An Efficient Cooperative Parking Slot Assignment Solution

Naourez Mejri, Mouna Ayari, Farouk Kamoun
RAMSIS, CRISTAL Lab.
University of Manouba
Manouba, Tunisia
{naourez.mejri; mouna.ayari}@cristal.rnu.tn; frk.kamoun@planet.tn

***Abstract*—Finding a vacant parking place is one of the major concerns of drivers on the road. Hence, good parking management policies are required to efficiently assign solicited parking places to drivers especially in highly solicited urban environments. In this context, we propose and study an efficient semi-centralized parking slot assignment approach with two variants: without/with complete knowledge. In our proposal, each parking lot, in a given urban zone, is monitored by a local authority entity called Parking Coordinator (PC). Its main task is to process received requests from vehicles and offer them accurate parking slots. This selection takes into account the preferences of each vehicle as specified in its request. Through comparing the two variants, we investigate the impact of communication (or lack of it) between these parking coordinators on the good distribution of assigned parking places and the requests rate satisfaction. We study the efficiency of the proposed schemas in various contexts using the mathematical programming solver for linear programming CPLEX and compare them with the centralized approach. Results show, on one hand, that the centralized approach provides the highest rate of request satisfaction. However, as known, this approach heavily suffers from the scalability problem. On the other hand, experiments show that the solution with complete information outperforms the semi-centralized one and its performances matches those obtained in the centralized solution.**

Keywords-coordination; parking slot assignment; vehicles; service-based systems

I. INTRODUCTION

In densely populated urban areas, parking is one of the non negligible causes of congestion and travel delays. In fact, searching for a vacant parking place can become a time consuming and frustrating task for drivers in a hurry. Hence, good parking management policies are required and efficiently assigning solicited parking places to drivers is a priority. In this paper, we propose an new efficient and practical approach to guide drivers to parking. We aim to guarantee drivers satisfaction with the parking assignment and to improve the fairness among parking zones by balancing their occupancy-load. The idea behind our proposal is to use Parking Coordinators (PC) for each lot of parking places. These authorities are responsible for assigning the empty spots based on the vehicles' demands and drivers' preferences. Drivers looking for parking spots send requests over the Vehicular Ad hoc NETwork (VANET). Each request specifies among others a parking place location at a preferred distance from the driver's final destination. The PCs collect the requests of the drivers over a certain time window and assign a free parking slot to each one. We distinguish two variants of our proposed semi-centralized strategy for parking slot assignment: without and with complete knowledge. In the first variant, each PC acts independently and makes its decision without caring about other authorities decisions. However, in the second variant, each PC considers its surrounding PCs possible alternatives.

We compare our proposal with the centralized strategy and we investigate the impact of communication between PCs on the accuracy and efficiency of the parking assignment. We notice that the main contribution of our paper is a scalable localized solution that assigns parking places based on drivers' preferences and ensures a good distribution of available parking places. This task can be fulfilled before vehicles reach the parking zone. Hence, this considerably reduces waiting and search time. The remainder of this paper is organized as follows: Section II discusses related work addressing parking slots assignment. In Section III, we introduce the system model and outline the main assumptions and goals on which our solution is designed. We describe the two variants of our proposal in Section IV and compare their performances with the centralized solution in Section V. Finally, the last section concludes the paper.

II. RELATED WORK

Last years, the parking slot assignment problem has received particular attention in both industrial and academic levels. Different smart applications have been developed in order to assist drivers in their parking search. Proposed solutions can be classified in two big categories: parking solutions with infrastructure assistance and parking solutions relying on estimated/predicted information. In the first category, existing or added facilities gather accurate data about parking occupancy and capacities in order for drivers to find efficiently their parking places. Whereas, in the second category, such privilege no longer exists and the status of the parking places is either predicted or estimated with other methods. [1][2][3][4] and [5] are examples of

 ISBN: 978-1-61208-289-9

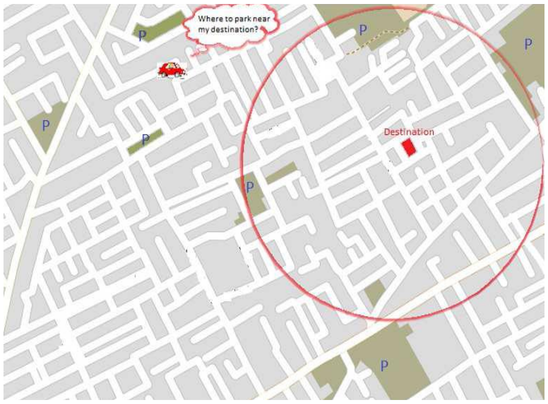

Figure 1. A vehicle looking for a parking within its zone of interest

solutions with infrastructure assistance. In SF-Park [1] and SmartParking [2], each parking spot is equipped with a fixed sensor to determine its occupancy/freeness. The infrastructure then advertises the available spots and manages their reservation. Moreover, a penalty mechanism is proposed in [2] to ensure that vehicles respect their assigned spots. However, these solutions require a large cost in order to adequately monitor the parking spaces even at the level of a downtown area. Hence, for more scalability, some parking garages manage their set of parking places with in/out counters at the entry and exit points to count the number of additional vehicles they can accommodate at any given time.

Such is the case of IrisNet [3] where web cameras are used to monitor individual parking spaces thus allowing users to query the system for vacant spaces on a web front-end. The same concept is used in the SPARK scheme [4]. Road Side Units (RSUs) were installed across a parking lot in a manner they could supervise the whole parking lot. They provide drivers with services such as real-time parking navigation, intelligent anti-theft protection and friendly parking information dissemination. Nevertheless, since such solutions are only valid on closed parking spaces and cannot be applied to curbside parking places, authors in ParkNet [5] proposed reducing the number of infrastructure/sensors required by equipping some special vehicles (such as cabs), instead of the parking places, with ultrasonic sensors to determine and reserve vacant places even in isolated areas of the road. However, the precision of these ultrasonic devices isn't very accurate and the solution concept requires the designated vehicles to permanently monitor the road state to check for parking availability. The solutions presented in [6] and [7] are examples of solutions based on predictability. In [6], although Caliskan et al. use the parking lot information disseminated by the parking automats, they rely more on the inter-vehicle broadcasts to allow the drivers find their preferred free parking lot, whereas in [7], each vehicle which leaves its parking place, becomes its coordinator. After collecting information among interested neighbors, it decides with which vehicle to share the parking coordinates. This process aims to reduce competition between vehicles in search for parking where only the elected vehicle knows the parking place exact location. However, this solution suffers from scalability since this process need to be iterated repeatedly for each freed parking place. Besides, it doesn't address how free parking places are being assigned at the initial process. We note here that other parking assignment solutions exist, however they assume that the information about parking occupancy is already available or that all the parking places are initially vacant and open for competition. They just worry about how to efficiently assign the parking places. An example of these solutions can be found in [8], where a parking slot assignment game was proposed with the vehicles being the rival players. Our proposal belongs to the second category since we aimed for a scalable solution without depending on existing infrastructure to ensure its compliancy with several topologies.

In our proposal, we aim at providing vehicles with parking spots in advance. In fact, in order to gain time, a vehicle can emit its request for a parking place in a specific region before reaching it. By the time it arrives, it will find a spot already reserved to it. Nevertheless, we note here that a vehicle can ignore this in advance request-answer mechanism and specify its desire for a parking place only when it reaches the parking garage (which can be accomplished through an online application).

III. SYSTEM MODEL AND DESIGN GOAL

A. System Model

Our network is composed of:

- A set $V = \{v_1, v_2, .. , v_N\}$ of N vehicles. Each vehicle v_i is looking for a free parking spot within its zone of interest ZI_i. The zone of interest ZI_i is defined as a circle of center the point of interest PI_i of the vehicle (i.e., its final destination) and of radius r_i. The radius value can be dynamically adjusted according to the day time, the network congestion, the geographic location and the willingness of the driver to walk once parked. This zone delimits the location of the parking places that can be assigned to the driver. It aims for him to be close enough of his destination and hence do not incur further time loss due to extra walking (see Fig. 1).
- A set $S = \{s_1, s_2, .. , s_M\}$ of M vacant parking slots.
- A set $PC = \{PC_1, PC_2, .. ,PC_L\}$ of L parking coordinators (PC). Each PC_k is a local authority managing a set SPC_k of size M_k. It consists of non overlapping slots of M such as:

$$\sum_{k=1}^{L} M_k = M \tag{1}$$

These PCs are responsible of allotting the empty parking places under them to the vehicles while optimizing their own social welfare i.e., the cost/the benefit induced by this parking slot assignment.

We assume that all received vehicles requests are collected over a periodic time window and processed periodically in the same time. Moreover, we assume that each PC knows about the locations of the other PCs. Hence, it can determine the requests simultaneously processed by it and them. Such information can be easily obtained through an up-to-date topology map or dynamically by periodic broadcast through the existing infrastructure.

B. *Design Goal*

Our main goal is to provide drivers with available parking spots in advance. Hence, each vehicle can express its desire for a parking place in a request issued beforehand. This request contains information about the preferences of the driver. It includes its point of interest coordinates and the radius of its zone of interest. Moreover, the driver specifies its current location at the moment of the request preparation. Such information is used when processing requests to give priorities to approaching vehicles. While assigning slots to vehicles, each parking coordinator has to make sure that this assignment meets the following criteria:

- A spot can be assigned only once to a vehicle. It stays unavailable until the parked vehicle leaves it. Each spot offered to a vehicle v_i has to lay within its zone of interest ZI_i.
- Each PC aims to satisfy the maximum number of requests for parking places in the limit of the available ones.

IV. SEMI CENTRALIZED PARKING ASSIGNMENT SOLUTION

In the following section, we present both variants of our proposal and explain how both solutions work.

A. Semi-centralized solution without complete knowledge

Periodically, upon reception of requests, each parking coordinator processes the requests that are relevant to it and discards the rest. A request of a vehicle v_i is relevant for the parking coordinator PC_k if it satisfies (2).

$$\exists s_j \in SPC_k \,|\, dist(s_j, PI_i) \leq r_i \qquad (2)$$

The measure dist refers here to the Manhattan distance between two points.

Each parking coordinator PC_k periodically updates an association matrix called A_k where $A(i, j)_k$ is set to one if the request of vehicle v_i can be satisfied by the slot s_j (element of the set SPC_k). $A(i, j)_k$ is set to zero otherwise. Then, the parking coordinator computes for each vacant slot s_j, its solicitation factor. This latter displays how many requests the slot can possibly satisfy. This factor is obtained from (3).

$$F(v_i, s_j) = \alpha \frac{dist(v_i, s_j)}{\sum_{k=1}^{M_k} dist(v_i, s_k)} + \beta \frac{dist(PI_i, s_j)}{\sum_{k=1}^{M_k} dist(PI_i, s_k)} + \gamma \frac{\sum_{i=1}^{N_k} A(i,j)_k}{N_k} \qquad (3)$$

where:

- The factors α,β and γ are variables, for which the values are selected according to whether the distance to the parking spot, the distance to the destination or the less solicited slots are to be privileged.
- The distances $dist(v_i,s_j)$ and $dist(PI_i,s_j)$ are respectively the Manhattan distances between the vehicle v_i and the slot s_j and between the vehicle's destination PI_i and the slot s_j . Both distances are normalized by the sum of distances among the whole slots belonging to PC_k.
- N_k is the total number of requests relevant to PC_k.

We stress here that although in real life, the Manhattan distance can be different from the real driving distance; it is not easy to determine the latter in advance unless the vehicle's trajectory is predetermined in advance. Besides, it is obvious that the driver can change his mind anytime depending on traffic/road and weather conditions. Finally, in our simulations, we used a grid model where the Manhattan distance matches well the real distance. Our objective can then be modeled as an integer linear program optimization for each PC_k as shown in (4).

$$\text{minimize} \sum_{i=1}^{N_k} \sum_{j=1}^{M_k} y_{i,j}\, F(v_i, s_j) \qquad (4)$$

subject to

- $dist(PI_i,s_j) \leq R_i$
- $y(i,j) = \begin{cases} 1 & \text{if vehicle } v_i \text{ was assigned to spot } s_j \\ 0 & \text{otherwise} \end{cases}$
- α,β and $\alpha \geq 0 \mid \alpha+\beta+\gamma=1$

As proven in [8], a system optimal solution can be computed for such problem in a (strongly) polynomial time. In this variant, each PC's decision is made independently of the other PCs decisions. This can obviously result in one request being answered more than once by different PCs while other vehicles can be left with no response to their requests. In order to remedy these possible solution defects, we opted in the second variant toward a solution where PCs are aware of each other.

The following section explains how this is accomplished.

 ISBN: 978-1-61208-289-9

B. Semi-centralized solution with complete knowledge

In this variant, the parking coordinators will share with each other information about the availability/occupancy of their parking places. Moreover, each PC will keep track of the set of relevant requests for the other PCs. On that account, each PC_k fills a request-coordinator correspondence matrix $Corr_k$ with L columns (referring to the total number of PCs and r rows (referring to the total number of requests being processed at the current period) as shown in (5).

$$\mathrm{Corr}_x(i,j) = \begin{cases} 1, & \text{if } PC_j \in ZI_i \\ 0, & \text{otherwise} \end{cases} \qquad (5)$$

From the obtained matrix $Corr_k$, PC_k can extract two indicators:

- A coverage indicator Cov_i for each request req_i representing the number of coordinators within the zone of interest of vehicle v_i and eligible to answer its request, it is computed from (6).

$$\forall i \in \{1,..,r\},\ Cov_i = \sum_{j=1}^{L} Corr_k(i,j) \qquad (6)$$

- A solicitation ratio Sol_j of each parking place coordinator PC_j indicating the number of relevant requests for the parking coordinator PC_j as shown in (7).

$$\forall j \in \{1,..,L\},\ Sol_j = \sum_{i=1}^{r} Corr_k(i,j) \qquad (7)$$

Each PC_k will then compute, for each request req_i, the cost of positively answering that request based on (8).

$$U_k(req_i) = \alpha \frac{dist(v_i, PC_k)}{\sum_{j/Corr_k(i,j)=1} dist(v_i, PC_j)} + \beta \frac{dist(PC_k, PI_i)}{\sum_{j/Corr_k(i,j)=1} dist(PC_j, PI_i)} + \gamma RFC_k \qquad (8)$$

where $RFC_k = Empty_k/Sol_k$ is a ratio indicating the capacity of the parking coordinator PC_k to fulfill the received requests. $Empty_k$ refers to the number of vacant parking places for PC_k. A ratio $RFC_k > 1$ indicates that there are enough vacant places to answer positively all the received requests.

For each request req_i, if the cost satisfies the condition stated in (9), then PC_k offers a parking slot to the vehicle v_i. Otherwise, it discards the request and processes the next one. This process goes on until all the relevant requests are satisfied or there are no more vacant parking slots.

$$U_k(req_i) = \underset{x \in L, PC_x \in ZI_i}{\mathrm{argmin}} \{U_x(req_i)\} \qquad (9)$$

Once the PCs decide on the vehicles to which the spots are going to be assigned, a reservation mechanism will start between them and the vehicles in question. At the end of it, each spot will be reserved for a single vehicle. If by chance, the latter decline the offer, the spot will still be considered vacant and reassigned to the next prior vehicle. The reservation mechanism is beyond the scope of this paper and will be detailed in future work.

Also, we stress here that we assume all vehicles in the zone are using the designed scheme. Problems might occur if other vehicles in the area decide to act on their own and might take a spot already reserved to another vehicle. To reduce the magnitude of such problem, a penalty mechanism like the one presented in [2] can be adopted. If the infringing vehicle still ignores the penalty fee and occupies the spot, the PC will have to emergently assign another spot to the vehicle which originally reserved it. It has to forward it to the closest free spot among its own, if there are any available. In the negative case, it will check among its neighbors to see if any of them can offer it a free spot. Then, it forwards the new location of the parking spot to the vehicle in question.

V. PERFORMANCE EVALUATION

In this section, we present the simulation environment and describe the used scenario. Then, we analyze the obtained results of our variants. We note here, that the two variants of our solution were compared with the centralized model of reference. In this model, a single centralized authority assigns slots to vehicles. It aims to minimize the total network cost while satisfying the maximum number of requests with consideration of the parking places capacity. It abides to the same constraints listed in the previous section.

A. Simulation Environment

Our parking assignment problem can be seen as a variant of the reference task assignment problem. Thus, we ran our simulations by using the linear programming tool CPLEX [10]. The data files with the constraints were exported from MATLAB [11]. We chose our network topology as a grid of 500*500m on which respectively 100,150 and 200 vehicles are looking for parking places. The total number of available slots was fixed to 200 slots. We note here that we were unable to increase the number of vehicles and slots because of MATLAB and CPLEX computation limitations. However, our choices are realistic and match vehicles' density in some countries with a density varying from 20 vehicles per km of road (in countries were vehicles are affluent transport means) to 10 vehicles per km of road (in less crowded countries). The reader can refer to [9] for more details about the classification of countries by their vehicles' density per km of road. The positions of vehicles were randomly generated over the grid by using SUMO [10] to have realistic dispersion of vehicles over the network. The vehicles randomly aim for one of the 4 interest points. These interest points were dispersed randomly each in one of the network quarters which we refer to as regions. Hence, the number of considered regions is $nr = 4$.

Each region is managed by a single parking coordinator, which manages 50 of the total 200 slots.

 ISBN: 978-1-61208-289-9

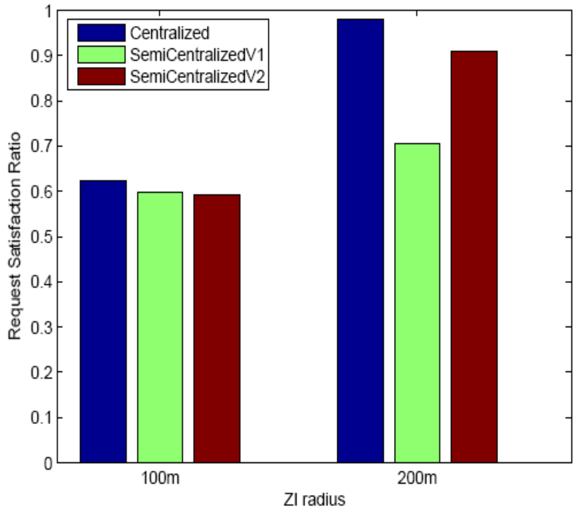

Figure 2. Impact of the variation of ZI radius on the Request Satisfaction Ratio

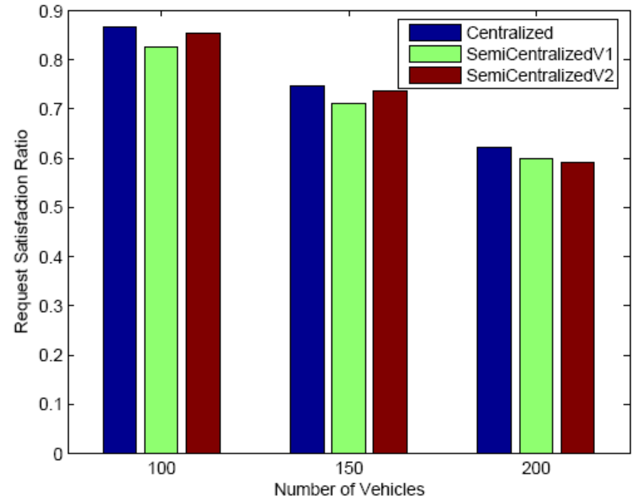

Figure 3. Impact of the vehicles' density on the request satisfaction ratio

B. Evaluation parameters

In order to evaluate the performance of our two proposals, we used the following parameters as performance indicators:

- The request satisfaction ratio: it is the quotient of the total answered requests divided by the total number of received requests.
- The fairness index: it indicates whether the slots are fairly shared inside the network regions among the vehicles. It is equivalent to Jain's Fairness Index introduced in [12] and is obtained from (10).

$$\text{FairnessIndex}=\left(\sum_{i=1}^{nr} x_i\right)^2 \Big/ \left(nr^* \sum_{i=1}^{nr} x_i^{\,2}\right) \qquad (10)$$

where nr is the network regions, and x_i=nbReqAnsReg$_i$/nbReqRecReg$_i$ is the fraction of answered requests with slots in region i over the total number of requests initially aiming for a point of interest inside the region i.

- The variance: it measures how far a set of numbers is spread out. It is computed by the formula in (11), where the expression of the variable x_i is the same as defined in the fairness index.

$$\text{Variance}=\left[\sum_{i=1}^{nr} x_i^{\,2} - \left(\sum_{i=1}^{nr} x_i\right)^2\right]/nr \qquad (11)$$

C. Simulation Results

The values given in the following section are the average values over multiple simulations ran for each point with a randomly generated position of both the slots and the point of interests. The vehicles' choice of the interest point to aim for is also randomly varied in each of the simulations.

1) Impact of varying the radius of the zone of interest:

First, we investigated the impact of varying the radius of the zone of interest of vehicles. We varied the radius for the same scenario consisting of 200 vehicles requesting slots in a network initially containing 200 vacant slots. We can see from Fig. 2 that increasing the radius from 100 meters to 200 meters increases the request satisfaction ratio. In fact, it increases the number of slots inside the zone of interest and hence more requests can be satisfied. We remark also that when the radius is of 100 meters, results given by both solutions are pretty similar. However, when increasing the radius to 200 meters, the second variant becomes very efficient and its satisfaction ratio is very closer to the centralized solution with almost 90% of the requests satisfied. In opposition, the first variant has a lower percentage of only 70 %. This difference between the two approaches is due to the fact that PCs act independently in the first variant. This results in having the same vehicle receiving several slots' offers while other vehicles' requests are left unanswered. Each vehicle getting more than one proposal will eventually choose the closest slot to it among them and reject the other offers. Thus, the network will end up with unassigned slots, which could have satisfied other vehicles' requests. In the second variant, the problem is solved since a parking coordinator would not offer a slot to a vehicle if it knows that another parking coordinator has a better offer. Henceforth, the request satisfaction ratio is higher for the second variant.

 ISBN: 978-1-61208-289-9

TABLE I. IMPACT OF VARYING THE RADIUS OF THE ZONE OF INTEREST

ZI Radius	***100m***			***200m***		
Solution	***Centralized***	***SemiCentralizedV1***	***SemiCentralizedV2***	***Centralized***	***SemiCentralizedV1***	***SemiCentralizedV2***
Fairness Index	0.977	0.973	0.838	0.984	0.977	0.974
Variance	0.014	0.0154	0.088	0.023	0.017	0.029

TABLE II. IMPACT OF VARYING THE VALUES OF THE FACTORS Α,Β AND Γ

α	1	0	0	0.5	0.25	0
β	0	1	0	0.5	0.25	0.5
γ	0	0	1	0	0.5	0.5
Request Satisfaction Ratio	0.7362	0.7364	0.7358	0.7369	0.7369	0.7364
Fairness Index	0.8628	0.8626	0.8632	0.8625	0.8625	0.8626
Variance	0.1045	0.1048	0.1036	0.1050	0.1050	0.1048

Increasing the radius of the zone of interest, increases the conflict between the parking coordinators and thus the strength of the second variant results over the first one becomes clearer.

Also, as shown in Table I, we remark that all three solutions give a fairness index values close to 1. This indicates a fair distribution of the resources (here the parking slots) among the vehicles. The low values of the variance close to zero, support the same conclusion. In fact, in the three solutions, the authorities (whether central or local) try to optimize the social welfare while ensuring a good distribution of the requests among the available slots with respect to the constraints.

2) Impact of varying the number of vehicles:

Next, we studied the impact of varying the number of vehicles on their requests' satisfaction ratio. For the same number of slots, and a radius of 100 meters, we varied the number of vehicles between 100, 150 and 200 vehicles to simulate respectively low, medium and high density. The obtained results are shown in Fig. 3. Obviously, a better request satisfaction ratio is obtained when the density is lower (the case for all solutions). In fact, when the number of vehicles is lower than the available vacant spots, the authorities are capable of satisfying most of the received requests that meet the constraints stated above. We also found that for the three densities, the request satisfaction ratio for the second variant of the semi centralized approach are slightly better than those of the first variant and hence tend toward the reference solution, i.e., the centralized one. This highlights the positive effect of exchanging information between the parking coordinators to make their decisions more efficient and avoid assigning several slots to the same vehicle while leaving others without any.

3) Impact of varying the values of the factors α, β and γ:

Finally, as described in Table II, we measured our performance parameters for different values of α, β and γ. The obtained results are almost similar to each other for the different values. This is probably due to two main factors:

- First, the value of the function as described in (1) is obtained by normalizing all the values, which highly reduces the differences between each fraction.
- Moreover, the size of our grid is not very big to show big difference between the distances between vehicles and the slots from one hand and the distance between the vehicles' point of interest and the slots from the other hand.

However, as we stated above, our choices were limited by capacities of MATLAB and CPLEX and chosen in order to have a realistic scenario as much as possible. A bigger scenario will probably show more clearly the impact of varying these factors.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a solution for parking slot assignment. Two variants of the solution were studied. In the first one, the parking coordinators decided independently of their surroundings. In the second variation, these PCs exchanged information with their neighboring ones and tuned their decisions accordingly. Simulation results showed that the second variant outperformed the first one especially when the number of conflicting requests between PCs increased. The results obtained for the second proposal were very close to the reference centralized solution while being more scalable than it. Preliminary results proved that cooperation between parking coordinators is highly recommended and its impact will be further studied in future work. Moreover, we still need to carry out further experiments to fine tune the values of α, β and γ parameters in order to improve the performance of our scheme. Furthermore, in the presented results only one single period (i.e., the time window during which PCs process received requests) was considered. However, the length of such period can highly affect the amount of requests processed. A short period will incur unnecessary load on the PCs whereas a relatively long period can affect the efficiency of the proposal since some requests might become obsolete by the time they are answered. Hence, the time window duration needs extensive simulation as to determine its best value. We note also that, in the case of the first proposed variant, the filling of the matrices would probably incur a high load on the network. Thus, we investigate to study the overhead induced by the proposed mechanisms. Finally, we plan to study the impact of learning mechanisms on the enhancement of our solution.

REFERENCES

[1] http://www.sfpark.org, [retrieved : July, 2013].

[2] G. Yan, W. Yang, D. B. Rawat, and S. Olariu, “Smartparking: a secure and intelligent parking system using notice”, In proceedings of the international IEEE Conference on intelligent transportation systems. Beijing, China, October 2008, pp. 569-574.

[3] J. Campbell, P. B. Gibbons, and S. Nathand, “Irisnet: an internet-scale architecture for multimedia sensors”, 13th annual ACM international conference on Multimedia(Multimedia’05), Singapore, 2005, pp. 81-88.

M. Caliskan, D. Graupner, and M. Mauve, “Decentralized discovery of free parking places”, 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET’06), Los Angeles, CA, USA, September 2006, pp. 30-39.

[4] S. Mathur and al., “Parknet: drive-by sensing of road-side parking statistics”,8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys 2010), San Francisco, USA, June 2010, pp. 123-136, doi:10.1145/1814433.1814448.

[5] T. Delot, N. Cenerario, S. Ilarri, and S. Lecomte, “A cooperative reservation protocol for parking spaces in vehicular ad hoc networks”, 6th International Conference on Mobile Technology, Applications and Systems, ACM Press Library, Nice, France, September 2009, pp. 1-8, doi:10.1145/1710035.1710065.

[6] R. Lu, X. Lin, H. Zhu, and X. Shen, “Spark: a new vanet-based smart parking scheme for large parking lots”, 28th Conference on Computer Communications (Infocom 2009), Rio de Janeiro, Brazil April 2009, pp. 1413-1421.

[7] D. Ayala, O. Wolfson, B. Xu, and B. Dasgupta, “Parking slot assignment games”, 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACMGIS 2011), Chicago, USA, November 2011, pp. 299-308, doi:10.1145/2093973.2094014.

[8] http://data.worldbank.org/indicator/IS.VEH.ROAD.K1, [retrieved : July, 2013].

[9] http://sumo.sourceforge.net/ , [retrieved : July, 2013].

[10] www.ampl.com/CPLEX, [retrieved : July, 2013].

[11] www.mathworks.com/products/matlab/, [retrieved : July, 2013].

[12] R. Jain, D. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems”, DEC Research Report TR-301, September 1984.

 ISBN: 978-1-61208-289-9