evaluation of parking recommendation methods.

6 Acknowledgements

This material is partially based upon work supported by the National Science Foundation under grants I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, and FRP IIP-1230661, as well as U.S. Department of Transportation under a 2013 TIGER grant.

References

[1] Asakura, Y., Kashiwadani, M.: Effects of Parking Information on System Performance - A Simulation Model Approach. In: Proceedings of the 1994 Vehicle Navigation and Information Systems Conference, 1994, pp. 251-254.

[2] Ayala, D., Wolfson, O., Xu, B., DasGupta, B., Lin, J.: Pricing of Parking for Congestion Reduction. Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Redondo Beach, CA, November 6-9, 2012, pp. 43-51.

[3] Ayala, D., Wolfson, O., Xu, B., DasGupta, B., Lin, J.: Parking Slot Assignment Games. Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Chicago, IL, November 1-4, 2011.

[4] J. Banks, J. S. Carson, B.L. Nelson, D. M. Nicol. Discrete-Event System Simulation, 5th edition. Pearson, 2010.

[5] Benenson, I. Martens, K., Birfir, S.: PARKAGENT: an agent-based model for parking in the city. In: Computers, Environment and Urban Systems, Vol. 32, No. 6, 2008, pp. 431-439.

[6] Choi, B.K., Kang, D.H.: Modeling and simulation of discrete-event systems. Wiley, 2013.

[7] Dieussaert, K., Aerts, K., Steenberghen, T., Maerivoet, S., Spitaels, K.: SUSTAPARK: an agent-based model for simulating parking search. In: Proceedings of AGILE International Conference on Geographic Information Science, 2009.

[8] Gallo, M., D'Acierno, L., Montella, B.: A multilayer model to simulate cruising for parking in urban areas. In: Transport Policy, Vol. 18, 2011, pp. 735-744.

[9] Horni, A., Montini, L., Waraich, R. A., Axhausen, K. W.: An Agent-Based Cellular Automaton Cruising-For-Parking Simulation. In: Proceedings of the 13th International Conference on Travel Behaviour Research, Toronto, Canada, July 15-20, 2012.

[10] Kaplan, S.; Bekhor, S.: Exploring en-route parking type and parking-search route choice: Decision making framework and survey design. In: Proceedings of 2nd International Choice Modelling Conference, 2011.

[11] Kanev, K., Mirenkov, N., Urata, A.: Parking Simulation and Guidance in a Model Environment. Proceedings of the 2006 Japan-China Joint Workshop on Frontier of Computer Science and Technology (FCST), 2006.

[12] Kuckertz, P.; Ullrich, O.; Linstädter, A.; Speckenmeyer, E.: Agent based modeling and simulation of a pastoral-nomadic land use system. In: Simulation Notes Europe (SNE), Volume 21, Number 3-4, Dezember 2011, pp. 147-152.

[13] Levy, N.; Martens, K.; Benenson, I.: Exploring cruising using agent-ased and analytical models of parking. In: Transportmetrica A: Transportation Science, Vol. 9, No. 9, 2012, pp. 773-797.

[14] Macal, C. M.; North, M. J.: Tutorial on agent-based modeling and simulation. In: Journal of Simulation 4 (2010), pp. 151-162.

[15] Martens, K.; Benenson, I.: Evaluating Urban Parking Policies with Agent-Based Model of Driver Parking Behavior. In: Transportation Research Record, No. 2046, 2008, pp. 37-44.

[16] Saltzman, R. M.: An Animated Simulation Model for Analyzing On-Street Parking Issues. In: Simulation, August 1997, pp. 79-90.

[17] Sargent, R.G.: Verification and Validation of Simulation Models. In: Proceedings of the 2010 Winter Simulation Conference, 2010, pp. 166-183.

[18] Shoup, D. C.: The high cost of free parking. Planners Press, Vol. 206, 2005.

[19] Shoup, D. C.: Cruising for parking. In: Transport Policy 13 (2006), pp. 479-486.

[20] Ullrich, O.; Proff, I.; Lückerath, D.; Kuckertz, P.; Speckenmeyer, E.: Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation. In: ITS for Connected Mobility - Selected Papers from the mobil.TUM 2013 International Conference on Mobility and Transport, Hrsg: Busch, F., Spangler, M., 2014, pp. 89-98.

[21] Ullrich, O.; Rishe, N.: Emissions Reduction in Vehicle Parking. U.S. Patent pending, Application No. 14/660,283, filed March 17, 2015.

[22] van der Waerden, P., Timmermans, H., Borgers, A.: PAMELA - Parking Analysis Model for Predicting Effects in Local Areas. In: Transportation Research Record 1781, Paper No. 02-2078, 2002, pp. 10-18.

[23] Xu, B., Wolfson, O. , Yang, J., Stenneth, L., Yu, P. S., Nelson, P. C.: Real-time Street Parking Availability Estimation. Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM). Milan, Italy, June 3-6, 2013, pp. 16-25.

[24] Young, W., Weng, T. Y.: Data and Parking Simulation Models. In: Kitamura, R., Kuwahara, M. (Ed.): Simulation Approaches in Transportation Analysis, Springer Operations Research/Computer Science Interface Series, Vol. 31, 2005. pp. 237-267.