Billhardt, H.; Fernández, A.; Lujak, M.; Ossowski, S.; Julian Inglada, VJ.; Paz, JFD.; Hernández, JZ. (2016). Towards smart open dynamic fleets. En Multi-Agent Systems and Agreement Technologies. Springer. 410-424. doi:10.1007/978-3-319-33509-4_32.

Towards Smart Open Dynamic Fleets

Holger Billhardt[1], Alberto Fernández[1], Marin Lujak[1], Sascha Ossowski[1], Vicente Julián[2], Juan F. De Paz[3], Josefa Z. Hernández[4]

[1]CETINIA, University Rey Juan Carlos, Madrid, Spain
`{holger.billhardt,alberto.fernandez,marin.lujak,sascha.ossowski}@urjc.es`
[2]DSIC, Universidad Politécnica de Valencia, Valencia, Spain
`vinglada@dsic.upv.es`
[3]BISITE, Universidad de Salamanca, Salamanca, Spain
`fcofds@usal.es`
[2]DIA, Universidad Politécnica de Madrid, Madrid, Spain
`phernan@fi.upm.es`

Abstract. Nowadays, vehicles of modern fleets are endowed with advanced devices that allow the operators of a control center to have global knowledge about fleet status, including existing incidents. Fleet management systems support real-time decision making at the control center so as to maximize fleet performance. In this paper, setting out from our experience in dynamic coordination of fleet management systems, we focus on fleets that are open, dynamic and highly autonomous. Furthermore, we propose how to cope with the scalability problem as the number of vehicles grows. We present our proposed architecture for open fleet management systems and use the case of taxi services as example of our proposal.

Keywords: Multiagent systems, coordination, open systems, dynamic fleet management, dynamic optimization

1 Introduction

The increase of human mobility and freight transportation in urban environments presents one of the challenges major urban cities in Europe and all over the world are faced with in today's society. It is one of the causes of congestion problems, inefficiencies in logistics and energy use, and air pollution in modern cities [1, 2]. To approach this challenge, innovative transportation solutions are required that allow for a more efficient use of resources (vehicles, energy resources, roads, etc.) but that assure at the same time flexible mobility solutions for both citizens and freight distribution. The idea of smart cities presents new challenges and requires new solutions related to traffic and transport. As a direct result, in the last years more and more systems that promote the shared use of vehicles have begun to emerge [3]. Solutions like public bicycle services, bike or car sharing systems, or applications like UBER, providing taxi services through "free" drivers, have the objective to improve human mobility

adfa, p. 1, 2011.

and at the same time reducing its cost. Also in the domain of freight distribution in the business sector the idea of "flexible" fleets that are composed on the fly by vehicles from possibly different owners has emerged. The goal is again to optimize the use of available resources but also to increase the flexibility in providing services with more and more demand fluctuations.

We call this type of solutions *open fleets*. Similar to the traditional fleet concept, an open fleet is operated by some entity that manages and coordinates the use of a limited set of resources in order to provide a specific transportation service. However, open fleets extend the traditional fleet concept towards a new dimension of openness: vehicles may join and leave the fleet at any time, and the capacity of the operator to control the fleet in its entirety may vary considerably. Both of those aspects imply the need for new solutions in the field of fleet management and fleet coordination.

In this paper we discuss the concept of open fleets and present preliminary work towards new solutions for management systems for these type of fleets. In section 2 we specify our notion of open fleet as compared to static and dynamic fleets. Section 3 presents an initial proposal for an architecture for a management system for open fleets. Section 4 defines an algorithm for assigning service tasks in a fleet management system in an efficient manner. Due to its decentralized nature, we believe that this algorithm is especially applicable for (very) large open fleets with high service demands. Finally, section 5 presents some conclusions and future work.

2 Fleet coordination: from static to open fleets

In this section we analyze different notions of fleets and discuss the requirements and possibilities for their efficient coordination. We proceed from simpler to ever more complex types of fleets, ending up with open fleets, which constitute the main focus of this paper.

In general, we conceive a fleet as a set of vehicles, possible of different types, that is used by some organization (*fleet operator*) with the aim of providing a specific transportation service. A transportation service comprises the fulfillment of several *service or transportation tasks*, in a given geographical region and distributed over time. And a transportation task consists in transporting objects (goods, humans, …) from one geographical position to another.

Usually the objective of any fleet operator is to improve the efficiency of the fleet operation. In particular, the aim is to maximize the quality of the service that is provided while minimizing the operational costs.

Regarding the quality of service, the objective is usually to reduce waiting and transportation times. However, other aspects may also be important, like reducing traffic congestions, an equalitarian and fair usage of the resources, etc.

The operation cost of a fleet, is composed of two components: i) a fixed cost of each vehicle, and ii) a cost for each transportation service. The latter depends on the type of vehicle that has done the service and the required distance of movements.

The efficiency of a fleet based transportation service depends on long term strategic decisions (e.g., the fixed number of vehicles in the fleet; the distributions of vehicle bases in the geographical region, etc.) and on the coordination of the fleet at the operational level. And different types of fleets require different coordination mechanisms for assuring an efficient operation.

Fleet Management Systems (FMS) are usually used to implement such coordination mechanisms. They have been used in a wide range of vehicle fleet-related applications in the fields of transportation, distribution and logistics. They can either support operators at a control center to take decisions both, at long term and at the operational level, or they can implement control and coordination strategies directly, without human intervention. In general, the objective of such systems is to improve the efficiency through an efficient coordination of the fleet, adapted to the specific transportation service that is provided.

2.1 Static Fleets

Static fleets operate in situations, where the number of vehicles and also the number of transportation tasks is constant (or almost constant) during the normal operation of the fleet. Typical examples of this kind of fleets are traditional public transport systems, like bus fleets, trains, metro, airplanes, etc.

For such systems, the goal of FMSs is to support planning and scheduling of the fleet (at long term), maintain the performance of the system as close as possible to the preschedule plans (e.g. timetables) and monitoring and actuating in case of unexpected events.

A key problem is the design of a transit route network. It consists in optimizing the (fixed) routes and schedules of the service under constraints such as the number and length of public transportation routes, allowable service frequencies, and the number of available vehicles. Furthermore, at the tactical level, the challenges also include supporting decision-making regarding the modification of the routes and schedules in order to adapt to seasonality, changing trends, or changing customer demands.

Static route network planning has been studied widely in the past. Good reviews can be found in [4,5,6]. Typically, the approaches focus on the development of optimal or near-optimal plans using various types of effective vehicle routing algorithms. Fleet schedules designed a priori with *static route planning* assume the following: all relevant data is known before the planning starts, and the time required for creation, verification, and implementation of route plans is of minor importance (e.g., offline planning).

The use of an initially created fleet schedule, is usually not sufficient to assure efficient operation, since it may not cope adequately with unexpected events during execution like, e.g., traffic delays, vehicle breakdowns, road works, and other, which may cause fleet delays, unexpected costs, and poor customer service. Thus, at the operational and real-time level, the challenge is to respond to such events in an adequate way, i.e. to detect deviations from the initial dispatch plan and adjust the schedule accordingly by suggesting effective re-routing immediately.

We consider semi-static fleets as fleets that are used for transportation services where the planning of routes and schedules is repeated at certain time intervals. This occurs, for example, in many logistics scenarios, like fright distribution or parcel delivery services. There, routes for a given set of transportation tasks are planned on a daily basis. The planning phase is still static since all service tasks are known beforehand, and exact routing algorithms can be applied. Usually, in such environments, the incidence of unforeseen events is greater, e.g. due to the cancellation of service tasks, time restrictions for delivery, etc. Therefore, real-time management systems that are able to treat such situations are of greater importance [7].

2.2 Dynamic Fleets

Dynamic fleets operate in an environment where transportation tasks appear on-the-fly and, thus, their operation cannot rely on pre-defined schedules. Usually, the objective is to provide transportation services “on demand”. Typical examples for the application of such types of fleets are taxi services, certain commercial delivery services, courier fleets, fire trucks, police cars, emergency medical services, and so on. Thus, dynamic fleets are characterized by a fixed number of vehicles but a dynamically changing and a priori unknown number of transportation tasks.

For dynamic fleets, the management tasks focus mainly on the real-time operational level. The main goal of FMSs is to solve the allocation or dispatching of vehicles, that is, assigning vehicles to transportation tasks in an efficient way. Usually, efficiency means minimizing the global travel distance of the fleet (the sum of the required travel distances for all transportation tasks). Minimizing the global distance implies reducing the operational costs, as well as improving the service quality (by reducing the average time required to fulfill all service tasks). This problem, known as the dynamic vehicle routing problem, has been studied widely in the literature. Good surveys in this regard are [8,9,10]. Many approaches are based on an adaptation of static algorithms. Here the main challenge is a rather short time horizon for decision making: as the degree of dynamicity of the environment increases, usually time-consuming optimization algorithms are less applicable.

In addition to the allocation of vehicles, the problem of vehicle deployment and redeployment is of importance. This problem refers to the task of distributing the available resources (vehicles) both at spatial (in the geographic area of influence) and temporal level. The underlying idea is to distribute vehicles in the area of interest based on the current and the expected demand, such that new appearing service tasks can be completed in a fast manner. Especially for services that require a quick response to certain events, effective deployment strategies can improve the service quality considerably. This is the reason why deployment approaches have been extensively studied in the area of emergency medical services where short response times are of foremost importance.

Some reviews of the research in the field emergency medical services are [11,12], concentrating on covering models and optimization techniques for facility location, and [13] analysing the use of simulation models in emergency medical service operations. Early deployment approaches treated the problem in a static long-term way trying to find optimal distributions of vehicle base stations in the region of interest, according to observed or estimated demand patterns and possible changes in the environment (e.g. planned population variation or urban developments), e.g., [14,15].

More recent approaches propose short-term dynamic deployment and redeployment models so as to adapt the fleet to the demands in any moment. Here, vehicles are either redeployed among a set of base stations (e.g., [16,17]), or in a patrol like way without using fixed stations (e.g., [18,19]).

Finally, in certain environments, dynamic allocation and re-deployment strategies may be combined with a priori planning. In such a context, timely close decisions are more important than the ones more remote in time.

2.3 Open (Dynamic) Fleets

During the last decades, vehicle sharing systems have proliferated. The main idea behind such systems is to maximize the utilization of vehicles for transportation tasks by reducing the times vehicles are idle. Instead of using private vehicles for a limited number of (private) transportation tasks, vehicles are used by different users, maximizing in this way their utilization.

Sharing systems may be of different types. On one hand, a number of vehicles, owned or operated by some organization, may be used by different users for their individual transportation needs, like it is the case in bicycle or car sharing services for human mobility. Here, the advantage is a reduction of the number of vehicles and, thus, of the operational costs, necessary for providing a transportation service. On the other hand, private users or organizations that own vehicles may offer a partial use of those vehicles to others, or may participate with their own vehicle in the provisioning of a given transportation service. This is for instance the case in “free” taxi or courier services, like UBER[1], where private people participate on an irregular basis in the provisioning of a certain service. In this case, the advantage is again a reduction of the number of vehicles exclusively dedicated to a transportation service. But also the possibility to have at disposal a “flexible” fleet that can adapt its size to varying service demands.

We call the type of fleets that are used in sharing systems *open fleets*. Open fleets are characterized by the following aspects:

- *Dynamic service demand:* Like dynamic fleets, open fleets are dynamic in the sense that service tasks may appear dynamically at any time and at any location (within the region of operation).
- *Dynamic number of vehicles:* The number of vehicles that participate in the fleet may also be dynamic. New vehicles may join or leave the fleet at any time and this should not affect normal fleet operation. It should be noted that also in sharing systems with an a priori fixed number of vehicles (like public bicycle services) the systems are conceived to operate regardless the actual number of vehicles in a given moment. Vehicles may be retained (e.g., for reparation) or new vehicles may be put into the system at any time, and this should not affect fleet coordination at the operational level.
- *Autonomy / limited control*: The capability of the fleet operator to regulate and control the fleet’s behaviour may be limited. In open fleets, the usage of a par-

---

[1] https://www.uber.com/

ticular vehicle, its availability in a given moment at a specific location does not only depend on the operator's decisions, but also on the user or owner of that vehicle. Here, individual preferences, objectives and needs of owners and users have to be balanced with the global objectives and goals of the system in its entirety. Depending on the particular application, there may be fleets that are more controllable, and others that allow only for very limited control, as it is the case, for instance, in a public bicycle service.

- *Size:* Whereas static and dynamic fleets typically operate in rather small environments with a limited size, open fleets are conceived to potentially work on a larger, maybe unlimited, scale.

Regarding long-term fleet management, at the tactical level, techniques that are applied in static and dynamic fleets may also be applied in open fleets, e.g., for calculating the adequate number of vehicles, or identifying good locations for base stations (where it applies). However, as for dynamic fleets, the efficiency of an open fleet depends much more on the coordination at the operational, real-time level. Here, however, we believe that the methods and techniques used in FMSs for static and dynamic fleets are not sufficient.

Operational management of open fleets must focus on coordination and regulation mechanisms that deal with the problem of balancing global and individual objectives. The aim is to maximize the achievement of individual needs and preferences but at the same time assuring an efficient operation with regard to some globally desirable parameters. Also, the type of global objectives may be different to static or dynamic fleets. Especially for public mobility services, parameters like energy efficiency, egalitarian and fair usage of resources, traffic reduction in a city, etc. will usually be of importance.

The basic decision tasks that have to be solved in operational management are the same as in dynamic fleets: i) task or vehicle allocation, and ii) vehicle (re-)deployment. However, due to the characteristics of open fleets, research on new solution approaches is required. We consider that there are essentially two new aspects that have to be considered.

The first aspect refers to the lack of control capabilities of the fleet operator and the autonomy of the vehicle drivers or users. Whereas in a dynamic fleet it is assumed that the orders regarding task assignment or re-deployment of vehicles are always fulfilled, in open fleets the autonomy (of greater or lesser degree) of the vehicles with respect to the fleet operator implies that the latter cannot impose a certain behavior. For instance, in a bicycle sharing system, a user will usually decide by himself which bike to take and he will return it at the station he likes to. In a "free" taxi service, where private drivers accomplish transportation tasks, the drivers may be able to reject task assignments and may also leave the fleet at their will, thus, not following a certain re-deployment strategy. Instead of using coercive strategies for imposing a certain behavior, fleet management should rely on soft, persuasion techniques. Or it may be necessary to combine both, coercive and soft enforcement mechanisms. Thus, the task of an FMS system consists not only in computing an optimal assignment and deployment solution in each particular moment (like for dynamic fleets), but also in convincing the drivers and/or users to adopt such a solution. In addition, the optimality criterion has to be changed to a utility criterion. Optimal solutions in the fleet context, usually involve the joint actions of several vehicles. This means that there are multiple possible points of failure (drivers not accepting the assigned task). In this sense, a best solution is not any more one that minimizes some global parameters, but a solution that has the highest utility; combing both the minimization of global parameters and the probability of being successfully executed.

Depending on the application domain, different persuasion techniques can be applied in order to convince users /driver to act in a specific way, such as incentives, argumentation, social reputation, etc. [23,24,25] Furthermore, trust and reputation mechanisms may be used to estimate the future behavior of drivers/users based on an analysis of their historic behaviors. Such information may be helpful when deciding on how to persuade a specific person, or when estimating the probability of success of a given assignment and/or deployment solution.

With regard to the (re-)deployment task, a common problem in large scale vehicle sharing applications for human mobility is that the flow of vehicles is usually not the same between different areas and, thus, the vehicle distribution may become unbalanced with respect to the demand in the near future. There have been different proposals for supporting fleet operators with operational strategies for relocation and redistribution of vehicles in order to meet future demand [20,21,22]. In addition to such techniques, we believe that persuasion mechanisms may help to avoid unbalanced distributions of vehicles (at least partially) and even may be used to adopt a given distribution to a changing demand pattern. The idea is to use persuasion techniques (like recommendation, argumentation or incentives) to convince the users to adapt their travel routs slightly towards a situation that represents a better distribution of the vehicles with regard to future demand.

The second aspect we consider, that should be taken into account in an FMS for open fleets, is scalability. As we mentioned before, open fleets are often conceived for large-scale problems with potentially many transportation tasks and vehicles. Usually also the dynamicity of such systems, in term of the frequency of new service task demands, is quite high. Furthermore, the fleets should be robust with regard to the appearance or disappearance of vehicles as well as with regard to local incidents or problems. That is, such situations should not affect the global operation. In order to cope with this aspect, distributed and scalable coordination approaches that rely on local computations of assignment and (re-)deployment strategies should be used at the operational level.

In the next section we propose a preliminary architecture for management systems for open fleets that takes the above-mentioned aspects into account.

3 An Architecture for Smart Open Fleets

There are two main problems fleet operators are faced with: task allocation and redeployment. The *allocation* problem consists in determining which vehicle should be sent to serve a given task. *Redeployment* consists in relocating vehicles in the region of influence in a way that new tasks can be reached fast and/or with low costs. Both issues are particularly challenging in dynamic environments, as continuously upcoming new tasks may require attendance, and the current situation of the fleet may change due to external influences. In order to maximize vehicle utilization and to improve service quality in such environments, task allocation and vehicle redeployment should as well be accomplished in a dynamic manner, adapting the coordination of the fleet seamlessly to upcoming events and changing demands. In order to adequately capture the real-time requirements in such a scenario, we set out from an event-driven approach.

Fig. 1 depicts our architecture for open fleet management. It contains three basic layers: the top layer contains the vehicles; the second layer represents the fleet coordination modules; while the third layer includes other components that are necessary for the normal operation of a fleet operator (e.g., components for monitoring, global fleet control, etc.).

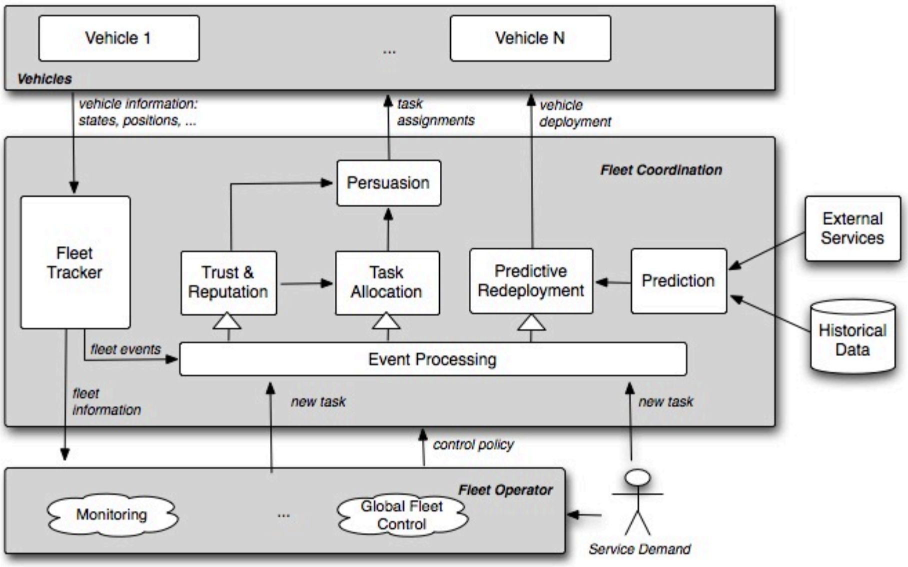

Fig. 1. Architecture for open fleets

In the fleet coordination layer, a *Fleet Tracker* follows the operational states and positions of the vehicles[2]. It informs the *Event Processing* module about any changes in the fleet that would require an adaptation of the task allocations and/or the deployment of idle vehicles. This module analyses the incoming events (state changes of vehicles and new task events) and determines whether or not a re-calculation of task assignments and/or deployment of idle vehicles should be done. If necessary, it trig-

---

[2] We assume that vehicles have capabilities to send their current positions on a regular basis and to inform about changes in their operational states.

gers the execution of the task allocation and predictive redeployment modules. The *Task Allocation* module, when executed, re-calculates the optimal global assignment of all pending tasks (in the current moment) to vehicles, based on a set of assignment criteria (depending on the application domain). The *Predictive Redeployment* module, calculates adequate positions for all idle vehicles at the current moment taking into account predictions concerning the appearance of new tasks (based on historical data) and the current state of the fleet.

Prediction is carried out taking into account historical data and other external sources (e.g. weather forecast, leisure events, etc.). Depending on the application domain, new tasks can be triggered by fleet users (clients) by communicating with the fleet operators or directly with the fleet coordination layer.

In order to deal with the (possible high) autonomy of vehicles, we include persuasion and trust and reputation modules in the architecture. The *Persuasion* module is in charge of providing actions for inducing agents (vehicles) to carry out the actions that tend to improve the overall performance measure of the system.

Information about previous experience of vehicles within the system can be exploited so as to take better decisions. The *Trust and Reputation* module is in charge of modeling the expected behavior of agents[3] in the system using feedback provided by the *fleet tracker*. That information is used at least in two different though related ways: (i) what actions have to be chosen and (ii) how agents can be influenced accordingly. For instance, in a vehicle renting scenario, the information about liability of users to return vehicles at the expected time can be used for estimating the number of available resources, which is important for task allocation decision. Likewise, the information given to a particular user in the course of an explanation or persuasion dialogue can be different depending on his/her expected behavior.

It is important to note that, depending on each particular case, not all modules described in the architecture are necessarily implemented.

As the number of vehicles increases, an important aspect to take into account when designing an architecture for open fleet management is scalability. The approach followed to this respect highly depends on the application domain: coordinating a fleet of about ambulances (for instance, there are less than 30 advanced life support ambulances in the Spanish town of Madrid [19]) is obviously quite different from orchestrating taxis as open fleets (Madrid can count on 15000 registered taxis). To address scalability, in many approaches the environment is divided into (generally overlapping) areas, and the control is applied locally in each area. Coordination is needed in case there are conflicts for using shared resources or services. That kind of approaches has been used, for instance, for public transportation management [26]. In the next section, we present a distributed coordination algorithm for taxi service assignment.

---

[3] Depending on the domain, agents can represent vehicles (e.g. taxi) or users/clients (person renting a bike).

4 Example: Taxi fleet coordination

In this section we apply the aforementioned architecture in a system for coordinating a fleet of "free" taxi services in a big city, where there are thousands of taxis (as mentioned before, some 15000 taxis in the case of Madrid). Usually, there are several taxi companies which taxis are affiliated to. They coordinate service calls, either assigning a taxi to the client or asking those taxis nearby who is interested in doing the service.

One of the main goals of the taxi company is to reduce the response time (e.g., the time between a client call and the moment a taxi arrives).

This scenario has the main features we used to characterize open fleets: (i) taxis join and leave the fleet anytime during the day, and (ii) taxis are autonomous since they decide whether they take a service and it is not possible to enforce them to carry out their commitments. In this example, we focus on the task allocation part of the architecture. We assume that event management and fleet tracking are processed by the corresponding modules. We do not use predictive redeployment in this scenario.

A naïve method many companies use for taxi assignment is the closest method rule based on the first-come/first-served (FCFS) principle. That is, the first client in the system is assigned first, then the next client, and so on. In each case, a client is assigned to the closest available taxi (using GPS) in that particular moment.

Imagine the scenario shown in Fig. 2, where there are three available taxis, and two clients. *c1* asked for a taxi a few seconds before *c2*. Fig. 2a shows the locations of taxis and clients, numbers represent the distance[4] between them. Fig. 2b shows the assignment resulting from applying the naïve strategy: when *c1* entered the system the closest taxi (*t1*) was assigned, and then *t3* was the closest to *c2* (with a total distance 3+10 = 13). However, there is a better assignment, as shown in Fig. 2c, where both clients get a taxi at distance 4 (total 8), which is better for a global point of view (*c1* has a lightly worse taxi but there is a high improvement for *c2*).

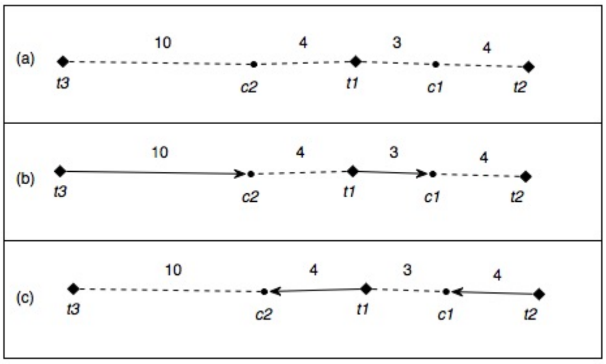

Fig. 2. Taxi assignment strategies: numbers represent distances, t1, t2 and t3 are available taxis, c1 and c2 are clients. (a) without assignment; (b) FCFS and shortest path with c1 appearing first; (c) optimal assignment

---

[4] At this point it is not important the distance function used.

The situation (b) was due to the FCFS strategy although both clients arose very close in time. That could be avoided using time windows so that several clients could be considered together and a better global assignment could be obtained.

We follow a different approach for the assignment of taxis to clients (task allocation). Our proposal is inspired by Bertsekas' auction algorithm [27]. We already developed an extension of Bertsekas' auction algorithm for efficiently coordinating the fleet of ambulances of the Emergency Medical Assistance Service SUMMA112 in the Autonomous Region of Madrid in Spain [19].

However, unlike the ambulances scenario, taxi drivers have higher autonomy. In particular, they have to accept the assignment proposed by the system. Furthermore, they might not fulfill the agreed service, so incentives/penalties are necessary to enforce their commitments.

The process of deciding which taxi is proposed to be assigned to a given client is based on the following idea. Clients bid for the available taxis in an auction process. Every taxi has a "virtual" price[5]. First, the prices of all taxis are initialized to 0. Then, the auction process starts. In each iteration, a bidding and an assignment phase take place. During the bidding phase, each client c that is not currently assigned to any taxi determines the taxis t_i and t_k with the least cost (p_1) and second least cost (p_2), respectively. The cost of a taxi t for a given client c is computed as proportional to the expected travel time for t to reach client c plus the current cost of t (other functions can be used). Then, client c issues a bid for its best taxi (t_i), where the bid value is the difference between the cost of the second best and the best taxi for c plus a constant ε. The rationale behind this bid value is that, at the current prices and up to a price increment of $p_2 - p_1$ for taxi t_i, client c would prefer this taxi with regards to its second choice (t_k), i.e. it represents how important for the client is to get that taxi compared to get the second choice. For instance, in the example of Fig. 2, if client $c2$ does not get taxi $t1$ its price increment would be 6 (10 – 4), while for $c1$ would be only 1 (4 – 3). ε is a (positive) constant (the minimum price increment), necessary to assure termination of the auction process. After all unassigned clients have issued their bids the assignment phase takes place. Each taxi t_i that received a bid is assigned to the client c that issued the highest bid for that taxi. If t_i was already assigned to another client, it is deassigned previously. Finally, the price of t_i is incremented by the highest bid value. The bidding and assignment phases are repeated until all clients are assigned to a taxi.

Dealing with Scalability

As discussed before, in fleets with high number of vehicles, scalability becomes a real problem. This is the case of taxi coordination in big cities, where thousands of taxis circulate daily.

Our proposal for coping with that problem is a distributed execution of the method described above. It consists of three type of components running in different devices: a taxi application that participate in the auction and finally accept or reject services, a

---

[5] Do not confuse with the price a client has to pay for a taxi service

client application that runs on the client device (e.g. smartphone) and a central server that manage a registry with basic information of taxis (such as location and cost).

The *central server*[6] is in charge of maintaining the location, price and status (available, occupied) of each taxi, and provides a set of functionalities such as calculating the closest taxis to a given location. Taxis send periodically their location to the server, and update their status and cost.

Fig. 3 shows the algorithm running on the taxis. It basically manages the participation in the auction for deciding which taxi is in charge of accepting the service a client needs.

```
1:  price = 0
2:  current_client = null
3:  Wait for bid<c, $p_c$>
4:  tmp_client = c
5:  price = $p_c$
6:  updatePrice(t, price)
7:  Start time window
8:  repeat Wait for bid<c, $p_c$> or end_of_time_window
9:      if $p_c$ > price then
10:         reject(tmp_client)
11:         tmp_client = c
12:         price = $p_c$
13:         updatePrice(t, price)
14:     else
15:         reject(c)
16:     end if
17: until end_of_time_window
18: if taxi driver accepts then
19:     notifyAssignment(t,tmp_client)
20: else
21:     reject(tmp_client)
22: end if
```

Fig. 3. Taxi algorithm executed whenever a taxi *t* gets available

The algorithm is started when the taxi becomes available (it joins the fleet or finishes a service). Initially the *price* is established to 0 and waits for a bid from a client (line 3). The bid includes the client *c* issuing that bid and the price p_c it offers. After receiving the first bid, the taxi establishes a time window during which it will accept bids from other potential clients. The variable *tmp_client* stores the client that is temporally assigned to the taxi during the time window. Then, the taxi waits for a new bid or the end of the time window (lines 8-17). If a new bid is received, then there are two

---

[6] We consider the server "conceptually" centralized, we do not focus in this paper on the distributed implementation of the registry.

possibilities: (i) if the price of the new bid is lower than the current price then the client is rejected (line 15); (ii) if the price is higher then the client is temporally chosen, the price is updated and the previous temporal client is rejected (lines 10-13). When the time window finishes the temporal client is definitively chosen and, after receiving the confirmation from the taxi driver (line 18), a notification (line 19) or rejection (line 20) is sent to him/her. Assignment notification (*notifyAssignment(t,c)*) and price update (*updatePrice(t,p)*) communicate with the central registry, which update the information of taxi *t* accordingly.

Fig. 4 shows the algorithm executed when a client asks for a taxi service. It asks the central registry for the two taxis with lowest costs (line 2). The registry returns (function *searchCheapestAvailableTaxis*) basic information of such taxis including their current prices (according to the algorithm) and their "costs" (taxi *price* + *distance* to client). Then, the client issues a bid to the cheapest taxi (line 3) and waits for an answer. If the bid is rejected, then the process is repeated until a bid is accepted. Functions *cost* and *price* return information of cost and price of taxis, respectively.

```
1:  repeat
2:      <t1,t2> = searchCheapestAvailableTaxis(c)
3:      Bid(t1, cost(t2) – cost(t1) + price(t1))
4:      Wait for answer
5:  until bid accepted
```

Fig. 4. Client algorithm

Persuasion

As mentioned previously, autonomy is a characteristic of this kind of systems. In particular, taxi drivers are free to accept or reject client assignments. As detailed in Fig. 3, at the end of the auction process the taxi driver has always the option to reject the assignment proposal.

However, it might be interesting to use mechanisms to foster taxis to follow the assignments recommended by the system, or even worst if they do not actually fulfill the service they committed. In section 3 we pointed out this aspect by including trust and reputation and persuasion in the architecture.

Even though in the proposed algorithm we did not deal with the problem of convincing driver to accept the clients the system recommends, this aspect could be integrated by manipulating the prices of taxis in the auction process. In particular, a trust model could be used to determine "reliable" drivers. During the auction, the central server could increase the price of "unreliable" taxis such that clients will be less inclined to bid for such taxis (if they have other similar options).

5 Conclusions

In this paper we have discussed the coordination of transportation fleets. The main contributions of this work are (i) an analysis and classification of different types of fleets ending up with the introduction of the notion of *open fleets*, (ii) an architectural framework for the management of open fleets, and (iii) some preliminary work on a decentralized algorithm for vehicle assignment that could be applied, for instance, in a large "free" taxi service.

In the future, we plan to evaluate the proposed decentralized algorithm for taxi assignment. In particular, we will compare its efficiency against the approaches that are currently applied in real world applications (e.g. FCFS with shortest path). In addition, we will explore other decentralized options (for instance based on spatial division of the region).

Finally, we will also like to analyze in more detail the relation between the autonomy of taxi drivers and the system performance.

Acknowledgments. Work partially supported by Spanish Government through the project iHAS (grant TIN2012-36586-C03), the Autonomous Region of Madrid through grant S2013/ICE-3019 ("MOSI-AGIL-CM", co-funded by EU Structural Funds FSE and FEDER), URJC-Santander (30VCPIGI15).

References

1. Dablanc, L.: Goods transport in large european cities: Difficult to organize, difficult to modernize. Transportation Research Part A: Policy and Practice, 41(3):280–285, (2007).
2. Shaheen, S.A., Cohen, A.P: Growth in worldwide carsharing: An international comparison. Transportation Research Record: Journal of the Transportation Research Board, 1992(1):81–89, (2007).
3. Mont, O.: Institutionalisation of sustainable consumption patterns based on shared use. Ecological Economics, 50(1):135–153, (2004)
4. Kepaptsoglou, K., Karlaftis, M.: Transit Route Network Design Problem: Review. Journal of Transportation Engineering, 135(8):491–505, (2009).
5. Wren, A., Carr, J.D.: Computers in Transport Planning and Operation, Ian Allan Publishing, Limited, London, (1971).
6. Chua, T.A.: The planning of urban bus routes and frequencies: A survey. Transportation, 12(2):147–172, (1984).
7. Zeimpekis, V., Minis, I., Mamassis, K., Giaglis, G.M.: Dynamic management of a delayed delivery vehicle in a city logistics environment. Dynamic Fleet Management. Springer US. 197-217 (2007)
8. Gendreau, M., Potvin, J.Y.: Dynamic vehicle routing and dispatching. in T.G. Crainic and G. Laporte (eds.). Fleet Management and Logistic, pp. 115–226, Springer, (1998)
9. Ichoua, S., Gendreau, M., Potvin, J.Y.: Planned route optimization for real-time vehicle routing. In V. Zeimpekis, C.D. Tarantilis, G.M. Giaglis and I. Minis (eds.). Dynamic Fleet Management. Volume 38 of Operations Research/Computer Science Interfaces. Pp. 1–18. Springer. (2007)

10. Pillac, V, Gendreau, M., Gueret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. European Journal of Operational Research. 225(1):1–11. (2013)
11. Brotcorne, L., Laporte, G., Semet, F.: Ambulance location and relocation models. European Journal of Operational Research. 147(3): 451– 463, (2003)
12. Li, X., Zhao, Z., Zhu, X., Wyatt, T.: Covering models and optimization techniques for emergency response facility location and planning: a review. Mathematical Methods of Operations Research. 74(3): 281– 310, (2011)
13. Aboueljinane, L. et al. A review on simulation models applied to emergency medical service operations. Computers & Industrial Engineering, 66(4): 734–750, (2013)
14. Toregas, C., Swain, R., ReVelle, C., Bergman, L.: The location of emergency service facilities. Operations Research 19(6): 1363–1373, (1971)
15. ReVelle, C., Hogan, K.: The maximum availability location problem. Transportation Science 23(3):192–200, (1989)
16. Naoum-Sawaya, J., Elhedhli, S.: A stochastic optimization model for real-time ambulance redeployment. Computers & Operations Research 40(8): 1972–1978, (2013)
17. Ibri, S., Nourelfath, M., Drias, H.: A multi-agent approach for integrated emergency vehicle dispatching and covering problem. Engineering Applications of Artificial Intelligence. 25 (3): 554 – 565, (2012)
18. Andersson, T., Varbrand, P.: Decision support tools for ambulance dispatch and relocation. Journal of the Operational Research Society. 58(2): 195 – 201, (2007)
19. Billhardt, H., Lujak, M., Sánchez-Brunete, V., Fernandez, A., Ossowski, S.: Dynamic Coordination of Ambulances for Emergency Medical Assistance Services. Knowledge-Based Systems. 70: 268 – 280, (2014)
20. Nair, R. et al. Large-scale vehicle sharing systems: Analysis of Vélib, International Journal of Sustainable Transportation, Vol. 7, pp. 85-106, (2013)
21. Kek, A.G.H., Cheu, R.L., Meng, Q., Fung, C.H.: A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation, 45(1): 149–158, (2009)
22. Nair, R., Miller-Hooks, E.: Fleet management for vehicle sharing operations. Transportation Science 45(4): 524—540, (2011)
23. Modgil, S. et al. The Added Value of Argumentation. Agreement Technologies. Springer. pp. 357-403. (2012)
24. Koster, A., Sabater-Mir, J., Schorlermmer M.: Argumentation and Trust. Agreement Technologies. Springer. pp. 441-451. (2012)
25. Castelfranchi, C., Falcone, R.: Trust Theory: A Socio-Cognitive and Computational Model. Wiley. (2010)
26. Ossowski, S., Hernandez, J.Z., Belmonte, M.V., Fernandez, A., García-Serrano, A., Pérez-de-la-Cruz, J.L., Serrano, J.M., Triguero, F.: Decision support for traffic management based on organisational and communicative multiagent abstractions. Transportation Research Part C, 13, pp. 272-298, (2005)
27. Bertsekas, D.: The auction algorithm: a distributed relaxation method for the assignment problem, Annals of Operations Research 14 (1): 105–123. (1988)