ABSCEV: An agent-based simulation framework about smart transportation for reducing waiting times in charging electric vehicles

Iván García-Magariño[a,b,*], Guillermo Palacios-Navarro[c], Raquel Lacuesta[a,b], Jaime Lloret[d]

[a]*Department of Computer Science and Engineering of Systems, University of Zaragoza, Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain*
[b]*Instituto de Investigación Sanitaria Aragón. University of Zaragoza.Zaragoza*
[c]*Department of Electronic Engineering and Communications, University of Zaragoza, Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain*
[d]*Integrated Management Coastal Research Institute, Universitat Politècnica de València, 46022, València, Spain*

Abstract

Fuel has been the main source of energy for cars for many years, but the non-renewable resources are limited in the planet. In this context, electric vehicles (EVs) are increasingly replacing the previous kind of cars. However, as the number of EVs increases, some challenges arise such as the reduction of waiting times in the queues of fast charging stations. The current work addresses this challenge by means of social coordination mechanisms. In particular, this work presents an agent-based simulation framework for simulating the effects of different coordination policies in the route planning of EV drivers for charging their vehicles on their trips. In this manner, researchers and professionals can test different coordination mechanisms for this purpose. This framework has been experienced by simulating an adaptive strategy based on the implicit communication through booking systems in the charging stations. This strategy was compared with another common strategy, which was used as the control mechanism. This comparison was

*Corresponding author, Telephone: +34 978645348, Fax: +34 978618104
Email addresses: ivangmg@unizar.es (Iván García-Magariño), guillermo.palacios@unizar.es (Guillermo Palacios-Navarro), lacuesta@unizar.es (Raquel Lacuesta), jlloret@dcom.upv.es (Jaime Lloret)

done by simulating several scenarios in two Spanish cities (i.e. Madrid and Zaragoza). The experimental results show that the current approach was useful to propose a route planning strategy that had statistically significant improvements in the reduction of waiting times in charging stations and also in the global trip times. In addition, the evolutions of pathfinding execution times and the numbers of interchanged messages did not show any overloading pattern over the time.

Keywords: agent-based framework, agent-based simulator, agent-based social simulation, electric car, electric vehicle, multi-agent system

---

1. Introduction

The renewable energies are the basis for a sustainable future [1]. Hybrid cars are getting to be popular in Europe. However, there is a large part of the population that use fuel cars, and there are very few citizens that use pure electric cars. In fact, there are some barriers that prevent most population from using non-hybrid electric cars. In general, one of the main drawback of electric vehicles (EVs) is that these need much more time to recharge their batteries than common fuel vehicles need for refilling gas. Thus, some citizens may choose not to buy EVs due to their concern about the possible waiting time in the queues of charging stations. Although increasing the number of charging stations could be a solution for decreasing this waiting time, this solution could bring other serious problems such as the impact on the distribution networks [2] or the necessary economical costs.

Vehicular ad hoc networks (VANETs) allow vehicles to communicate among each other without needing a fixed infrastructure [3]. The street awareness can be useful to keep vehicle positions updated. For instance, a geographical forwarding protocol was proposed based on street awareness for a VANET, and it reduced the average delay by communicating the relevant information when vehicles arrived to intersections [4]. In VANETs, model-driven engineering has shown to be useful for modeling the interchanged information for facilitating the collaboration among vehicles [5].

In this context, multi-agent systems (MASs) have proven to be useful for managing different traffic and transport situations in the last decades [6]. More concretely, MASs have been used for analyzing different impacts of EVs in aspects such as power grid [7, 8] or the best pricing strategies for charging station companies for making high economical profits [9]. In particular, agent-based simulators (ABSs) are a specific kind of MASs, aimed at simulating different decentralized realities. In the recharge of EVs, the comfort of drivers has also been analyzed for slow charging parking stations [10] and fast charging stations [11]. Model-driven engineering has also been useful for modeling MASs for facilitating the collaboration. The MAS based on the Delphi process is an example of multi-agent based collaboration developed with this approach [12]. Model-transformation by examples have allowed developers to easily define model transformations in this kind of development [13].

This article focuses on how to tackle the problem from a practical way of coordinating drivers for making the best use of charging stations. In this line of research, this article proposes an ABS framework for testing coordination policies so engineers and practitioners can assess their performance through simulations before actually using these in real scenarios. This framework is configurable so it can be applied to different city maps, by introducing these maps from text files.

It is worth noting that the presented framework is a first step towards the application of Intelligent Transportation Systems (ITSs) for reducing the trip times of EV drivers. Once the most appropriate strategies are found, these could be integrated in the EV software, so drivers can ask their vehicles to offer the most convenient charging stations for stopping on their way to certain destinations.

The remainder of the article is organized as follows. The next section introduces the most similar existing works, and highlights the gap of the literature that the current work covers. Section 3 presents the ABS framework for simulating different route planning and coordination strategies for EVs that need to be charged on their trips. Section 4 assesses the current framework by evaluating and comparing certain strategies with different simulations in two different-sized Spanish cites. Section 5 discusses the most relevant aspects of the results, and section 6 mentions the conclusions and the future lines of research.

2. Related work

Simulations have been widely applied to address challenges of ITSs in general, as the current work has done for a particular problem. For instance, the evacuation scenarios have been simulated in different network topologies to analyze the repercussions of the correlated network intersections on the evacuation scenarios [14]. Another simulator was used to assess a route planning approach based on mobile crowdsensing for determining the number of vehicles in different road segments [15]. In addition, another work explored the benefits of ride-sharing considering pollution, energy consumption and congestion among others [16]. They presented a hybrid approach that firstly used a greedy algorithm and then improved the solution with constrained optimization, in order to obtain routes that could be shared by several citizens. This ride-sharing approach was simulated and validated with information from about 3 million rides extracted from a public taxicab dataset about New York. Like in these works, many ITS works ignore the specific problems that are being raised by the large use of EVs.

Nonetheless, there is a growing number of ITS works that address specific challenges of EVs. Most of these works fall into one of several categories. The first category is about simulating and analyzing the effects of the charging of EVs on the power networks. Secondly, some works address economic issues around the usage of EVs. Finally, some works focus on several aspects of the drivers' comfort, such as facilitating the charge of their EVs.

In the first category, several works used MASs for assisting the management of EVs regarding power grids. [17] introduced a MAS for controlling the charging of large populations of EVs. This work presented a simulation over realistic scenarios to evaluate their approach. Their main goal was to ensure the power networks performance considering the users' preferences. In a similar way, [8] proposed an agent-based approach for EVs that considered the state of the power grid, in order to recharge in a balanced way to not overload it. [7] proposed a MAS composed of several agents, each of which was integrated in a Linux system embedded in a different EV. These agents cooperated to integrate these EVs in a power grid system. In their approach, these EVs charged when the electricity was less demanded (with a low price), and they even sold energy in the peak hours (with a high price) supporting the grid. This work was mainly focused on assisting the power grid and the trade of energy with different prices reducing the costs for EV owners or even making them earn money. Thus, these works are mainly concerned with the charge of EVs avoiding congestions on power grids. By contrast, the current work is mainly focused on reducing queue waiting times on charging stations for the comfort of drivers and the popularization of EVs.

The second category of works address economic issues around the charge of EVs. For example, a simulator considered different charging strategies of EV drivers, such as mainly considering charging prices to get the lowest ones assuming the cases in which drivers are not in hurry [18]. From the perspective of charging station owners, [9] presented a MAS for estimating the demand of the charging stations from EV drivers in order to determine the better prices for maximizing the profits. These works and the current one have in common the execution of simulations for testing different strategies. However, the difference is that these works are mainly concerned with the prices and profits, while the current one is concerned with the reduction of queue waiting times.

The third category is related to EV drivers' comfort, and is the most related one with the current approach. These works are mainly aimed at facilitating the charge of EV drivers. In particular, [10] proposed an approach for providing an appropriate infrastructure of slow charging parking stations (about 12 hours each charge) with standard wall outlets (110 V - 220 V), so that EV drivers can be supplied when they park their EVs. They also evaluated their approach with a simulation. The current work is also aimed at facilitating the charge of EV drivers. However, the current work deals with fast charging stations (about 30 minutes each charge), in which the user is assumed to be there waiting and the number of stations is much more limited.

In addition, in this same category, [11] presented an anticipatory coordination mechanism for the fast charging of EVs. In their approach, the EVs were coordinated so that each driver selected the best charging station on their way. However, this work did not consider possible deviations in EV driver paths, as the current work does.

Although the number of simulations about ITSs is increasing steeply, there is still not an agreement about which are the best technologies for this kind of simulations. For example, [18] used Matlab for simulating the effects of EVs on electric distribution systems. Other approaches used general-purpose object-oriented programming languages such as Java [19], or even web-based languages like JavaScript alongside Cordova [20]. However, there are other works that propose domain-specific technologies for ITSs such as ontologies [21], facilitating the intercommunication between different devices. In this line, another proposal used model-driven engineering principles for conforming a domain-specific modeling language about the simulation of ITSs [22]. In this context, cross-platform game-based development technologies are not so much explored as is the case of Unity, which is the technology used in the current work.

3. ABS about smart transportation for reducing waiting times in charging EVs

The proposed model of the *ABS framework about smart transportation for reducing waiting times in Charging EVs* (ABSCEV) has been defined with the updated version of ODD (Overview, Design concepts, and Details) protocol [23]. The presentation of the current ABS model follows the structure of sections recommended by this protocol. In addition, section 3.9 introduces the user interface of the simulator tool.

The development of the ABS framework has followed the Process for developing Efficient Agent-Based Simulator (PEABS) [19]. In general, PEABS allows developers to rapidly develop ABS that are normally efficient in terms of response time and memory usage. PEABS achieves these benefits by relying in some core ideas such as (a) using implicit communications, (b) avoiding communication platforms like JADE, and (c) using method invocations for communications. In particular, ABSCEV uses implicit communications among EVs through the booking systems in charging stations. ABSCEV has been developed without any time-consuming communication platform like JADE. ABSCEV simulates the communications with method invocations in the C# programming language. We have also incorporated the recommendations about spatial locations of agents from TABSAOND (a technique for developing agent-based simulation apps and online tools with nondeterministic decisions) [20].

Regarding the technology, this ABS uses the widespread cross-platform game-based engine Unity [24], which has allowed us to create a user-friendly simulator that can be used by both researchers and non-specialized users.

3.1. Purpose

The purpose of this ABS framework is to simulate the outcomes of different pathfinding strategies and coordination mechanisms in the selection of paths of EVs that need to be charged on their ways from certain departure locations to certain destinations. The main outcomes are measured with the average waiting time of EVs in fast charging stations and the average time of EV trips. Another goal of this ABS framework is that researchers can extend it for assessing more elaborated strategies and richer representations of real-world scenarios.

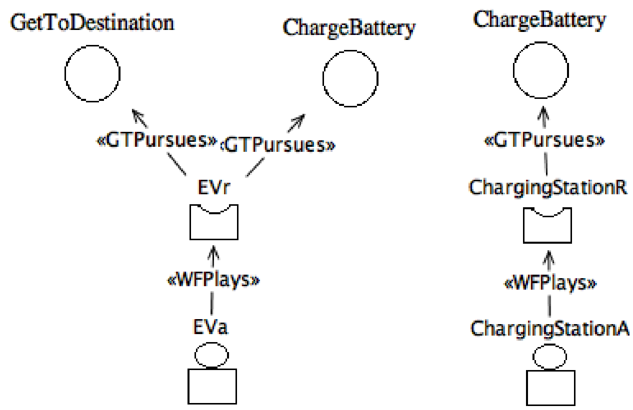


Figure 1: Agent diagram of ABSCEV with Ingenias notation

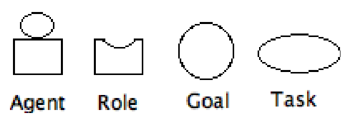


Figure 2: Main symbols of Ingenias notation

3.2. Entities, state variables and scales

The main entities of the presented ABS model are the "EV" agents and the "Charging Station" agents, which are presented in Figure 1 with an agent diagram with the Ingenias notation [25]. This diagram also includes the agent roles and their goals. The architecture of this ABS was designed considering the common MAS guidelines about keeping the cohesion high and the coupling low [26]. It is worth mentioning that this diagram uses the -R (or -r) suffix for roles and the -A (or -a) suffix for agents, to avoid conflict of names in an abbreviated way. In order to make this diagram and the following one understandable, Figure 2 determines the representation of the main concepts in the Ingenias notation. The agent types of this ABS are further introduced below:

- *EV agents*: Each of these agents simulates the behavior of an EV driver. This agent could also simulate a software assistant installed in the EV navigator system. These agents consider their current locations and their destinations. Each of these agents is aware that it needs to charge its vehicle before finishing the trip, as this assumption belongs to the problem definition. Each submodel can establish a different strategy for selecting a route that includes a stop for charging in a station.

- *Charging Station agents*: Each of these agents manages a charging station, and controls which EV has access to the fast charging plug-in at each time slot. Regarding the submodel, it can use different strategies, such as (a) implementing just a waiting queue for sequentially charging the EVs in the arrival order, or (b) having a booking system to let EV agents know the availability of the station in advance.

Regarding the state variables, some of these are common to both agent types. For example, any agent has a 2D position. EV agents change their positions during the simulations, since EVs move from some places to others. However, the positions of charging station agents are just set at the beginning of the simulation from a given map.

In addition, the places of agents of both types are related to their positions in the graph that represents the current map. Thus, any agent knows in which road (graph edge) or intersection joint (graph node) it is placed.

Each EV agent has variables for its heading direction and its speed. EV agents has another variable that indicates the necessary time for charging their vehicles in a fast charging station. Each EV agent also has an internal variable that determines whether it has charged its battery in the current simulation.

Each charging station agent has a variable that indicates the EV that is currently charging if any. It also has different data structures regarding the policy for establishing the order of charging EVs.

Concerning the scales, the positions are established as relative distances from the map center measured in kilometers. The speed is determined in kilometers per hour. The necessary time for charging is measured in minutes.

3.3. Process overview and scheduling

Among the parameters of the simulator is the time interval between the starting times of drivers. The simulator starts an EV and then waits for the established time. Then, it starts the driving of another EV and waits again. The simulator repeats this until it reaches the established number of EVs in the simulation. This simulator was built using Unity, which is a cross-platform engine for developing applications. Unity was originally created as game-based engine, but its flexibility facilitates the development of general-purpose applications. Unity allows designers to define, create and destroy dynamically visual elements arranged in a scene. These visual elements were used to represent the different kinds of agents such as the EVs with drivers and the charging stations. In Unity, visual elements can periodically call functions from scripts representing object-oriented classes. More concretely, the "Update" function is invoked every frame, and we implemented this function to impersonate the behavior of each agent type. Algorithm 1 represents the implementation of this function showing how the simulator keeps track of the elapsed time and generates EV agents sequentially.

Algorithm 1 Procedures for generating the EV agents

```
procedure Start
    elapsedTime ← 0
    counterEVs ← 0
procedure Update(timeInterval, numEVs)
    if counterEVs<numEVs then
        elapsedTime ← elapsedTime + Time.deltaTime
        if elapsedTime ≥ timeInterval then
            (origin,destination) ← SelectOriginAndDestination(graph)
            ev ← new EV()
            ev.Pathfinding(origin, destination)
            ev.StartDriving()
            elapsedTime ← 0
            counterEVs ← counterEVs+1
```

When an EV agent is created, firstly the manager selects its origin and destination. Then, the EV agent searches for a path considering these two locations with the restriction of charging the battery. Section 3.7 will introduce several mechanisms for this search of paths. Once the path is found, the EV agent starts driving following the path, updating its position visually in the user interface every frame.

It is worth mentioning that there are two modalities for establishing the origins and destinations of the paths of the drivers. In the first option, all the EVs share the same origin and the same destination. In this case, the origin and the destination nodes are the first and the last ones in the definition of the map. Normally these two locations are recommended to be selected far from each other, so that the possible paths can cover a great part of the map. This modality is especially useful for analyzing and understanding behaviors in a simple, deterministic and reproducible way.

The second modality is that each EV selects two random positions for respectively the origin and the destination forcing that these are different between each other, with the Algorithm 2. Each EV agent searches a path for its particular origin and destination. This modality is more realistic and relevant for obtaining simulations that are more similar to real scenarios. The analysis of this kind of simulations requires a representative number of executions to draw meaningful conclusions, due to its stochastic behavior.

Algorithm 2 Selection of two different random positions for respectively the origin and the destination

```
procedure RandomOriginAndDestination(graph)
    indexOrigin ← Random.randInt(graph.numNodes)
    origin ← graph.GetNode(indexOrigin)
    indexDestination ← Random.randInt(graph.numNodes-1)
    if indexDestination ≥ indexOrigin then
        indexDestination ← indexDestination+1
    destination ← graph.GetNode(indexDestination)
    return (origin, destination)
```

3.4. Design concepts

3.4.1. Basic Principles

The goal of ABSCEV is to analyze the waiting and trip times of EVs given certain conditions, for comparing different strategies in the selection of paths and stations for charging EVs. The basic principles of ABSCEV are summarized as follows:

- Several EVs are simulated in the same map.
- Each EV is assigned to a certain origin and destination.
- Each EV must go from the origin to the destination with the mandatory restriction of charging its battery in a station before finishing the trip.
- There are several charging stations in the map.
- Two different EVs cannot simultaneously charge their batteries in the same station.

The proposed framework allows analyzing different strategies and incorporate different elements to implement useful coordination strategies. In particular, the framework included the possibility of booking time slots in the charging stations, in order to analyze a pathfinding algorithm with coordination based on these bookings.

3.4.2. Emergence

In the case of using the booking systems in the charging stations, the resulting behavior of the EV agents is an emergent coordination. For example, when all the agents share the same origin and destination locations, the emergent behavior is that the EVs distribute between the different paths with charging stations in a coordinated way to reduce the waiting times in the charging stations. This happens even if some of the paths are larger than others, as long as they achieve their goal of reducing the trip times. This fact reveals coordination, since normally each EV would not select a path if there was another shorter possibility, unless the EVs are coordinated to distribute the paths for avoiding coincidences in the stations.

When applying the booking systems in simulations with different origins and destinations, one can also observe that EV agents choose paths that are not the shortest ones given their beginning and target locations and the restriction of needing to charge. However, these selections also reveal coordination, since these reduce not only the average waiting time in charging stations but also the average time of the trips. Section 4.2 indicates the results of the experiments that corroborate these assertions.

3.4.3. Adaptation

In the strategy in which EV agents use booking systems, EV agents are adaptive since they decide their paths based on the current status of bookings in charging stations. A prove of the adaptiveness of EV agents is that, when they share the same departure and destination locations, they take different decisions based on the particular state of the simulation. This adaptive booking strategy will be properly introduced in section 3.7.2.

By contrast, in the strategy used as control mechanism introduced in section 3.7.1, EV agents always take the same decision given some particular departure and target locations in a specific map, regardless the current state of the simulation.

3.4.4. Objectives

The primary objective of EV agents is to arrive at their destinations from their origins in the shortest time possible with the condition that each of these charges its EV in a charging station on its way.

The use of the current framework is illustrated with a pathfinding mechanism in which charging stations integrate booking systems. In this particular version, the objective of the charging station agents is to manage the bookings of time slots in the plug-in so that EV agents can efficiently use the service of charging stations.

This strategy is compared with another pathfinding mechanism, as the control version. Although the primary objectives EV agents remain the same, they accomplish it in a different way. They simply assume that the shortest path in distance will be the fastest one ignoring any estimation of the possible delays because of the waiting queues in the corresponding charging stations. In this case, the goal of charging station agents is just to manage the corresponding waiting queues so the EV agents respect the order of arrivals in the use of the charging service.

3.4.5. Prediction

The current ABS framework predicts the average waiting times and the average trip times in the trips of EVs when these need to be charged on their ways.

The prediction is based on certain input parameters such as the expected number of EVs for a certain time period, the charging time, the average EV speed and the frequency of EVs starting their trips (indirectly indicated as the number of minutes between the beginning of these trips).

One of the most influential factors on the waiting and trip times is the strategy used for selecting the paths and coordinating among the different agents. The current framework is illustrated with a basic control mechanism and a different strategy based on booking charging station slots. The current framework can predict the repercussions of each of these strategies. In addition, the framework can also be used to explore new strategies and predict their repercussions.

3.4.6. Sensing

The charging station agents sense the arrivals of EV agents. They assign these to a queue and then sequentially supply these with energy. In the case of using the booking strategy, charging station agents inform EV agents about the soonest available time slot from certain time. They also book an available time slot when an EV agent requests so.

The EV agents sense (a) the whole map, as this is normally public static knowledge, and (b) the positions of the charging stations. In one of the strategies of the current approach, EV agents can also know about the available time slots for charging in the stations.

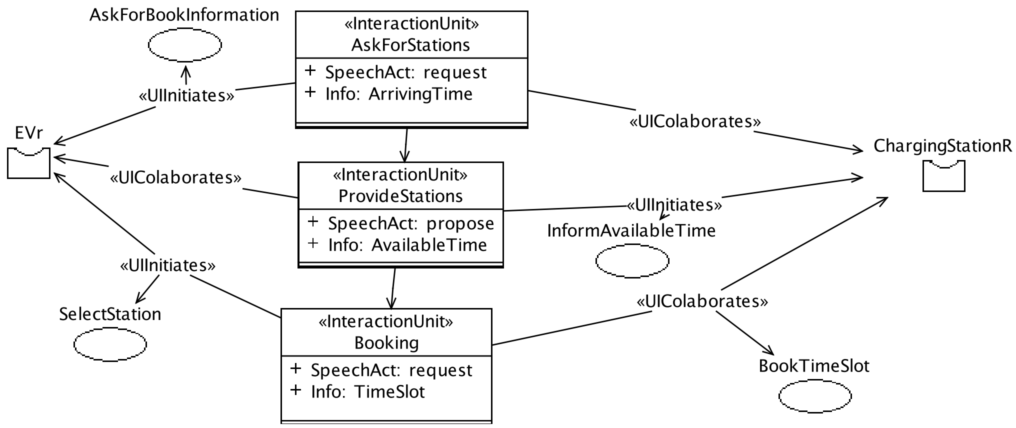


Figure 3: Interaction between EV agents and charging station agents

3.4.7. Interaction

The explicit communication is performed between EV agents and charging station agents, and its purpose is twofold. First, the EV agents can ask about information of the current bookings in the different stations. Second, the EV agents can actually book a time slot in a particular charging station, just right after it has selected a path and a stop for charging.

Figure 3 shows the interaction between an EV agent and a charging station agent. This figure uses the Ingenias notation that was previously introduced in Figure 2. Firstly, the EV agent asks for information about available time slots from a given arriving time. The charging stations replies with the first available one, considering the necessary charging time. The EV agent assesses its option. In case, it actually selects this particular charging station, it informs so to the charging station agent, so that it actually books a time slot in this station. Notice that the EV agent normally assesses several charging station options before actually choosing one. In case it does not choose a particular charging station, it just omits the third message (i.e. interaction unit) of the interaction.

It is worth mentioning that there is also an implicit communication between EV agents. Although these agents do not interchange explicit messages between each other, each EV agent takes a decision (i.e. a path) and lets this information available to their peers through an actual booked time slot in one of the charging stations. This implicit communication among EV agents is what makes the emergent coordination behavior of this ABS.

3.4.8. Stochasticity

In the modality of selecting different origin and destination locations, these locations are selected randomly from the different nodes of the graph that represents the city map. The corresponding mechanism was determined in Algorithm 2.

3.5. Initialization

The initialization of the simulations depends on the input parameters set by the user. More concretely, the user can establish the number of electric cars of the simulation, the necessary time for charging a vehicle in a station, the speed of cars, the time interval between the start time points of cars, the modality for selecting the origins and destinations of trips, and the kind of strategy applied in the simulation.

In the time zero of the simulation, the map selected by the user is loaded into the simulator. All the charging station agents are created, assigning one agent to each station of the map. These agents initialize empty waiting queues and/or empty list of bookings, regarding the used strategy.

Normally only one EV agent is created at the beginning since all the EV agents start at different time points equally distributed among the timeline of the simulation, even if the start points are separated with very short time. Notice that the current simulator is executed with 30 frames per second, so 30 agents could start in different time points in each real-time second of the execution. The simulator assigns an origin and a destination to the EV agent. This agent chooses a path with a pathfinding algorithm, and starts following it at the speed indicated by the user.

3.6. Input data

This simulator receives input from data files that represent any kind of road or city maps. These files contain the positions of the road joints (that represent the nodes of a graph) in one list, and the way these locations are interconnected (i.e. the edges of the graph) in another list. Figure 4 shows an example of a simple map represented with this notation. Notice, that in the former list each node is represented with the two coordinates of its location, measured as the distances from the map center in km. When a node has a charging station, it is denoted with the word "station". In the later list, the edges are denoted with the two integer indexes of the nodes it connects, starting counting the indexes at zero. Both lists are headed with meaningful titles to remind their meanings and to determine the limit between them.

SimpleMap.txt

```
Positions of nodes
-2,0
-1,1
-1,-1
0,1,station
1,0
2,2
2,-2,station
3,0
Edges indicated with indexes
0,1
0,2
1,3
2,3
3,4
3,5
4,5
4,6
2,6
5,7
6,7
```

Figure 4: Example of a map represented with a text file for being simulated

3.7. Submodels

This section presents the two possible strategies that are included in this framework for illustrating some different mechanisms for selecting paths in the current approach. Section 3.7.1 presents a basic static strategy based on the distances of the corresponding map, while section 3.7.2 presents a dynamic adaptive strategy based on the use of booking systems in the charging stations.

3.7.1. Basic strategy based on physical distances

In this strategy, each EV agent selects the shortest path from the beginning location to the destination that satisfies the restriction of going through a charging station. This strategy is called "Distance" as it only focuses on the physical distance of the path, besides satisfying the charging restriction.

We defined a modification of the A* (A-Star) algorithm for this strategy. Note that in general the variations of the A* algorithm have been widely used in the literature about pathfinding [27]. In the presented modification of the current work, the condition of achieving the right path is not only to reach the destination but also going through a charging station.

Basically, this algorithm uses a priority queue of paths ordered by their distances. It starts with a path of only the origin. From each path in the queue, the algorithm pops it, and push new paths generated from it. More concretely, the algorithm generates all the paths obtained from adding the unvisited neighbors to the popped path. If the last node of the treated path is a charging station, the path is marked as "charging", and adds a new edge to the path that symbolizes the recharge. When analyzing each first path of the priority queue (ordered by global distance), the algorithm determines whether both of the two conditions are satisfied: (a) the last location is the destination and (b) whether the path is marked as charging. If both conditions are met, the algorithm returns this path.

Once a car obtains a path with this algorithm, then the car starts following this path. Notice that even if the path is selected as the shortest one, then the simulation considers the restriction that only one car can charge in a station at a given time. Thus, a trip may suffer delays from the expected time because of the waiting time to be attended in a charging station due to the possible coincidences of cars.

3.7.2. Adaptive strategy based on booking systems

This adaptive strategy is based on the dynamic booking of charging stations. Figure 5 shows an overview of this strategy with a functional block diagram. This indicates the actions triggered in each simulation frame for respectively the simulation manager and each EV agent. In the creation of EVs shown in the left side of the diagram, the pathfinding incorporates the communications for asking some stations which are their first available slot from the expected arrival time. The pathfinding algorithm is further detailed with pseudocode and the C# programming code later in this section. After selecting one path, each EV books the corresponding station. The right side of the diagram introduces the behavior of each EV in each frame of the simulated animation. Each EV checks whether it has arrived to the destination after charging in its way. If not, it checks whether it is in a charging station. In this case, if the EV has already the battery full of energy, it leaves the station. Otherwise, it waits for its turn in the queue of the station, or charges its battery on its turn.

In the pathfinding of this strategy, EV agents use a further modification of A* algorithm that departs from the strategy introduced in the previous section. This modification is presented in the Algorithm 3. When each EV agent is managing the paths from the priority queue, it creates new paths from the neighbors. In the case of being a charging station, it creates a new edge with the same node. This edge does not only consider the charging time but also an estimation of the waiting time.

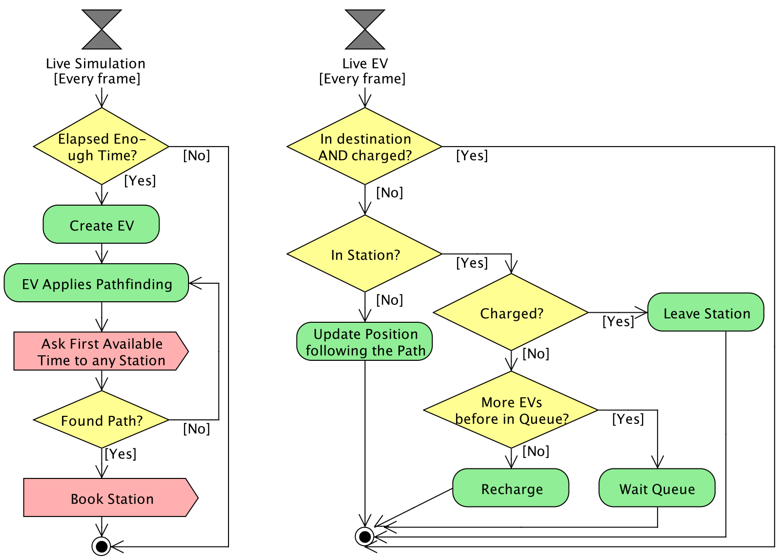


Figure 5: Functional block of the Booking strategy

Algorithm 3 Pathfinding algorithm using charging stations with booking systems

```
procedure PathfindingWithBooking(origin, destination, chargingTime, speed)
    queue ← new PriorityQueue()
    path ← new Path(origin)
    path.hasCharged ← false
    while (path.last ≠ destination) OR NOT path.hasCharged do
        if path.last.isStation then
            arrivingTime = Time.CurrentTime() + path.Duration()
            availableTime = Interaction.AskForBookInformation( path.last.chargingStationAgent, arrivingTime)
            waitingTime ← availableTime - arrivingTime
            duration ← waitingTime + chargingTime
            newPath ← path.Clone()
            isCharginInThisNode ← true
            newPath.Add(path.last, duration, isChargingInThisNode )
            newPath.hasCharged ← true
            queue.Insert(newPath)
        for edge ∈ path.last.neighbors do
            if edge.target ∉ path.NodesFromLastChargeOrOrigin() then
                newPath ← path.Clone()
                duration ← edge.distance / speed
                isChargingInThisNode ← false,
                newPath.Add(edge.target, duration, isChargingInThisNode )
                queue.Insert(newPath)
        path ← queue.PullFirst()
        queue.RemoveFirst()
    return path
```

In order to be accurate with the waiting time, the EV agent interacts with the corresponding charging station agent, and asks about the first available time slot for charging from the earliest time that the EV agent can get there. The charging station agent indicates the soonest available time slot considering the previous bookings and the necessary charging time. The algorithm calculates the waiting time based on this response, and adds a new path fragment considering the duration (i.e. the waiting time plus the charging time), and marks this path fragment as charging. Since this strategy uses a booking system, it is referred as "Booking" strategy.

The algorithm finishes when it detects a path that reaches the destination going through a charging station. When found, the path is the shortest in time because of the properties of the A* algorithm [28] and since the priority queue is sorted by the durations of paths. In that moment, the EV agent selects this path, and contacts the corresponding charging station agent of the path to actually book the specific time slot. This booking is considered by subsequent agents in the simulation.

The original A* algorithm discards the neighbors that are already visited for a given path, to avoid paths with cycles that could hinder the performance and the termination property. In the presented modification, this slightly changes since sometimes a EV may need to charge in a particular station and then come back through a part of the same path. For this reason, the proposed modified version only guarantees separately that (a) the path does not contain repeated nodes between the ones before charging, and (b) the same for the nodes after charging. Thus, when checking the visited nodes, this algorithm only uses the list of nodes from the last charging stop (if the EV has been charged in this path) or from the origin (otherwise).

Since both pathfinding mechanisms share most of the code, we implemented a general solution for supporting both pathfinding submodels. This facilitates the maintainability of the programming code for possible extensions. Figure 6 shows this general solution with the C# programming language. The "Find Shortest Path" method is used for both pathfinding strategies, and all the programming lines are the same for both pathfinding strategies. The different between these is encapsulated in the "Correct Edge Distance" method. This modifies the distance of an edge representing a recharge only in the case of using the Booking strategy. If so, it communicates with the corresponding recharging station for adding a delay to the estimated time. This delay is calculated from the first available slot in the queue of bookings of the station.

```csharp
// It finds the shortest path from one node to another. The nodes are represented with
// the origin index and the target index. It returns an empty path in case of error.
// A paramater indicates whether the car needs to recharge.
public Path FindShortestPath(int indexOrigin, int indexTarget, bool needToRecharge=true){
    // Create the priority queue
    SortedList queue = new SortedList(new PathComparer());
    // Get the neighbors and add the paths to a list
    List<Edge> neighbors = graph.GetNeighbors (indexOrigin);
    if (neighbors.Count == 0)
        return new Path ();
    foreach (Edge edge in neighbors) {
        Path path = new Path ();
        path.AddEdge (edge);
        queue.Add (path, path.GetDistance ());
    }
    // Repeat until reaching the target or emptying the list
    while (queue.Count>0){
        Path firstPath = (Path) queue.GetKey (0);
        queue.RemoveAt (0);
        // Condition of goal accomplished
        if ((firstPath.GetIndexEnd () == indexTarget) &&
                (firstPath.hasRecharged || !needToRecharge))
            return firstPath;
        // Add the neighbors that has not been visited after the recharge if any
        neighbors = graph.GetNeighbors (firstPath.GetIndexEnd());
        foreach (Edge edge in neighbors) {
            // It modifies the edge distance if estimating delays for recharging
            CorrectEdgeDistance(edge, firstPath);
            // Allow the recharging edge to repeat node
            if (!firstPath.ContainsNode (edge.indexEnd) || edge.recharge) {
                Path path = firstPath.Clone ();
                path.AddEdge (edge);
                queue.Add (path, path.GetDistance ());  //
            }
        }
    }
    return new Path();
}

// This method corrects the distance of recharging edges when using the Booking strategy
public void CorrectEdgeDistance(Edge edge, Path firstPath){
    if (SimParameters.usingBooking && edge.recharge) {
        Node station = graph.GetNode (edge.indexInit);
        float fromTime = firstPath.GetDistance ();
        float firstAvailableTime = station.FirstAvailableTime (fromTime);
        float delay = firstAvailableTime - fromTime;
        edge.distance += SimParameters.speedCar * delay;
    }
}
```

Figure 6: Programming code of the pathfinding in C# language

3.8. Definition of new strategies with this framework

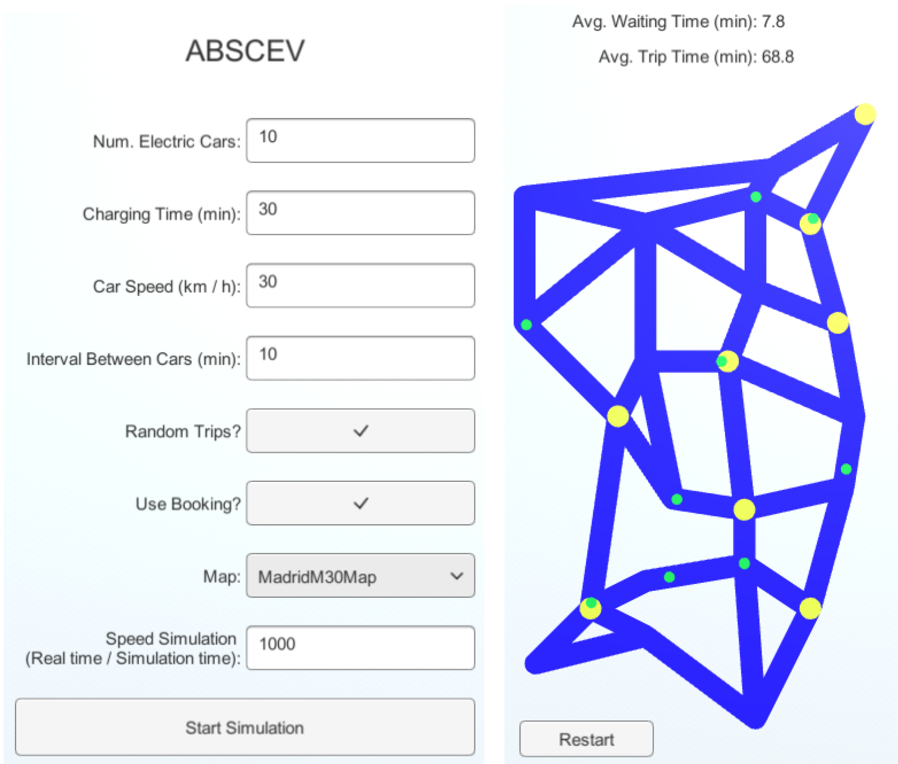
This framework provides the necessary tools for implementing different pathfinding and coordination algorithms and testing whether these can reduce waiting times in charging stations and the global trip times or improve in any other aspect. In particular, the new pathfinding algorithms can be added as new functions within the "Pathfinding" class of the framework.

In order to implement new coordination mechanisms, the developer can extend the EV agent and/or the charging station agents. For example, the coordination of EV agents can be implicit through charging station agents, but incorporating mechanisms different from booking systems. Another option can be that EV agents communicate through explicit messages. In this case, if the communication is based on the locations, for example each EV agent could communicate with each other EV agent nearby the path the former agent is considering to take.

3.9. User interface

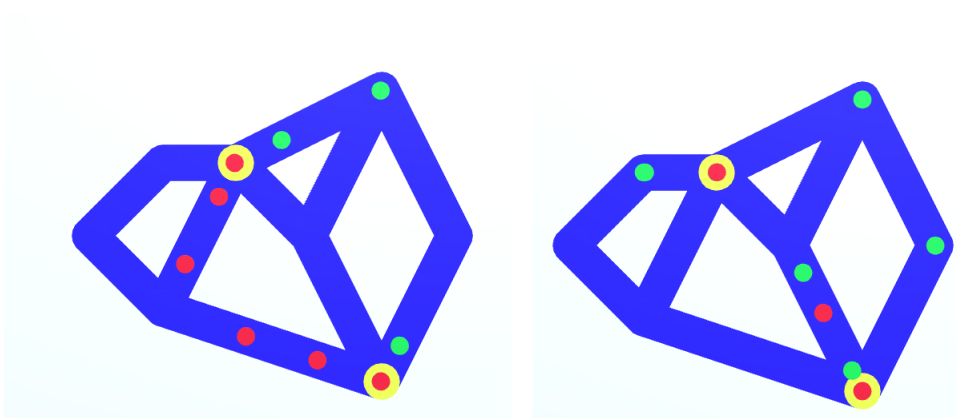
Figure 7 presents the user interface of ABSCEV. More concretely, Figure 7(a) shows the interface for setting the input parameters of simulations. These include (1) the number of electric cars, (2) the necessary time for charging a car battery, (3) the speed of cars, (4) the frequency of cars for starting driving, determined with the time interval between cars, (5) whether the cars start and end at different random points (denoted as "random trips" in the interface) or if all of them share the same origin and destination, (6) whether the simulation uses the Booking strategy, and (7) the map of the simulation. Notice that new maps can be easily added by means of text files. Finally, the user can establish the speed of the simulation as the ratio of the simulated real-time divided by the actual simulation time. For example, a speed of 1000 indicates that 1000 real-time seconds of the cars will be simulated in only one second in the simulator. Figure 7(b) shows an example of the final results of a simulation. One can observe that the top of the application shows the resulting average waiting time and the resulting average trip time.

Figure 8 shows examples of simulation executions in a basic map. The charging stations are represented as yellow circles. The red circles represent electric cars that have not charged their batteries in the current simulation yet. By contrast, green circles represent electric cars that have already charged their batteries in the simulation. The streets are represented with blue lines. In the simulations, one can observe that cars with low-level energy



(a) User input (b) Simulation results

Figure 7: User interface of ABSCEV



(a) Same origin and destination (b) Random origins and destinations

Figure 8: Example of simulations with different options about the origins and destinations

sequentially appear (i.e. red ones). Each car goes to a charging station, and stays there for a while, simulating both the possible waiting time and the charging time. Then, it changes its color to green representing that it has a high level of energy, and continues its path towards its destination.

The simulation of the same trips (referring only to the same origin and destination) is useful for understanding the system. Figure 8(a) shows an example for this kind of simulation. It uses booking systems, and one can observe that the vehicles alternatively take different paths even if one of these is slightly larger, in order to avoid the waiting time in charging stations and reduce the global trip time.

The simulation of trips with random beginning and target locations allows the user to obtain more realistic situations. Figure 8(b) shows an example of this kind of simulations. The study of these simulations requires to perform a representative number of executions to avoid bias of the results due to the nondeterministic behavior of the simulator.

This application was deployed as both as a mobile application and as a desktop application, thanks to the fact that the used engine (i.e. Unity) is cross-platform.

4. Evaluation

The validation of ABSCEV was performed in two different levels. First, the basic functioning of the simulator was checked with simple situations. Each of these situations was aimed at checking a specific property that the simulator should have to be consistent. The second level of validation used more complex cases that were inspired by real scenarios. The goal was to assess the global emergent behavior of the system to draw representative conclusions. The subsections of this sections present respectively these two levels.

4.1. Validation of individual rules

In order to guarantee the proper functioning of the ABS, we checked the following individual rules in two simple maps created for this purpose:

- We selected some origin location and a destination in each map. We executed both the Distance and the Booking strategies for only one car. We checked that in both cases the car departed from the origin location and ended its trip in the destination location.

- We selected an origin location and a destination location in a map so that the shortest path did not go through any charging station. In both strategies, we checked that (a) a simulated car followed a path that went through a charging station, (b) it actually stayed there for a while, and (c) afterwards it proceeded with a high level of energy.

- In the same previous scenario, we checked that the car with the Distance strategy followed the shortest path from the ones that went through a charging station.

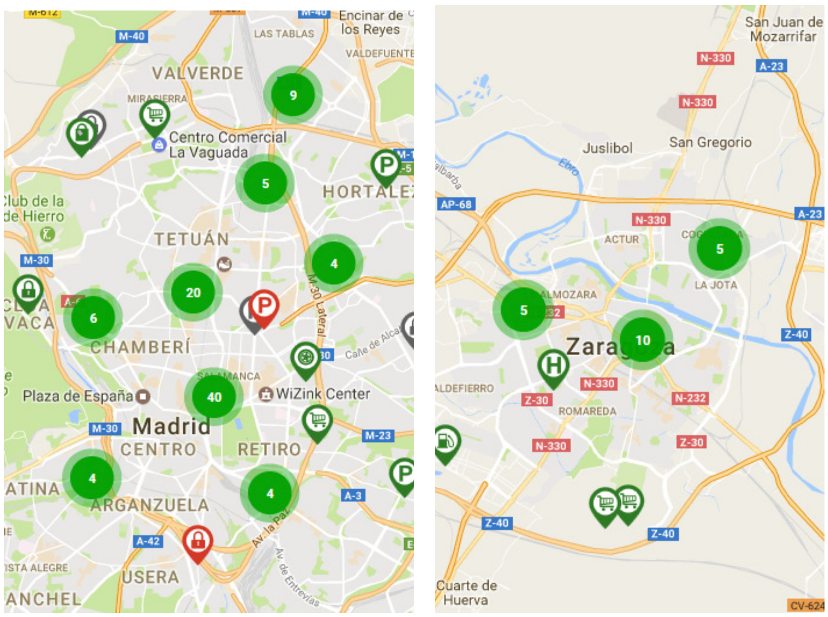
- In this same scenario, we also confirmed that a car with the Booking strategy selected the shortest path from the ones that went through a charging station. This behavior is appropriate since there could not be any waiting time for starting to charge, since there was only one car in the simulation.

- In each map, we chose an origin and a destination, so that there were several possible paths that went through different charging stations. These paths had different distances. In this scenario, we executed 10 cars with the Distance strategy, and we checked that all the cars selected the shortest path in distance that went through a charging station. We also confirmed that each car waited in the charging station until it was available. In this way, the first car charged directly without waiting, while the second one waited until the first one completed the charge. The third one waited until the two previous ones finished, and so on. We checked that every car arrived to the target location with the battery charged.

- In the same scenario as the previous rule, we executed 10 cars with the Booking strategy. We checked that the cars alternatively selected paths through different charging stations, to avoid the waiting times in the stations. We confirmed that the Booking strategy reduced both the average waiting time in the charging stations and the average trip times in comparison with the Distance strategy in the previous test. We also observed that every car arrived at the destination and had been charged.

4.2. Validation in two real scenarios

In order to assess the global emergent behaviors of the current approach, we applied it in two different Spanish cities. We used the information avail-



(a) Madrid within M30 road (b) Zaragoza up to Z40 road

Figure 9: Maps of the main charging stations of two Spanish cities

able from the website *electromaps.com* (see some examples in Figure 9). This website provides a geographical information system that informs of all the charging stations indicating some of its features. It aggregates the areas where there are several charging points when zooming out.

More concretely, Figure 9(a) shows a simplified version of the main areas or charging stations in the center of Madrid city, delimited by the surrounding M30 road. Each area of charging stations is represented with a green circle. In a similar way, Figure 9(b) shows a similar map for the Zaragoza city delimited at the south and east by the Z40 road. The current experiments used these two maps considering only the most relevant streets and places. These maps were represented as text files with the notation of the current approach, so that the simulator was able to load them.

First, we used the fixed origin and destination locations to compare both strategies. Thus, we ran the simulator respectively with (a) the Distance strategy, which followed the shortest paths in distance and (b) the Booking strategy with the proposed booking systems.

Then, we repeated the comparison but this time using random origin and

	Fixed trips (both cities)	Random Trips Madrid	Zaragoza
Num Cars	10	20	20
Charging time (min)	30	30	30
Car speed (km/h)	30	30	30
Interval between cars (min)	10	3	10

Table 1: Values of the input parameters.

destination locations. In this occasion, we executed each strategy 100 times, to avoid bias because of the nondeterministic behaviors.

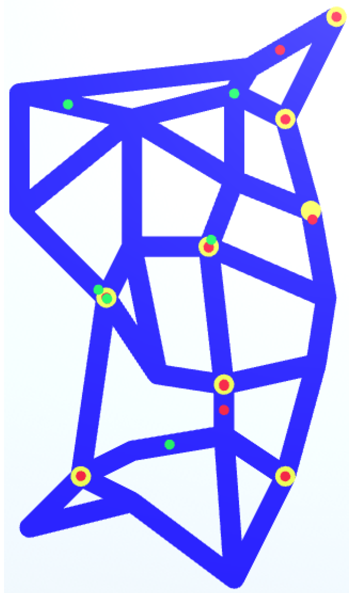
All these experiments were repeated in the same way for the two maps, respectively associated with the Madrid and Zaragoza cities.

Table 1 shows the input parameters used in the simulations. In the fixed trips (i.e. with fixed origin and destination), all the same inputs were used for both cities. In the case of the random trips, the number of cars was increased to have more representative data due to the nondeterministic aspect of this simulations. In addition, the number of cars was increased to maintain the possibility of having coincidences in the charging stations, since the coincidences normally decreased when having different origin and destination locations. In addition, since Madrid was represented with a higher number of charging stations than Zaragoza, we selected a higher frequency for starting cars in the former case (represented with a shorter time interval between generated cars) to simulate scenarios in which the charging stations needed to be shared efficiently.

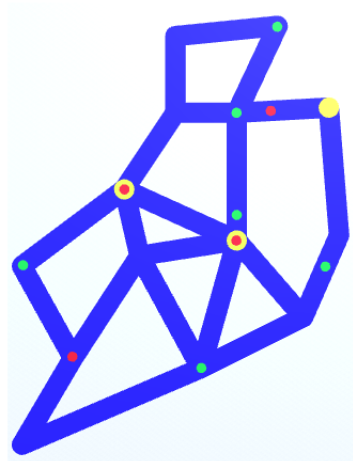
Since the maximum car speed allowed in Spanish cities is 50 km/h, we used a lower average speed (i.e. 30 km/h) to consider the possible traffic conditions and the stops because of traffic lights. In addition, we selected 30 min as the charging time of the fast charging stations, as it is one of the most common charging times for this kind of stations [29].

Figure 10 shows examples of simulations for both cities with the parameters of these experiments. In both examples, one can observe some EVs with low levels of energy (red circles), each of which either (a) going to a charging station on the way to its destination or (b) getting charged in one of the stations (yellow circles). One can also observe EVs already charged (green circles) going to their destinations. In these particular examples, both simulations were using the Booking strategy.

Figure 11 compares the waiting times of EVs in charging stations for the different simulation scenarios. Each scenario is determined by (a) the city, and (b) the way of selecting the current and target locations. In the



(a) Madrid



(b) Zaragoza

Figure 10: Examples of simulations of the experiments

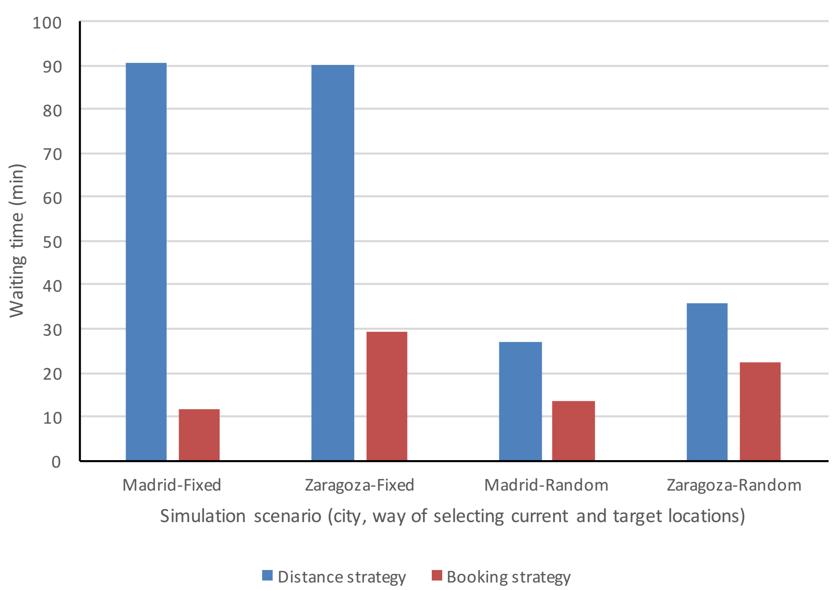


Figure 11: Comparison of waiting times between the strategies for different scenarios

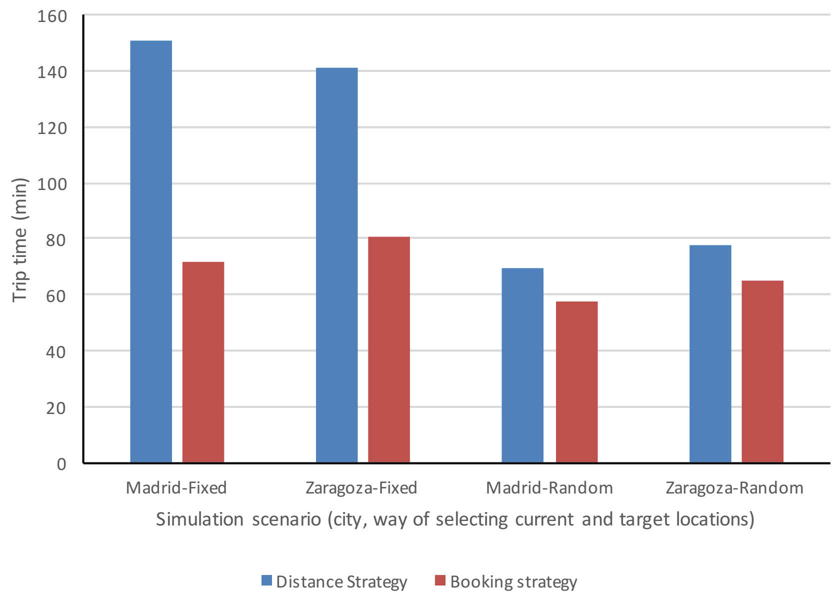


Figure 12: Comparison of trip times between the strategies for different scenarios

cases of random selection of current and target locations, each result shows the average of 100 simulations, to avoid bias due to its nondeterministic behavior. It is worth noting that the reductions of waiting times of Booking strategy over Distance were higher when using the same current and target locations (i.e. in Fixed scenarios) than when having a variety of origin and target locations (i.e. in Random scenarios). In particular, in the Fixed scenarios, the Booking strategy obtained reductions of 86.9% and 67.7% for respectively Madrid and Zaragoza. In the Random scenarios, the reductions were respectively 49.8% and 37.9 % for respectively Madrid and Zaragoza. One can also observe that the reduction of waiting time was higher in Madrid (68.4%) than in Zaragoza (52.8%), and the former city was represented with a bigger map than the latter one.

Figure 12 shows the comparison of trip times in the different scenarios. The trip time was measured between the departure and arrival of each EV. It is worth mentioning that Booking strategy sometimes used longer paths if these reduced the trip time by avoiding unnecessary waiting time in charging stations. Although Booking strategy obtained high reductions in waiting times over Distance strategy, the trip time reductions were usually lower

		Distance Strategy		Booking Strategy		Time reduction (%)	
Trip kind	City	Waiting time (min)	Trip time (min)	Waiting time (min)	Trip time (min)	Waiting time	Trip time
Fixed	Madrid	90.6	150.4	11.9	71.9	86.9	52.2
	Zaragoza	90.2	141.3	29.1	80.3	67.7	43.2
	Average	90.4	145.9	20.5	76.1	77.3	47.7
Random	Madrid	27.0 (8.7)	69.2 (8.8)	13.6 (3.4)	57.3 (4.0)	49.8	17.2
	Zaragoza	35.8 (14.9)	77.5 (15.0)	22.2 (7.2)	65.0 (7.4)	37.9	16.1
	Average	23.8	51.9	13.0	42.1	43.9	16.6
Total		50.5	89.5	16.0	55.7	60.6	32.2

Table 2: Results of the experiments.

as EVs frequently spent more time on the road for avoiding these occupied charging stations. In fact, the global trip time reduction in this chart (i.e. 32.2%) is lower than the global reduction in the previous chart about waiting times (i.e. 60.6%). In the case of trip times, the reductions in Fixed scenarios (i.e. 47.7% in average) were also higher than in Random scenarios (i.e. 16.6% in average). In the case of trip time, the city represented with a larger map (i.e. Madrid) also obtained a higher time reduction (34.7% in average) than the other city represented with a smaller map (29.7% in average).

Table 2 presents the results for all the aforementioned configurations of input parameters, indicating both the resulting average waiting times and average trip times. In the case of random trips, it includes the standard deviations of the 100 simulations between parentheses. This table also indicates the reduction percentages of respectively the waiting and trip times for each case. These reduction percentages are calculated with the formula $100 * (d - b)/d$ where d and b are respectively the times for the Distance and Booking strategies. One can observe that in all the configurations, the average waiting time was reduced at least 37% and the average trip time was reduced at least 16%.

Furthermore, we applied the Welch's t-test (also known as "unequal variances t-test") [30] to determine whether the reduction of the waiting time and the trip time was statistically significant in each configuration with random trips. We selected this test as it is appropriate for comparing two independent samples and it is more robust than Student's t-test for unequal variances. In addition, we also used the Brown-Forsythe test for equality of means, since it is also robust for unequal variances [31]. Table 3 shows the results of these tests. As one can observe, the improvements of the Booking strategy from the alternative are very statistically significant (under a .001 significance level) for both the waiting and trip times in the two cities.

			Statistic[a]	df1	df2	Sig.
Waiting Time	Madrid	Welch	207.888	1	128.358	.000
		Brown-Forsythe	207.888	1	128.358	.000
	Zaragoza	Welch	67.286	1	142.761	.000
		Brown-Forsythe	67.286	1	142.761	.000
Trip Time	Madrid	Welch	149.616	1	137.568	.000
		Brown-Forsythe	149.616	1	137.568	.000
	Zaragoza	Welch	55.441	1	144.479	.000
		Brown-Forsythe	55.441	1	144.479	.000

Table 3: Results of robust tests about equality of means for comparing the results of random trips. [a]Asymptotically F distributed.

	Waiting time	Trip time
Madrid	2.04	0.71
Zaragoza	1.16	0.64

Table 4: Cohen's d effect sizes of for the comparison of means between strategies.

The effect sizes were measured with Cohen's d as normally done for independent t-tests, and table 4 shows the results. These results can be interpreted with the Cohen's guidelines [32] that associated 0.2, 0.5 and 0.8 respectively with small, medium and large effects. In addition, [33] added the category of very large for the values above 1.3. Following these interpretations, the booking strategy reduced the waiting time in charging stations with large and very large effect sizes respectively for the two case studies. In addition, the trip time was reduced with medium-large effect sizes in both case studies.

In order to assess the fluctuations and possible overloads of the execution times of pathfinding, we measured these times for the two city scenarios and the two strategies, with simulations of 200 EVs in each case. We used the random selection of origins and destinations to test more realistic scenarios. Figure 13 shows the execution times in milliseconds of the pathfinding of each EV in the Zaragoza scenario. It compares the execution times for both strategies. The abscissa represents the position of the vehicle in the chronological order of simulated vehicles. One can observe that most pathfinding execution times had low values with averages of 6.55 ms and 5.17 ms for respectively the Booking and Distance strategies. These averages were substantially lower than the maximum execution times, which were respectively 19 ms and 18 ms. One can observe that the pathfinding executions recurrently had some peaks, but the execution times were not excessive. These time fluctuations may be due to the random selection of origins and destinations. Another reason might be that many charging stations could have the

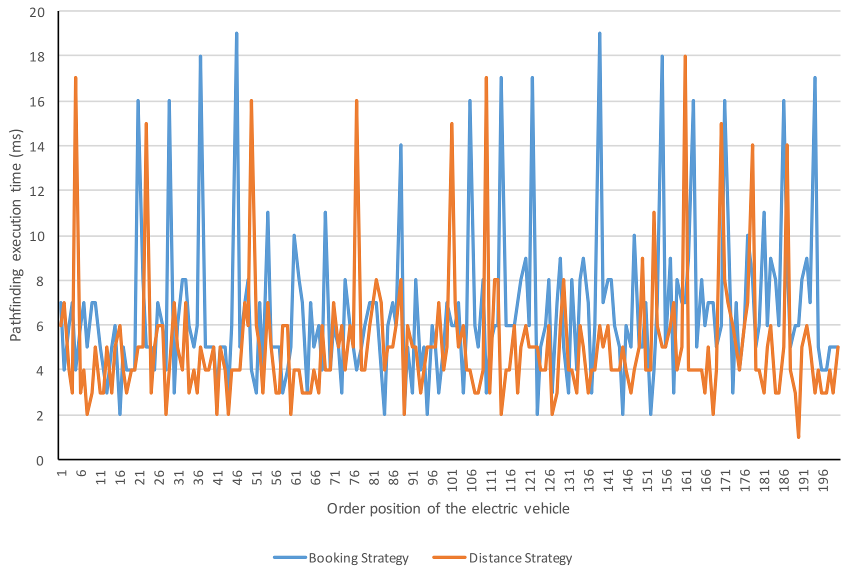


Figure 13: Pathfinding execution times in the Zaragoza scenario

queue full and then the algorithm had to search more alternatives.

The most relevant aspect is that the chart does not show any overloading patterns over the time. In other words, the values followed a neutral trend (i.e. neither increasing nor decreasing) despite their fluctuations.

The Booking strategy had an increase of the average execution time (26.5%) and the maximum execution time (5.6%) over the Distance strategy. However, it is probably worth this increase for the reduction of both waiting and trip times for the drivers, since the execution time increase of this amount of milliseconds would probably not be noticed by the final users.

Figure 14 shows the pathfinding execution times for the Madrid scenario. As one can observe, the conclusions of the previous scenario were confirmed in this one. The average execution times were low (118.7 ms and 99.4 ms for respectively Booking and Distance strategies) in comparison with the maximum cases (i.e. 907 ms and 872 ms). It is probably worth the increase of all these values from Distance to Booking strategies (19.4% and 4% respectively for averages and maximums) for the reduction of trip and waiting times. The increase of the execution times from the Zaragoza scenario to the Madrid one may be due to the differences of paths distances and map sizes. This scenario neither presented any overloading pattern over the time.

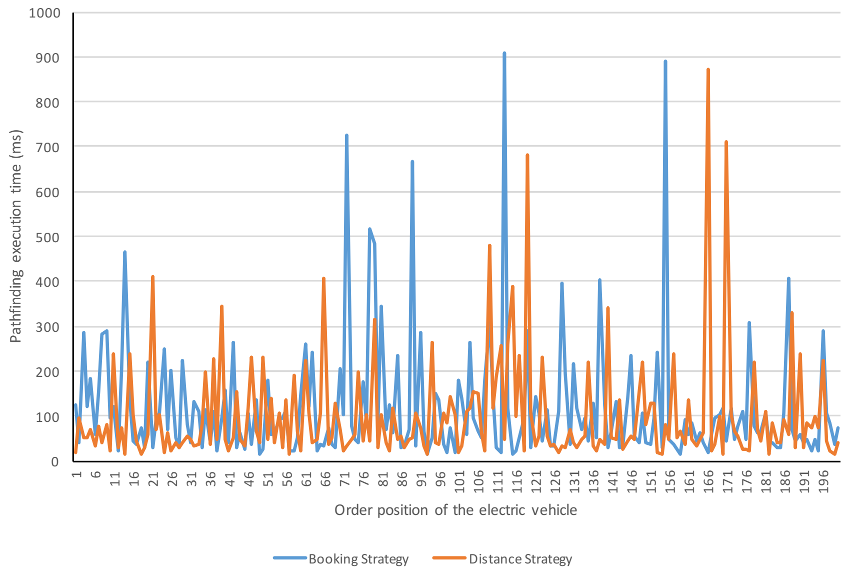


Figure 14: Pathfinding execution times in the Madrid scenario

Moreover, we also measured the number of messages interchanged in the pathfinding of the adaptive Booking strategy between cars and charging stations. The measurement of communications has shown to be especially relevant in both VANETs and MASs communities [34]. Notice that the basic Distance strategy does not communicate with charging stations so the number of messages was zero for all cases. We used simulations of 200 EVs as in the previous experiments. Figures 15 shows the results of the number of messages for the Zaragoza scenario. One can observe an average of 110.51 messages per pathfinding while the maximum was 208 messages. The fluctuations of the numbers of messages were lower than for execution time. Figure 16 presents the same experiments for the Madrid scenario. The average was 2247.06 messages and the maximum 8960 messages. The results were higher probably due to the bigger map and the greater number of charging stations. In both scenarios, one can observe that there was not any overloading patter over the time, since the numbers of messages followed neutral trends despite their fluctuations.

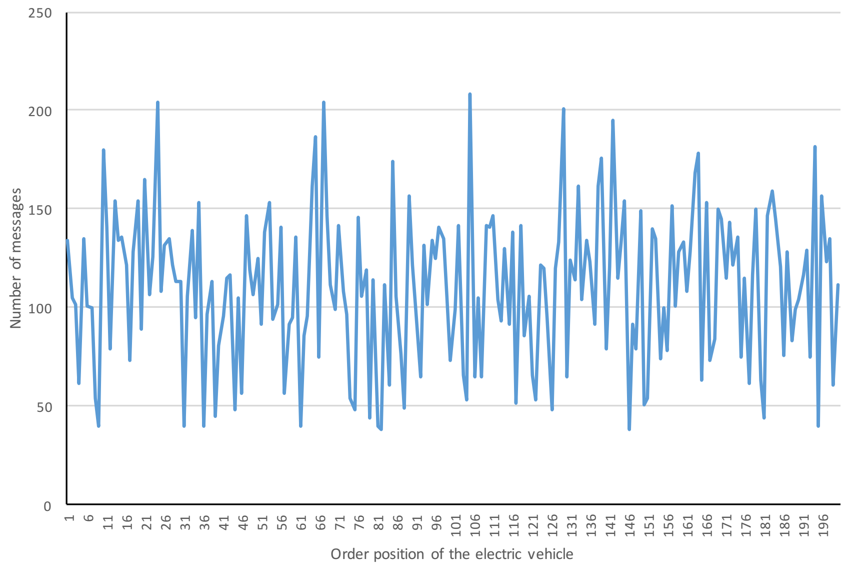


Figure 15: Number of messages in the Zaragoza scenario

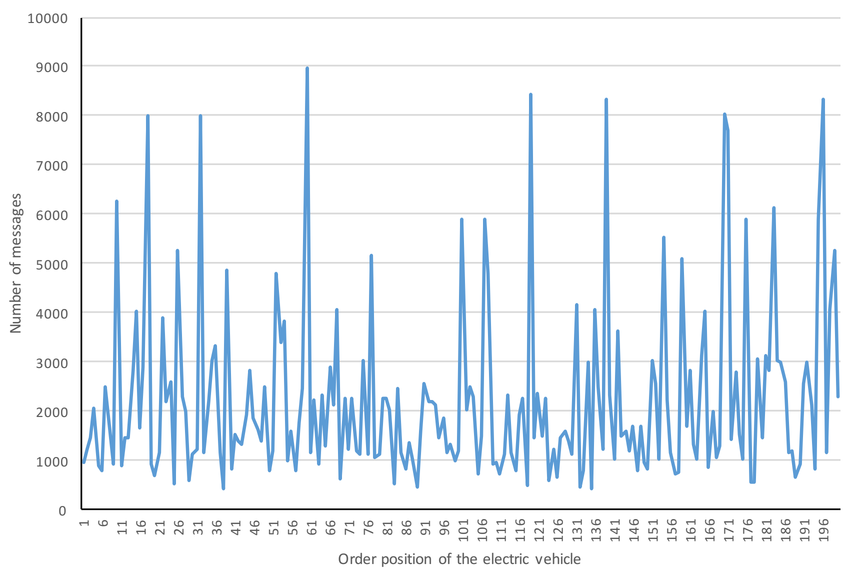


Figure 16: Number of messages in the Madrid scenario

5. Discussions

The proposed framework has shown its utility in simulating groups of EVs that need to travel from certain origins to some destinations and stop on their ways for charging their batteries in fast charging stations. In particular, this framework has simulated several strategies and has loaded different maps. For its illustration, a strategy based on booking systems has been compared with a control mechanism. The control mechanism represents a basic system for selecting the shortest path going through a charging station. However, it could also represent the drivers that do not use the proposed system and they just select the shortest path with a charging station. The results showed a statistically significant reduction of the average waiting times in charging stations and the average trip times in general. In addition, the evolutions of the pathfinding execution times and the number of messages did not show any overloading pattern over the time.

In the simulation results, the simulations in which the departure and destination locations were shared among the different EVs, the Booking strategy had a stronger repercussion in reducing waiting and trip times (i.e. with higher reduction percentages). That makes sense since when several EVs share the same path, there are normally more coincidences in the charging stations and consequently the waiting queues are usually longer. Therefore, the presented approach will probably be more useful in cities in which many citizens have similar paths at the same time. The analysis with the current framework can assist engineers in designing the appropriate mechanisms of coordination in certain cities for preventing users from suffering unnecessary waiting times.

The reduction percentage of waiting times is higher than the reduction percentage of global trip times. EVs can avoid waiting times by distributing with different charging stations in a balanced way. However, the trip time can never be reduced below a certain amount (i.e. the one needed for the shortest path). In addition, the Booking strategy sometimes decides to take a larger path in distance in order to reduce the waiting time and the global trip time. Hence, the trip time reduction is less as it also counts the time spent in this larger path. Even though, the algorithm of the Booking strategy prioritizes the reduction of the trip time, so it only takes longer paths if it reduces the total trip time. In fact, drivers normally prefer the reduction of the total trip time (including the time spent in the charging station and the one driving) in comparison to just reduce the waiting time, since in the former case drivers arrive sooner at their destinations.

It is worth mentioning that the presented global booking mechanism can be considered as a global greedy algorithm, since each EV agent chooses a path and establishes its booking without being able to be altered afterwards. If the information of all the drivers were known in advance, maybe a different scheduling mechanism could be theoretically better. However, in real scenarios drivers normally manifest the desire of taking the EVs and charging it at a given time, when the information of future drivers is normally not available. Thus, the proposed mechanism can be useful for common situations of drivers. The departure times can be accurate with the current approach, since the drivers or their software assistants would report the departure time when they were actually taking the car. This could be especially easy to use, if the algorithm was integrated with the navigation system of the car. The driver would introduce the destination and the need of charging, and the system would guide them through the path that is probably the shortest in time.

The current simulator can have more utilities than the ones initially presented in this article. For instance, the current simulator can be useful to assess which locations can be appropriate for building charging stations in a given city. For example, ABSCEV can simulate the repercussions of adding a new charging station in different locations in a given map with a certain frequency of cars needing to charge their batteries. The results can be compared to determine which of the considered locations will probably reduce more the average waiting and trip times.

It is worth mentioning that the current work is mainly dedicated to the coordination of charging EVs. However, this problem could be more complex when combining it with other factors such as traffic conditions. In this case, the estimated locations of traffic jams should be considered besides the information about charging stations.

6. Conclusions and future work

The current work has presented an ABS framework for testing different pathfinding and coordination strategies for EVs that go from one place to another with the restriction that they need to charge their batteries in a fast charging station on their ways. The ABS framework can simulate any map that is represented with the appropriate text format. The current work has compared two illustrating strategies, which respectively obtain the shortest path in distance and the shortest path in time. The later was founded on the use of booking systems in charging stations. Both strategies have been experienced in two different-sized Spanish cities. The results show that the strategy based on the booking of stations actually reduced both the waiting times in stations and the trip times with statistically significant differences. The pathfinding execution times and the number of interchanged messages did not have any overloading over the time. Therefore, the current ABS framework has proven to be useful for testing different pathfinding and coordination strategies for EVs in several city scenarios.

The current ABS is planned to be extended for including more realistic features. First, EVs will be able to have different charging times, regarding what is their exact levels of battery. In some cases, the EVs may have the restriction of a maximum distance that they can drive before they run out of battery energy. Furthermore, the simulator will consider the possibility that each charging station can have several plug-ins to charge simultaneously several EVs up to a certain number. This will be useful to perform larger simulations with more EV agents. In addition, we will consider the influence of temperature on the performance of EV batteries in a future version of the simulator. We will analyze whether all these changes influence the results somehow. In addition, the simulator will be experienced with more cities to determine whether the conclusions of the current work are generalizable. Moreover, we will encourage other researchers to use the provided ABS framework for testing different strategies for charging EVs, so they can give us feedback to improve the ABS framework. Furthermore, this simulator may be integrated with a traffic simulator, in order to provide appropriate solutions for considering both the traffic conditions and the need of charging batteries. In addition, we may incorporate the possibility of having different charging prices in the simulator, so that the simulations can also be influenced by economic factors.

Acknowledgments

We acknowledge the research project "Construcción de un framework para agilizar el desarrollo de aplicaciones mviles en el ámbito de la salud" funded by University of Zaragoza and Foundation Ibercaja with grant reference JIUZ-2017-TEC-03. This work has been supported by the program "Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores" funded by the Spanish Ministry of Education, Culture and Sport with reference CAS17/00005. We also acknowledge support from "Universidad de Zaragoza", "Fundación Bancaria Ibercaja" and "Fundación CAI" in the "Programa Ibercaja-CAI de Estancias de Investigación" with reference IT1/18. This work acknowledges the research project "Desarrollo Colaborativo de Soluciones AAL" with reference TIN2014-57028-R funded by the Spanish Ministry of Economy and Competitiveness. It has also been supported by "Organismo Autónomo Programas Educativos Europeos" with reference 2013-1-CZ1-GRU06-14277. We also acknowledge support from project "Sensores vestibles y tecnologa móvil como apoyo en la formacin y práctica de mindfulness: prototipo previo aplicado a bienestar" funded by University of Zaragoza with grant number UZ2017-TEC-02.

References

[1] K. Kaygusuz, Renewable energy: Power for a sustainable future, Energy exploration & exploitation 19 (6) (2001) 603–626.

[2] I. Karakitsios, E. Karfopoulos, N. Hatziargyriou, Impact of dynamic and static fast inductive charging of electric vehicles on the distribution network, Electric Power Systems Research 140 (2016) 107–115.

[3] K. Z. Ghafoor, J. Lloret, K. A. Bakar, A. S. Sadiq, S. A. B. Mussa, Beaconing approaches in vehicular ad hoc networks: a survey, Wireless personal communications 73 (3) (2013) 885–912.

[4] K. Z. Ghafoor, K. A. Bakar, J. Lloret, R. H. Khokhar, K. C. Lee, Intelligent beaconless geographical forwarding for urban vehicular environments, Wireless networks 19 (3) (2013) 345–362.

[5] R. Lacuesta, J. Gallardo, J. Lloret, G. Palacios, Integration of data from vehicular ad hoc networks using model-driven collaborative tools, Mobile Information Systems 2016 (2016) Article ID 4291040, 15 pages.

[6] S. Kim, Y.-J. Son, Y. Tian, Y.-C. Chiu, C. D. Yang, Cognition-based hierarchical en route planning for multi-agent traffic simulation, Expert Systems with Applications 85 (2017) 335–347.

[7] S. Kamboj, W. Kempton, K. S. Decker, Deploying power grid-integrated electric vehicles as a multi-agent system, in: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1,

International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 13–20.

[8] M. D. Galus, G. Andersson, Demand management of grid connected plug-in hybrid electric vehicles (PHEV), in: Energy 2030 Conference, 2008. ENERGY 2008. IEEE, IEEE, 2008, pp. 1–8.

[9] E. Xydas, C. Marmaras, L. M. Cipcigan, A multi-agent based scheduling algorithm for adaptive electric vehicles charging, Applied Energy 177 (2016) 354–365.

[10] X. Xi, R. Sioshansi, V. Marano, Simulation–optimization model for location of a public electric vehicle charging infrastructure, Transportation Research Part D: Transport and Environment 22 (2013) 60–69.

[11] K. Coninx, R. Claes, S. Vandael, N. Leemput, T. Holvoet, G. Deconinck, Anticipatory coordination of electric vehicle allocation to fast charging infrastructure, in: Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection, Vol. 8473 of Lecture Notes in Computer Science, Springer, 2014, pp. 74–85.

[12] I. García-Magariño, J. J. Gómez-Sanz, J. R. Pérez-Agüera, A multi-agent based implementation of a delphi process, in: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems, AAMAS, Vol. 3, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1543–1546.

[13] I. García-Magariño, S. Rougemaille, R. Fuentes-Fernández, F. Migeon, M.-P. Gleizes, J. Gómez-Sanz, A tool for generating model transformations by-example in multi-agent systems, in: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Springer, 2009, pp. 70–79.

[14] O. Tayan, Y. M. Alginahi, M. N. Kabir, A. M. Al BinAli, Analysis of a transportation system with correlated network intersections: A case study for a central urban city with high seasonal fluctuation trends, IEEE Access 5 (2017) 7619–7635.

[15] D. Cerotti, S. Distefano, G. Merlino, A. Puliafito, A crowd-cooperative approach for intelligent transportation systems, IEEE Transactions on Intelligent Transportation Systems 18 (6) (2016) 1529 – 1539.

[16] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, D. Rus, On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment, Proceedings of the National Academy of Sciences 114 (3) (2017) 462–467.

[17] E. L. Karfopoulos, N. D. Hatziargyriou, A multi-agent system for controlled charging of a large population of electric vehicles, Power Systems, IEEE Transactions on 28 (2) (2013) 1196–1204.

[18] J. Xiong, K. Zhang, Y. Guo, W. Su, Investigate the impacts of pev charging facilities on integrated electric distribution system and electrified transportation system, IEEE Transactions on Transportation Electrification 1 (2) (2015) 178–187.

[19] I. García-Magariño, A. Gómez-Rodríguez, J. C. González-Moreno, G. Palacios-Navarro, PEABS: a process for developing efficient agent-based simulators, Engineering Applications of Artificial Intelligence 46 (2015) 104–112.

[20] I. García-Magariño, G. Palacios-Navarro, R. Lacuesta, TABSAOND: A technique for developing agent-based simulation apps and online tools with nondeterministic decisions, Simulation Modelling Practice and Theory 77 (2017) 84–107.

[21] D. Gregor, S. Toral, T. Ariza, F. Barrero, R. Gregor, J. Rodas, M. Arzamendia, A methodology for structured ontology construction applied to intelligent transportation systems, Computer Standards & Interfaces 47 (2016) 108–119.

[22] A. Fernández-Isabel, R. Fuentes-Fernández, Analysis of intelligent transportation systems using model-driven simulations, Sensors 15 (6) (2015) 14116–14141.

[23] V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, S. F. Railsback, The odd protocol: a review and first update, Ecological modelling 221 (23) (2010) 2760–2768.

[24] D. V. De Macedo, M. A. Formico Rodrigues, Experiences with rapid mobile game development using Unity engine, Computers in Entertainment (CIE) - Theoretical and Practical Computer Applications in Entertainment 9 (3) (2011) 14.

[25] J. Pavón, C. Sansores, J. J. Gómez-Sanz, Modelling and simulation of social systems with ingenias, International Journal of Agent-Oriented Software Engineering 2 (2) (2008) 196–221.

[26] I. García-Magariño, M. Cossentino, V. Seidita, A metrics suite for evaluating agent-oriented architectures, in: Proceedings of the 2010 ACM Symposium on Applied Computing, ACM, 2010, pp. 912–919.

[27] J. Yao, C. Lin, X. Xie, A. J. Wang, C.-C. Hung, Path planning for virtual human motion using improved a* star algorithm, in: Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, IEEE, 2010, pp. 1154–1158.

[28] W.-J. Seo, S.-H. Ok, J.-H. Ahn, S. Kang, B. Moon, An efficient hardware architecture of the a-star algorithm for the shortest path search engine, in: INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on, IEEE, 2009, pp. 1499–1502.

[29] L. Dickerman, J. Harrison, A new car, a new grid, IEEE Power and Energy Magazine 8 (2) (2010) 55–61.

[30] G. D. Ruxton, The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test, Behavioral Ecology 17 (4) (2006) 688–690.

[31] M. B. Brown, A. B. Forsythe, The small sample behavior of some statistics which test the equality of several means, Technometrics 16 (1) (1974) 129–132.

[32] J. Cohen, Statistical power analysis for the behavioral sciences (2nd ed.), Hillsdale, NJ: Lawrence Earlbaum Associates, 1988.

[33] J. A. Rosenthal, Qualitative descriptors of strength of association and effect size, Journal of social service Research 21 (4) (1996) 37–59.

[34] C. Gutierrez, I. Garcia-Magariño, A metrics suite for the communication of multi-agent systems, Journal of Physical Agents 3 (2) (2009) 7–14.