Multi-agent resource allocation for road passenger transportation⋆

Beatriz Lopez and Pere Urra and Xavier Abelló

Institut d'Informàtica i Aplicacions, Universitat de Girona,
Campus Montilivi, edifice P4, 17071 Girona, Spain

Abstract. Inter-urban road passenger transportation requires the allocation of drivers and buses to transport people. In such allocation process, several constraints on driving time are being imposed by the governments in order to assure citizens safety. Such constraints, however, are posing a lot of difficulties to the allocation process, usually generated by human operators. In this paper we formalize the problem and provide a solution approach based on a multi-agent environment. Particularly, we propose the use of combinatorial auctions. The first results obtained in the first prototype developed are provided and discussed.

1 Introduction

Road passenger transportation has been for years a matter of concern for the traffic responsible in order to minimize bus accidents. Traffic accidents in general are one of the major mortality rates in developed countries. In this line, several European governments are campaigning for better driving practices. Regarding buses, the European law is also evolving in order to control professional driving licences and driving times, with the aim of assuring the maximum guarantees to the citizen that use road passenger transports.

This new laws and regulations are posing a lot of requirements to the companies related to this economic field. The challenge is not so much related to regular and down town services that can be scheduled once a year, but to just-in-time services. That is, services required within a short period of time, usually, from one day to the next one. This kind of services are often related to conference events, holidays, excursions, etc., which are provided by inter-urban transport companies.

In the past, human operators in the inter-urban transport companies were in charge of allocating drivers to required services once a day. For example, at night, when all the customers have already performed they requests, the operator dedicate so much time to the allocation process. New laws and regulations, however, are posing too many constraints for being manually managed. As a by-pass solution, operators elaborate schedulers in which drivers have unoccupied hours. The economic consequences for the benefits of the companies are evident: with the same amount of drivers, they can provide less services, and so they earn

---

less money. Moreover, there is no guarantee that all the constraints imposed by the law are satisfied, so the company is assuming the risk to be billed by the traffic authorities.

Trying to face this problem, we are working in the development of new scheduling techniques. Particularly, from our experience on the use of multi-agent resource allocation techniques in the RoboCup Rescue environment [1], and ambulance transportation [2], we have modelised the problem following a multi-agent approach, and applied combinatorial auctions to solve the problem. In this paper we present our first results.

This paper is organized as follows. First, we give a description of the road passenger problem in section 2. We continue by giving our approach to solve the problem in a multi-agent environment in section 3. We proceed by giving our first results in section 4 and we end with some discussion and conclusions in 5.

2 Problem description

In the road passenger transportation domain we are given with two set of resources, drivers $D = \{d_1, \ldots, d_n\}$ and buses: $B = \{b_1, \ldots, b_m\}$, and a set of tasks (services) to be performed by using the resources $S = \{s_1, \ldots, s_l\}$. The problem consists on assigning to each service a driver and a bus, subject to the constraints and preferences provided by the government .

The problem can reduced to driver allocation. There are two main reasons for that. First, each driver has a bus assigned by default, so the second allocation process is trivial when solving the first one. And second, in case that additional buses were required, there is no problem to rent extra ones. The critical resources are drivers.

The time unit used in the allocation process is the hour. However, in order to verify the different constraints imposed by low, the definition of a *sliding time window* of one month is also required. For convenience, we consider a month composed by 28 days organized in four weeks: week 1 (from day 1 to 7), week 2 (from day 8 to 14), week 3 (from day 22 to 28) and week 4 (from day 22 to 28). All the definitions that follows are contextualized within this sliding time window.

2.1 Services

Definition 1. A *service* is a tuple

$$s_i = < ti_i, tf_i, dur_i, orig_i, dest_i, n_i, D^i, b_i >$$

where ti_i is the initial time, tf_i the final time ($tf_i > ti_i$), dur_i the service duration, $orig_i$ the place where the service starts, $dest_i$ the destination place, n_i the number of passengers, D^i the drivers assigned ($D^i = \{d_1^i, \ldots, d_{p_i}^i\}$ and $|D^i| \geq 1$) and bi the bus allocated.

Further refinements of the problem should include itineraries, that is, $Iti = < track_{i_1}, \ldots track_{i_k} >$, where $track_{i_1} = orig_i$ and $track_{i_k} = dest_i$.

2.2 Drivers

Definition 2. A *driver* is a tuple

$$d_i = < \mathcal{T}_i^d, \mathcal{T}_i^p, \mathcal{T}_i^b, \mathcal{T}_i^w, p_i, pkm_i >$$

where $\mathcal{T}_i^d, \mathcal{T}_i^p, \mathcal{T}_i^b$, and $\mathcal{T}_i^w$ are four different time measures (effective working time, presence time, break time and weekly-break time, see below), p_i is the basic cost and pkm_i is the cost per kilometer.

The *effective working time* $\mathcal{T}_i^d$ measures the time the driver i is effectively driving a bus. This time includes auxiliary works.

Definition 3. The *effective working time* $\mathcal{T}_i^d$ for driver i is defined as the set of all dairy effective working times within the sliding time window:

$$\mathcal{T}_i^d = \{T_i^{d_1}, \ldots, T_i^{d_{28}}\}$$

where: $T_i^{d_j}$ is the dairy effective working time for day j.

Definition 4. The *dairy effective working time for day j*, $T_i^{d_j}$, is the sequence of all time slots assigned to driver i for driving a bus along journey j:

$$T_i^{d_j} = t_{i_{j_1}}^{d_j} < \ldots < t_{i_{j_d}}^{d_j}$$

Each time slot $t_{i_k}^{d_j}$ represents the initial time in which the driver should start a given service. The duration of the time slot $t_{i_k}^{d_j}$ is noted as $|t_{i_k}^{d_j}|$. Note that $|t_{i_k}^{d_j}|$ cannot necessarily equals the duration of the service. According to the different constraints, several drivers can be assigned to a service, so a time slot can partially cover the service duration. It is the addition of the time slots of all the drivers assigned to a service that should totally cover the service duration (see problem formulation at section 2.4).

Definition 5. The accumulated effective working time for day j is defined as:

$$T_i^{\Sigma d_j} = \sum_{k=i_{j_1}}^{i_{j_d}} |t_k^{d_j}|$$

The *presence time* $\mathcal{T}_i^p$ measures the time the driver is in the bus but not driving.

Definition 6. The *presence time* $\mathcal{T}_i^p$ for driver i is defined as the set of all dairy presence times within the sliding time window:

$$\mathcal{T}_i^p = \{T_i^{p_1}, \ldots, T_i^{p_{28}}\}$$

where $T_i^{p_j}$ is the dairy presence time in day j.

Definition 7. The *dairy presence time for day j*, $T_i^{p_j}$, is the sequence of all time slots assigned to driver i along journey j in which he/she is not driving:

$$T_i^{p_j} = t_{i_{j_1}}^{p_j} < \ldots < t_{i_{j_p}}^{p_j}$$

There should be a relationship between two consecutive effective working time slots, t_i^d and t_{i+1}^d and a presence time slot in between, t_j^p. That is, if two consecutive effective time slots have some time gap, such time gap should correspond to a presence time slot.

Definition 8. The accumulated presence time for day j is defined as:

$$T_i^{\Sigma p_j} = \sum_{k=i_{j_1}}^{i_{j_p}} |t_k^{p_j}|$$

The *break time* T_i^b measures the time the driver is out of the vehicle along its journey. The minimum length is one hour.

Definition 9. The *break time* T_i^b for driver i is defined as the set of all dairy break times within the sliding time window:

$$T_i^b = \{T_i^{b_1}, \ldots, T_i^{b_{28}}\}$$

where $T_i^{b_j}$ is the dairy break time for day j.

Definition 10. The *dairy break time for day j*, $T_i^{b_j}$, is the sequence of all time slots assigned to driver i along journey j in which he/she is out of the car:

$$T_i^{b_j} = t_{i_{j_1}}^{b_j} < \ldots < t_{i_{j_b}}^{b_j}$$

where each $|t_{i_{j_k}}^{b_j}| \geq 1$.

Definition 11. The accumulated break time for day j is defined as:

$$T_i^{\Sigma b_j} = \sum_{k=i_{j_1}}^{i_{j_b}} |t_k^{b_j}|$$

The *weekly break time* T_i^w measures the time the driver has continuous break along a week (week ends, holiday). Weekly break time includes dairy break time. Both concepts, break and weekly-break should be considered as separated entities related to constraints required by the UE.

Definition 12. The *weekly break time* T_i^b for driver i is defined as the set of all four break times corresponding to the four weeks within the sliding time window:

$$T_i^w = \{T_i^{w_1}, \ldots, T_i^{w_4}\}$$

where $T_i^{w_j}$ weekly break time in week j

Definition 13. The *weekly break time for week j*, $T_i^{w_j}$, is the sequence of all time slots assigned to driver i along week j in which he/she is either out of the office:

$$T_i^{w_j} = t_{i_{j_1}}^{w_j}, \ldots, t_{i_{j_w}}^{w_j}$$

Definition 14. The accumulated week break time for week j is defined as:

$$T_i^{\Sigma w_j} = \sum_{k=i_{j_1}}^{i_{j_w}} |t_k^{w_j}|$$

The break time is included in the weekly time, and such relationship is formalized according the following expressions:

$$t_{i_{j_1}}^{w_j} = \sum_{k=1}^{7} T_i^{\Sigma b_k}, t_{i_{j_1}}^{w_j} = \sum_{k=8}^{14} T_i^{\Sigma b_k}, t_{i_{j_1}}^{w_j} = \sum_{k=15}^{21} T_i^{\Sigma b_k}, t_{i_{j_1}}^{w_j} = \sum_{k=22}^{28} T_i^{\Sigma b_k} \qquad (1)$$

2.3 Constraints and preferences

The following constraints should be satisfied in any allocation solution:

Coverage The addition of all the time slots of effective working time of the drivers allocated to a service should cover the duration of the service.

Overlapping Different services with common drivers assigned should not have overlapping times.

Constraints on effective working time The maximum effective working time within a day is 12h, with some exceptions.

Constraints on effective working time There are several maxima on the effective working time: 12 hours in a day, 90 hours in a week, and 4.5 h of continuous driving time

Constraints on presence time : the maximum is 20h per week in average inside the sliding time window.

Constraints on break time : the minimum continuous break time between two consecutive journeys is 11h. In case that the time is split in several bits at least one of the bits should be 8h long, the remaining bits should be at least 1h long, and the total amount of all the bits should be 12h. Vehicles with two drivers are allowed to have a minimum continuous break time of 8h within 30hours.

Constraints on weekly break time : the minimum continuous weekly break time is 36 hours.

All such constraints are subject to different exceptions. For the sake of length, we do not include the formal specifications of the constraints and preferences here (see [3] for a full description and formalization of the problem).

Regarding preferences, two main issues should be addressed:

Cost Drivers with low cost are preferred than expensive drivers. Low cost drivers means 0 basic cost, since they are employer of the company. Otherwise, drivers are hired as required. The same is applicable to buses.

Continuity Time slots of effective working time for a given driver are preferred to be continuous.

2.4 Problem formulation

Definition 15. *Driver's allocation problem.* Given a set of services $S = \{s_1, ..., s_l\}$ required in day x, and a set of drivers D, assign a set of drivers $D^i \in 2^D$ for each service s_i subject to the constraints and preferences described above.

Each driver $d^i_k \in D^i$ should then have allocated at leat one time slot in his effective working time $t^x_{k_j} \in T^{d_x}_k$ for service i, and eventually some time slot in his presence time $t^x_{k_y} \in T^{p_x}_k$. The remaining time of the day is allocated as break and weekly-break time for the corresponding driver.

Different solutions are feasible. Each solution *sol* has a global cost $c(sol)$ that relates the number of constraints violated, the number of preferences unsatisfied, the drivers cost and the buses cost. The optimization problem consists then on finding the best solution. Formally:

Definition 16. *Driver's optimization problem.* Given a set of services $S = \{s_1, ..., s_l\}$ required in day x, a set of drivers D, and a function cost c, find the set of drivers $D^i \in 2^D$ for each service s_i subject to the constraints and preferences described above so that c is maximized.

The complexity of the problem is known to be exponential, regarding the number of services requested.

3 Multi-agent resource allocation approach

The road passenger transportation problem described in the previous sections has a lot of information regarding drivers, namely different time variables. In addition, constraints are mainly focussed on the different time variables of the drivers. So we believe that the problem is naturally distributed according to the drivers information. For this reason, we propose a multi-agent system as the supporting system for the allocation problem, in which each agent represents the interest and preferences of a driver. Then, each agent is responsible of a single resource, a driver, and the multi-agent resource allocation is the process of distributing a number of items (services) amongst a number of agents according to [4].

We also adopt from [4] the interpretation of the social welfare term. If each agent has its individual preferences, measured by an utility function (cost of a solution), the concept of social welfare is the sum of the individual utilities and can be used to measure the quality of the allocation from the viewpoint of the system as a whole.

From our experience, we propose the use of combinatorial auctions as the allocation procedure. It is a centralized procedure, in which a single agent, the auctioneer, decides upon the allocation from the bids reported by the remaining agents (drivers).

A picture of the system is shown in Figure 1. The remaining of this section is devoted to explain the driving and auctioneer agents.
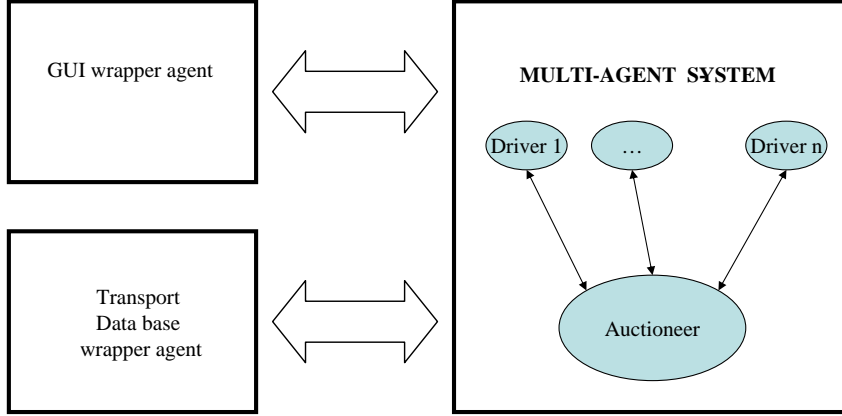

Fig. 1. The multi-agent resource allocation system

3.1 Driver agent

A driver agent keeps information about a single driver (resource), that is, their occupancy, their constraints and preferences. For this purpose, each driver agent has an agenda. The agenda of driver d_i for day x contains the assigned services to this driver. It is composed of several time slots, each one with the following attributes: identification, duration, starting time, end time, service (see table 3.1).

Table 1. Driver agent agenda.

slot id	duration	start time	end time	service
t_1^x	dur_1^x	ti_1^x	tf_1^x	s_{j_1}
t_2^x	dur_2^x	ti_2^x	tf_2^x	s_{j_2}
...	...	...	...	...
t_y^x	dur_y^x	ti_y^x	tf_y^x	s_{j_y}

The starting time ti_k^x of each slot t_k^x assigned to a service s_{j_k} is later than

the starting time of the service ti_{j_k}.

$$\forall k, ti_k^x \in t_k^x \geq ti_{j_k} \in s_{j_k}$$

Analogously, the end time tf_k^x of each slot t_k^x assigned to a service s_{j_k} is earlier than the end time of the service ti_{j_k}.

$$\forall k, tf_k^x \in t_k^x \leq tf_{j_k} \in s_{j_k}$$

Each day, the driver agent participates in the auction process supervised by the auctioneer agent, in order to find the allocation of the services for the given day. For this purpose, each driver agent generates all possible agendas according to the required services and measures the goodness of each agenda according to a cost function $c(agenda)$. The cost function takes into account the number of violated constraints, closeness of effective working time slots, and drivers preferences. For example, agendas with lots of gaps between different effective working time slots have a high cost than agendas with continuous time slots.

3.2 Auctioneer agent

The auctioneer agent is responsible of the allocation procedure. Particularly, it starts a combinatorial auction process from which all the drivers agents are informed of the current requested services. Then, the drivers agents answers with the corresponding bid.

According to the services requested, each driver agent d^i generate a possible partial solution or alternative regarding the tasks that it can perform. Each alternative is a pair (Seq_j^i, c_j^i), where Seq_j^i is the sequence of services that the driver can accomplish according to the agenda and the corresponding cost. For each service, the corresponding initial and end time are also provided.

$$Seq_j^i = \{< s_{j_1}, ti_{j_1}, tf_{j_1} >, < s_{j_2}, ti_{j_2}, tf_{j_2} >, \ldots, < s_{j_{n_j^i}}, ti_{j_{n_j^i}}, tf_{j_{n_j^i}} >\}$$

Since there are n drivers, each of which generating n^i alternatives, the total amount of alternatives are $\sum_{i=1}^{n} n_i$. The best combination of all of them should be selected. This is what has been call the winner determination problem. Currently, there are two algorithms showing the best results: CABOB [5] and CASS [6]. We has chosen CASS, mainly because of our experience on it.

4 Results

We have implemented a first prototype of the system in JADE (see [7] for details) and we have performed several experimental tests with real data coming from a inter-urban transport company, that for confidential reasons we cannot mention here. There were 70 drivers and in average 40 services per day should be scheduled. In each experiment different parameters have been changed : amount
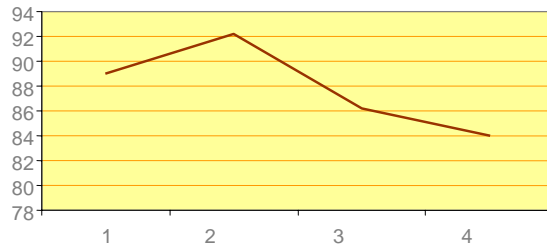
Fig. 2. First results. X-axis shows the service/bid ration, while the y-axis the percentage of services covered.

of bids per driver, amount of services per bid, and rate between drivers and services. That is, when both, the amount of drivers and services is huge (close to the maximum), the response time of the system is too high. So, we have established a tradeoff between both concepts: when there are few number of drivers, a higher amount of bids per driver are allowed; otherwise, a lower amount of bids per driver are generated.

Figure 2 shows our first results. The x-axis corresponds to the number of services per bid, and the y-axis the percentage of services from which an allocation has been achieved. That is, due to the constraint on the number of bids allowed to the drivers, not all the services are covered by the system. As the number of services is increased, each driver submits less number of bids in the auction process. The best results obtained are when each driver submits in average 2 services per bid.

We think that this preliminary results can be improved with the development of better bidding policies for driver agents. Moreover, we are thinking on developing some kind of recursive auction techniques in order to start a new allocation process with the remaining services.

5 Discussion and conclusions

In this paper we have presented the road passenger transportation problem, from a formal point of view, and an approach to find a solution by means of multi-agent resource allocation. Transportation problems are a matter of concern from the Intelligent Agent research community, as the recent publication of [8] has shown. However, most of the problems are related to logistics for industrial procurement [9] , traffic control [10], and even bus routes for cities [11]. However, the inter-urban transport poses particular challenges to the research community that have not been tackled before. The definition of different time measures, as effective working time, presence time, break and weekly break time, characterize the problem with quite complex constraints and preferences not handled in other domains.

The preliminary results shown on this paper point out the computational complexity of the the multi-agent system when dealing with real problems. Further improvements and research effort is required. Regarding improvements, we are thinking of using the CASS source code that is free from the author's web page. For sure, his code is more effective than our re-implementation. Second improvement is related to the agent platform. JADE has several drawbacks regarding computational costs, and probably, other platforms as RePast can be more suitable.

Finally, we are also interesting on researching bidding policies that guarantees the finding of a complete solution without the necessity of dealing with all the combinations. In this sense, and according to [12], the complexity of the problem can be reduced depending on the topological space of the bids.

References

1. López, B., Suárez, S., de la Rosa, J. In: Task allocation in rescue operations using combinatorial auctions. IOS Press (2003) 233–243 CCIA 2003.
2. Lopez, B., Aciar, S., Innocenti, B., Cuevas, I.: How multi-agent systems support acute stroke emergency treatment. In: Proc. of the IJCAI workshop on Agents Applied in HealthCare. (2005) 51–59
3. Lopez, B.: Time control in road passenger transportation. Problem description and formalization. IIIA 05-04 Research Report, University of Girona (2005)
4. Chevaleyre, Y., Dunne, P., Endris, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-Aguilar, J., Sousa, P.: Issues in multiatgent resource allocation. In: Agentlink III- Technical Forum Group on Multi-agent Resource Allocation. (2005) http://www.doc.ic.ac.uk/ ue/MARA/mara-may-2005.pdf.
5. Sandholm, T.: Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence (2002) 1–54
6. Leyton-Brown, K.: Resource Allocation in Competitive Multiagent Systems. PhD thesis, Stanford University (2003)
7. Abello, X.: Estudi dún mètode dśchedulling distribuit. Aplicació en una empresa de transport terrestre. BsC Thesis, University of Girona (2005)
8. Klugl, F., Bazzan, A., Ossowski, S., eds.: Applications of Agent Technology in Traffic and Transportation. A Birkhuser book (2005)
9. Sousa, P., Ramons, C., Neves, J.: The fabricaire scheduling prototype suite: Agent interaction and knowledge base. Jorunal of Intelligent Manufacturing **14** (2003) 441–455
10. Tomas, V., Garcia, F.: A cooperative multiagent system for traffic management and control. In: AAMAS Industry Track. (2005)
11. Cantillon, E., Pesendorfer, M. In: Auctioning bus routes: The London experience. MIT Press (2006) To appear.
12. Rothkopf, M., A., P.: AComputationally Manageable Combinatorial Auctions. DIMACS Technical Report 95-09 (1995)

This article was processed using the LaTeX macro package with LLNCS style