

# To Bid or not To Bid

## Agent Strategies in Electronic Auction Games

Javier Béjar<sup>1</sup> and Juan A. Rodríguez-Aguilar<sup>2</sup>

Departament de Llenguatges i Sistemes Informàtics<sup>1</sup>  
Universitat Politècnica de Catalunya  
c/ Jordi Girona 1-3. 08034 Barcelona, Spain  
bejar@lsi.upc.es  
Phone: 34 3 4015653 Fax: 34 3 4017014

Institut d'Investigació en Intel·ligència Artificial<sup>2</sup>  
Consejo Superior de Investigaciones Científicas  
08193 Bellaterra, Spain  
jar@iia.csic.es

**Abstract.** This paper presents the results and analysis of the *Fishmarket* tournament held this spring at the Technical University of Catalonia (UPC) by a group of undergraduate students as a course work for an artificial intelligence applications course.

In the tournament participated sixteen different agents that competed in a three phase eliminatory competition. The agents were divided in groups of four and competed in a number of Downward Bidding Protocol (DBP) auctions for boxes of fish.

We present the information analyzed by the students in order to build their agents, what information was considered relevant, and the different strategies of the agents.

**Keywords:** Autonomous Agents, Multiagent Systems, Electronic Institutions, e-Auctions

## 1 Introduction

This work presents the results of the spring auction tournament held at the Technical University of Catalonia. The participants of the tournaments are students from the undergraduate course on applications of the artificial intelligence from the Barcelona School of Informatics.

This kind of tournaments have been held during the past five years with very fruitful results. The agents implemented has been used as a test for the Fishmarket platform and had aid to tune and extend its possibilities.

This competition consists in set of auctions of goods (fish boxes) using the Downward Bidding Protocol (DBP) as auction protocol. The goal of the agents is to pursue the greater benefit.

Notice that our initiative shares many commonalities with the *Double auction* tournaments held by the Santa Fe Institute[1] where the contestants competed for developing optimized trading strategies.

The agents used in this last tournaments, a total of sixteen, were developed in groups of three students to introduce issue about electronic markets and their relationship with autonomous agents.

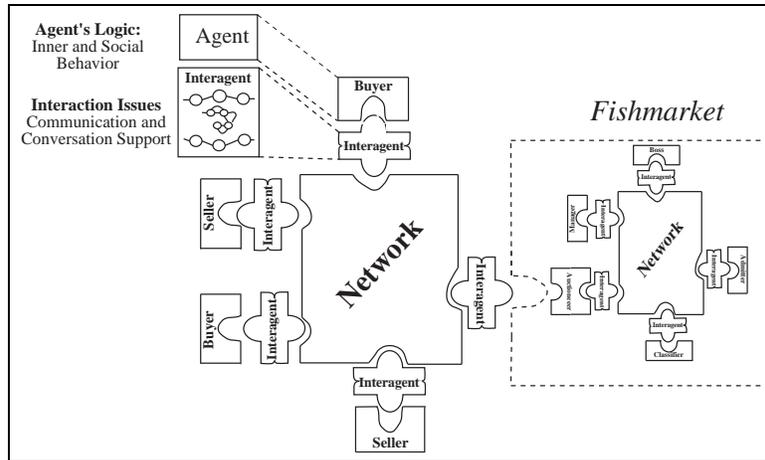
The fishmarket platform provides all the implementations needs (data structures, market information, communication, etc.), so the only problem to solve is the strategy to deal with the auctions. The students had all the available information about how the market works and the parameters that the platform provides. As way to stimulate competition among the different groups, a part of the mark of the course is related to the performance of their agents in the tournament.

This article is organized as follows. In section 2 we will briefly describe the development framework and the characteristics of the auctions that can be held with the Fishmarket platform. Section 3 will be devoted to the characteristics of the spring tournament, its parameters, the different scenarios that the agents had to face and the goals that were pursued. In section 4, the different agents will be analyzed, describing its strategies and the different market information that were used. Section 5 will summarize the results of the tournament and the explanation of the success of the different strategies. Finally, the section 6 will summarize all the conclusions drawn from the tournament.

## 2 The Experimental Framework

In order to obtain an auction tournament environment, more functionality has been added to the FM96.5 agent-mediated electronic auction house[14] to turn it into a *domain-specific* test-bed that models and simulates an e-auction house that henceforth we shall refer to as *FM*. A distinguishing feature of the resulting test-bed is that it is *realistic* since it has been built out of a complex real-world application. Being an extension of FM96.5, FM inherits *interagents*, the mechanism of interaction between trading agents and the market. As introduced in [7]interagents are a particular type of facilitators conceived as autonomous software agents devoted to mediating the interaction among agents in an agent society in general and in an agent-mediated institution in particular. Thus, interagents constitute the unique mean through which agents interact within a multi-agent scenario as depicted in Figure 1. Interagents are all *owned* by the institution but *used* by external agents. As a major role, interagents are responsible for guaranteeing the enforcement of institutional norms to external agents.

Consequently FM shows a crisp distinction between agents and the simulated world, a desirable requirement for any multi-agent test-bed. Furthermore, the use of interagents permits also to consider FM as an *architecturally neutral* environment since no particular agent architecture (or language) is assumed or provided. However, some support for agent developers is provided by including a library of agent templates in various languages (C, Java, and Lisp) for



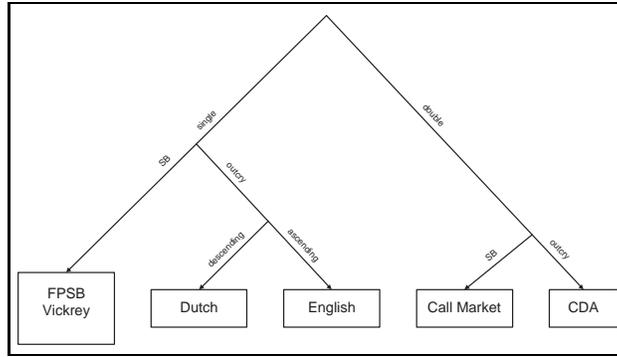
**Fig. 1.** Interagents in an Agent-mediated Institution.

building agents. Furthermore, the test-bed also offers the possibility of generating customisable *dummy agents* at the aim of providing agent developers with contenders for training purposes.

FM inherits also all the auction protocols included in FM96.5, namely Dutch, English, First-price sealed-bid and Vickrey. All these auction protocols are classified as *single-sided* since bidders are uniformly of type buyer or uniformly of type seller<sup>1</sup>. *Double-sided* auctions admit multiple buyers and sellers at once. Figure 2 depicts a possible taxonomy for a small part of the auction space. The classification is made on the basis of whether the auction is single or double, bids are sealed (SB) or public (outcry), and prices are called in either ascending or descending order. FM contains the auction protocols hanging along the left branch, i.e. the classic auction types. Consequently FM can be classified as a multi-agent test-bed for classic auctions. As to the systematisation of our experiments, the complete parametrisability of FM allows for the generation of different market scenarios. This capability of *scenario generation* appears as a fundamental feature of any multi-agent test-bed if it intends to guarantee the *repeatability* of the experiments to be conducted. Concretely, the customisability of FM allows for the specification, and subsequent activation, of a large variety of market scenarios: from simple toy scenarios to complex real-world scenarios, from carefully constructed scenarios that highlight certain problems to randomly generated scenarios useful for testing trading agents' average performance. Figure 3 displays a snapshot of the graphical display provided by FM to specify the particular features of a tournament scenario.

As to the matter of evaluating a trading agents' performance, FM keeps track of all events taking place during an experimental session, so that a whole

<sup>1</sup> Particularly single auctions have been the main focus of theoretical studies of auction [8].



**Fig. 2.** A classification of classic auction types[15].

auction can be audited step-by-step, and the evolving performance of all the agents involved in a tournament can be traced, calculated, and analysed. On the one hand, FM records all information produced during an experimental session onto a database. On the other hand, FM counts on monitoring capabilities. A monitoring agent receives all the events distributedly coming about in the marketplace thanks to interagents, which collect and convey *carbon copies* of all external and institutional agents' utterances so that the monitoring agent can order them to reconstruct the dynamics of a market session.

Lastly it is worth mentioning a very important feature that seems to be somewhat skipped by test-bed designers: the problem of scalability. When running multi-agent experiments, an experimenter usually faces serious resource limitations that may prevent him from having all agents up and running. We say that FM is *scalability-aware* in the sense that it provides support for distributing an experimenter's agents across several machines in a network. This does not mean that all agent involved in a tournament must belong to the very same user. Tournament designers are free to define *open tournaments* accessible to agents owned by multiple users.

Notice that the resulting environment, FM, thus constitutes a multi-agent testbed where a very rich variety of experimental conditions can be explored systematically and repeatedly, and analysed and reported with lucid detail if needed. Table 2 summarises the features of FM.

### 3 The Tournament Scenario

A trading scenario will involve a collection of explicit conventions that characterise an artificial market. Such conventions define the bidding conditions (timing restrictions, increment/decrement steps, available information, etc.), the way goods are identified and brought into the market, the resources buyers may have available, and the conventions under which buyers and sellers are going to be evaluated. Next we introduce the elements needed to make precise specifications

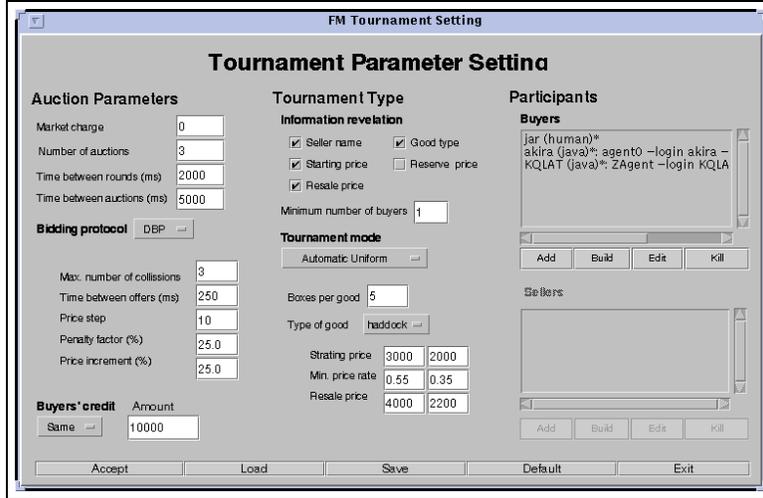


Fig. 3. FM Tournament Definition Panel

of actual tournament scenarios in general and of the actual UPC tournament scenario. In general terms, a tournament scenario specification is intended to comprise all the information necessary for a trading agent to participate in a tournament along with the way they are to be evaluated.

We shall start by studying the characterizing parameters of auction protocols. In particular, although FM supports the classic auction protocols (Vickrey, First-price Sealed-bid, English and Dutch), we shall solely consider a slight variation of the Dutch bidding protocol—henceforth referred to as Downward Bidding Protocol or DBP for shorter—since it was the unique auction protocol employed in the UPC tournament<sup>2</sup>. Each auction protocol can be characterised by a set of parameters that we refer to as *bidding protocol dynamics descriptors*, so that different instantiations of such descriptors lead to different behaviours of their corresponding bidding protocols.

With a chosen good  $g$ , the auctioneer opens a *bidding round* by quoting offers downward from the good's starting price,  $(p_\alpha)$ , as long as these price quotations are above a *reserve price*  $(p_{rsv})$  previously defined by the seller. For each price called by the auctioneer, several situations might arise during the bidding round:

- *Proper sale*. When a single buyer submits a bid that his credit can support, it is turned into a sale.
- *Unsupported bid*. When a buyer submits a bid that his credit cannot guarantee. The buyers' manager fines this bidder and the round is restarted by the auctioneer who calculates the new starting price by increasing by some percentage  $I_{sanction}$  the price within the bid.

<sup>2</sup> A thorough characterization of the rest of bidding protocols is provided in [10].

Test-bed Features
<ul style="list-style-type: none"> <li>■ domain-specific</li> <li>■ realistic</li> <li>■ architecturally neutral</li> <li>■ scenario generation and repeatability capabilities</li> <li>■ monitoring and evaluation facilities</li> <li>■ library of agent templates (C,Java,Lisp)</li> <li>■ dummy agents</li> <li>■ scalability aware</li> <li>■ open (multi-user) and closed (single-user) tournaments</li> <li>■ market scenarios as tournament scenarios</li> </ul>

**Table 1.** Features of the FM test-bed.

- *Collision*. When two or more buyers simultaneously submit the same bid. The auctioneer declares a collision and restarts the round. Again, the new starting price is calculated by increasing by some percentage  $\Pi_{rebid}$  the collision price.
- *Expulsion*. When a buyer is overdrawn and cannot back up a fine, he is sent off the market and the round is restarted as usual
- *Withdrawal*. Each good is assigned a minimum price when passing through the sellers' admitter office. If minimum prices are reached, the round is restarted as usual.

The algorithm in Figure 4 codifies the downward bidding protocol. The description helps us to explicitly identify the parametrisation of the bidding protocol.

Six parameters that control the dynamics of the bidding process are implicit in this protocol definition. We shall enumerate them now, and require that they become instantiated as part of a tournament scenario definition.

**Definition 1 (DBP Dynamics Descriptor).** We define a Downward Bidding Protocol Dynamics Descriptor  $\mathcal{D}_{DBP}$  as a 5-tuple  $\langle \Delta_{price}, \Delta_{offers}, \Sigma_{coll}, \Pi_{sanction}, \Pi_{rebid} \rangle$  such that

- $\Delta_{price} \in \mathcal{N}$  (price step). Decrement of price between two consecutive quotations uttered by the auctioneer.
- $\Delta_{offers} \in \mathcal{N}$  (time between offers). Delay between consecutive price quotations.
- $\Sigma_{coll} \in \mathcal{N}$  (maximum number of successive collisions). This parameter prevents the algorithm from entering an infinite loop as explained above.
- $\Pi_{sanction} \in \mathcal{R}$  (sanction factor). This coefficient is utilized by the buyers' manager to calculate the amount of the fine to be imposed on buyers submitting unsupported bids.

```

Function round ( $B_r^i, g_r^i, p, coll, \mathcal{D}_{DBP}$ ) =
let Function check_credit( $b_i$ ) =
  if  $C_r^i(b_i) \geq p$  then
    update_credit( $b_i, p$ );
    sold( $g_r^i, b_i, p$ );
  else if  $C_r^i(b_i) \geq p * \Pi_{sanction}$  then
    update_credit( $b_i, p * \Pi_{sanction}$ );
    round( $B_r^i, g_r^i, p * (1 + \Pi_{rebid}), 0, \mathcal{D}_{DBP}$ );
  else
    round( $B_r^i - \{b_i\}, g_r^i, p * (1 + \Pi_{rebid}), 0, \mathcal{D}_{DBP}$ );
in
offer( $g_r^i, p$ );
wait( $\Delta_{offers}$ );
let  $B = \{b_i \mid \mathbf{bid}(b_i) = \mathbf{true}, b_i \in B_r^i\}$  in
  case
     $\|B\| = 0$  : if  $p = p_\omega$  then withdraw( $g_r^i$ );
                 else round( $B_r^i, g_r^i, p - \Delta_{price}, 0, \mathcal{D}_{DBP}$ );
     $B = \{b_i\}$  : check_credit ( $b_i$ );
     $\|B\| > 1$  : if  $coll < \Sigma_{coll}$  then
                  round( $B_r^i, g_r^i, p * (1 + \Pi_{rebid}), coll + 1, \mathcal{D}_{DBP}$ );
                  else check_credit(random_select( $B$ ));
  end case
end
end

DBP( $B_r^i, g_r^i$ ) = round( $B_r^i, g_r^i, p_\alpha, 0$ )

```

**Fig. 4.** Downward bidding protocol

- $\Pi_{rebid} \in \mathbb{R}$  (price increment). This value determines how the new offer is calculated by the auctioneer from the current offer when either a collision, or an unsupported bid occur.

Note that the identified parameters impose significant constraints on the trading environment. For instance,  $\Delta_{offers}$  and  $\Delta_{rounds}$  affect the agents' time-boundedness, and consequently the degree of situatedness viable for bidding strategies.

By *auction round* we shall refer to the ontological elements involved in each bidding round.

**Definition 2 (Auction Round).** For a given round  $r$  of auction  $i$  we define the *auction round*  $\mathcal{A}_r^i$  as a 4-tuple  $\langle B_r^i, g_r^i, C_r^i, d_r^i \rangle$  where

- $B_r^i$  is a non-empty, finite set of buyers' identifiers such that  $B_r^i \subseteq \mathcal{B}$ , the set of all participating buyers.
- $g_r^i = \langle \iota, \tau, p_\alpha, p_{rsv}, s_j, p_\omega, p_{rsl}, b_k \rangle$  is a good where  $\iota$  stands for the good identifier,  $\tau$  stands for the type of good,  $p_\alpha \in \mathbb{N}$  stands for the starting price,

$p_{rsv} \in \mathbb{N}$  stands for the reserve price,  $s_j \in \mathcal{S}$ —the set of all participating sellers—is the seller of the good,  $p_\omega \in \mathbb{N}$  stands for the sale price,  $p_{rsl}$  stands for the expected resale price, and  $b_k \in \mathcal{B}_r^i$  is the buyer of the good. Notice that  $g_r^i$  is precisely the good to be auctioned during round  $r$  of auction  $i$ , and that  $p_\omega$  and  $b_k$  might take on empty values when the round is over, denoting that the good has been withdrawn.

- $C_r^i : \mathcal{B}_r^i \rightarrow \mathbb{R}$  assigns to each buyer in  $\mathcal{B}_r^i$  his available credit during round  $r$  of auction  $i$ .
- $d_r^i$  stands for an instance of a bidding protocol dynamics descriptor.

Each auction is devoted to the auctioning of a particular *lot* of goods by opening an auction round for each item within the lot. Typically a tournament session (and a market session too) will be composed of a sequence of auctions.

**Definition 3 (Auction).** We define an auction  $\mathcal{A}^i$  as a sequence of *auction rounds*  $\mathcal{A}^i = [\mathcal{A}_1^i, \dots, \mathcal{A}_{r_i}^i]$

On the basis of these definitions, we are ready to determine what elements and parameters are necessary to wholly characterise a tournament scenario, i.e. all the relevant information needed by an agent to participate in an auction-based tournament, compiled in the definition of *tournament descriptor*. A tournament descriptor is intended to be the sole information on which trading agents count prior to the starting of a tournament session.

**Definition 4 (Tournament Descriptor).** We define a Tournament Descriptor  $\mathcal{T}$  as the 11-tuple  $\mathcal{T} = \langle n, \Delta_{\text{auctions}}, \Delta_{\text{rounds}}, \mathcal{D}, \mathcal{P}_B, \mathcal{P}_S, \mathcal{B}, \mathcal{S}, \mathcal{F}, C, M, \epsilon, E \rangle$  such that:

- $n$  is the tournament length expressed either as the number of auctions to take place during a tournament or the closing time.
- $\Delta_{\text{auctions}}$  is the time between consecutive auctions.
- $\Delta_{\text{rounds}} \in \mathbb{N}$  (time between rounds) stands for the delay between consecutive rounds belonging to the same auction.
- $\mathcal{D}$  is a finite set of bidding protocols' dynamics descriptors.
- $\mathcal{P}_B$  is the conversation protocol that buyer agents must employ in their interaction with their interagents.
- $\mathcal{P}_S$  is the conversation protocol that seller agents must employ in their interaction with their interagents.
- $\mathcal{B} = \{b_1, \dots, b_p\}$  is a finite set of identifiers corresponding to all participating buyers.
- $\mathcal{S} = \{s_1, \dots, s_q\}$  is a finite set of identifiers corresponding to all participating sellers.
- $\mathcal{F} = [\mathcal{F}^1, \dots, \mathcal{F}^n]$  is a sequence of supply functions. A supply function  $\mathcal{F}^i$  outputs the lot of goods to be auctioned during auction  $i$ .
- $C : \mathcal{B} \rightarrow \mathbb{N}$  is the credit initially endowed to each buyer. For some tournaments, all buyers are assigned the same credit, while for others they may either have assigned different credits or alternatively declare themselves the credit they want to have available.

- $M = \langle b, s, r, r' \rangle$  where  $b, s, r, r' \in \{0, 1\}$  is the information revelation mask. It determines whether the identity of buyers ( $b$ ) and sellers ( $s$ ) is revealed to the contenders, and whether the reserve price ( $r$ ) and expected resale price ( $r'$ ) of a good are revealed too.
- $\epsilon$  stands for the fees charged to an agent for participating in a bidding round.
- $E = \langle E_b, E_s \rangle$  is a pair of evaluation functions that permit to calculate respectively the score of buyers and sellers.

From the definition follows that a tournament descriptor contains:

- all the relevant parameters that characterise the dynamics of the auctioning process;
- the procedural information that allows trading agents to participate in the market by means of their interagents;
- the degree of information revelation (transparency) (i.e. the degree of uncertainty concerning the identity of traders and some particular, relevant features of goods); and
- the way the performance of trading agents is evaluated.

It is the task of the tournament designer to conveniently set up the parameters of the tournament descriptor in order to generate the desired type of tournament scenario. For this purpose, FM provides the graphical configuration tool shown in Figure 3 to assist the tournament designer to configure tournament scenarios.

Additionally FM incorporates the so-called *tournament modes* that constrain the type of tournament descriptor that can be defined. The purpose of this standard tournament modes is to allow an experimenter to define tournament scenarios of different degrees of complexity: from toy scenarios where, for instance, the same lot of goods is repeated over and over with complete information to actual-world auction scenarios. Thus, in FM tournament designers can choose among the following standard tournament modes:

**One auction (data set)** This mode permits a tournament designer to specify a fixed set of goods to be repeatedly auctioned a finite number of times. Notice that no sellers are involved in this type of tournament.

**Automatic** The lots of goods to be auctioned are artificially generated by the sellers' admitter based on supply functions of arbitrary complexity specified by the tournament designer in the set  $\mathcal{F}$ . Notice that likewise *one auction (data set)* no sellers are allowed to participate in these tournaments.

This tournament mode allows to artificially generate a large variety of markets. For instance, markets with more demand than supply or the other way around, markets with high quality goods more appropriate for restaurant owners, or markets with large supply of low-quality goods more appropriate for wholesale buyers<sup>3</sup>. In general, this tournament mode allows to create tournaments focusing on particular market scenarios.

---

<sup>3</sup> Note that for all the examples we consider fishmarket-like tournament scenarios.

**Uniform** This mode is a particular case of the preceding tournament mode. Lots of goods are randomly generated by the sellers' admittor based on uniform distributions in  $\mathcal{F}$  defined by the tournament designer. Notice that again no sellers are involved in the resulting tournaments either. Table 2 shows some examples of uniform distributions that can be employed for generating lots of goods.

This tournament mode is intended to generate scenarios wherein the average performance of buyer agents can be tested. Along with *one auction (data set)* it must be considered as a mode to generate game-like scenarios.

**One auction (with sellers)** Once all participating sellers have submitted their goods, the same auction is repeated over and over with the same lot of goods. This tournament mode is particularly useful to test the adaptivity of trading agents to an actual market scenario.

**Fishmarket** The mode closest to the workings of an actual auction house<sup>4</sup>. The tournament designer simply specifies the starting and closing times. During that period of time buyers and sellers can enter, submit goods, bid for goods, and leave at will. *Fishmarket* is the more realistic mode, standing for an actual market scenario.

Depending on the tournament mode chosen by the experimenter, some features of the tournament descriptor will be either enabled or disabled in the *parameter setting panel* at Figure 3. Notice that all parameters identified as part of the tournament descriptor lie down on the *parameter setting panel*.

Finally the UPC tournament can be fully characterised by the tournament descriptor in Table 2. Some comments apply to the resulting scenario:

- All buyer agents were assigned the same credit (17.500 EUR) at the beginning of each auction of the tournament.
- Because the tournament mode was set to *uniform*, the number of fish boxes for each type of fish ( $\tau$ ) were randomly generated for each auction  $\mathcal{A}^i$ , and the starting price ( $p_\alpha$ ), resale price ( $p_{rst}$ ), and reserve price ( $p_{rsv}$ ) of each box were also randomly generated according to the uniform distributions in Table 2.
- As to information revelation, whereas the identity of buyers and the expected resale price of each good were made publicly available, the reserve price was kept as private information.
- The chosen evaluation function ( $E_b$ ) calculates the performance for each buyer at round number  $r$  of auction number  $k$  based on the accumulated benefits ( $B_k(b)$ ) of buyer  $b$  at auction  $k$ . The goal of this evaluation function is to weigh higher the fact of winning the auctions which are closer to the end of the tournament. In this way, bidding strategies that learn to improve an agent's performance as the tournament goes by are more valued.

---

<sup>4</sup> We name it *fishmarket* for historical reasons, though the term must not be misleading since under this mode goods can be auctioned through several auction protocols.

$n$	10																														
$\Delta_{auctions}$	8000 msec																														
$\Delta_{rounds}$	4000 msec																														
$\mathcal{D}$	$\{d_{DBP}\} = \{\langle 50.0EUR, 500msec, 3, 0.25, 0.25 \rangle\}$																														
$P_B$	$P_{DBP}$																														
$P_S$	$\emptyset$																														
$\mathcal{B}$	$\{b_1, b_2, b_3, b_4\}$																														
$\mathcal{S}$	$\emptyset$																														
$\mathcal{F}^i(i = 1 \dots n)$	<table border="1"> <thead> <tr> <th><math>\tau</math></th> <th>#Boxes</th> <th><math>p_\alpha</math></th> <th><math>p_{rsi}</math></th> <th><math>p_{rsv}</math></th> </tr> </thead> <tbody> <tr> <td><i>cod</i></td> <td><math>U[1, 15]</math></td> <td><math>U[1200, 2000]</math></td> <td><math>U[1500, 3000]</math></td> <td><math>U[0.4, 0.5]</math></td> </tr> <tr> <td><i>tuna fish</i></td> <td><math>U[1, 15]</math></td> <td><math>U[800, 1500]</math></td> <td><math>U[1200, 2500]</math></td> <td><math>U[0.3, 0.45]</math></td> </tr> <tr> <td><i>prawns</i></td> <td><math>U[1, 15]</math></td> <td><math>U[4000, 5000]</math></td> <td><math>U[4500, 9000]</math></td> <td><math>U[0.35, 0.45]</math></td> </tr> <tr> <td><i>halibut</i></td> <td><math>U[1, 15]</math></td> <td><math>U[1000, 2000]</math></td> <td><math>U[1500, 3500]</math></td> <td><math>U[0.4, 0.6]</math></td> </tr> <tr> <td><i>haddock</i></td> <td><math>U[1, 15]</math></td> <td><math>U[2000, 3000]</math></td> <td><math>U[2200, 4000]</math></td> <td><math>U[0.35, 0.55]</math></td> </tr> </tbody> </table>	$\tau$	#Boxes	$p_\alpha$	$p_{rsi}$	$p_{rsv}$	<i>cod</i>	$U[1, 15]$	$U[1200, 2000]$	$U[1500, 3000]$	$U[0.4, 0.5]$	<i>tuna fish</i>	$U[1, 15]$	$U[800, 1500]$	$U[1200, 2500]$	$U[0.3, 0.45]$	<i>prawns</i>	$U[1, 15]$	$U[4000, 5000]$	$U[4500, 9000]$	$U[0.35, 0.45]$	<i>halibut</i>	$U[1, 15]$	$U[1000, 2000]$	$U[1500, 3500]$	$U[0.4, 0.6]$	<i>haddock</i>	$U[1, 15]$	$U[2000, 3000]$	$U[2200, 4000]$	$U[0.35, 0.55]$
$\tau$	#Boxes	$p_\alpha$	$p_{rsi}$	$p_{rsv}$																											
<i>cod</i>	$U[1, 15]$	$U[1200, 2000]$	$U[1500, 3000]$	$U[0.4, 0.5]$																											
<i>tuna fish</i>	$U[1, 15]$	$U[800, 1500]$	$U[1200, 2500]$	$U[0.3, 0.45]$																											
<i>prawns</i>	$U[1, 15]$	$U[4000, 5000]$	$U[4500, 9000]$	$U[0.35, 0.45]$																											
<i>halibut</i>	$U[1, 15]$	$U[1000, 2000]$	$U[1500, 3500]$	$U[0.4, 0.6]$																											
<i>haddock</i>	$U[1, 15]$	$U[2000, 3000]$	$U[2200, 4000]$	$U[0.35, 0.55]$																											
$C$	$C(b) = 17.500EUR \forall b \in \mathcal{B}$																														
$M$	$\langle 1, 0, 0, 1 \rangle$																														
$\epsilon$	0																														
$E$	$\langle E_b, E_s \rangle = \langle \sum_{k=1}^n \ln(k+1)B_k(b), \emptyset \rangle$																														

**Table 2.** UPC'2000 Tournament Descriptor

At this point, it is time to make explicit how trading agents and interagents interact in practice and the conversation protocol that they employ. An interagent works as a Java process which uses its standard input and standard output to communicate with trading agents via pipes. In adopting such a simple convention, software agents written in any programming language can interact with the auction house via interagents. Thus, a trading agent firstly spawns the interagent received from the auction house as a child process and subsequently plug to it. Thereafter trading agent and interagent communicate in a rather straightforward way by exchanging string-based illocutions according to the protocol depicted in Figure 5 as an FSM. Tables 3 and 4 list respectively the possible contents of the illocutions labelling the arcs in Figure 5, while Table 5 lists their intended meanings. In Figure 5 numbers followed by / stand for a buyer's agent utterance while messages following / stand for a buyer's agent reception.

#Message	Predicate	Parameters
1	<i>admission</i>	<i>buyerlogin password</i>
2	<i>bid</i>	
3	<i>exit</i>	

**Table 3.** Messages that (software) buyer agents can utter during a tournament

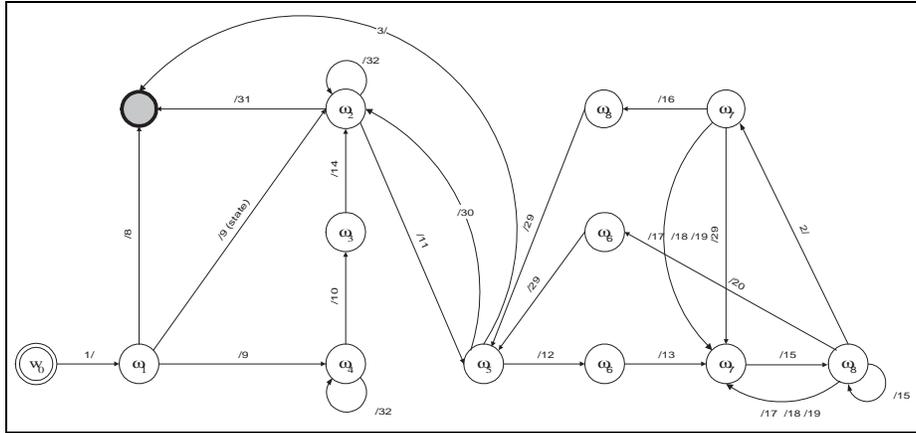


Fig. 5. Protocol used by buyer agents to interact with interagents.

## 4 The agents

As said before, a total of sixteen agents participated in the tournament. All the students had time to study the environment, and to experiment with toy agents provided by the platform and agents from previous tournaments. The code of the agents from previous tournaments was not available, so, they only could observe their behavior against other agents.

From their study of the platform and private tournaments, the different groups observed the information that could be helpful in the problem. They reported the following possibly relevant information:

**From the market:** Number of rounds, number of boxes, initial market money, remaining market money, last benefit.

**From the goods:** Starting price, resale price, reserve price if the good is retired from the market, name of the good, ratio between buying price and resale price, name of the buyer,

**From the competitors:** Mean benefit, benefit of the best agent, remaining credit, behavior of the agent.

**From the agent state:** Own benefit, remaining credit, number of boxes bought.

Due to the time restrictions, not all this information could be used during the tournament. Each group reduced the information available to just what they thought could be relevant on deciding the bidding price for a good.

There was a great consensus between the agents about what information had to be considered. The first of it was the length of the auction. Almost all the agents considered a classification of the auctions from its length. The number of classes considered ranged from two to four, but the most used value was three. The classification characterized short sized auctions (approximately 20 boxes),

#Message	Predicate	Parameters
8	<i>deny</i>	<i>deny_code</i>
9	<i>accept</i>	<i>admission</i>
10	<i>open_auction</i>	<i>auction_number</i>
11	<i>open_round</i>	<i>round_number</i>
12	<i>good</i>	<i>good_id good_type</i> <i>starting_price resale_price auction_protocol</i>
13	<i>buyers</i>	{ <i>buyerlogin</i> }*
14	<i>goods</i>	{ <i>good_id good_type</i> <i>starting_price resale_price</i> }* <i>protocol</i>
15	<i>offer</i>	<i>good_id price</i>
16	<i>sold</i>	<i>good_id buyerlogin price</i>
17	<i>sanction</i>	<i>buyerlogin fine</i>
18	<i>expulsion</i>	<i>buyerlogin</i>
19	<i>collision</i>	<i>price</i>
20	<i>withdrawn</i>	<i>good_id price</i>
29	<i>end_round</i>	<i>round_number</i>
30	<i>end_auction</i>	<i>auction_number</i>
31	<i>closed_market</i>	<i>closing_code</i>
32	<i>tournament_descriptor</i>	<i>auction n</i> $\Delta_{auctions}$ $\Delta_{rounds}$ $\epsilon$ <i>bidding_protocols</i> <i>dbp</i> $\Delta_{price}$ $\Delta_{offers}$ $\Sigma_{coll}$ $\Pi_{sanction}$ $\Pi_{rebid}$ <i>UBP</i> $\Delta_{price}$ $\Delta_{offers}$ $\Pi_{sanction}$ $\Pi_{start}$ <i>FPSB</i> $B_t$ $\Pi_{sanction}$ <i>vickrey</i> $B_t$ $\Pi_{sanction}$ ( <i>buyers</i> { <i>buyerlogin</i> * # <i>buyers</i> } <i>credit</i> { <i>credit</i> *  <i>credit</i>   <i>unkown</i> } <i>sellers</i> { <i>sellerlogin</i> *  <i>Market</i> } <i>mode</i> { <i>automatic</i> , <i>uniform</i> <i>one_auction_data</i> , <i>one_auction_sellers</i> , <i>fishmarket</i> }

**Table 4.** Messages that (software) buyer agents can receive during a tournament.

medium sized auctions (approximately 45 boxes) and long sized auctions (up to 75 boxes).

It is a strategy observed in this tournament and previous, to classify the auction by length. Each kind of auction lead to a different strategy:

- In short auctions an aggressive strategy is used, trying to buy almost at starting price. If the credit is enough, this is an admissible strategy because the total money is more than the cost of all the goods. There are no time to consider the characteristics of the goods, because probably not all the money could be spent. The better good is that with a better ratio between starting price and resale price.
- In medium auctions a more deliberative strategy is necessary. The total money of the agents is almost enough to buy all the goods, so the agents had to be selective and compete for the best goods. The last goods of the auction can be interesting because their price can be lower.
- In long auctions the planning is very important. The cost of the goods are more than the total market money. The agent has to decide what goods are interesting because its price and its position in the auction. It could be

<b>Predicate</b>	<b>Semantics</b>
<i>exit</i>	Leave the marketplace.
<i>admission</i>	Request for admission.
<i>bid</i>	Bid at the current price.
<i>deny</i>	Refuse requested action.
<i>accept</i>	Accept access to scene.
<i>open_auction</i>	The auctioneer opens a new auction.
<i>open_round</i>	The auctioneer opens a new bidding round.
<i>good</i>	Features of the good in auction.
<i>buyers</i>	List of participating buyers.
<i>goods</i>	Lot of goods to be auctioned.
<i>offer</i>	Current offer called by the auctioneer.
<i>sold</i>	The good in auction has been sold.
<i>sanction</i>	Sanction imposed on a given buyer.
<i>expulsion</i>	Buyer expelled out of the market.
<i>collision</i>	Multiple bids at the same price (DBP).
<i>withdrawn</i>	Reserve price reached. Good withdrawn.
<i>end_round</i>	Bidding round over.
<i>end_auction</i>	Auction over.
<i>closed_market</i>	End of market session.

**Table 5.** Semantics of the messages exchanged between a buyer and the auction house.

an interesting strategy to wait until all the competitors had spent all their money in order to obtain better prices. In this kind of auctions an accurate estimation of the reserve price is very important.

The other information from the auction that had almost all the consensus was the quotient between the total resale value of the goods of the auction and the total market money. This value can be used as an estimation of the mean expected benefit. To outperform or underperform this value is an indicator of the performance of the agent. This measure is correlated to the behavior of the auction and allow to not to observe individually to each competitor.

This expected benefit can be updated during the auction by the bought of the agents. This allow to change the behavior of the agent because the raise or fall of the expected benefit.

Almost all the agents used this ratio as base value in order to decide its bid. If the initial benefit of the good is lower than the mean benefit, then the good is not interesting and, either the bid is not done, or the agent wait until the price drops to a more interesting one.

The agents used other complementary values to correct the bidding price obtained from the calculation of the mean benefit. For example, the benefit of the best agent, the remaining credit of the competitors and heuristical factors obtained by experimentation during the private auctions that were held before the official tournament.

Due to that in long auctions to wait until almost the end of the auction is a profitable policy, the estimation of the reserve price becomes important. Every agent has a way to estimate the reserve price. Some agents do the estimation dynamically, trying to learn this price from the auction, others used a constant percentage from the initial price. Obviously, the agents that try to estimate the reserve price dynamically obtained better results.

The strategies to determine the reserve price were diverse, but based on statistical estimation. Because the real reserve price is unknown, the difference between starting price and the lower price payed for the goods is a good initial estimation. This estimation can be corrected using the price observed when a good goes out of market, circumstance that can be observed in long auctions. Some agents tried to estimate the reserve price for each kind of good. Due to the relative shortness of observations those estimations were less accurate than those from the agents that tried to estimate a global reserve price.

Planning and learning were rare among the agents of the tournament. Some agents tried to plan beforehand the goods more attractive, estimating the optimal bid and distributing the available money among them. All allowed a dynamical redistribution of the bids if the chosen goods were bought by another competitor.

Just two agents tried to use learning between auctions to improve their performance. The first, used the comparison between the benefit obtained and the benefit of its competitors in order to reduce or increase the bidding price in the next auction. the second used a more sophisticated learning mechanism based on reinforcement learning. This strategy used Q-learning in order to decide the optimal benefit for each good from the own actions and the actions of its competitors.

## 5 The Results

The competition was organized in three eliminatorary rounds. The first round divided the agents randomly in four groups. Each group competed in a tournament as specified in section 3. From each group only the two best were chosen.

In this round the agents with a weak strategy obtained a significant less performance than the more elaborated agents. This year, in contrast with previous tournaments, the level of cooperation between the groups were very low. Only a small number of agents participated on private tournaments. Most of the agents that were eliminated in this round were the non cooperative ones. This gives an idea of how important is cooperation and interaction during the development of agents.

The second round paired the winning agents of the first and second group and the agents of the third and fourth group. In this round also the two best of each group passed to the final round.

In this round the competition was hardest. In the first group the difference among the three firsts agents were very short. In the second group there was a clear difference between the first two agents and the other two competitors.

The strategies of the winners of this round were not significantly different from the rest, but included some the agents that used some kind of learning and adaptation.

Surprisingly, the winner of the final round was the agent with the simplest strategy of the four competing agents. Those are the four agents of the final round and their strategies:

**HumbleJES:** This is the winner agent. The basis of this agent is the ratio between the resale price of the remaining goods and the total credit of the agents. This ratio is weighted using a value that is an estimation of the desired benefit. This expected benefit is a constant that is not changed during the competition.

This value is used to estimate the bid for the actual good. This price is corrected with the information about the money available for the other agents. If this value is greater than the price that can be paid by their competitors, it is adjusted to a little more than this quantity. If the competitors can not buy the good, then the price is adjusted to the estimated reserve price.

**garsa:** This is the second agent. The basis of this agent is also the expected benefit obtained as a ratio of the resale price of the goods and the money available, but in this case, this ratio is calculated at the beginning of each auction. This value is modified using the behavior of the other agents. If the rest of agents bid to a price higher than the calculated, the value is not touched. If the other agents bid to a lower price, the benefit is adjusted to obtain a bid slightly higher than the bid of the competitors, increasing the own benefit.

This agent detects when the competitors have not enough money to buy more goods. When this happens, the bid is adjusted to a statistically estimated reserve price.

**The Pretender:** This is the third agent. This is the more sophisticated agent, it uses reinforcement learning (Q-learning) in order to learn what is the better price for a good. It uses a probability matrix indexed by resale price and expected benefit. This matrix stores the probability distribution of the optimal benefit for a given resale price. The matrix was initialized with a priori probability distributions obtained from the private tournaments.

Three different reinforcements are used during the auction. A positive reinforcement if the current bid is successful and is considered a good bid, a negative reinforcement if it is considered that the actual bid benefit has to change and a negative reinforcement if the actual did benefit of the agent is high. A set of rules allow to decide what kind of reinforcement is necessary. These rules evaluate different information, as the number of remaining rounds, the performance of the competitors or the number of competitors with enough money. The learning is done in each auctions, so the information of the previous auctions is not maintained.

This probability matrix adapts to the behavior of the market, and predicts the most probable benefit that the competitors desire to obtain. This information allow to advance the bid and to buy before than the competitors.

**TokOchons:** This is the fourth agent. The strategy of this agent uses two information. The first is a variation of the ratio between the resale price of the remaining goods and the remaining market money. This information allow to guess the expected benefit. The second source of information is a function that give a measure of how interesting is a good. This function combines the relative and absolute benefit obtained for a given bid.

This bid is corrected using different parameters. The more interesting is a value that measures the proportion of the market money that the agent owns. If the proportion is great, this means that the agent almost has not competitors, so, the expected benefit can be raised.

This agent stores the past auctions in order to analyze them. If the current auction has a similar number of rounds that a past auction, its information is recalled. If in this past auction some money was not spent, the bids are raised in order to spent all the money, increasing the benefit by buying more goods. If the winner of this past auction obtained a benefit higher than ours, the expected benefit for the current auction is raised in order to pay less for the goods.

In the figure 6 can be seen the evolution of the objective function (see section 3) that measured the performance of the agents. It can be seen that the agent **HumbleJES** performs significantly better that the others from the start of the competition, the other tree agents are in a tie until auction number seven, in this point the agent **garsa** starts outperforming the other two agents. It seems that the learning procedures of this two agents are not a real advantage against the other two strategies.

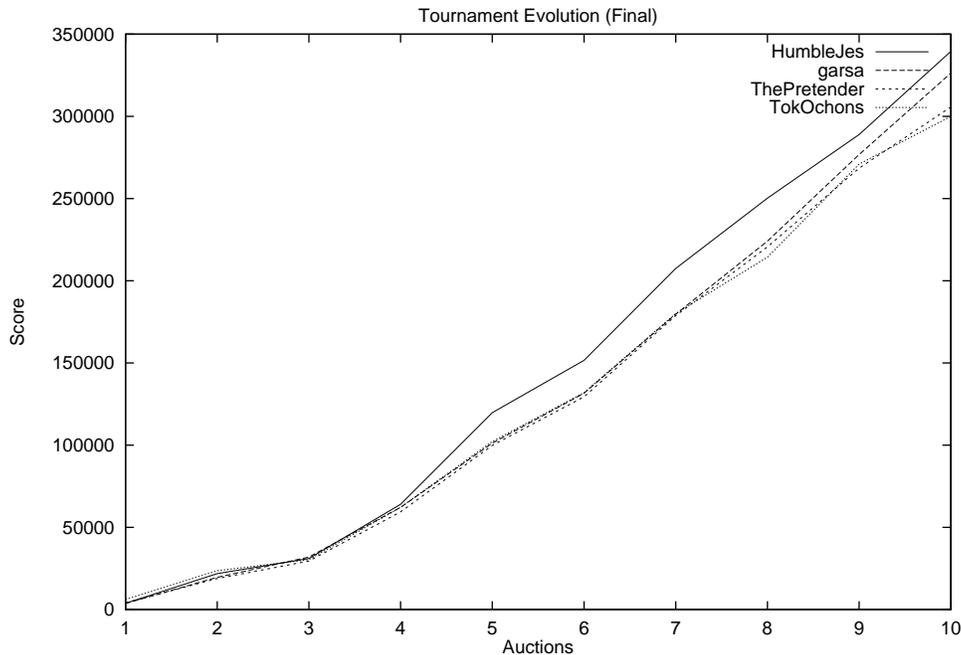
## 6 Conclusions

Some conclusions can be drawn from this tournament. First of all, that more sophisticated strategies has not evident advantage against simple ones. The best agents use an strategy based on market information without neither trying to model the other agents not use learning from experience to improve their performance. This does not means that this characteristics are not desirable. An adequate learning policy could overperform simple strategies in a more dynamic environment.

The other conclusion is the significance of competition in the developement of this kind of agents. At has been said, only the agents from the people that decided to share their knowledge and competed in private tournaments were successful. The need to test a strategy are crucial for its developement. It is difficult to have success without interaction.

## 7 Acknowledgments

We want to thank the IIIA (CSIC) to allow us to use the *Fishmarket* platform for the tournaments. Also we want to acknowledge all the competitors of this spring tournament.



**Fig. 6.** Evolution of the last round of the tournament

## References

1. M. Andrews and R. Prager. *Genetic Programming for the Acquisition of Double Auction Market Strategies*, pages 355–368. The MIT Press, 1994.
2. A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, 1996.
3. U. Cortés and J. A. Rodríguez-Aguilar. Trading agents in auction-based tournaments. *INFORMATIQUE*, 1(1):39–50, 2000.
4. Facultat d'Informàtica de Barcelona. <http://www.fib.upc.es>.
5. The FishMarket Project. <http://www.iiia.csic.es/Projects/fishmarket>.
6. N. R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1995.
7. F. J. Martín, E. Plaza, and J. A. Rodríguez-Aguilar. An infrastructure for agent-based systems: An interagent approach. *International Journal of Intelligent Systems*, 15(3), 2000.
8. R. P. McAfee and J. McMillan. Auctions and bidding. *J. Ec Lit.*, XXV:699–738, jun 1987.
9. P. Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor*. Number 8 in IIIA Monograph Series. Institut d'Investigació en Intel·ligència Artificial (IIIA), 1997. PhD Thesis.
10. J. A. Rodríguez-Aguilar. *On the Design and Construction of Agent-Mediated Institutions*. PhD thesis, Institut d'Investigació en Intel·ligència Artificial, 2000. (forthcoming).

11. J. A. Rodríguez-Aguilar, F. J. Martín, F. J. Giménez, and D. Gutiérrez. Fm0.9beta users guide. Technical report, Institut d'Investigació en Intel·ligència Artificial. Technical Report, IIIA-RR98-32, 1998.
12. J. A. Rodríguez-Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Competitive scenarios for heterogeneous trading agents. In *Proceedings of the Second International Conference on Autonomous Agents (AGENTS'98)*, pages 293–300, 1998.
13. J. A. Rodríguez-Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11(1):5–19, 1998.
14. J. A. Rodríguez-Aguilar, P. Noriega, C. Sierra, and J. Padget. Fm96.5 a java-based electronic auction house. In *Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology(PAAM'97)*, pages 207–224, 1997.
15. P. R. Wurman, , M. P. Wellman, and W. E. Walsh. The Michigan Internet Auction-Bot: A Configurable Auction Server for Human and Software Agents. In *Second International Conference on Autonomous Agents (AGENTS'98)*, pages 301–308, May 1998.