Smart Parking Reservation System Based on Distributed Multicriteria Approach

Imen Boudali & Meriam Ben Ouada

Published online: 31 Oct 2017.

Submit your article to this journal ⬚

View related articles ⬚

View Crossmark data ⬚

Taylor & Francis
Taylor & Francis Group

Check for updates

Smart Parking Reservation System Based on Distributed Multicriteria Approach

Imen Boudali [iD][a] and Meriam Ben Ouada[b]

[a]Department of Information and Communication Technologies (ICT), National Engineering School of Tunis (ENIT), University of Tunis ElManar (UTM), Tunisia; [b]Department of Software Engineering and Information Systems, High Institute of Computer Sciences – ISI, University of Tunis ElManar (UTM), Tunisia

ABSTRACT

In metropolitan area, finding a parking space is a difficult task for drivers especially in rush hours. This causes waste of time and fuel and results in traffic congestion. We propose a smart parking system based on multi-agent approach to provide a real-time decision-aid for drivers by handling their preferences. The system ensures an online space allocation based on real-time information by optimizing drivers' preferences with respect to operational constraints. The online allocation problem is considered as a multicriteria decision-aid problem for which we present a mathematical formulation: multicriteria parking reservation problem. The solution is an optimal compromise from the set of efficient solutions which is determined by means of a multicriteria ranking method ELECTRE III. An update of resource allocation is performed to avoid reservation conflict and to ensure constraints satisfaction. After the reservation process, the driver is assisted via a guidance module to reach the reserved place through the shortest path. Simulation results show the wide applicability of the approach in real cases.

Introduction

With the population growth in different metropolises, the parking problem has aroused most interest during the last decade given its socioeconomic impacts. In fact, for convenience and comfort, most of people prefer travelling in their vehicles rather than public transport. The search for a parking space is a difficult task for drivers especially during rush hours. Actually, finding a vacant parking space is not only a time consuming but also affects the environment by the important carbonic gas emission, the social relations, costs and the traffic congestion. Therefore, there is a critical need for an intelligent, efficient, and reliable support system that can be employed for finding the parking vacancy and for guidance toward the space. Such systems

CONTACT Imen Boudali ✉ imen.boudali@gmail.com; imen.boudali@isi.utm.tn ⊡ Department of Information and Communication Technologies (ICT), National Engineering School of Tunis (ENIT), University of Tunis ElManar (UTM), B.P. 14, Menzah 6, Ariana 2091, Tunisia.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/UAAI.

are often known as parking guidance information (PGI) system, which are a part of intelligent transportation systems (ITS).

PGI systems are based on telecommunication and information technologies, through the use of sensors which are located at each parking space. The current PGI systems simply acquire the availability information of parking spaces from deployed sensor networks, and only publish the parking information to direct drivers (Wang and He 2011), to road panels (Srikanth et al. 2009), or dispatch it through the Internet. Although current PGI systems enhance the probability of finding vacant parking space, they present several shortcomings. First, they cannot assist drivers to reach their parking destination with guaranteed availability of the space. So, drivers may not really find vacant parking space by following guidance. In addition, when the number of vacant spaces in an area is limited, most of drivers who get the parking information are heading for these spaces. Therefore, the "multiple-car-chasing-single-space" phenomenon may probably occur and serious traffic congestion could be caused (Wang and He 2011). So, we notice that the driver behavior is changed from searching to competing for parking. Furthermore, these systems help find any available parking space to the detriment of missing the opportunity for a better space (Geng and Cassandras 2011). In fact, the preferences of drivers in terms of distances, time, and parking fees are not significantly considered. In addition, parking space utilization is generally not optimized, since some spaces are more utilized than others and space residues are typically not addressed.

In the present paper, we propose a smart parking system for space reservation and guidance in response to parking requests of drivers. Our system is based on multi-agent paradigm which is considered as one powerful technology for the development of large-scale distributed systems and which deals effectively with uncertainty in a dynamic environment.

Our system holds two modules. The first one is reservation module which is in charge of space reservation according to drivers' preferences and by considering some exploitation objectives. During the reservation process, an optimal parking space has to be found out among the available spaces by optimizing a number of reservation criteria representing exploitation objectives and drivers' preferences. We notice that a number of operational constraints such as bound of parking fees, bounds of distances, time interval have to be also satisfied for each reservation.

Given this multi-criteria aspect of the reservation problem, we designate it as a multicriteria parking reservation problem (MPRP), for which we propose here a mathematical formulation. At each user request, a set of compromise solutions is generated by optimizing the different criteria separately and simultaneously on the base of Pareto-optimality and non-dominance concepts. Then, the best compromise solution is determined by considering a

ranking method, ELECTRE III. So, at the end of this first step, an optimal parking space is reserved for the driver for a prefixed period of time.

The second module is responsible for a real-time guidance of drivers toward the selected space. In this module, the shortest path to the reserved space is determined via the Dijkstra algorithm.

Literature review

In this section, we study the existing parking reservation approaches and present their limitations. Blind search is the basic strategy used by drivers when there is no parking information. So, drivers keep cruising for parking spaces near their destination until finding an available space. Otherwise, the searching area will be further extended (Wang and He 2011). Hence, the blind search system is an open loop strategy, where drivers make decisions without looking at the state of the system.

The following strategy is the parking information sharing (PIS) which is based on publishing the parking availability information to drivers in a specific area (Lu et al. 2009). So, the drivers make decisions based on the system state such as the parking availability information. However, when the number of vacant spaces is very limited, especially in rush hours, the phenomenon of multiple-car-chasing-single space will occur and will cause serious traffic congestion.

In order to overcome the limitations of the previous strategies, other approaches have been developed for an intelligent parking system. We distinguish two categories: centralized approaches and distributed approaches. For the first category, few works have been proposed. We mainly mention the contribution of Geng and Cassandras (2011), who propose a smart parking system based on a dynamic resource allocation. In their approach, the parking process is viewed as a sequence of mixed integer linear programming (MILP) problems solved over time at specific decision points. The space assignment is based on an allocation center and queue mechanism (waiting queue and reserve queue). Hence, the center collects all driver requests over a certain time window and makes an overall allocation at decision points by optimizing a combination of proximity to destination with parking cost. In the same category, we can also find the works of Wang and He (2011) who designed a prototype of reservation-based smart parking system (RSPS) in order to broadcast real-time parking information (occupancy, prices) to the drivers and to provide reservation service as part of user-target service. RSPS is built on advanced sensing and mobile communication techniques to process streams of time-stamped sensing data from sensor network in parking lot and published parking availability information. The drivers can retrieve parking information and reserve their desired vacant spaces via

WIFI or Internet. So, the system serves as the centralized decision-making body in a planned economy. It makes all pricing decisions regarding the state of parking spaces and user demands.

In Iordache, Nemtanu, and Cormos (2015), the authors expose an idea of autonomic integrated parking system for smart cities. The authors intend to implement and apply it in the areas of Bucharest.

In a distributed context, many parking systems based on multi-agent paradigm have been suggested. Among these systems, we can mention the InfoStation-based system which offers a car parking locator service provision within a University Campus area (Ganchev, O'Droma, and Meere 2008). This service allows registered users that are equipped with mobile wireless devices, to locate available parking spaces throughout the campus.

Another multi-agent system known as agent-based intelligent parking negotiation and guidance system (ABIPNGS) was proposed by Chou, Lin, and Li (2008). The system acts as a bargaining platform between park and drivers, which facilitates the search for available parking spaces, dynamic negotiation of parking fees, reservation of parking spaces, and the derivation of optimal paths from the current location to the intended car park as well as from the car park to the final destination. This approach presents some shortcomings due to the increasing number of exchanged messages during the negotiation process as well as the possibility of non-convergence of the negotiation toward a viable solution. In order to overcome the first weakness, the system was extended in Longfei, Hong, and Yang (2009) by integrating mobile agents. This technology helps reduce data transmission over a network and enhance flexibility, adaptability, and stability to the original multi-agent system.

Bessghaier, Zargayouna, and Balbo (2012) proposed a decentralized approach based on multi-agent system by employing an inter-vehicular communication (V2V) which allows vehicles to receive and broadcast information on space availability to the other vehicles of the same community. The system works without prior information on the parking spots and without central storage of information. In fact, vehicles of the same district exchange messages including the list of occupied and free spots at each parking allocation and each discharge to update their local knowledge.

Then, each vehicle determines the cost of each available space in terms of distance from the current position in order to select the best one. The main weakness of this approach is the scalability especially when many vehicles are present in a district and are looking for available spots. Another major limitation is the probable allocation conflict of spots since no central information is available.

Discussion

Although these current centralized and distributed approaches increase the probability of finding vacant parking spots, they present major limitations. In fact, the guidance toward the reserved space is often neglected (Geng and Cassandras 2011; Ganchev, O'Droma, and Meere 2008; Wang and He 2011; Bessghaier, Zargayouna, and Balbo 2012). So, drivers may not reach the space within the allowed time interval, since they have no idea on the shortest path. Second, a reservation conflict may happen, especially in a decentralized information storage for the parking availability (Bessghaier, Zargayouna, and Balbo 2012). So, a parking space could be reserved by more than one vehicle at an instant t. Finally, an important feature of the parking problem that was neglected in these works is the multicriteria aspect. In fact, the searching for a space must be performed by optimizing many criteria that are related to drivers' preferences or exploitation objectives of the parking lot. Among these criteria, we can mention the security level of the parking lot, the distance from the current position to the parking space, the distance from the parking space to destination, the space residue resulting from the vehicle parking in a given space, the parking fees, and the waiting time for a confirmed reservation.

In our work, we chose to consider all these criteria in order to optimize the reservation process. In addition, we aim at handling the guidance process to assist the drivers in reaching the reserved space through the shortest path without waste of time.

Given the multicriteria aspect of the parking problem, we chose to designate it as the MPRP for which we propose a mathematical formulation.

Multicriteria reservation problem

Before detailing the proposed formulation for the MPRP, we will first present some notations related to vehicles (or drivers), parks, and parking spaces (or spots). We assume that each locality (or district) contains 0 or n parks. So, each park belongs to only one locality or to the intersection of two localities.

Notation

For the vehicle side, we denote by:

- $V = \{i \mid i = 1, 2, 3, \ldots, N\}$ the set of vehicles (or drivers) with N the number of vehicles having parking requests.
- $Size_i$: the size of vehicle i
- $ReqTime_i$: the request time of vehicle i
- Pos_i: the current position of vehicle i determined by its longitude and its latitude

- $Dest_i$: the destination of vehicle i
- Sec_i: the required security level for vehicle i
- $CMax_i$: the maximum cost that vehicle i could afford
- $ResTime^k_{ij}$: the reservation time of space k in park j by vehicle i
- $DMax1_i$: the maximal distance allowed by vehicle i from its current position to a parking space
- $DMax2_i$: the maximal distance allowed by vehicle i from the parking space to its destination
- $WaiTimeMax_i$: the maximal waiting time for a confirmed reservation since the request launching
- $ResDur_i$: the estimated parking duration for vehicle i
- $State_i$: the state of vehicle i

$$State_i = \begin{cases} p^k_j & \text{if vehicle } i \text{ has reserved the space } k \text{ of park } j \\ 0 & \text{otherwise} \end{cases}$$

For the parking side, we denote by:

- $P = \{p^k_j \mid k = 1, 2, 3, \ldots, K; j = 1, 2, 3, \ldots, J\}$ the set of parking spaces p^k_j, with J the number of parks and K the number of spaces in park j;
- $Sec(P_j)$: security level of park j. So, each spot p^k_j has the same security level as the belonging park P_j. Formally, $Sec(P_j) = Sec(p^k_j)$ for each $p^k_j \in P_j$ with $k = 1, 2, \ldots, K$

$$Sec(P_j) = \begin{cases} 2 \text{ high security level (with security camera and security guard)} \\ 1 \text{ medium security level (with only security guard)} \\ 0 \text{ low security level (without any security measure)} \end{cases}$$

- $ResTime_{max}$: the maximal duration of reservation for a parking space.

  For each parking space, we designate by:
- p^k_j: identifier of space k in park j
- Pos^k_j: position of space p^k_j
- $Size^k_j$: size of space p^k_j, defined by its $width^k_j$ and $length^k_j$
- $State^k_j$: state of space p^k_j

$$State^k_j = \begin{cases} -1 \text{ if space } p^k_j \text{ is occupied} \\ 0 \text{ if the space is free} \\ i \text{ if the space is reserved by vehicle } i \end{cases}$$

- $Sec(p^k_j)$: security level of the p^k_j, such that $Sec(P_j) = Sec(p^k_j)$
- $Cost(p^k_j)$: parking cost of the p^k_j
- x^k_{ij}: the decision variable related to a parking reservation

$$x^k_{ij} = \begin{cases} 1 \text{ if space p}^k_j \text{ is occupied by vehiclei} \\ 0 \text{ otherwise} \end{cases}$$

Reservation criteria

In order to optimize drivers' satisfaction and to enhance the parking management, our system has to consider some objectives that we called reservation criteria. Here we will consider the following criteria: the waiting time for a confirmed reservation (WaiTime), the distance toward the parking space (DToSpace), the distance toward destination (DToDest), the space residue (SpaceRes), and the parking cost (Cost). All of these criteria have to be minimized in the reservation process.

Waiting time

This criterion is relative to the waiting of driver after launching his reservation request. Formally, it is defined by Eq. (1):

$$\text{WaiTime} = \left(\text{ResTime}^k_{ij} - \text{ReqTime}_i\right) \tag{1}$$

Distance toward the reserved place

The drivers often prefer a parking space which is near their current position for time saving. So, optimizing this criterion corresponds to minimizing the travelling route. Formally, it is described by Eq. (2).

$$\text{DToSpace} = \text{Distance}\left(\text{Pos}^k_j - \text{Pos}_i\right); \forall j \in J; \ k \in K \tag{2}$$

Distance toward the destination

This criterion is relative to the Euclidian distance between the current position of the driver and its destination (Eq. (3)):

$$\text{DToDest} = \text{Distance}\left(\text{Dest}_i - \text{Pos}^k_j\right); \forall j \in J; \ k \in K \tag{3}$$

Cost

It corresponds to parking fees for the reserved period of time. Formally, this criterion is described in Eq. (4):

$$\text{Cost} = \text{Cost}\left(P^k_j\right)^* \text{ResDur}_i; \forall j \in J; \ k \in K \tag{4}$$

Space residue

In order to satisfy as many parking requests as possible, we intend to optimize the space residue resulting from a space reservation p^k_j by a specific vehicle i. This criterion is formally described in Eq. (5).

$$\text{SpaceRes} = \text{Size}^k_j - \text{Size}_i; \forall j \in J; \ k \in K \tag{5}$$

Reservation constraints

Several constraints that are related to reservation criteria have to be satisfied during the reservation process. We distinguish the following constraints:

$$\text{ResTime}^k_{ij} - \text{ReqTime}_i \leq \text{WaiTimeMax}_i \tag{6}$$

$$\sum_k^K \sum_j^J x^k_{ij} = 1, \ \ i = 1, 2, \ldots, N \tag{7}$$

$$\sum_{i=1}^N x^k_{ij} \leq 1, \ \ \forall j \in J, \ \forall k \in K \tag{8}$$

$$\text{Distance}\left(\text{Pos}^k_j - \text{Pos}_i\right) \leq \text{DMax1}_i \tag{9}$$

$$\text{Distance}\left(\text{Dest}_i - \text{Pos}^k_j\right) \leq \text{DMax2}_i \tag{10}$$

$$\text{SysTime} - \text{ResTime}^k_{ij} \leq \text{ResTimemax} \tag{11}$$

$$\text{Size}^k_j > \text{Size}_i \tag{12}$$

$$\text{Sec}\left(p^k_j\right) \geq \text{Sec}_i \tag{13}$$

$$\text{Cost}\left(p^k_j\right)^* \text{Resdur}_i \leq \text{CMax}_i \tag{14}$$

The first constraint in Eq. (6) is relative to the maximal waiting time for a space reservation p^k_j by a vehicle i. Eq. (7) guarantees that only one vehicle is assigned to a single space. However, Eq. (8) ensures that a space p^k_j is at most occupied by one vehicle i. Eq. (9) and Eq. (10) are relative to the upper bound of distance toward the parking space and toward the destination, respectively. Eq. (11) ensures that the vehicle i must not exceed ResTimemax for the space reservation p^k_j by considering the current system time SysTime. Eq. (12) corresponds to the size constraint of a specific space p^k_j in order to fit the vehicle i. Eq. (13) ensures that a minimal security level in p^k_j is required by a vehicle i. Finally, the cost constraint grantees that the estimated cost of vehicle i is not exceeded in Eq. (14).

Multi-agent system for reservation and guidance (MASRG)

Our proposed smart parking system is based on multi-agent approach. The main objective of this system is to deal with the multi-criteria reservation problem presented above. So, the multi-agent system for reservation and guidance (MASRG) aims to assist the driver in reserving the most convenient parking space according to reservation criteria. Then, the system has to provide route guidance toward the chosen space. In this section, we will first provide a description of the multi-agent architecture. Then, the multi-criteria decision aid process relative to space reservation is detailed. Finally, the guidance process is presented.

Multi-agent architecture

In order to reduce the complexity of the MPRP and to reach a considerable number of efficient alternatives in real time, the problem knowledge is divided into subsets that are affected to a number of agents by coordinating their activities. The system architecture is illustrated in Figure 1. We distinguish four types of agent: Vehicle agent, GIS agent, Locality agent, and Park agent. Each agent class belongs to a specific layer as shown in Figure 1. In the first layer, we find the Vehicle agents which are generated by the vehicle information and communication system (VICS) to handle requests of drivers. We notice that each vehicle has an integrated vehicular system VICS which is in charge of communication with the environment. The second layer models the parking information management center (PIMC), which acts as an interface between route layer and park layer. We notice that for each region in a city, a PIMC is associated to manage all the local parking lots.
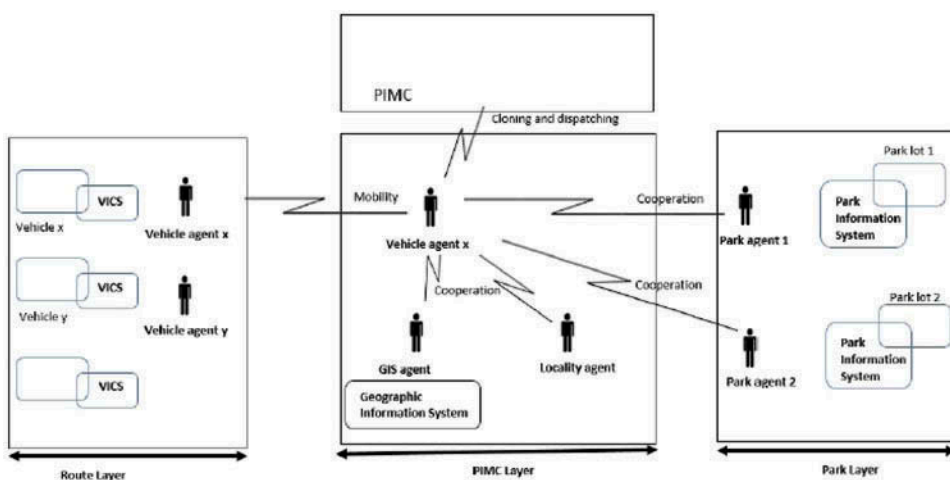

Figure 1. Architecture of MASGR.

The PIMC holds locality agent and geographic information system agent (GIS agent), which cooperates with the other agents during the decisional process.

The third layer is relative to parking lots of a specific region. For each lot, a park agent is generated to provide a real-time management of slot information (features of slots, status, parking fees, security level, etc.). Let us notice that the only mobile agent of the system is the Vehicle agent. It is able to move from the VICS to the PIMC in order to look for an efficient parking reservation according to reservation criteria.

Dynamic of MASGR

In the first layer, each driver asking for a parking space uses the integrated VICS to indicate its preferences (security level, vehicle size, destination, parking duration, upper bound for fees, distance toward the parking, and waiting time). The VICS generates a vehicle agent and initiates it with these parameters. Then, the Vehicle agent is dispatched to the PIMC, in the second layer where it directly communicates with the GIS agent by sending its destination. The GIS agent evaluates whether the vehicle destination is in the same region or not and informs the vehicle agent of the result. If the region is the same, then the vehicle agent will begin the cooperation with the locality agent of the same PIMC. Otherwise, the Vehicle agent will be cloned and dispatched to the concerned PIMC.

Once the Vehicle agent is in the appropriate PIMC, it asks the locality agent for the list of relevant park agents in order to cooperate with them. So, the multicriteria reservation process is started by organizing the relevant Park agents of the third layer and the Vehicle agent. The steps of the multicriteria reservation process will be detailed in the next section. Given the result of this process, the Vehicle agent determines the shortest path to the reserved space through a guidance process and by cooperating with the GIS agent.

Finally, the Vehicle agent will be sent back to the original VICS with the parking reservation and routes guidance. Then, space localization as well as the route guidance result will be displayed on the VICS interface. We notice that the main interest of using mobile agent in this system is to reduce the average communication traffic in the network, to improve the communicating time, and to guarantee a security of the whole process.

Multi-criteria reservation process

As we mentioned above, this process is carried out by the Vehicle agent and the pertinent Park agents within the appropriate PIMC. The Vehicle agent starts by asking the Park agents for the available slots via the message *Request_availability*. When receiving this message, each Park agent

determines the set of available spaces with the required features (position, security level, cost, size, etc.) from the park information system. Then, the determined set is sent to the Vehicle agent via the message *Answer_Availability*. So, the set of solutions S (or available spaces) is progressively constructed by the Vehicle agent at each reception of the message *Answer_Availability* from the Park agents.

Once the set S is defined, the Vehicle agent computes the value of each reservation criteria (WaiTime, DToSpace, DToDest, SpaceRes, Cost) for each solution and then verifies the satisfaction of operational constraints (upper bound of distances, the minimal security level, upper bound of cost, of waiting time, size constraint). So, the subset of non-satisfying solutions S_{NSat} which are violating the operational constraints is eliminated. Formally we obtained $S = S \backslash S_{NSat}$.

From the obtained solution set S, the subset of Pareto optimal solutions is found out by assigning a Pareto rank to each solution (see Algorithm 1) (Collette and Siarry 2002). These ranks are assigned to the solutions iteratively, after performing pair-wise comparisons between solutions. In fact, once the solutions with a Pareto rank 1 are determined, they are placed in a temporal set and the same thing is done for the remaining solutions (see Algorithm 1). This procedure stops when the initial set of possible solutions become empty. The solutions with rank 1 are considered Pareto optimal ones, since they are non-dominated. So, we can stop this ranking assignment after obtaining the set of solutions with rank 1. In Algorithm 1 shown here, we denote by N the cardinality of the initial set S (the set of satisfying solutions X_i). Let us notice that the set of solutions with rank 1 corresponds to the set of Pareto solution S_{Pareto}. Once the set S_{Pareto} is defined, the best compromise solution is selected by applying a multicriteria ranking method. In our case, we chose the ELECTRE III that will be justified and discussed below. Afterward, the Vehicle agent sends a reservation request via the message *Reserve_Request* including the selected space to the responsible Park agent. So, two scenarios are possible:

- If the space is still available, the Park agent will confirm the reservation for the Vehicle agent via the message *Confirm_Reservation*.

Algorithm 1 Assignment of Pareto rank

```
Assignment_ Pareto_ Rank()
   CurrentRank=1
   N= | S |
   m=N
   While N ≠ 0
   For i=1 to m do
   If Xᵢ is Non-dominated
   Rank(Xᵢ)= CurrentRank
   For i=1 to m do
   If Rank(Xᵢ)=CurrentRank
   put Xᵢ in a temporal set
   N=N-1
   CurrentRank=CurrentRank+1
   m=n
```

- If the corresponding space was already confirmed for another Vehicle agent, then the Park agent will ask the sender of current request to choose another available space. So, the Vehicle agent will select from the ranked solutions, the following one and will send a reservation request to the relevant Park agent.

When, the Vehicle agent receives the confirmed reservation, it deals with the guidance process through cooperation with the GIS agent in order to compute the shortest path toward the space.

Guidance process

As a result of the multicriteria reservation process, the Vehicle agent obtains a confirmed parking reservation from a Park agent via the message *Confirm_Reservation*. In order to assist the driver in reaching the reserved space, the Vehicle agent launches the guidance process in cooperation with the GIS agent. In fact, the vehicle agent starts by sending the message Route_Guidance_Request to the GIS agent with the space information as well as the current position of vehicle. At the reception of this message, the GIS agent computes the shortest path from the current position to the reserved space by using the Dijkstra algorithm (Schulz, Wagner, and Weihe 2000; Tirastittam and Waiyawuththanapoom 2014). This algorithm aims to finding the shortest paths between nodes in a graph which represents in our case road networks. The edge path costs represent driving distances between pairs of cities connected by a direct road. We notice that the stopping criterion of the algorithm is when the shortest path to the destination node has been determined. The resulting shortest path is then sent to the Vehicle agent through the message *Notify_route*. Finally, the Vehicle agent is sent back to the original VICS with information about the reserved space as well as route guidance.

Solution ranking method: ELECTRE III

In order to address solution ranking, we opted for a multi-criteria decision method ELECTRE (elimination and choice expressing reality) (Collette and Siarry 2002). We particularly chose the variant ELECTRE III, since it is based on fuzzy logic which introduces incertitude and imprecision in the evaluation of solutions via the pseudo-criteria instead of true criteria and the definition of credibility index. The introduction of fuzzy logic also offers a fine and precise solution ranking in addition to possibility of weighting the various criteria to express the preference of decision-makers. ELECTRE III relies upon two major phases (see Figure 2): the construction and the exploitation
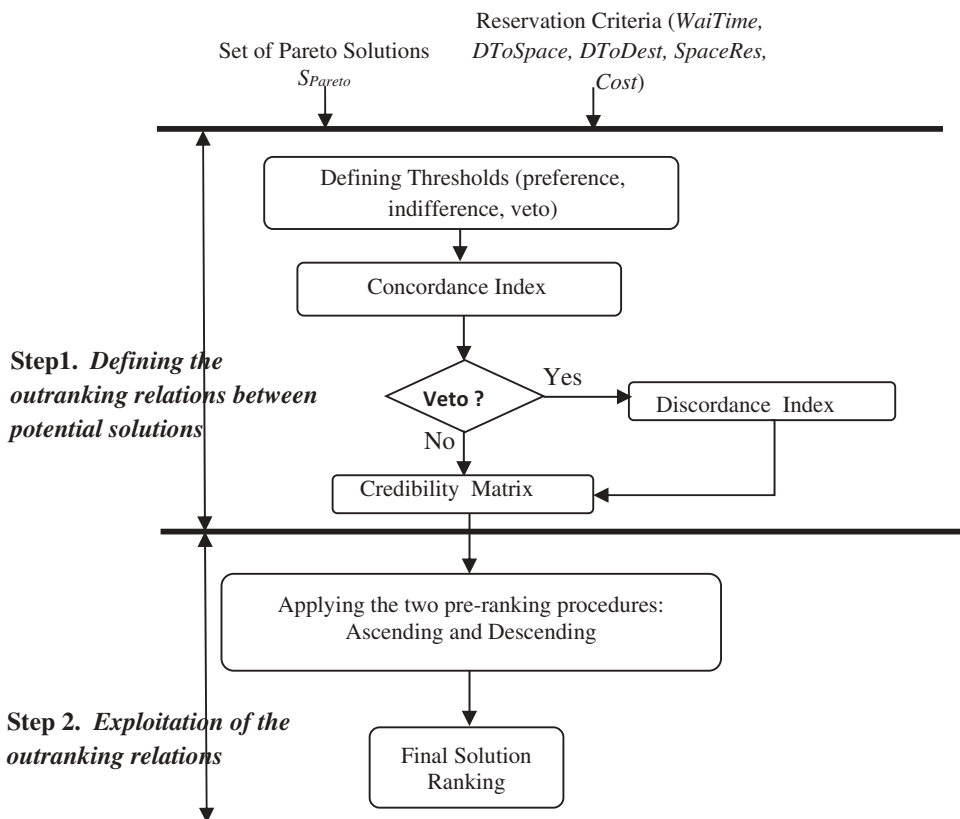
Figure 2. ELECTRE III process flow.

of the outranking relations (Giannoulis and Ishizaka 2010; (Fancello, Carta, and Fadda 2014).

Construction of the outranking relation

Alternatives are pair-wise compared (A,B). Each pair-wise comparison is characterized by an outranking relation. To say that "alternative A outranks alternative B" means that "A is at least as good as B in most of the criteria." Therefore, three outranking relations exist: A is "indifferent (I)," "weakly preferred (Q)," or "strictly preferred (P)" to B depending on the difference between the performance of alternatives and the thresholds given by the user (the indifference threshold q, the preference threshold p and veto v) (see Figure 3). We notice that the difference between performances of alternatives is indicated through the index of concordance and discordance.

Let us notice that the different parameters (concordance index, discordance index, credibility index, thresholds, etc.) are computed on the base of alternatives performances and according to formulas that can be found in
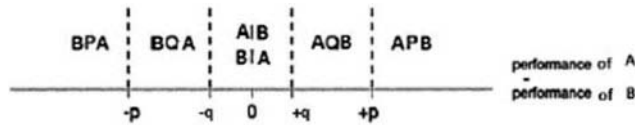
Figure 3. Outranking relations between solutions *A* and *B*.

Collette and Siarry (2002) and Giannoulis and Ishizaka (2010). Here, we only focus on the application of the ELECTRE III in the parking problem.

Exploitation of the outranking relation

The second stage of the method ELECTRE III consists in establishing a final ranking of alternatives on the base of the generated indexes. This final ranking is defined through the combination of two pre-rankings which are constructed with two antagonist procedures (ascending and descending distillation) in order to provide a finite and precise solution ranking.

In the multicriteria reservation process, this outranking method is employed by Vehicle agent in order to establish a classification of the computed Pareto solutions. As shown in Figure 2 (Giannoulis and Ishizaka 2010), the Vehicle agent starts by building the outranking relation among the Pareto solutions by taking into account their performances for each reservation criteria (WaiTime, DToSpace, DToDest, SpaceRes, Cost).

The construction of outranking relation involves defining thresholds (preference, indifference, and veto), pair-wise comparisons of alternatives, and indexes computing (index of concordance and discordance, credibility). Once the outranking relations are defined, the distillation procedure is launched. The distillation is an automated procedure which is used to rank the Pareto alternatives. The algorithm for ranking all alternatives yields two pre-orders:

- The first pre-order is obtained with a descending distillation, selecting the best-rated alternatives initially and finishing with the worst. This algorithm operates iteratively by starting with the initial set $S_{init} = S_{Pareto}$. The subset of best solutions S_{Best} is found out on the base of computed indexes and then is extracted from S_{init}. The same process continues in the following iteration but on the updated set $S_{init} = (S_{init}/S_{Best})$. At each iteration, the extracted alternative(s) S_{Best} will be ranked on a lower position. The procedure terminates when only one alternative remains or a group of alternatives that cannot be separated. Hence, the compromise solutions are progressively classified from the best compromise solutions to the worst ones.

- The second pre-order is constructed with an ascending distillation. In this case, the worst rated alternatives are selected first and the distillation terminates with the assignment of the best alternative(s). Hence, the ascending distillation procedure works in a similar way to the descending distillation with the exception of an opposite pre-order.

Once the two procedures are performed, the final ranking is obtained through the combination of the two pre-orders. The results from the partial pre-orders are aggregated into a ranking matrix. We have four possible cases:

- A is higher ranked than B in both distillations or A is better than B in one distillation and has the same ranking in the other distillation, then A is better than B: $A P^+ B$
- A is higher ranked than B in one distillation but B is better ranked than A in the other distillation, then A is incomparable to B: $A R B$
- A has the same ranking than B in both distillations, then A is indifferent to B: $A I B$
- A is lower ranked than B in both distillations or A is lower ranked than B in one distillation and has the same rank in the other distillation, then A is worst than B: AP_B

The final ranking is obtained according to the score of adding the number of $P+$. In case of tie, the comparison between the two alternatives with the same score decides between an indifferent or incomparable relation. Consequently, the Vehicle agent obtains a classification of Pareto solutions from the best ones to the worst. Hence, the Vehicle agent is not confused in the choice between several compromise solutions.

Simulation result

In order to assess the performance of the suggested MASRG model, we implemented it by using the multi-agent plate-form JADE (2008). In this section, we illustrate the reservation and guidance processes of MASRG through the following example inspired from real cases in the cities of Tunis and Ariana. Each city holds many parking lots as shown in Figure 4. We assume a driver coming from Ariana city and going to Mohamed V Avenue (Tunis city). We notice that the two positions belong to different regions and that each region is represented by a PIMC. The driver specifies some information through the graphic user interface (GUI) of the VICS: its destination, the estimated arrival time at destination, the upper bound of fees, security level, parking duration. The VICS automatically adds the vehicle size and its current position into the interface. Once the required information are introduced in the GUI, a Vehicle agent is launched and
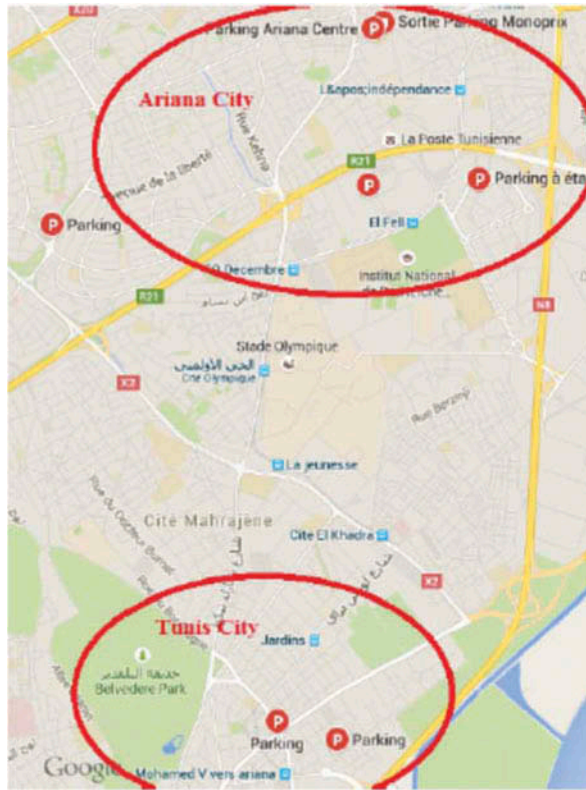
Figure 4. Parking location of the considered cities.

initialized with these parameters in order to undertake the driver request. The Vehicle agent moves to the PIMC of the current locality and begin cooperation with the GIS agent by sending its destination. The GIS agent checks whether the vehicle destination is in the same region or not and informs the vehicle agent of the result. Given that the current position and the destinations belong to two different localities, the Vehicle agent will be cloned and then dispatched to the appropriate PIMC. When arriving at the adequate PIMC, the Vehicle agent cooperates with the locality agent in order to get the list of relevant park agents. Hence, the Vehicle agent sends a parking request to the different park agents. When receiving all the parking offers from the park agents, a multicriteria reservation process is begun. In Table 1, we show the available parking spaces that may be received.

The Vehicle agent determines the performances of each feasible solution in order to compute the Pareto optimal solutions (see Table 2). Let us notice that feasible solutions correspond to available spaces that respect the reservation constraints.

Given the efficient solutions, the outranking method is applied to assist the Vehicle agent in selecting the adequate space. So, for each pair of Pareto

Table 1. Parking spaces availability.

Spaces	ParkName
S1,S2,S3,S4,S5,S6,S7,S8,S9,S10	Parking Mohamed V, Tunis
S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11, S12,S13,S14,S15,S16	Parking Mont-Plaisir, Tunis
S1,S2,S3,S4,S5,S6	Parking of Jordanie, Tunis
S1,S2,S3,S4,S5,S6,S7,S8,S9	Parking lot, Tunis

Table 2. Performance matrix of solutions (S1– S10).

Spaces	Cost	DToSpace	DToDest	SpaceRes
S1	3358.33	17.94	17.99	12.84
S2	3141.66	17.91	17.97	17
S3	1300	22.54	22.50	11.45
S4	1625	20.65	20.67	9
S5	1191.66	22.48	22.47	12.5
S6	1733.33	18.32	18.36	5.12
S7	1440.83	19.97	20.03	11.72
S8	4333.33	18.92	18.97	3.98
S9	2383.33	20.30	20.33	11.07
S10	1950.0	17.50	17.55	19.54
Weight(w)	0.5	0.2	0.2	0.1
Indifference(q)	4	1	2	0
Preference(p)	10	2	4	5
Veto(v)	90	4	8	15

solutions, the indexes of concordance, discordance, and credibility are computed and the outranking relations are determined (see Table 3).

Then, the Vehicle agent performs the ascending and descending distillations in order to get the final ranking of solutions. In Figure 5, we show the result of this two distillations and the obtained final ranking. Given the solution ranking (S8 of parking Mohamed V, Tunis), the Vehicle agent sends a reservation request to the Park agent in charge of the space.

When a confirmation is received, the Vehicle agent cooperates with the GIS agent during the guiding process in order to determine the shortest path to the space (see Figure 6).

Conclusion

In this paper, we first provided a literature review of the existing smart parking systems and we highlighted their major shortcomings. One of the

Table 3. Credibility matrix.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	1	0.91	0.6	0.73	0.6	1	0.79	0	0.76	0.9
S2	0	1	0.6	0.72	0.6	1	0.79	0	0.76	0.94
S3	0	0	1	0	0.97	0	0	0	0	0
S4	0	0	0.77	1	0.76	0	0.94	0	0	0
S5	0	0	0	0	1	0	0	0	0	0
S6	0	0	0.5	0.69	0.5	1	0.76	0	0	0
S7	0	0	0.75	0	0.74	0	1	0	0	0
S8	0.9	0.9	0.54	0.75	0.54	0.97	0.88	1	0.82	0.9
S9	0	0	0.77	1	0.75	1	0.98	0	1	0.9
S10	0	0	0.6	0.68	0.6	1	0.75	0	0	1
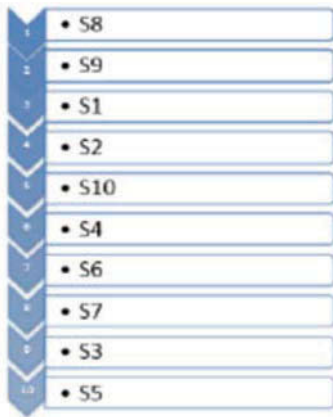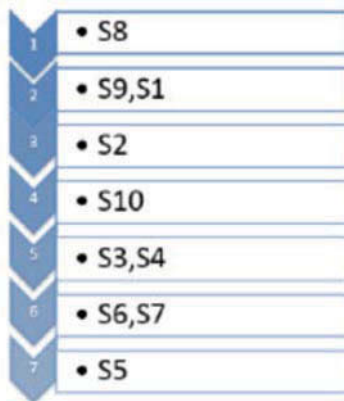
Figure 5. Results of the two distillation procedures and the final ranking of solutions.
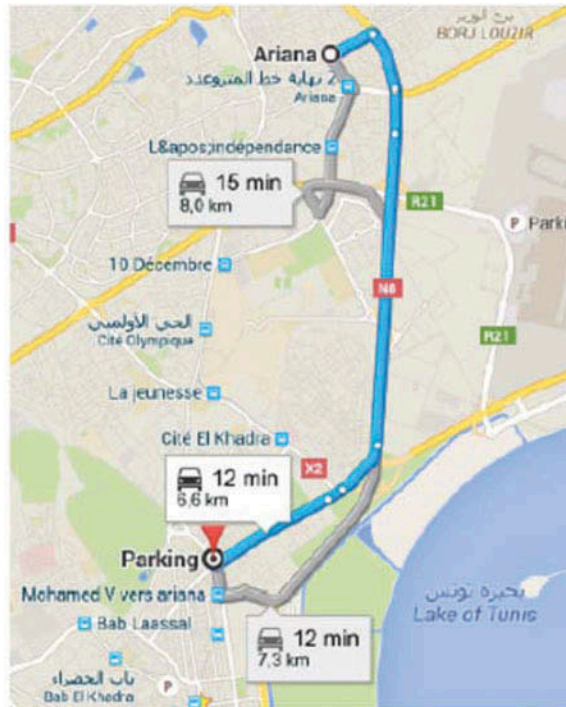
Figure 6. Shortest path from Ariana city to Mohamed V Avenue.

most neglected aspect in the existing approaches is the multicriteria aspect of the parking problem. In fact, the different preferences of the driver have to be considered during the solving process, and some exploitation constraints have to be satisfied.

Therefore, we proposed a multicriteria formulation of the problem which we designate as the (MPRP). Then, we suggested a MASRG in order to deal with the problem. The architecture of our system is based on multilayer approach defining three layers of decision: route layer, PIMC layer, and park layer. Agents of these layers act cooperatively during the two solving processes: reservation and guidance processes. In the first process, a multicriteria parking reservation is handled by applying an outranking method: ELECTRE III. The method guarantees a finite ranking of Pareto solutions according to the driver's preferences. Then, in the second process a guidance process is performed to assist the driver in reaching the reserved space. In order to show the solving process of the system, we presented an illustrative example based on real cases in the cities of Tunis and Ariana. Since no previous works have addressed the multicriteria aspect of the parking reservation problem and no benchmarks are available for this kind of problem, we did not perform a comparative evaluation of our research. However, we intend to study the scalability of our multi-agent system especially during rush hours when many parking requests are dispatched from vehicles.

ORCID

Imen Boudali 🔴 http://orcid.org/0000-0003-2885-0085

References

Bessghaier, N., M. Zargayouna, and F. Balbo 2012. Online localized resource allocation application to urban parking management, In Proceedings of the International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2012), IEEE/WIC/ACM, 67–74.

Chou, S.-Y., S.-W. Lin, and -C.-C. Li. 2008. Dynamic parking negotiation and guidance using an agent-based platform. *International Journal of Expert Systems with Applications* 35 (3):805–17. Elsevier. doi:10.1016/j.eswa.2007.07.042.

Collette, Y., and P. Siarry. 2002. *Optimisation multiobjectif.* France: Eyrolles.

Fancello, G., M. Carta, and P. Fadda. 2014. A decision support system based on Electre III for safety analysis in a suburban road network. *International Journal of Transportation Research Procedia* 3:175–84. Elsevier. doi:10.1016/j.trpro.2014.10.103.

Ganchev, I., M. O'Droma, and D. Meere. 2008. Intelligent car parking locator service. *International Journal of Information Technologies and Knowledge* 2:166–73.

Geng, Y., and C.-G. Cassandras 2011. Dynamic resource allocation in urban settings: A smart parking approach. In Proceedings of IEEE International Symposium on Computer-Aided Control System Design (CACSD-2011), 1–6.

Giannoulis, C., and A. Ishizaka. 2010. A web-based decision support system with Electre III for a personalised ranking of British universities. *International Journal of Decision. Support Systems* 48:488–97. Elsevier. doi:10.1016/j.dss.2009.06.008.

Iordache, V., F.-C. Nemtanu, and A.-C. Cormos 2015. Autonomic integrated parking system for smart cities. Meeting of Autonomic Road Transport Support Systems (ARTS-ECR 2015), Malta.

JADE. 2008. Java agent development framework. Accessed December 29, 2015. http://jade.tilab.com/

Longfei, W., C. Hong, and L. Yang 2009. Integrating mobile agent with multi-agent system for intelligent parking negotiation and guidance. In Proceedings of the 4th Conference of Industrial Electronics and Applications 1704-1707, IEEE.

Lu, R., L. Xiaodong, Z. Haojin, and S. Xuemin 2009. Spark: A new Vanet-based smart parking scheme for large parking lots. In Proceedings of IEEE - INFOCOM'2009, 1413–21.

Schulz, F., D. Wagner, and K. Weihe. 2000. Dijkstra's algorithm on–line: An empirical case study from public railroad transport. *Journal of Experimental Algorithmics* 5:12. ACM. doi:10.1145/351827.384254.

Srikanth, S. V., P. J. Pramod, K. P. Dileep, S. Tapas, M. U. Patil, and C. B. N. Sarat 2009. Design and implementation of a prototype smart parking (SPARK) system using wireless sensor networks. In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA-2009), IEEE, 401–06.

Tirastittam, P., and P. Waiyawuththanapoom. 2014. Public transport planning system by dijkstra algorithm: Case study bangkok metropolitan area. *International Journal of Social, Behavioral, Educational, Economic and Management Engineering* 8:1.

Wang, H., and W. He 2011. A reservation-based smart parking system. In Proceedings of the Workshop of Computer Communications of IEEE-INFOCOM Conference (2011), 690–95, IEEE.