Postprint

This is the accepted version of a paper published in *European transactions on telecommunications*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

---

Citation for the original published paper (version of record):

Atif, Y., Kharrazi, S., Ding, J., Andler, S F. (2020)
Internet of Things data analytics for parking availability prediction and guidance
*European transactions on telecommunications*
https://doi.org/10.1002/ett.3862

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

This is the peer reviewed version of the following article: Atif Y, Kharrazi S, Jianguo D, Andler SF. Internet of Things data analytics for parking availability prediction and guidance. Trans Emerging Tel Tech. 2020;e3862. https://doi.org/10.1002/ett.3862, which has been published in final form at https://doi.org/10.1002/ett.3862. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-18081
Internet of Things Data Analytics for Parking Availability Prediction and Guidance

Yacine Atif†, Sogol Kharrazi†, Ding Jianguo*, Sten F. Andler*

*University of Skövde †National Road and Transport Research Institute
Sweden Sweden
Email: {Yacine.Atif, Ding.Jianguo, Sten.F.Andler}@his.se, Sogol.Kharrazi@vti.se

Abstract—Cutting-edge sensors and devices are increasingly deployed within urban areas to make-up the fabric of TCP/IP connectivity driven by Internet of Things (IoT). This immersion into physical urban environments creates new data-streams which could be exploited to deliver novel cloud-based services. Connected-vehicles and road-infrastructure data are leveraged in this paper to build applications that alleviate notorious parking and induced traffic-congestion issues. To optimize the utility of parking-lots, our proposed SmartPark algorithm employs a discrete Markov-chain model to demystify the future state of a parking-lot, by the time a vehicle is expected to reach it. The algorithm features three modular sections. First a search process is triggered to identify the expected arrival-time periods to all parking lots in the targeted Central Business District or CBD area. This process utilizes smart-pole data-streams reporting congestion rates across parking-area junctions. Then, a predictive analytics phase uses consolidated historical-data about past parking-dynamics, to infer a state-transition matrix showing the transformation of available spots in a parking-lot over short periods of time. Finally, this matrix is projected against similar future seasonal-periods to figure out the actual vacancy-expectation of a lot. The performance evaluation over an actual busy CBD area in Stockholm (Sweden) shows increased scalability capabilities, when further parking-resources are made available, compared to a baseline case algorithm. Using standard urban-mobility simulation packages, the traffic-congestion aware SmartPark is also shown to minimize the journey duration to the selected parking-lot while maximizing the chances to find an available spot at the selected lot.

Keywords—Smart parking; Markov chain; Internet of Things; optimization; traffic management; search; Artificial Intelligence;

I. INTRODUCTION

The increased volume of individuals in a limited land area to seek some activity provides grounds for a parking problem to arise. Motorists hope to locate an available spot that is close to business and/or administrative attractions in order to minimize walking distances to desired activity. Urban-cores house Central Business District (CBD) areas where human activity thrives and hence raises the demand for further parking spaces. Municipalities need to attract office tenants, shopping consumers, and tourists to CBD areas, and many of them expect to use their vehicles which they park upon arrival from home. However, on-street parking lowers road-capacity in crucial fluid CBD areas, and hence off-street parking-lots spread across 1-2 km radius within hotspot CBD areas or districts are recommended [1]. These parking structures require less land and can better be screened compared to their on-street counterparts.

Ultrasonic and magnetic sensors are already mounted on the ceiling to detect the availability of a parking spot, particularly in parking lots. The sensor conveys the parking-spot availability information in real-time to a gateway to be processed by a parking system which for instance displays the rate of available spots on screens. The gateway could also communicate that information to a Cloud-based service that is further exploited by third-party parking-service providers. This trend is part of the current Internet-of-Things (IoT) evolution that is powering the reach to contextual information of a wide range of future smart-city sub-systems, such as energy, waste and traffic management. Smart-poles are part of this evolution whereby junctions are able to identify traffic patterns across outgoing edges [2, 3, 4, 5, 6]. Using traffic sensors, a pole can detect the movements of individual vehicles, leading to a range of analytical applications to improve traffic-induced issues including parking-related ones.

The streams of data that literally "senses the city" is increasingly driving open Cloud-services to stimulate technology and business innovations [7, 8]. These services are expected to be exploited for better infrastructure management and new added-value services to both users, city managers and businesses. Smart parking service providers are potential beneficiaries of this evolution, particularly with the expected progression towards connected and driverless vehicles. In this foreseeable context, parking land is a commodity used by the real-estate owner and/or by tenants for a fee, while new parking-service provider (PSP) intermediaries supply necessary IoT infrastructures to bring that parking information to a Cloud service. Fig. 1 illustrates this Parking as A Service that is driven by IoT integration in traffic subsystems of smart cities. For example, supermarkets may outsource their parking-facility to a PSP in off-peak periods to generate new revenue streams.

Driven and autonomous vehicles may cruise through streets and need to park when dropping passengers, and taken out of service. At this time, previously-registered PSPs relay parking-lot availabilities and bind their navigation services to vehicles. This is where our proposed SmartPark approach comes into action to run as a Cloud service using PSP-provided data, offering an available spot in a parking-
The parking selection problem addressed in this paper makes some assumptions, states research questions and targets specific objectives. The assumptions driving the proposed solution in this paper include the pervasive integration of sensing and communication devices in parking lots, whereby the rate of entering and departing vehicles are known at anytime, as well as parking-lot occupancy. This assumption drives the expectation that such historical data is available, which could be supplied by contemporary parking-lot operators or future PSPs seeking to leverage their services quality and increase the utilization rate of owned parking lots. We also assume that a vehicle’s GPS-location is known when entering the parking area. This allows the proposed algorithm in this paper to be triggered and use the vehicle’s entry point as a navigation startup to the selected parking-lot. However, motorists may specify the desired parking-area entry point, and expected entry-time into the area offline to trigger the algorithm, as well. Alternatively, these parameters are determined online as vehicles proceed into a parking-lot, as well as a congestion-aware navigation route. The foreseeable rise of parking real-estates, which may be driven by profit incentives from PSPs and land owners takes the parking-selection problem to a higher-complexity level. The complexity of numerous parking options to pick from, is amplified by balancing stakeholder needs. These include city operators who are wary about carbon-emission from cars roaming around to find an available parking spot, and PSPs pushing for an increased service utilization, as well as drivers seeking a suitable spot.

The combination of data streams from parking-lot gateways and traffic junction smart-poles are particularly of interest in this paper. We argue that the accumulation of such data provides historical grounds for optimizing future decision-marking. Furthermore, we contribute a parking-selection and navigation algorithm named SmartPark which performance evaluation shows increased scalability capabilities when further parking resources are made available in the parking area. The traffic-congestion aware SmartPark also minimizes the journey duration to the selected parking-lot, while maximizing the chances to find an available spot there.

The remaining sections in this paper are organized as follows. Section II describes further the problem addressed in this paper and explores some related works. Section IV reveals the first part of our proposed SmartPark algorithm that deals with routing vehicles to designated parking lots, while the second part of the algorithm that demystifies SmartPark’s approach to predict future parking states and optimizes parking-lot selection is further elaborated in Section V. Section VI shows the experiment setup and the performance results of the proposed SmartPark approach, against a greedy approach labelled BlindPark. Finally, Section VII concludes the paper with a summary of our contributions and some suggested future works.

II. Problem Statement

The parking-selection problem is confined within urban cellular areas, named "parking areas" which are instances of a 1-2km radius of land within CBD regions. This modular approach allows the proposed parking-selection algorithm to operate within some preset attributes that are specific to each particular area. There may be several parking-areas with vehicles cruising across them, in which case separate execution-instances of the algorithm are triggered.

A. Problem description

Thousands of motorists are daily faced with the dilemma to choose among numerous parking-lots when entering a city area in the urban core to reach many destinations of interest. While motorists are mostly aiming at minimizing walking-distance to destination, driving-time to parking lots, and parking-fees, there are other stakeholders such as city and parking operators who would like to factor-in the traffic-induced while searching for parking, and the distribution of parking-lots’ utilization. On the other hand, Internet-connected components are generating streams of real-time data about both traffic-flow and parking usage, which could be exploited to judiciously reveal the actual utility of selected parking-lots. The search for parking lots suggests examining policies that influence parking-lot selection decisions, which take into account concurrent parking requests in a way that spreads traffic, unlike traditional navigation applications which drive motorists along the same path to a common parking destination.

Furthermore, parking lot availability and traffic situation fluctuate across different seasonal periods, raising the need for a data approach to the parking selection problem that learns from past historical seasonal data to predict current parking- and traffic-dynamics. IoT enabled parking systems capture the dynamics of parking-lots across different times of the day, and smart-poles provide traffic information during different hours. City managers and parking-lot owners can leverage the accumulated data warehouses to improve sustainability, service-quality and overall business profits.

The parking selection problem addressed in this paper makes some assumptions, states research questions and targets specific objectives. The assumptions driving the proposed solution in this paper include the pervasive integration of sensing and communication devices in parking lots, thereby the rate of entering and departing vehicles are known at anytime, as well as parking-lot occupancy. This assumption drives the expectation that such historical data is available, which could be supplied by contemporary parking-lot operators or future PSPs seeking to leverage their services quality and increase the utilization rate of owned parking lots. We also assume that a vehicle’s GPS-location is known when entering the parking area. This allows the proposed algorithm in this paper to be triggered and use the vehicle’s entry point as a navigation startup to the selected parking-lot. However, motorists may specify the desired parking-area entry point, and expected entry-time into the area offline to trigger the algorithm, as well. Alternatively, these parameters are determined online as vehicles proceed into a parking area.
Smart-poles are available at each junction of the parking-area, recording traffic patterns through real-time data that can contribute to reduce congestion and carbon emission in the urban core, as influenced by the proposed parking-lot selection approach. Proxy agents support the communication of data from Internet-connected components, which in this case consist of smart-pole and parking-lot mounted-sensors. This assumption allows the algorithm to use standard APIs to query proxy-agents for IoT data. The proposed algorithm operates within the boundaries of a parking-area, and thus any parking-lot within that area is a plausible solution candidate. This assumption allows the algorithm to utilize all parking lots in the parking area as part of the solution domain.

The main problem addressed in this paper is how to choose the best parking-lot in a given parking area to maximize a utility function involving some weighted criteria. We contribute an algorithm of a parking-lot selection that takes into account the current traffic situation, so that the parking selection problem does not contribute to exacerbate a traffic-congestion problem. The main problem can be further decomposed into some research questions investigated throughout this paper. How to integrate parking selection and traffic congestion problems? How to predict the future state of a parking-lot when the request is triggered at an entry point of the parking-area? How to navigate vehicles separately across separate routes, although possibly destined to the same parking lot? How to exploit IoT-data to produce analytical insights that support congestion-aware parking selection algorithms?

For each triggered parking-request, the proposed algorithm strives to achieve some objectives, including predicting parking availability in lots within the parking area, that are deemed to fulfil motorists’ destination requirements. This parking-lot selection involves a vehicle routing approach that maximizes the utility of the selected parking lot considering a multicriteria decision-making process.

**B. Multicriteria decision-making**

How to choose the best parking lot in a given parking-area is generally subject to a range of criteria that may include user, municipality and parking-operator preferences. Users may have some preferences with respect to walking distance to destination. Municipalities prefer to spread the traffic to reduce congestion in the urban core. Parking-operators seek to maximize parking-lots utilization in order to increase profits on real-estate investments. Fig. 2 illustrates the combined optimization process involving data sources from these stakeholders as input. Different stakeholders may influence the outcome of the parking and routing outcomes.

The generic expression of the optimization formula may be contextually instantiated to exhibit some relevant criteria and corresponding weights to guide the selection of parking-lots and the navigation towards them. For example, municipalities may prioritize congestion. On the other hand, parking-lot operators may wish to raise parking-lots occupancy. And finally, users may trade walking-time to destination with corresponding parking-fees. Different functions may have preset criteria and weights according to different parking-areas.

![Fig. 2. Utility function for a given parking-area](image)

A cloud-based service receives data sources and related weighted values to optimize parking-selection and vehicle-navigation decisions through SmartPark algorithm discussed later in this paper. Despite the generic formulation of the utility-function, the scope of this paper emphasizes these two latter criteria. User preferences are already integrated in some existing mobile and in-vehicle parking system applications, where the focus has been on the vehicle and its passengers.

**III. BACKGROUND AND RELATED WORKS**

The current smart city trend stemmed essentially from unleashing new data sources used to improve inhabiting citizens and housed businesses [9]. Ubiquitous computing frameworks involving a plethora of technological devices are incorporated across the urban environments’ fabric, leading to diverse emerging-applications, which in turn generate massive data that await effective utilization [10]. Urban mobility is one of the major beneficiary of this evolution, particularly in metropolitan cities due to the foreseeable proliferation of connected vehicles, which are equipped with Internet connectivity [11], and smart-poles that are equipped with sensors to enable intelligent transportation systems [12]. Subsequently, a wealth of fertile research opportunities emerged in urban-traffic ecosystems, including parking solutions [13].

Some of the advantages of data unleashed from IoT-embedded urban infrastructures, is the opportunity to engage into predictive analytics to forecast traffic and parking dynamics that tune decision-making processes [14]. There are two mainstream bodies of literature to represent state-variables of interest for predictive analysis: continuous and discrete models. The difference tells about the amount of state-variable instances or measurements that need to be collected or represented to perform the predictive analysis.
Continuous measurements enable state-variables to take any value on a number line, whereas discrete measurements are confined to integer instances. Since predictive analysis is often probabilistic, this distinction results in different distributions. Discrete instances are described through a probability distribution which lists all potential instances and the associated probability of occurrence. Continuous predictors however, can take unlimited instances between lowest and highest measurement points, and thus are not restricted to specific instances. The multi-attributes parking selection problem was addressed under both considerations to predict the availability-state instances in a parking lot. Continuous queuing-theory approaches [15, 16, 17], as well machine-learning approaches [18, 19, 20, 21] including deep-learning approaches [22, 23, 24] have been reported, with varying degrees of effectiveness and considerations for both parking-lot and traffic dynamics, simultaneously. Discrete approaches to the parking-selection problem like our proposed approach, have been more active to quantify uncertainty over future parking-lot states, as discussed next.

Discrete approaches to the parking-selection problem have been quite extensively investigated, and can be summarised into three main categories. Earlier studies focused on the parking-type selection addressed issues such as off-street vs. on-street parking with respect to pricing dynamics [25, 26]. Subsequent research thrust combines travel experience and parking issues, such as park-and-ride optimization practices [27, 28, 29]. Finally, contemporary research directions combine navigation and parking information to alleviate congestion-issues associated with the cruising process towards a designated parking lot. Our work is positioned within this scope to investigate decision-making approaches for parking-selection that includes the routing process. The targeted issues varied between cruising time [30, 31, 32], parking-search space [33, 34] and the incurred search complexity [35, 36, 37]. We propose a stochastic process centred on parking-availability expectations within an actual urban area and using parking-dynamics data aggregated from IoT-enabled parking-gateway platforms.

IV. COLLABORATIVE PATH-FINDING

Parking is a distributed search problem to find the best parking-lot in the parking area. As illustrated in Fig. 3, a parking area is represented as a directed graph, where nodes represent junctions and edges represent road lanes. Each junction is linked to other junctions or parking lots by a single or a double-lane road. Each smart-pole collects congestion data via traffic-sensors, while parking-lots accessed via dedicated junctions are equipped with availability-sensors of parking-lot spots. Each parking lot detects and reports parking occupancy state by accessing smart parking-spot gateways. IoT-enabled components in a parking-area interact together to collaborate in the path-finding process. Proxy agents are associated with junctions and parking-lots to facilitate a distributed decision-making approach in an IoT network context [38].

The decentralized system design provided by IoT enable junctions and parking lots to sense, compute and communicate to collaborate in the fulfillment of path-finding requirements. The agent-based computing paradigm has been widely advocated to support IoT systems to develop smart environments of varying scale degrees [38, 39, 40]. Our agent-based collaborative path-finding methodology is a follow-up to our previous IoT enabling by sensing research to guide software architects realizing IoT applications [41]. Hence the middleware and related technological aspects of this enablement are beyond the scope of the present paper. Within this environment, several agents can be integrated to provide a framework for agents to communicate and collaborate.

A. Parking-area representation

Each IoT component of the parking area is managed by an autonomous agent. The goal of the search process is to locate a parking junction and then to compute the expected arrival time considering the traffic congestion from a given entry point. The search is carried out along two processes. A forward search starts from the entry point node which scouts the parking area for an available parking lot. When found, the predicted availability of the corresponding parking is calculated, considering the estimated arrival time. A backtracking process carries back the availability-rate and congestion attributes to the entry node. While doing so, the utility of the parking lot is re-evaluated at each junction point and the maximum utility is relayed back to antecedent node, until it reaches back the vehicle at the entry point node.

To illustrate the above process, Fig. 4 shows an example of vehicles being routed to designated parking lots upon entering a congested parking area. We adopted a simplified congestion model that is consistent with the conventional traffic-flow theory [42]. Each road edge between two junctions is directed and has a capacity of vehicles driven over
the edge lanes. Entering vehicles onto an edge \( i \) of the road are captured by junctions’ smart-poles to determine the density \( d_i \) of the traffic, formulated as \( d_i = \frac{N}{K_i} \), where \( N \) is the number of vehicles driven over the road edge, and \( K_i \) is the maximum number of vehicles that could be accommodated over that edge, expressed as percentages in the graph-based representation of the parking area in Fig. 4. This local information is worked out in real-time by smart poles and communicated to the search process triggered by a vehicle entering a parking area to figure out a global congestion rate. The congestion model is generalized over the parking area with a global value \( C = \sum_{i=1}^{j} \frac{d_i}{L} \), where \( L \) is the total number of road edges in the parking area. This global congestion rate estimate is used later by the parking-state prediction algorithm to infer parking states, considering expected arrival time.

![Fig. 4. Routing vehicles to parking-lots in a congested parking-area](image)

**B. Vehicle routing**

Given a parking lot, the path-finding problem consists in finding the best path to that lot from a given entry point, in terms of driving time in order to derive the expected arrival time to the parking lot. The search space consists of a graph where the root node represents the entry point, and the nodes at the next level represent all junctions that could be visited first from the entry point, whereas the nodes at the following level represent all junctions that could be visited from outgoing junctions in the previous level, etc. In this tree-like view of the graph traversal, the maximum depth is the number of junctions, and the candidate parking junction occur at this level of depth. Dijkstra algorithm guarantees to find an optimal path which minimizes travelling time to a given parking-lot. However, this suggests that multiple vehicles coming through the same entry point and heading for the same parking lot, may not follow the same “shortest” path. This is because a query to smart-poles junctions enable the algorithm to work out a new path for each parking-request considering the new \( d_i \) values of congestion-rates returned by smart-poles, resulting in a new estimate of the parking-area’s congestion rate \( C \). For illustration, Fig. 5 shows near real-time congestion-rates worked-out from smart-poles for different vehicles entering a parking-area. At time \( t_1 \), an entering vehicle receives Congestion-rate \( C_1 \), whereas a following vehicle which enters the same parking-area, receives a compiled Congestion-rate \( C_2 \). The differences in congestion-rates result in different shortest-paths to a given parking-lot, since edge-weights of the road-map graph changed.

![Fig. 5. Example of congestion-rates worked-out from IoT-enabled smart-poles across time](image)

Algorithm 1 depicts the path-finding search process triggered initially by SmartPark algorithm to obtain optimal paths to given parking lots considering a list of congestion-rate inputs from smart-poles’ traffic sensors. The algorithm is executed upon each vehicle invocation for a promising parking-lot, and thus a subsequent invocation implicitly collaborates with a previous invocation considering the contribution to congestion each entering vehicle makes within the parking area.

**Algorithm 1** SmartPark - Path-finding

1: procedure FindRoute\( (E, p, PA, LC) \)
   \( \triangleright \) \( E \) is the entry point to parking area \( PA \) where there is a parking lot \( p \) and a list of edge congestion-rates \( LC \)
2: \( G \leftarrow \text{ConvertToGraph}(PA) \)
3: \( G \leftarrow \text{AddWeights}(G, LC) \)
4: return ShortestPath\( (G, E, p) \)
   \( \triangleright \) Returns the shortest-path in the weighted graph \( G \), from Node \( E \) to Node \( p \)
5: end procedure

V. PREDICTIVE ANALYTICS

Data analytics predictive models’ accuracy rely on fed data, which may increase their effectiveness overtime. One way to effectively monitor prediction [43], is accumulate data from historical repositories, run it through existing situations with predictive analytics algorithms and visualize the results. If results deviate from expectations, parameters
within the algorithms may need to be adjusted further and/or additional historical/current data is sought out. First data across historical repositories need to be consolidated into a format suitable to the employed analytics approach. This step includes data cleaning and accumulating sufficient data for analytical inferences. Parking data is usually available in transactional databases format, that is converted into a flattened view which emphasizes predictor variables used to make actionable decisions. The "available parking spots" in a parking lot is the variable of interest here, which is captured at periodical seasons and time snapshots, to infer a state-transition matrix showing the transformation of available spots in a lot over a short period of time. This matrix is employed to figure out the vacancy probability of a lot when moving from one state to another. Transition matrices with different input parameters but across adjacent time-periods are multiplied to represent the parking dynamics over an extended time-span duration, and determine trends and patterns. Subsequently, the transition matrix is used to guide parking requests to lead vehicles to the most promising lot, given the anticipated arrival-time expectation.

A. Historical data representation

There exists a number of applications currently used to simplify the quest for an available parking spot, which use publicly available data [44]. User-provided data or parking-lot sensors are potential source of information used by these applications. Low-power, water-resistant and high communication-range sensors can be affixed to the ground of parking spots, providing an affordable and highly accurate technology to connect physical real-estate parking areas to a cloud-based application or end-user mobile apps. ParkoPedia is an example of user-provided parking data [45]. Contemporary trends promote connected and automated parking solutions [46, 47]. Connected services support the search for parking by guiding vehicles straight to available parking spots.

Connectivity technologies or users make it possible to generate and store historic data of parking occupancy. Using this data repository, a categorisation based on seasonal periods across several instances of historical data, is inferred. This categorisation is used to match current parking considerations with similar historical situations for a given parking lot. The seasonal classification aims at capturing similar periods, which is chosen to be small enough to reduce variations in parking dynamics. This data representation is meant to model the randomly changing parking’s available spots whereby a future state depends only on the current state, and does not depend on any event that may have occurred before it. This property is called the Markov property whereby we can assume a stochastic model based on a Markov process to describe parking dynamics. Hence, it is the availability of a particular parking lot that is the variable of interest here, and Fig. 7 illustrates averaged values of such variable over a number of similar instances. The parking availability variable is averaged over multiple historical one-minute period observations across similar seasons. The time period is chosen sufficiently small to assume discrete state changes of a parking lot modelled over a time-span of 5 minutes during which arrival and departure rates are assumed fixed.

The next 5-minutes cycle uses a similar discrete states change model, but with different arrival and departure rate values. The small 5-minutes interval assumes a stationary arrival/departure rates Poisson process with fixed mean value of entering $\lambda_i$ and departing $\mu_i$ vehicles to/from parking lot $P_i$ over a single observation period. The Poisson distribution is used in our simulation to represent the arrival process into parking lots. The simulation uses the arrival times generated by this distribution to fill parking lots at run-time. The Poisson process changes dynamically the availability status of parking lots following the arrival of vehicles to parking-area lots.

B. State modelling

Based on the availability distribution collected from historical data classified over observation periods as illustrated in the previous section, parking states are derived for each parking lot $P_i$ of a certain capacity $C_i$. A state represents the parking availability range. Following Kendal’s notation, the classic Markov chain model with exponentially distributed inter-arrival times and and parking durations $M/M/C$ queue is used to predict the future availability state of a parking lot $S_i$ given that the current state is $S_j$, and denoted $P(S_i|S_j)$. A fixed arrival rate $\lambda_i$ of vehicles entering $P_i$ lot with maximum capacity $C = C_i$, and departure rate $\mu_i$, are used to describe $P_i$’s queueing model, over a 5-minutes observation period. Arrival/departure rates variation is handled across observation windows but considered fixed within each observation window that is deliberately chosen small.
enough to justify this assumption. Each state corresponds to
an availability-rate range. Fig. 8a illustrates a 6-states
model within some availability-ranges for a sample parking
lot with capacity \( C_1 = 1000 \). Subsequently, a frequency
matrix is elaborated from similar periods across historical
data illustrated in Fig. 7 to the currently 3-minutes observed
period. A dummy-data example of the frequency matrix is
sketched in Fig. 8b which for example says that the transition
from State \( S_0 \) to State \( S_1 \) occurred 30 times during the
observed-window period in a parking lot across the available
historical data. A normalized version of the frequency matrix is
inferred in Fig. 8c, which sketches the parking dynamic
availability patterns within the observed period in terms of
probabilistic weights.

C. Decision inference

The previous illustrations show our methodology to de-
sign a Markov Chain to forecast future parking states using
past information from historical data. Subsequently, the
state that displays the highest transition probability from
the current state is forecasted as the next state of the
corresponding lot. Our proposed model employs 6 states:
\( S = \{S_0, S_1, S_2, S_3, S_4, S_5\} \), ranging from the highly-
occupied state \( S_0 \) to the highly-available state \( S_5 \) of a
parking lot. The collected historical data facilitate the elab-
oration of the transition probabilities relationship between
states as depicted in the process of Fig. 8. Diagonal values
account the number of times the same availability range has
been registered within the same observation window, across
several similar intervals (e.g. Mondays) of past observation
windows (e.g. 8:00 AM to 8:05 AM). Parking availability
of a lot fluctuates from state State \( S_i \) as a vehicle enters
a parking area to State \( S_j \) as the vehicle is expected to
reach that parking lot. The likelihood of a sequence of
states’ fluctuation in a lot can be computed using the Markov
property (where \( S_k \in S \)):

\[
P(S_i, ..., S_j) = \prod_{k=i}^{j} P(S_k|S_{k-1}), \quad i, j \in \{0, 1, ..., 5\}
\]

The Markov model provides an inference approach
through connecting the dependencies of current period infor-
mation (in this case, parking-lot availability) with historical
information (previous availabilities in similar periods). For
each parking lot \( P_q \), and a current observation time-window
\( w_t \), an arrival rate \( \lambda_{qr} \), a departure rate \( \mu_{qr} \) and a state
transition-matrix \( A_{qr} \) are established as follows:

\[
A_{qr} = \begin{bmatrix}
a_{00} & a_{01} & \cdots & a_{0n} \\a_{10} & a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\a_{n0} & a_{n1} & \cdots & a_{nn}
\end{bmatrix}
\]

where \( a_{ij} = P(S_i = S_j|S_{t-1} = S_i) \), in which \( S_i \)
designates the observed state at time \( t \) that falls within
the time-window \( w_t \). That is \( a_{ij} \) represents the transition
probability from State \( S_i \) to State \( S_j \) for a given lot in
a given time-window. Given an observation-window set
\( W = \{w_1, ..., w_k\} \), we generate transition-matrices \( A_{qr} \), \( 1 \leq r \leq k, \) for each parking lot \( P_q \) in the parking area following
the approach discussed earlier in the previous subsection.
When entering a parking area at time \( t_0 \) that falls within
period \( w_t \), the expected arrival-time \( t_1 \) to a targeted parking-
lot is computed and the corresponding arrival period \( w_{t+1} \) is
identified, \( m \) shifts forward the entering period \( w_t \). If
the trip time exceeds the duration of a single observation-
window period (i.e. \( w_i \neq w_j \)), the corresponding transition
matrices for each window since \( t_0 \) and up to the arrival
time \( t_1 \) are multiplied to figure out the state of the targeted
parking lot \( P_q \). Hence, using the multi-step transition matrix
\( A_{q,t-n} = \prod_{l=m}^{n} A_{qr} \), we can look-up \( P(S_{t+1}|S_0 = S_l) \)
where \( n = l + m \) is the distance in \( m \) steps between the
periods spanning \( t_0 \) to \( t_1 \), and \( S_i, S_j \in S \). \( A_{q,t-n} \) represents
the transition states of lot \( P_q \) when the expected arrival time
spans a sequence of observation-windows \( w_k, 1 \leq k \leq n \).
However, under short periods of time which fit within a
single observation window size (i.e. 5 minutes), a single
states-transition matrix holds the parking dynamics infor-
mation. First, the state with highest transition probability
is predicted as the forthcoming availability of each parking
lot \( P_q \), and then the parking with the highest forthcoming
availability state, or the equivalent lowest failure rate is the
selected parking lot with the most promising vacant spots.

Algorithm 2 shows the driver framework of our parking-
selection approach while Algorithm 3 reproduces into steps
the above state-prediction process of the selected parking lot.
The parking selection process is triggered once a vehicle
enters a parking area from a given entry-point \( E \). The
geographical map of the parking area \( PA \) is an input to
the parking-selection algorithm as well as the list of parking
lots and their related attributes, such as capacity and current
availabilities. The traffic data returned by smart-pole sensors
within the parking area is conveyed to the algorithm in the
form of congestion rates compiled in a list \( LC \). Algorithm 2
figures out the parking-lot with the least chances of failing
to find a vacant spot once the vehicle entering Area \( PA \) reaches
the selected lot. The best route to reach each parking lot in
Area \( PA \) is searched considering Entry-Point \( E \), and the list
of current congestion-rates for road edges \( LC \). The search
process delivers a route for which the expected arrival time
\( T \) is worked out in the next step of the algorithm. Next,
the algorithm queries the parking lot for which Route \( R \) has
been chosen to retrieve the current parking state \( CS \).
The future state of the parking lot by the expected arrival-time
is forecasted considering Algorithm 3. The availability rate is
computed out of the forecasted stated and the corresponding
failure-rate is inferred. After compiling every parking-lot’s
failure rate in Area $PA$, the lot with minimum chances to fail including a vacant spot upon the arrival-time of the vehicle controlled by Algorithm 2 is returned.

**Algorithm 2** SmartPark - Parking-lot selection

```
1: procedure FINDPARK($E$, $P$, $PA$)
   $\triangleright$ $E$ is the entry point to parking area $PA$ where there is a list of parking lots $P$
2:   $LC \leftarrow$ TrafficSensors($PA$)
   $\triangleright$ $LC$ is the list of congestion rates in Parking Area $PA$
3:   for each $p \in P$ do
4:      $R \leftarrow$ FindRoute($E$, $p$, $PA$, $LC$)
5:      $T \leftarrow$ GetArrivalTime($R$, $LC$)
6:         $CS \leftarrow$ GetCurrentState($p$)
   $\triangleright$ Query current state of Parking lot $p$
7:      $FS \leftarrow$ ForecastFutureState($T$, $p$, $CS$)
   $\triangleright$ The availability rate of Lot $p$ in State $FS$ is averaged from past historical instances
8:      $p.FailureRate \leftarrow 1 - p.AvailabilityRate[FS]$
9:   end for
10:  return ArgMin($P$, $FailureRate$)
   $\triangleright$ Returns the parking-lot index corresponding to the lowest failure rate
11: end procedure
```

To forecast the state of a parking lot, Algorithm 3 matches a current state with similar historical situations for a target parking lot $q$. Given the time-sliced segmentation of a parking lot into periods mapped to discrete states as shown earlier, the length of a period is first set to 5 minutes which is used to figure out the actual forecasted period $n$. The transition matrix $A_l$ for the current period $l$ is retrieved from at a similar past period using historical data as illustrated in the process of Fig. 8, and then multiplied by the sequence of transition matrices up to the one corresponding to the future period $n$ where the vehicle is expected to reach the parking lot. Using the resulting transition matrix $A_{q,l\rightarrow n}$, the future state of parking $q$ is predicted as the argument of the column that maximizes the availability of the current state row $S$.

**Algorithm 3** SmartPark - Parking state prediction

```
1: Parameter
2:   PeriodLength = 5
3: end Parameter
4: procedure FORECASTFUTURESTATE($S$, $T$, $q$)
   $\triangleright$ $T$ is the future time duration in minutes starting form the entry-time to parking area, at which the state of parking-lot $q$ is sought given current state is $S$
5:   $n \leftarrow T \mod PeriodLength$
   $\triangleright$ $n$ is the observation window for the expected arrival time $T$
6:   $A_{ql} \leftarrow$ TransitionMatrix($q$, $l$)
   $\triangleright$ $A_l$ is the transition matrix of parking lot $q$ in a past period that is similar to the current-period index labelled $l$
7:   $A_{q,l\rightarrow n} \leftarrow \prod_{r=l}^{n} A_{qr}$
   $\triangleright$ $A_n$ is the transition matrix of parking lot $q$ for the future period $n$ corresponding to the vehicle’s arrival time $T$
8:   return ArgMax($A_{q,l\rightarrow n}$, $S$)
   $\triangleright$ From the current state $S$, return the next state that maximises the availability of parking-lot $q$, using the trend matrix $A_{q,l\rightarrow n}$
9: end procedure
```

VI. EXPERIMENT AND EVALUATION

A. Simulation environment

We developed a simulation environment made up of three modules as illustrated in Fig. 9. A driver Python code positions parking lots in a city road-network defined by NetEdit utility of the traffic simulator SUMO [48]. While NetEdit is used to describe road junctions and edges on a map, it receives a variable list of parking lots that are randomly placed on particular junctions to evaluate SmartPark ability to scale its performance within varying parking resource situations.

A parking area is subsequently generated by NetEdit to zoom on a snapshot of a road-network and related infrastructure within a city area of interest, hereby labelled...
as parking area. Any parking lot in the parking area is a valid candidate to park a given vehicle that enters the parking area. This spatial view of the urban core defined by the parking area shapes the road topology in an experimental setting that is to be investigated in a particular vehicular-traffic context. The driver code supplies traffic and parking parameters to evaluate the performance of candidate algorithms under varying traffic-density degrees and parking-lot instances. We captured an actual city area using OpenStreetMap [49] and imported it into NetEdit. Kista is busy hub district of Stockholm with a variety of business and shopping outlets as well as a booming science and innovation cluster. Fig. 10 shows the parking area of interest in this experimental evaluation of the candidate algorithms, as a simplified snapshot of the actual Kista’s road-network. The list of parking lots is incrementally augmented and sparsely distributed across locations in the parking area as featured in Fig. 10. In this experimental study, each parking lot has a maximum capacity of 200 spots, with arrival and departure rates set to 10 and 5 vehicles per minute. A parking lot is deemed full if its occupancy threshold exceeds 80%. The simulation driver injects a varying degree of traffic density to evaluate the performance of candidate algorithms under different traffic considerations. Models used to generate data sets for parking dynamics and traffic flows are described further next.

B. Data sets

To run the experiment, two sources of data are generated, namely parking and traffic data. The first data set simulates the parking dynamics in terms of historical and current activity within a given parking-lot. The second data set models the traffic-flow over the parking-area.

1) Parking dynamics: While historical data could be made available from proprietary parking service providers, in this experiment historical data is artificially crafted as follows. Parking availability data reflect a parking lot state minute by minute over, over a period of 5 minutes. Subsequently, the parking-dynamics process follows a 6-states discrete Markov model State $S_0$ to State $S_5$, where $S_0$ designates the initial state of a parking lot, labelled in Fig. 10 as Initial Parking Availability. The available spots for each state is averaged from 10 instances of historical parking-data simulating 10 similar observation periods. For example, 8:00 AM to 8:05 AM on Mondays. The resulting data set is illustrated in Fig. 11, where Sample$_0$ represents a 5-minutes observation period with minute by minute available spots in a parking lot. In the context of our experiment, the initial availability is randomly generated following a uniform distribution, but upper-bounded by 90% maximum initial availability. Each data sample of Fig. 11 represents the parking dynamics over successive temporal states, where State $S_0$ represents the initial availability state, $S_1$, represents the availability rate after 1 minute, etc.

2) Traffic dynamics: All routes in the parking area network are subject to congestion, and congestion on a route translates into a time delay to traverse it. This delay is simulated by increasing the total number of vehicles in the parking area. The congestion rate represents the proportion of vehicles that is injected into the road network of the parking area. Vehicles enter the parking area at a mean
rate representing the congestion rate parameter used in this experiment. The mean rate is the average number of vehicles that enter the parking area per minute. Given a total-vehicles threshold that can be accommodated by the parking area, the congestion rate in this experiment ranges from 10% to 90% of that threshold. Fig. 12 shows an averaged number of injected vehicles normally distributed around the congestion-rate mean, into the parking area when the total vehicle threshold is set to 100.

![Fig. 12. Traffic dynamics data set](image)

C. Candidate algorithms

SmartPark is compared to a base case algorithm labelled BlindPark which is illustrated in the flowchart of Fig. 13. This base-case algorithm imitates an instinctive parking search procedures drivers commonly adopt by seeking the nearest lot, hoping to reduce driving-time. Since any lot is deemed suitable once entering the parking area, the nearest lot appears a natural choice. However, it may turn out to be full, and thus the path is augmented with a new segment leading to the following nearest lot, starting from the current vehicle location.

![Fig. 13. BlindPark: Base-case parking algorithm](image)

SmartPark and BlindPark are compared with varying parking-lot instances in the parking area, and under varying congestion rate situations. To create a controlled experimental environment, by studying the scalability of the candidate algorithms when parking resources increase on one hand, and the ability to adjust to varying traffic considerations on the other hand, the entry point to parking area is fixed.

D. Performance metrics

Two prominent metrics are targeted in the proposed experimental setup. The failure rate and arrival time. The failure rate reflects the blocking probability, which is the chance that a vehicle finds a designated lot, by the candidate algorithm full. This metric reveals the rate by which the algorithm fails to lead vehicles to a parking lot with available spots. The parking dynamics model described earlier, is used to determine the available spots in the designated lot. Subsequently, the blocking probability or failure rate for each parking lot is obtained from the availability rate, as follows:

\[
\text{FailureRate} = 1 - \text{AvailabilityRate} = 1 - \frac{\text{VacantSpots}}{\text{TotalSpots}}
\]

\text{VacantSpots} is retrieved from the corresponding state at which the parking lot is found upon arrival of the vehicle controlled by the candidate algorithms, and \text{TotalSpots} is the parking lot capacity.

The arrival-time metric used also in our experiments, measures the time duration a vehicle controlled by candidate algorithms, spends within the parking-area since entry till it reaches an available parking-lot. Vehicles may pass through unavailable parking lots until a successful one is found. A clock is maintained throughout the simulation to pick vehicles’ arrival time to a parking-lot, which availability rate falls below a common threshold, that reflects the lot’s exhaustive use parameter.

E. Experiment results

1) Parking scalability: In this experiment, we investigate the scalability performance of the candidate algorithms to decrease the failure rate when taking advantage of an increasing instances of parking lots in the parking area. The injected traffic in this experiment is fixed at a high rate of 0.8, while parking-lot instances range from 1 to 9. Fig. 14 shows the average result of 20 simulation runs, with error-bars showing the deviation of the sample means. The figure shows that SmartPark takes better advantage of the available lots in the parking area, scaling down gracefully the failure rate as more parking lots are provided. The clairvoyant SmartPark outperforms the greedy BlindPark, which routes vehicles naively to the nearest parking lots, while facing consistently high failure probabilities.

2) Routes congestion: This experiment compares SmartPark and BlindPark failure rate performance with a varying degree of congestion rates in the parking area. Hence, the performance metric is still the probability to find an available spot in the designated lot when the vehicle controlled by
the candidate algorithms reach the designated parking lot. In this experiment, the number of parking lots is fixed to a high value of 8 lots in the parking area. Fig. 15 shows the average result of 20 simulation runs, with error-bars showing the deviation of the sample means. The figure shows that SmartPark is not sensitive to congestion fluctuations due to its ability to use real-time traffic data from sensors across smart-poles in the parking area junctions when planning a route to an available parking lot. BlindPark on the other hand, is unaware about the traffic situation, driving cars to the nearest but highly occupied lots.

3) Arrival time: Another metric of interest is the arrival-time to parking-lots. While varying the instances of parking lots, SmartPark’s quest for the “best” parking lot with higher availability-rate, incurs a lower arrival-time cost as illustrated in Fig. 16. The figure reveals the arrival times in seconds since entering the parking area, while the dots in the curves show the failure-rate of the parking lot, once the vehicle arrives there. SmartPark consistently leads vehicles to a parking lot with lower failure rates, yet those vehicles arrive earlier than the ones led by BlindPark to promised lots. With only one parking lot in the parking area, both algorithms have no choice but to lead vehicles to the same lot, but the arrival-times gap expands as more parking lots are made available. In this experiment, the congestion rate is fixed at a high value of 0.8, which allows parking dynamics to fluctuate into blocking states faced by BlindPark roaming around full parking lots. Eventually, as parking resources increase, chances to blindly fall into an available parking lot increase too, allowing BlindPark to lower arrival times, after a peak of about 20 minutes delay difference compared to SmartPark. The arrival times gap narrows as further lots are provided, yet SmartPark always keeps a lower failure rate, which means even-though arrival times converge with increasing parking lots, the probability to find available spots in a lot chosen by SmartPark is always higher than BlindPark. SmartPark cars consistently arrive earlier to the designated lot, yet with a lower failure rate (at most 0.42), whereas BlindPark failure rate appears to be consistently higher than SmartPark counterparts. The figure shows the average result of 20 runs, with error-bars showing the deviation of the sample means. BlindPark error-bars are much wider than SmartPark ones, which indicate the degree of arrival-times variation is much higher across multiple simulation runs for BlindPark, whereas SmartPark appears to be more predictive ensuring the arrival-time remains within a narrower interval.

Another dimension of the arrival time is the influence of congestion rates. Fig. 17 shows that the arrival times generally increase within an increasingly congested parking area. Traffic congestion data are captured by real-time sensors in junction smart-poles and used by SmartPark search algorithm to derive an optimal parking lot. SmartPark’s integration of traffic data trades routing and parking-allocation problems in a way that vehicles reach a highly promising parking lot with failure rates ranging from 0.34 to 0.41. However BlindPark vehicles arrive later to parking...
lots with failure rates ranging from 0.5 to 0.65. While parking-availability chances are always in the advantage of SmartPark-led vehicles, the arrival-time gap narrows further with congestion rates, making it difficult for both algorithms to move vehicles at a higher-speed as the parking area becomes packed.

![Fig. 17. Arrival time to parking lot with varying congestion rates while parking instances is fixed at 8](image)

**VII. CONCLUSION**

We described an approach to the parking selection and related navigation problems, considering real-time traffic situation within a delimited scope of a busy CBD area. Entering vehicles within the area controlled by the proposed SmartPark algorithm and requesting a parking-spot trigger the search process which first estimates the arrival time to all parking-lots within the area. The estimated arrival times are used to predict the parking-lot states using a discrete Markov-chain model that utilizes historical data from past and similar seasonal instances. Using that prediction model, we project the current parking-state to infer the future parking-state considering the expected arrival time to each parking lot. The best parking lot is then picked, followed by a routing process which makes use of smart-pole data collected earlier. The performance results applied to an actual busy area of Stockholm CBD show improved parking-availability expectations when using our proposed SmartPark algorithm compared to intuitive BlindPark that consistently attempts the nearest parking lot. The simulation results collected from standard traffic-simulation APIs, show also higher scalability of parking-resource utilization, and lower cruising-time in favour of SmartPark.

Future research considers incorporating machine-learning approaches to further learn from newly emerging data streams enabled by IoT immersion into urban landscapes to optimize further parking-utility, while incorporating additional criteria, such as user-profiled preferences.

**ACKNOWLEDGEMENT**

This work has been funded by Vinnova-FFI, as part of the research project on "sustainable SmartParking management for connected and autonomous vehicles" Dnr 2017-03028.

**REFERENCES**


[45] Parkopedia.


