

Fair Mechanisms for Recurrent Multi Unit Combinatorial Auctions

Javier Murillo and Beatriz López¹

Abstract. Auctions have been used to deal with resource allocation in multi-agent systems. In some environments like service-oriented electronic markets, it is advisable to use recurrent auctions since resources are perishable and auctions are repeated over time with the same or a very similar set of agents. Recurrent auctions are a series of auctions of any kind where the result of one auction may influence the following one. As a drawback some problems do appear that could cause the market to collapse at mid-long term. Previous works have dealt with these problems by adding fairness to the auction outcomes. Those works dealt with multi-unit auctions, in which several units of an item are sold, and they do not assure that agents cannot manipulate the auctions for their own benefit. In this paper, we present new fair mechanisms that goes further. First we focus on combinatorial auctions, in which different items, and several units per item are sold in each auction, which poses additional challenges when they are recurrent. And second, the mechanisms are shown to prevent some agents' manipulation of the auction outcomes.

1 INTRODUCTION

Auctions have become a well known mechanism to share resources among agents in distributed environments. Traditionally, the auctioneer or the owner of the resources exhibits an utilitarian behavior in which he tries to maximize his own revenue by giving the resources to the agents that offer the highest bids. Recent work, such as [6, 9] has shown that this utilitarian behavior can cause the collapse of the market at the mid-long run when dealing with recurrent markets. Thus, this rational behavior exhibited in a single shot game, becomes unsuccessful when repeating the game or auction several times. The reason for that is that this behavior produces the inevitable starvation of certain buyers that could leave the market [7]. So, when poor agents leave, richer bidders have the power to set the prices, provoking a fall in prices that can collapse the market in the mid-term.

Some authors have equipped the traditional mechanisms with reservation prices, which consist in the auctioneer setting a minimum price for the resources and bids below the reservation price cannot be winners. However this adds a new problem: resource waste. With reservation prices, many resources are not sold, and if the resources cannot be stored, they are lost. This is the case of the domain of network bandwidth allocations with which our work is concerned. Thus, if the auctioneer wishes to obtain the maximum revenue from the resources along all the auctions, he needs to look for alternative mechanisms to solve the auction. Particularly, [6, 9] show that if each auction is solved with a fair mechanism, at the end of the recurrent auction the highest revenue is obtained. Thus, all of the agents are

behaving selfishly; and particularly the auctioneer, that by exhibiting a fair behavior also obtains the maximum total revenue.

However, such works consider that the auctioneer sells a constant quantity of units of a unique resource and bidders can only send a unique bid to obtain one unit of the resource. The real world, however, is different, and agents can compete for bandwidth of different quality: voice, high quality voice, video, high quality video, and data. Thus, we are dealing with a multi-unit combinatorial auction (MUCA), in which different kinds of resources, and several units of each resources (e.g. two units of data bandwidth) are auctioned. The extension from a unique kind of resources to multiple kinds of resources in a recurrent scenario is not straightforward, and this is the main contribution of this paper.

In this article we propose two new fair mechanisms for MUCA. In this case, the auctioneer auctions several units of several resources and bidders can bid for several resources and several units of each resource. Moreover, we study the impact of fairness regarding agent manipulation. That is, since the auctions are repeated, bidders can learn about the fairness behavior, and act malevolently in order to obtain the resources at lower prices. We prove that our mechanism, unlike to the previous fair ones, avoids some kinds of manipulation.

2 BACKGROUND

This section shows some necessary background on recurrent multi unit auctions, fairness and trust to understand the following sections.

2.1 RECURRENT MUCA

Recurrent auctions are a series of auctions (of any kind) that are solved individually, but the outcomes of the auctions could be influenced by previous auctions. For example, if a bidder wins an auction, it could reduce the bid in the next one, trying to obtain a higher utility. In this paper we use the recurrent multi-unit, multi-item, sealed-bid combinatorial auction. A *combinatorial* auction is an auction in which bidders can place bids on combination of items, or packages, rather than just individual items [3]. In a *multi-unit* auction the auctioneer offers several (identical) units of each auctioned item. In a *multi-item* auction the auctioneer offers several items or resources. Finally a *sealed-bid* auction is one where the auctioneer is the only agent that knows the bids sent by the bidders.

In an auction process we can distinguish the following three main components. First, the *bidding policies* are the methods used by the agents to decide which items to bid for and the price they are willing to pay for them. Although the auction is sealed bid and bidders do not know the bids of other agents, in a recurrent scenario an agent has information since he knows the result of his previous auctions

¹ University of Girona, Spain, email: jmurillo@eia.udg.edu

and the bid that he has sent. From this information the agent can intuit the global state of the market and choose an appropriate bid. So, we assume that agents have a behavior that tries to maximize their profit using the adaptation strategy defined in [7]. This strategy consists in varying the bid price depending on whether they have won or lost in the previous auction, thus trying to get as close as possible to the optimal bid price with which they can obtain the resources. The second component is the *winner determination problem*, that is the method the auctioneer uses to select the winning bids. The third component is the *pricing mechanism*, that is how the auctioneer decides the price to be paid by the winners. In our case each bidder pays the price he bids (*first-price* auction). Our research is concerned mainly with the winner determination, which poses an optimization problem for auctioneers that try to maximize their revenue [10].

The winner determination algorithm strategy is conditioned on the kind of resource being auctioned. In this paper, we focus on perishable resources. A resource is perishable if it vanishes or loses its value when held over an extended period of time (e.g., the network bandwidth not used is not accumulable for the future).

When perishable resources are auctioned in recurrent auctions some problems appear. First, perishable resources cannot be stored in warehouses for future sales; if the resources are not allocated, they lose their value or vanish completely. This is the *resource waste* problem in recurrent auctions, because if the auctioneer does not sell the resource in an auction, it cannot be sold in the forthcoming auctions. The second problem is the *asymmetric balance of negotiation power*. In most traditional auction mechanisms, the bid prices depend only on the customer's willingness to pay for the traded goods. This means that only the intentions of customers, not those of the auctioneers, are reflected in the auction winning prices [6]. In the long run, the effect of this problem may cause the auction to collapse. And third, the *bidder drop* problem occurs when bidder agents participating in many auctions always lose. These bidders could decide to leave the market because they are not making any profit. The reduction in the number of bidders gradually decreases the price competition because the probability of winning increases for the remaining bidders. Bidders can decrease their bids without losing the chance to win, provoking the overall drop in bid price².

Finally, an important drawback that can appear in this kind of auction is the manipulation of the mechanism by bidders. That is, different trials of bidders in a recurrent auction could be planned in order to take advantage of the auctioneer's mechanism. This problem is worse when we are dealing with several resources and several units as happens in combinatorial auctions. Thus, the winner determination algorithms should be strategy-proof in the sense that no bidder can manipulate the outcome of the auction. Our mechanisms are based on reinforcement mechanisms, so that agents have priorities and reservation prices that are dynamically adapted according to the auction outcomes, including agent's honesty.

2.2 Fairness

The application of fairness in recurrent auctions can be considered from either a local or a global point of view. A local point of view means treating each auction separately and trying to find a fair solution to each one. With a global point of view we can find that all the solutions of a set of auctions are fair. In this paper we focus on the global point of view. Fairness can also be analyzed in light of the

agent who is fair. From the point of view of the bidder agent, fairness means that the objective of the agent is not only to maximize revenue in the short term but also to consider fairness at the time of making decisions. Fairness can benefit the whole society and consequently it allows to obtain greater individual benefits. In this case the solution to the allocation of resources is given by the interaction, negotiation and coalition between agents, etc. [5]. It is also possible to consider fairness from the point of view of the seller agent. In this case the behavior of bidders could be totally utilitarian and the seller agent imposes fairness to distribute the resources [8]. In this article we focus on the seller point of view. Other fairness studies take the bidder point of view, such as in an envy-free allocation, where none of the agents would prefer to exchange his allocated goods with those allocated to another agent [1]. However, this criteria alone is not sufficient since the efficiency is also important. Other studies [2] have shown how such efficient envy-free allocations can be attained through distributed negotiation among the agents. In such a setting, there is no central agent or authority to decide what the optimal allocation is, but the agents themselves perform a sequence of deals (exchanges of resources and payments) to find an efficient and envy-free allocation. The work presented by these authors, however, deals with an isolated allocation problem, and does not mention how distributed negotiation could be applied when the agents are faced with a sequence of allocation problems. In another study [4] the concept of egalitarian social welfare is defined to measure the degree of fairness in this envy-free scenario, as well as for a general purpose.

3 RELATED WORK

There are two main previous works on fair mechanisms. First, Lee and Szymanski proposed a method based on agent classification and reservation prices [6]. Agents are classified into three categories: definite winners, definite losers, and the "winner or loser" ones. This classification varies from auction to auction and depends on a reservation price that defines the separation between definite winners and "winner or loser" class. The reservation price changes according to an *ad hoc* mechanism and is updated in each auction. Based on a dropout probability model maintained by the auctioneer about the agents, resources are assigned to winner agents and "winner or loser" agents with the aim of keeping agents interested in the auction. After Lee and Szymanski's work, in [9] a reinforcement learning method was used to determine the reservation price from run to run, based on the previous reservation prices and the price paid by the bidder. Thus, reservation prices are set one per agent instead of a general value for all the bidders. Both approaches have been tested to demonstrate that fairness is able to reduce the problems of traditional recurrent auctions, but they have been developed for non-combinatorial auctions.

4 FAIR MUCA MECHANISMS

Our work focuses on developing a recurrent auction mechanism that clears the auction with fairness from the point of view of the auctioneer along time. Thus, after introducing some notation, in this section we first explain the mechanism based exclusively on priorities and then we extend the mechanism with dynamic reservation prices.

4.1 Notation

The recurrent auction A is formed by a succession of one-shot auctions $A = \{a_0, \dots, a_{|A|-1}\}$. In each auction a_i , $|R_i|$ resources or

² We assume that we are working in a closed environment where there are not arrivals of new bidders.

items are auctioned where R_i is a set composed by tuples that indicates the resource r_x and the number of units of each resource q_x ($R_i = \{\langle r_0, q_0 \rangle, \dots, \langle r_{|R_i|-1}, q_{|R_i|-1} \rangle\}$). A set R_i of an auction a_i can be different of the set R_j corresponding to auction a_j , $i \neq j$.

Auctions are ordered temporally and we associate to each auction a_i a time step t_i where T is the set of time steps in which the auctions of A occur ($T = \{t_0, \dots, t_{|A|-1}\}$ where $t_0 < t_1 < \dots < t_{|A|-1}$). G is the set of all bidder agents, $G = \{g_0, \dots, g_{|G|-1}\}$, that participate in the recurrent auction A . A bidder agent g_i sends to the auctioneer agent a set $V_{i,j}$ of bids in auction a_j . Each bid is composed by a package and the price. A package is formed by one tuple indicating the items required by the agent and the number of units of each item (e.g. $\{\langle bandwidth, 2 \rangle, \langle memory, 5 \rangle\}$). We define as $\mathbb{S}_i$ the set of all packages sent by agent g_i in the recurrent auction, $\mathbb{S}_i = \{S_{i,0}, \dots, S_{i,|\mathbb{S}_i|-1}\}$. Therefore a bid is defined as $b_{i,j,x} = \langle S_{i,m}, c_{i,j,x} \rangle$ where x is the number of bid in the auction a_j sent by agent g_i to obtain the package $S_{i,m}$ ($0 \leq m < |\mathbb{S}_i|$) paying $c_{i,j,x}$. We define the functions $\mathcal{S}(b_{i,j,x})$ to know the index of the package of bid $b_{i,j,x}$ and $\mathcal{C}(b_{i,j,x})$ to know the bid price.

We have chosen the XOR bidding language. This language allows to bidders express any preference over resources [3]. The bidder sends a set of bids XOR, consequently only one or none can be winner. Solving a combinatorial auction is NP-hard [10] and in the case of a recurrent auction, an auction must finish after the start of the following one and before the perish of resources too. For the sake of simplicity, we do not deal with the way of solving the combinatorial auctions in this paper.

4.2 Prioritized MUCA (PMUCA)

The PMUCA mechanism takes into account the past history of each agent. The auctioneer assigns to each bidder a set of priorities, one priority for each different package that the bidder agent has bid for in the recurrent auction. The value of each priority depends on the number of lost auctions since the bidder won a package for the last time. The more auctions the bidder has lost the greater the priority. The priority values are calculated by the auctioneer and are only known by him. The priority values are calculated after the reception of bids and are used to solve the auction modifying the prices $c_{i,j,x}$ of the bids $b_{i,j,x}$ generating new bids $b'_{i,j,x}$ with new prices $c'_{i,j,x}$. The auctioneer stores in a set L_i the counters of lost auctions for each bidder g_i , one counter for each element of $\mathbb{S}_i$ ($L_i = \{l_{i,0}, \dots, l_{i,|\mathbb{S}_i|-1}\}$).

However, the outcome from such fair winner determination, that is, changing the price from $c_{i,j,x}$ to $c'_{i,j,x}$, could be learnt easily by dishonest agents. That is, a bidder that is not actually interested in an auction run could send a fake bid with the aim to lose and thus obtain a greater priority in the next auction where the agent is truly interested. Thus, a mechanism should not incentivize the participation in the auctions where the bidder is not actually interested.

On the other hand a mechanism that works with combinatorial auctions must take into account the different nature of items. A bidder could send a bid for an item of little value to obtain a high priority and then use the priority to bid for a high value item. For example a bidder could send a bid for a pencil until obtaining a high priority assuming the risk of winning the pencil since the cost of the pencil is low. Then, when the bidder obtains a high priority, he can use the priority to bid for a car.

In order to avoid these situations we define a *cycle* as a set of consecutive auctions where the priority of the agent g_i for the package $S_{i,m}$ is not decreasing. Each agent has a different cycle for each package. When a cycle ends the priority is initialized with an initial

value. A cycle ends and another starts when a bidder agent wins the package or when the agent sends a bid for the package with a price higher than the bid price that the agent sent at the first auction of the cycle. If a bidder agent sends more than one bid, the agent could have a different priority for each bid since each bid requests a different package. We define $k_{i,m}$ as the price of the bid in the first auction of the current cycle of agent g_i and package $S_{i,m}$. We define K_i as the set of all the $k_{i,m}$ of agent g_i , $K_i = \{k_{i,0}, \dots, k_{i,|\mathbb{S}_i|-1}\}$.

The priority $p_{i,m}$ of an agent g_i and a package $S_{i,m}$ is calculated with the following equation:

$$p_{i,m} = \begin{cases} l_{i,m} < \gamma & \lambda + \frac{1-\lambda}{\gamma} \cdot l_{i,m} \\ l_{i,m} \geq \gamma & 1 \end{cases} \quad (1)$$

Where λ is the initial priority and γ indicates the number of lost auctions necessary so that the priority arrives to its maximum value (1.0). $l_{i,m}$ is the number of lost auctions in the actual cycle for package $S_{i,m}$. The expression $\frac{1-\lambda}{\gamma}$ shows the increase of priority for each lost auction. After priorities are calculated, the modified bids $b'_{i,j,x}$ are generated. The bid $b'_{i,j,x}$ of an agent g_i that has sent a bid $b_{i,j,x}$ in the auction a_j is:

$$b'_{i,j,x} = \langle S_{i,\mathcal{S}(b_{i,j,x})}, \mathcal{C}(b_{i,j,x}) \cdot p_{i,\mathcal{S}(b_{i,j,x})} \rangle \quad (2)$$

Then the auction is solved in the traditional form with the $b'_{i,j,x}$ bids. When winners are determined, the mechanism updates the sets L_i and K_i (see Algorithm of Figure 1). For each bidder g_i if any of his bids has been winner, the actual cycle is ended for each package $S_{i,m}$ corresponding to the bids sent by agent g_i , therefore all counters $l_{i,m}$ of agent g_i that corresponds to bids sent in the auction are initialized. In the case of all bids being losers, then all counters corresponding to g_i are incremented one unit except if the price is lower than $k_{i,m}$.

```

{ 1 }   for each bidder  $g_i$ 
{ 2 }       if agent  $g_i$  has won in current auction  $a_j$ 
{ 3 }           for each bid  $b_{i,j,x}$  in  $V_{i,j}$ 
{ 4 }                $l_{i,\mathcal{S}(b_{i,j,x})} \leftarrow 0, k_{i,\mathcal{S}(b_{i,j,x})} \leftarrow \mathcal{C}(b_{i,j,x})$ 
{ 5 }           else
{ 6 }               for each bid  $b_{i,j,x}$  in  $V_{i,j}$ 
{ 7 }                   if ( $k_{i,\mathcal{S}(b_{i,j,x})} < \mathcal{C}(b_{i,j,x})$ )
{ 8 }                        $l_{i,\mathcal{S}(b_{i,j,x})} \leftarrow 0, k_{i,\mathcal{S}(b_{i,j,x})} \leftarrow \mathcal{C}(b_{i,j,x})$ 
{ 9 }                   else
{10 }                        $l_{i,\mathcal{S}(b_{i,j,x})} \leftarrow l_{i,\mathcal{S}(b_{i,j,x})} + 1$ 
```

Figure 1. Algorithm for update the L_i and K_i sets after auction a_j .

In order to show that our mechanism cannot be affected by the explained manipulation, let us suppose two consecutive auctions: a_0 and a_1 . An agent wants to win a package in the auction a_1 but not in a_0 . Suppose the following two situations: an agent that is honest (g_h) and an agent that is not (g_{-h}). So, g_h does not participate in the first auction, and in the second he sends a bid $b_{h,1,1}$ for the package with a price $\mathcal{C}(b_{h,1,1})$. While, g_{-h} sends a bid in the first auction with a low price $\mathcal{C}(b_{-h,0,1})$ with the aim of losing and raising his priority for the package in the next auction. In the second auction g_{-h} sends a bid with a price $\mathcal{C}(b_{-h,1,1})$, where $\mathcal{C}(b_{-h,0,1}) < \mathcal{C}(b_{-h,1,1})$. Assume that the price is the same for both agents in the second auction, $\mathcal{C}(b_{-h,1,1}) = \mathcal{C}(b_{h,1,1})$. Thus, in order to analyze which agent has

more chances to become a winner, we define $pow_x(y)$ as the probability of winning in the auction a_x with a bid price y . Then the probability of winning for the honest agent in the second auction a_1 is $pow_1(\mathcal{C}(b_{h,1,1}) \cdot pr)$ where pr is the priority of agent g_h . Analogously, the probability of winning for g_{-h} is $pow_1(\mathcal{C}(b_{-h,1,1}) \cdot \lambda)$. Note that $pr \geq \lambda$ since $\mathcal{C}(b_{-h,0,1}) < \mathcal{C}(b_{-h,1,1})$ and consequently the priority for g_{-h} is initialized with the initial value λ . Then the probability of winning for g_h is equal or greater than for g_{-h} :

$$pow_1(\mathcal{C}(b_{h,1,1}) \cdot pr) \geq pow_1(\mathcal{C}(b_{-h,1,1}) \cdot \lambda) \quad (3)$$

Regarding the price that the agents have to pay, g_h has an expected value of $pow_1(\mathcal{C}(b_{h,1,1}) \cdot pr) \cdot \mathcal{C}(b_{h,1,1})$. On the other hand g_{-h} has an expectation of $pow_1(\mathcal{C}(b_{-h,1,1}) \cdot \lambda) \cdot \mathcal{C}(b_{-h,1,1}) + pow_0(\mathcal{C}(b_{-h,0,1}) \cdot pr) \cdot \mathcal{C}(b_{-h,0,1})$. Then, assuming the worst case in which the honest agent was also the initial priority value ($pr = \lambda$), we obtain the results shown in Equation 4.

$$\begin{aligned} & pow_1(\mathcal{C}(b_{h,1,1}) \cdot \lambda) \cdot \mathcal{C}(b_{h,1,1}) \leq \\ & pow_1(\mathcal{C}(b_{-h,1,1}) \cdot \lambda) \cdot \mathcal{C}(b_{-h,1,1}) + \\ & pow_0(\mathcal{C}(b_{-h,0,1}) \cdot \lambda) \cdot \mathcal{C}(b_{-h,0,1}) \end{aligned} \quad (4)$$

As $\mathcal{C}(b_{h,1,1}) = \mathcal{C}(b_{-h,1,1})$ we can see that g_{-h} has more chances to pay equal or more than g_h supposing that they win the same number of times. Then, the best strategy for a bidder is not to participate in the auction if the agent is not really interested in winning it.

4.3 Dynamic Reservation Price MUCA (DRPMUCA)

Priorities are fair since they allow bidders from win time to time, independently of their wealth, and keep agents interested in the auction. However fairness can be improved if a minimum price can be defined for each agent according to his wealth. The main purpose of reservation prices is to incentive bidders to pay a higher value for resources. In DRPMUCA the auctioneer has a set RP_i of reservation prices for each bidder agent g_i . In RP_i there is a reservation price for each package sent by g_i ($RP_i = \{rp_{i,0}, \dots, rp_{i,|S_i|-1}\}$). The reservation price $rp_{i,m}$ is defined as the minimum price at which the auctioneer is willing to sell the package $S_{i,m}$ to the agent g_i . This means that the auctioneer will not accept a bid of agent g_i for package $S_{i,m}$ with a price lower than $rp_{i,m}$. The reservation prices are dynamically changed according to the bids sent in each auction.

DRPMUCA is composed by three phases. In the first phase the auction is solved in the traditional way but only taking into account the bids $b_{i,j,x}$ that fulfils the condition $\mathcal{C}(b_{i,j,x}) \geq rp_{i,\mathcal{S}(b_{i,j,x})}$. In this first phase the bids are not modified. Once determined the winners of the first phase, the idea of the second phase is to sell the resources that are not being distributed in the first phase between the loser bidders with higher priority without taking into account the bid price. But in order to an agent become a winner in the second phase, the priority of the bid has to be greater than a parameter ξ . This minimum priority value avoids having the rich agents obtaining the resources at low price due to the fair distribution. The priority is calculated using Equation 1. Then each bid $b_{i,j,x}$ that belongs to loser bidders with a priority greater than ξ is modified in order to obtain $b'_{i,j,x}$:

$$b'_{i,j,x} = \langle \mathcal{S}(b_{i,j,x}), p_{i,\mathcal{S}(b_{i,j,x})} \rangle \quad (5)$$

A new auction is solved with the modified bids. Finally, in the last phase of DRPMUCA, the information related to priorities (L_i

and K_i sets) are updated following the algorithm of Figure 1 and reservation prices as follows. For each agent, if the price of a bid that requires the package $S_{i,m}$ is greater than $rp_{i,m}$, $rp_{i,m}$ will be incremented for the next auction since the bidder is showing that he can pay more for the package. Conversely if the price is lower, $rp_{i,m}$ will be decremented. Note that changes in $rp_{i,m}$ are independent of whether the bids are winners or losers, or they participate or not (if they do not participate is equivalent to send a bid with price 0 for all of the packages, and the reservation price is decreased). We define $\delta \in [0, 1]$ as the minimum percentage of increment or decrement. When $rp_{i,m}$ is incremented, the increment is the maximum between the half of the difference between the price and the $rp_{i,m}$ and $rp_{i,m} \cdot \delta$. When $rp_{i,m}$ is decremented, the new $rp_{i,m}$ is $rp_{i,m} - (rp_{i,m} \cdot \delta)$. Reservation prices are updated at the end of each auction and are used to solve the next auction. Figure 2 shows the update of each RP_i .

```

{ 1 } For each bidder  $g_i$ 
{ 2 }   For each package  $rp_{i,m}$  in  $RP_i$ 
{ 3 }   If bidder  $g_i$  has sent a bid  $b_{i,j,x}$  for  $S_{i,m}$ 
{ 4 }     If  $\mathcal{C}(b_{i,j,x}) > rp_{i,m}$  then
{ 5 }        $rp_{i,m} = rp_{i,m} + \max(\frac{\mathcal{C}(b_{i,j,x}) - rp_{i,m}}{2}, rp_{i,m} + rp_{i,m} \cdot \delta)$ 
{ 6 }     else
{ 7 }        $rp_{i,m} = rp_{i,m} - rp_{i,m} \cdot \delta$ 
{ 8 }     else
{ 9 }        $rp_{i,m} = rp_{i,m} - rp_{i,m} \cdot \delta$ 

```

Figure 2. Update algorithm of RP_i after a_j auction

In order to show that this mechanism cannot be affected by the explained manipulation too, let us suppose two consecutive auctions: a_0 and a_1 . An agent wants to win a package in the auction a_1 but not in a_0 . Suppose an honest agent (g_h) and a dishonest agent (g_{-h}). g_h does not participate in the first auction and in the second sends a bid $b_{h,1,1}$ for the package with a price $\mathcal{C}(b_{h,1,1})$. In the second case, g_{-h} sends a bid in the first auction with a low price $\mathcal{C}(b_{-h,0,1})$ with the aim of losing and decreasing the reservation price for the package in the next auction. In the second auction g_{-h} sends a bid with a price $\mathcal{C}(b_{-h,1,1})$, where $\mathcal{C}(b_{-h,1,1}) = \mathcal{C}(b_{h,1,1})$. In both cases the reservation prices are decremented thus the resulting reservation prices are the same and consequently the probability of obtaining the package is the same in both situations.

Regarding the payments, analogously to the case of PMUCA, the honest agent has to pay the same or less than g_{-h} :

$$\begin{aligned} & pow_1(\mathcal{C}(b_{h,1,1})) \cdot \mathcal{C}(b_{h,1,1}) \leq pow_1(\mathcal{C}(b_{-h,1,1})) \cdot \mathcal{C}(b_{-h,1,1}) \\ & + pow_0(\mathcal{C}(b_{-h,0,1})) \cdot \mathcal{C}(b_{-h,0,1}) \end{aligned} \quad (6)$$

5 EXPERIMENTATION

In this section we show the experimentation done with the proposed mechanisms. First of all we define a fairness measure to know the level of fairness obtained by the different mechanisms. Then we describe our experimental scenario and finally we show the results obtained in a bandwidth allocation domain.

5.1 Fairness measure

We define a measure that takes into account the sequence of winning and losing bids of an agent in a recurrent auction. This measure provides an aggregation of bidder satisfaction values of each

auction round. We assume that if a bidder sends a price $\mathcal{C}(b_{i,j,1})$ in a bid $b_{i,j,1}$ and in other bid $b_{i,j,2}$ he sends a price $\mathcal{C}(b_{i,j,2})$, if $\mathcal{C}(b_{i,j,1}) > \mathcal{C}(b_{i,j,2})$, then we suppose that the bidder is more interested in winning the bid $b_{i,j,1}$. Consequently we assume that if he wins $b_{i,j,1}$, it has a greatest satisfaction degree than winning $b_{i,j,2}$. Then we define the satisfaction of an agent g_i in an auction a_j as the won value divided by the maximum that the agent could win. Formally:

$$s(i, j) = \frac{\sum_{\forall b_{i,j,x} \in V_{i,j}} (\mathcal{C}(b_{i,j,x}) \cdot \mathcal{W}(b_{i,j,x}))}{\mathcal{M}(V_{i,j})} \quad (7)$$

Where $V_{i,j}$ is the set of bids sent by agent g_i in the auction a_j . $\mathcal{W}(b_{i,j,x})$ indicates whether bid $b_{i,j,x}$ has resulted winner (value 1) or not (value 0) and $\mathcal{M}(V_{i,j})$ returns the highest bid price of the $V_{i,j}$ set. Thus the level of fairness of an auctioneer can be defined as the inverse of the standard deviation of the average satisfaction degree of all of the agents.

5.2 Experimental setup

We have created a bandwidth allocation scenario based on extending the one described in [6] to multi unit combinatorial auctions. Each auctioneer is a bandwidth provider that owns 60 units of bandwidth that are distributed in five groups. The first group is dedicated to voice (20 units), the second group is dedicated to high quality voice (10 units), the third group is dedicated to video (10 units), the fourth group is dedicated to high quality video (10 units) and the fifth group is dedicated to data (10 units). Then the auctioneer can provide five different services or resources: Voice (VO), High Quality Voice (HQVO), Video (VI), High Quality Video (HQVI) and Data (D).

There are 100 bidder agents that are requesting services. Bidders can be classified in six groups. The agents of the first group (20 agents) are doing a video conference and send a bid to obtain the package $\{\langle VI, 1 \rangle, \langle VO, 1 \rangle\}$. The second group (15 agents) is watching TV and want a good quality, and bid for $\{\langle HQVI, 1 \rangle, \langle HQVO, 1 \rangle\}$ xor $\{\langle VI, 1 \rangle, \langle HQVO, 1 \rangle\}$ xor $\{\langle HQVI, 1 \rangle, \langle VO, 1 \rangle\}$ xor $\{\langle VI, 1 \rangle, \langle VO, 1 \rangle\}$. Obviously for the first bid they offer the highest price but if they cannot win the first bid they want to win any of the alternatives. The third group (30 agents) is composed by bidders that are doing phone calls and they want $\{\langle VO, 1 \rangle\}$. The members of the fourth group (10 agents) want $\{\langle HQVO, 1 \rangle\}$ xor $\{\langle VO, 1 \rangle\}$ since they are listening to music. In the fifth group there are 15 agents that need $\{\langle D, 1 \rangle\}$. Finally the agents of the last group (10 agents) need $\{\langle D, 2 \rangle\}$ because they want to download files at a high velocity.

To compare the results of the proposed mechanisms, three unfair auction mechanisms have been considered. First the Traditional MUCA (TMUCA) mechanism in which the winners are the bidders with the highest bids. This is the simplest auction mechanism. Second the Reservation Price MUCA (RPMUCA) in which the auctioneer defines a reservation price (the same for all bidders) that indicates the minimum price the bidders should pay. Only bids higher than the auctioneer's reservation price are considered during the winner selection. And third, the Cancellable MUCA (CMUCA) where if the resulting revenue of an auction does not meet the minimum requirements of the auctioneer, the entire auction is canceled.

In order to implement the bidding policy of bidders we have adapted the strategy proposed in [6] to combinatorial auctions. For each possible package $S_{i,m}$ bidders have a $z_{i,m}$ value that indicates the current price that they offered by the package $S_{i,m}$ and a value

$w_{i,m}$ that indicates the maximum price that they can pay for the package. $z_{i,m}$ is dynamic during the auctions while $w_{i,m}$ is chosen at the beginning of the simulation and it maintains its value during all the recurrent auction. The initial value of $z_{i,m}$ is a random value between $w_{i,m}/2$ and $w_{i,m}$. If a bidder loses in an auction a_j , the bidder increments the value of $z_{i,m}$ multiplying $z_{i,m}$ by α for all the packages that the agent has requested in the auction a_j in order to increment the probability of winning in the next auction. We consider that a bidder loses the auction a_j if none of his bids results as winner. In the opposite, if the bidder won in the auction a_j , all the prices of the packages requested are decremented multiplying it by β the 50% of the times. The other 50% the prices remain with the same value. The values of α and β are fixed to 1.2 and 0.8 respectively. The minimum price that a bidder g_i can offer for a package $S_{i,m}$ is the 10% of the $w_{i,m}$ value. Before starting the simulation a value of wealth is assigned to each bidder following several statistical distributions. This value will be used to set the values of $w_{i,m}$. The value for each $w_{i,m}$ is the wealth of the bidder multiplied by the number of units of the package counting twice the high quality items.

In order to model the bidder drop problem we have used the same concept of Tolerance to Consecutive Losses defined in [6]. Each bidder has a TCL value randomly distributed between 2 and 10. If a bidder agent loses more than his TCL consecutive auctions he leaves the recurrent auction.

5.3 Results

Figure 3 shows the average resource price at which each of the 5 auctioneers has sold the resources during the 2000 auctions of one simulation. The x -axis represents the temporal sequence of the auctions, and the y -axis is the average price. The wealth of bidders has been distributed following a uniform distribution in [2, 10]. The numerical data is shown in Table 1, where the total revenue, the resource waste produced, the fairness level, and the number of bidders remaining with the auctioneer at the end are shown. The table values are the average of 20 simulations with the standard deviation below.

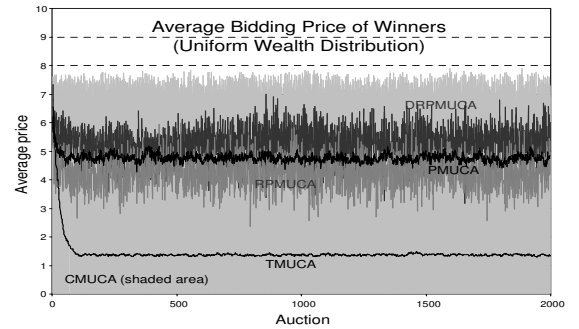


Figure 3. Average bidding price in scenario 1 (uniform wealth distribution). CMUCA appears in the plot as an area because the average prices are 0 when the auction is cancelled.

Results on Table 1 show that the bidder drop problem affects in a major way TMUCA, CMUCA and RPMUCA. In the case of TMUCA, the drop of bidders has provoked the collapse of the auction. Figure 3 shows that the average price of TMUCA has fallen to very low prices in less than 100 auctions. The bidders that have not dropped have obtained the power of negotiation and consequently have fixed the price at the minimum possible. The rest of the methods, despite the drop of bidders, has maintained the power of negoti-

ation thanks to the reservation prices. Regarding the resource waste, RPMUCA and CMUCA have been the mechanisms most affected. They have wasted 31.87% and 38.94% of resources respectively. Below these mechanisms, DRPMUCA has lost 15.32% of resources and finally the mechanism non based on reservation prices are the ones that have wasted less resources. Note that these methods can also have resource waste due to the combinatorial nature of the problem. Regarding the total revenue, DRPMUCA has been the mechanism that has obtained the first position followed by RPMUCA, PMUCA and CMUCA. Finally regarding fairness, DRPMUCA is the mechanism with a higher value followed by PMUCA. We can see that the methods with higher fairness obtains the highest revenue.

Table 1. Results for all mechanisms with a uniform distribution of wealth in scenario 1. Averaged results of 20 simulations

Name	◇	□	△	✕
PMUCA	550119.74	5729.30	68.60	3.51
	60183.99	1480.90	2.43	0.08
DRPMUCA	666314.48	18384.25	66.50	4.10
	41969.16	2283.99	2.18	0.12
TMUCA	178297.63	2439.00	52.25	2.71
	29690.01	1835.44	1.58	0.08
RPMUCA	549565.30	38255.10	45.25	3.10
	53360.16	6779.03	3.64	0.03
CMUCA	532021.85	46737.40	37.60	3.46
	19758.86	2932.32	1.04	0.06

◇ Total revenue □ Resource Waste △ Number of bidders ✕ Fairness

On the other hand, in a bandwidth communications domain it is feasible to expect malfunctions, technical problems with the network, and maintenance issues that decrease the available supply. In a second experiment we have reduced the supply in the auctions during the time steps between 1000 and 1100 by a 40%. Figure 4 (left) shows the average price in one simulation of this experiment. We can see how prices rise for all methods during the time with scarce units of resource (less supply, prices go up). During this period all methods suffer the drop of the poorest bidders, and as a consequence, some of the mechanisms are not able to return to their same or close steady-state when the supply is restored. Only DRPMUCA and RPMUCA achieve an acceptable average price when the supply is restored. The average total revenue in 20 simulations for RPMUCA is 508869.65 lower than the 635399.09 obtained by DRPMUCA.

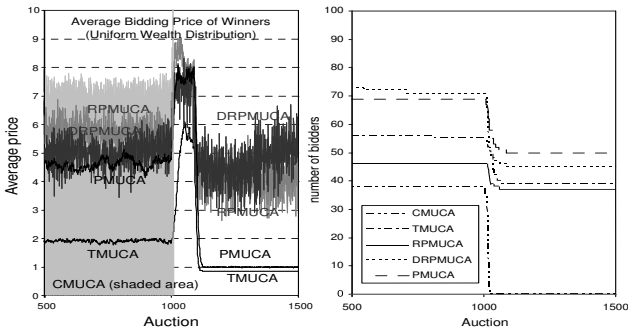


Figure 4. Average bidding price in scenario 2 (left). Number of bidders in the recurrent auction (right). Detail between time step 500 and 1500

In this experiment the effect of bidder drop problem is higher. All the methods have lost more bidders than in the first experiments. In Figure 4 (right) we can see the drop of bidders produced at time step

1000. PMUCA is the method that has maintained the greatest number of bidders but when the supply has returned to its normal value, the method has collapsed. We can find the explanation on the fair distribution of the resources. This mechanism focus on avoiding resource waste, and it is always trying to sell all the resources. If we classify the bidders in two classes, rich and poor agents, the bidders from the poor class put pressure on the upper class. If bidders from the high class relax, the bidders of the lower class get the resources, and the rich ones do not get the resources, consequently they maintain prices at an acceptable level. When bidders of the upper class get the resources at low prices due to the fair distribution, then they have no incentive to increase their bids, and prices begin to decrease. This does not happen with DRPMUCA, because they sacrifice resource waste for avoiding that rich agents obtain the resources due to the fair distribution thanks to the ξ parameter.

6 CONCLUSIONS

In environments where resources are perishable and the allocation of resources is repeated over time with the same set or a very similar set of agents, recurrent auctions come up. A recurrent auction is a series of auctions where the result of one auction can influence the following ones. However, these kinds of auctions have particular problems when the wealth of the agents is unevenly distributed and resources are perishable. Fair mechanisms have already been proposed in order to deal with this problem in non-combinatorial auctions. In this work we have presented two mechanisms for recurrent combinatorial auctions based on priorities and reservation prices that minimize these problems and prevents some kind of bidder's manipulation. The results show that fairness gives bidders incentives to stay in the auction and reservation prices help maintaining the equilibrium in negotiation power. Moreover the DRPMUCA proposed here avoids collapse by introducing a minimum priority value that prevents the fairly distributed resources from arriving to the middle-upper class and presents the highest averaged performance of all the simulated conditions and makes a profit for the auctioneer. However, we need to further explore different manipulation strategies to clearly claim about the impossibility of manipulating our mechanism.

REFERENCES

- [1] S.J. Brams and A.D. Taylor, *Fair division: From cake-cutting to dispute resolution*, Cambridge University Press, 1996.
- [2] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet, 'A short introduction to computational social choice', in *SOFSEM-2007*, number 4362 in LNCS. Springer-Verlag, (2007).
- [3] *Combinatorial Auctions*, eds., Peter Cramton, Yoav Shoham, and Richard Steinberg, MIT Press, 2006.
- [4] U. Endriss, N. Maudet, F. Sadri, and F. Toni, 'Negotiating socially optimal allocations of resources', *JAIR*, **25**, 315–348, (2006).
- [5] S. De Jong, K. Tuyls, and K. Verbeeck, 'Fairness in multi-agent systems', *Knowledge Engineering Review*, **23**(2), 153–180, (2008).
- [6] J-S. Lee and B.K. Szymanski, 'A novel auction mechanism for selling time-sensitive e-services', in *IEEE International Conference on E-Commerce Technology*, pp. 75–82. IEEE Computer Society, (2005).
- [7] J-S. Lee and B.K. Szymanski, 'Stabilizing markets via a novel auction based pricing mechanism for short-term contracts for network services', in *Integrated Network Management*, pp. 367–380, (2005).
- [8] M. Lemaître, G. Verfaillie, H. Fargier, J. Lang, N. Bataille, and J.-M. Lachiver, 'Equitable allocation of earth observing satellite resources', in *Proc. 5th ONERA-DLR Aerospace Symposium (ODAS 2003)*, (2003).
- [9] J. Murillo, V. Muñoz, B. López, and D. Busquets, 'A fair mechanism for recurrent multi-unit auctions', in *Lecture Notes in Computer Science*, volume 5244, pp. 147–158. Springer, (2008).
- [10] T. Sandholm, 'Algorithm for optimal winner determination in combinatorial auctions', in *Artificial Intelligence*, pp. 542–547, (2002).