

# USING SHADOW PRICES FOR RESOURCE ALLOCATION IN A COMBINATORIAL GRID WITH PROXY-BIDDING AGENTS

Michael Schwind

*Institute of Information Systems, J. W. Goethe University  
Mertonstrasse 17, D-60054 Frankfurt, Germany*

Oleg Gujo

*Institute of Information Systems, J. W. Goethe University  
Mertonstrasse 17, D-60054 Frankfurt, Germany*

**Keywords:** Resource allocation, combinatorial auction, grid computing, agent-oriented programming.

**Abstract:** Our paper presents an *agent-based simulation environment* for task scheduling in a distributed computer systems (grid). The scheduler enables the *simultaneous allocation of resources* like CPU time, communication bandwidth, volatile, and non-volatile memory while employing a combinatorial resource allocation mechanism. The resources are allocated by an *iterative combinatorial auction with proxy-bidding agents* that try to acquire their desired resource allocation profiles with respect to limited monetary budget endowments. In order to achieve an efficient bidding process, the auctioneer provides *information on resource prices* to the bidding agents. The calculation of explicit resource prices in a combinatorial auction is computationally demanding, especially if the bid bundles exhibit complementarities or substitutionalities. We therefore propose a new approximate pricing mechanism using shadow prices from a linear programming formulation for this purpose. The efficiency of the shadow price-based allocation mechanism is tested in the context of a *closed loop* grid system in which the agents can use monetary units rewarded for the resources they provide to the system for the acquisition of complementary capacity. Two types of proxy-bidding agents are compared in terms of efficiency (received units of resources, time until bid acceptance) within this scenario: An *aggressive bidding agent* with strongly rising bids and a *smooth bidding agent* with slowly increasing bids.

## 1 INTRODUCTION

We present an agent-based simulation environment for resource allocation in a distributed computer system that employs shadow prices as an information entity to optimize the allocation process. Our environment allows the simulation of a mechanism for the simultaneous allocation of resources like CPU time, communication bandwidth, volatile and non-volatile memory in the distributed computer system. In contrast to traditional grid allocation approaches, our allocation process considers production complementarities and substitutionalities for these resources making the resulting resource usage much more efficient. The central scheduling instance of our system is comparable to an auctioneer that performs an iterative combinatorial auction in which proxy-agents try to acquire the resources needed in computational tasks for the provision of *information services and information production (ISIP)*<sup>1</sup> by submitting package bids

<sup>1</sup>Examples for ISIP tasks are e.g. the provision of Web services, the customized retrieval and replication of cus-

for the resource combinations. The proxy-agents' *willingness-to-pay (W2P)* for these bundles is constrained by limited budgets of a virtual currency they are endowed with. The allocation system simulates a *closed-loop* grid economy in which the agents gain monetary units for resources they provide to other grid system participants via auctioneer. The earned virtual currency can be used for the acquisition of complementary resource capacity by submitting combinatorial bids. The simulation environment allows the utilization and benchmarking of different proxy-bidding strategies in various system load situations. Two main bidding strategies are compared in this paper:

- An *aggressive bidding agent* that submits combinatorial bids while trying to achieve quick bid acceptance by using a fast inclining bid pricing strategy.
- A *smooth bidding agent* that submits multiple bid bundles to the auctioneer waiting for bid acceptance of some of the alternative bids while increas-

tomized stock chart data or the broadcast of a public event to viewers via TCP/IP protocol.

ing the bid prices only slowly.

The bidding strategies are compared with each other regarding their allocation-related efficiency, which is measured in terms of received resource units per virtual currency unit spent by the agents and in terms of time to bid acceptance in the auction process.

## 2 COMBINATORIAL AUCTIONS FOR RESOURCE ALLOCATION IN DISTRIBUTED COMPUTER SYSTEMS

Various auction protocols have been proposed for resource allocation in distributed computer systems in the last decades. The transfer of economic principles to resource attribution in grid systems, like the *price controlled resource allocation (PCRA)*<sup>2</sup> used in our scenario, allows *flexible* implementation of allocation mechanisms in decentralized systems (Buyya et al., 2001).

Combinatorial auctions are a suitable tool to allocate interdependent resources because they can take their substitutionalities and complementarities into account. The production process for information services in distributed systems comprises an allocation problem with strong complementarities. For example, if an information service such as the provisioning of a video conference service via the web or the off-line calculation of distributed database jobs has to be processed on different computers and acquires CPU time without obtaining communication network capacity between the computers at the same time, the acquired CPU time is worthless. The application of combinatorial auctions for resource allocation in distributed computer systems is still in its infancy despite its excellent applicability to grid computing. A combinatorial auction-based mechanism for resource allocation in a *SensorNet* testbed was presented by (Chun et al., 2004) in a recent approach. The devices in this mechanism feature different capabilities in various combinations. The periodically performed combinatorial sealed-bid auction is implemented within the *microeconomic resource allocation system (MIRAGE)*. The system uses a very simple combinatorial allocation mechanism to achieve sufficient real time performance. MIRAGE users have accounts based on a virtual currency enabling a bartering process for the SensorNet resources. A consequent continuation of this work is the grid computing environment *Bellagio* by (AuYoung et al., 2004). Each bidder has a budget of a virtual currency available for task pay-

<sup>2</sup>The price is used as a control variable for the scheduling mechanism, requests with higher W2P are prioritized.

ment purposes. The assignment of the required resources to the particular tasks is done by a combinatorial second-price auction, which comes close to a strategy proof mechanism. In several experiments, the system is tested for scalability, efficiency, and fairness. Due to the simple greedy algorithm used in this system, the efficiency of the resulting resource allocation is not satisfying.

## 3 AN AGENT-BASED SIMULATION ENVIRONMENT FOR COMBINATORIAL RESOURCE ALLOCATION

Our combinatorial grid scheduling environment, realized in JADE 3.3, goes beyond the recent approaches in several points:

- The system allows the usage of several winner determination algorithms like simulated annealing, genetic programming, and integer programming methods according to the users' requirements in terms of allocation quality and computation time<sup>3</sup>.
- The simulator provides tools to investigate various bidding behaviors of the proxy agents in the resources acquisition process. We will concentrate on this aspect in this paper.
- The framework can simulate changing resource capacities to test the combinatorial grid scheduler's response with respect to allocation efficiency and system stability.

### 3.1 Scenario for a PCRA in a Combinatorial Grid Scheduler

This section gives a brief overview on the *resource allocation scenario* for ISIP provision used in our work. The scenario includes four resource types:

- *Central processing units (CPU)* that are mainly responsible for the data processing in the ISIP tasks.
- *Volatile memory capacity (MEM)* which is necessary to store short-term processing data for the central processing units.
- *Non-volatile storage capacity (DSK)* which is necessary to keep mass data on databases and to provide program codes for the execution of the ISIP processes.

<sup>3</sup>For algorithm description see (Schwind et al., 2003).

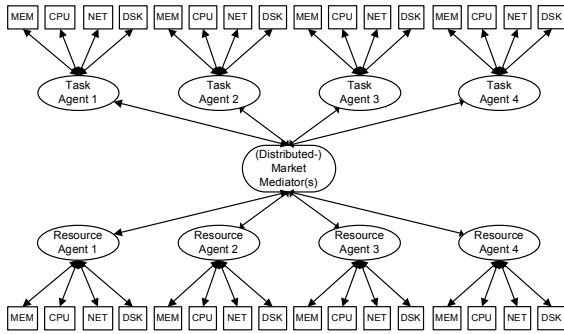


Figure 1: Scenario for the allocation of ISIP resources.

- *Network bandwidth (NET)* that is required for data interchange among the grid computer units<sup>4</sup>.

The general PCRA scenario used within the combinatorial grid simulator is constructed as follows:

- *Task agents* (bidders) are engaged in acquiring the resources needed to process the ISIP task in the distributed computer system on behalf of real world clients. They do this by bidding for the required resource combination via the mediating agent.
- A *mediating agent* (auctioneer) receives the resource bids and calculates an allocation profile for the available resources managed by the *resource agents* according to the allocation mechanism. After a successful auction process, bidders are informed about the acceptance of their bids.
- *Resource agents* collect information about available resources on their particular host IT systems through a network of distributed computers and provide this information to the market mediator. The resource agents offer the available capacities to the task agents via the mediating agent. If a bid is accepted via the auctioneer, the acquired resources are reserved for the corresponding winning agent in advance.

Figure 1 depicts the ISIP allocation scenario. Resource agents administrate available MEM, CPU, NET, and DSK capacities on their particular host computers systems on the supply side. On the demand side, task agents collect the required resource combinations including MEM, CPU, NET, and DSK capacity needed to accomplish their production tasks. Between resource and task agents, there is a market mediator that allocates the resources employing a combinatorial auction. For the formal representation of the bids, a two-dimensional *bid-matrix (BM)*

<sup>4</sup>Network connections themselves exhibit complementarities due to their peering character. For simplicity reasons, we assume that NET capacity can be managed as one single system resource.

is used. One dimension of the *BM* describes the time  $t \in \{1, \dots, T\}$  at which the resource is required within the request period  $T^5$ . The other dimension  $r \in \{1, \dots, R\}$  denotes the resource types MEM, CPU, NET, DSK. The request for a quantity of an individual resource  $r$  at time  $t$  is then denoted by a matrix element  $q(r, t)$ . A price  $p$  is assigned to each *BM* expressing the agent's W2P for the resource bundle.

Table 1: Example for a structured bid matrix *BM* submitted by a task agent.

<i>BM<sub>struct.</sub></i>	Time Slot $t$							
Resources	1	2	3	4	5	6	7	8
$r_1$	2	2			3		3	3
$r_2$		1	1	1				
$r_3$		2	2		1	1	1	1
$r_4$	3	3				2	2	

In addition to the *BM*, two other matrix types play an important role within our grid simulation framework:

- The *allocation matrix (AM)* describes the awarded allocation  $q(r, t)$  for resources  $r$  and time slots  $t$  within the following ISIP provision period  $T$ .
- The *constraint matrix (CM)* expresses the maximum quantity  $q_{max}(r, t)$  of resource  $r$  the auctioneer can assign to the task agents at time  $t$ . The maximum possible resource load of the *CM* represents the aggregated resource availability for the following time slots.

$q_{bmax}$  denotes the maximum resource load that can be requested by a bidder for a single *BM* element  $q_{i,j}(r, t)$ . In our matrix instances each entry in a *BM* is occupied with probability  $p_{tso}$ .

### 3.2 The Combinatorial Scheduling Auction

In the following paragraph, the course of the combinatorial grid auction is described in the light of an UML sequence diagram based on the FIPA definition of the English auction (steps are denoted in  $\circ$ )<sup>6</sup>:

1. The auctioneer requests the resource agents to evaluate the available resource capacities and informs the bidders about the bidding terms. Then he announces the start of the auction. Additionally, the auctioneer awards an initial budget to the task agents.

<sup>5</sup>Time period  $T$  is divided into equidistant intervals (time slots  $t$ ) within the simulation model. The time period  $T$  is identical to a following production time span  $t_{pro}$  at which the ISIP provision takes place.

<sup>6</sup>See foundation for intelligent physical agents [www.fipa.org/specifications/fipa00031D](http://www.fipa.org/specifications/fipa00031D).

2. Following the auctioneer's call-for-proposal, the task agents create their bids according to the desired resource combination. Bidders compute the associated bid price dependent on their actual pricing policy, their budget level, and the latest resource prices.
3. The auctioneer receives the bids and calculates the return-maximizing combinatorial allocation. He informs the task agents about any bid acceptance/rejection and requests the resource agents to reserve the awarded resources.
4. Resource agents inform the auctioneer about the status of the task execution.
5. The auctioneer propagates any task status information to the task agents, and the agents' accounts are debited with the bid prices of the awarded bids. Then a call-for-proposal for the next round is issued.
6. Task agents can renew their bids in the next round in case of non-acceptance or non-execution. The agents' bid pricing follows rules defined in the subsequent paragraph.
7. The process is repeated until the auctioneer announces the end of the auction.

In the following, the three crucial elements of the combinatorial grid scheduling system are described in more detail: the budget management mechanism, the combinatorial auctioneer, and the task agents' bidding behavior.

### 3.2.1 The System's Budget Management Mechanism

Each of the task agents  $a_i$  holds a monetary budget  $BG_i$  that is initialized with a fixed amount  $BG_i^{ini}$  of monetary units (MUs) at the start of the system. At the beginning of each round  $k$ , the task agents' budgets are refreshed (see figure 2,  $\circ 1$ ) with an amount of MUs enabling them to acquire the resource bundles  $b_{i,j}$  required for their ISIP provision task. The agents' budget refill can be done in two ways in our grid economy:

- A fixed amount  $BG_i^{inc}$  that is defined by the system user is added to the agents' budgets independently of the production capacity they provide to the grid system. This case, where task agents only behave as consumers, is denoted as an *open grid economy*. The *resource agents* which own the resources act independently from the task agents providing only resource availability and resource usage information to the auctioneer. The resource agents are compensated by the auctioneer for the capacity provided proportionately to the auctioneer's income  $Inc^{acc}$ .

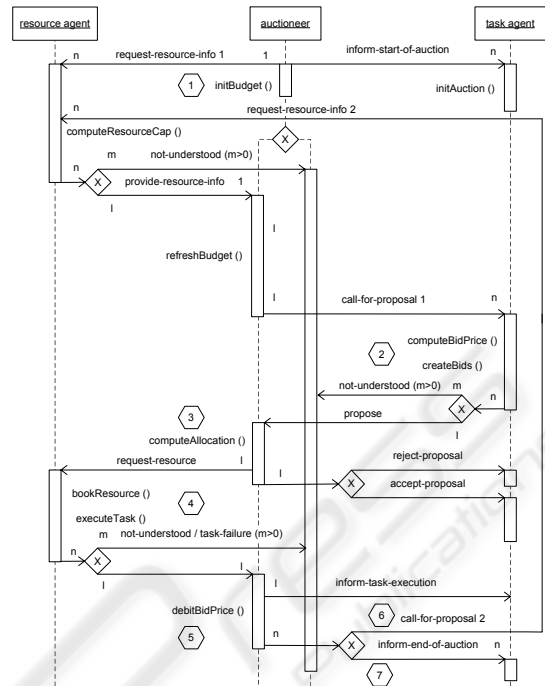


Figure 2: FIPA AUML diagram for the iterative combinatorial scheduling auction.

- Task and resource agents act as a unit of consumer and producer both owning the resources of their peer system. This means a task and a resource agent reside simultaneously on each peer computer in the grid. The resource agent does the reporting of resource usage and provisioning for the task agent owning the peer computer resources (see figure 2,  $\circ 1,4$ )<sup>7</sup>. The agents on the peer computer are compensated for the resources provided to the system. The compensation process is organized by the auctioneer. Starting with the initial budgets  $BG_i^{ini}$  the amount of MUs circulating in the system is kept constant for the *closed grid economy*.

The accounting of the agents' budgets in the grid system is done by the combinatorial auctioneer (see figure 2,  $\circ 1,5$ ).

### 3.2.2 The Combinatorial Auctioneer

The combinatorial auctioneer controls the iterative allocation process of the grid system. For this purpose, the auctioneer awaits the XOR-bundled bids  $b_{i,j}$  that have been submitted by the task agents  $a_i$  for the current round. The bids that are submitted in the form of *BMs* are shown in Table 1. They represent the task agents requests for resource capacity  $q_{i,j}(r, t)$  at

<sup>7</sup>In Figure 1 this implies that resource agent 1 and task agent 1 reside on the same peer computer.

a particular point of time  $t$ . After having received all alternative  $BM$ s submitted by the task agents, the auctioneer has to solve the *combinatorial auction problem (CAP)* which is  $\mathcal{NP}$ -hard (Parkes and Ungar, 2000; Fujishima et al., 1999). The CAP is often denoted as the *winner determination problem (WDP)*, according to the traditional auctioneers task of identifying the winner. The formal description of the CAP could be considered as a special variant of the *weighted set packing problem (WSPP)* (Vries and Vohra, 2001) and is formulated as:

$$\max \sum_{i=1}^I \sum_{j=1}^J p_{i,j} x_{i,j}$$

subject to

$$\sum_{i=1}^I \sum_{j=1}^J q_{i,j}(r, t) x_{i,j} \leq q_{max}(r, t), \quad (1)$$

where  $r \in \{1, \dots, R\}$ ,  $t \in \{1, \dots, T\}$  and

$$\sum_{j=1}^J x_{i,j} \leq 1, \quad \text{where } i \in \{1, \dots, I\}.$$

Resources:	$r$	$\in \mathbb{N}$
Time slots:	$t$	$\in \mathbb{N}$
Resource requests:	$q_{i,j}(r, t)$	$\in \mathbb{N}$
Price for bid $b_{i,j}$ :	$p_{i,j}$	$\in \mathbb{R}^+$
Acceptance variable:	$x_{i,j}$	$\in \{0,1\}$
Bid $j$ of agent $i$ :	$b_{i,j}$	$\in B$
Income for all accepted bids:	$Inc^{acc}$	$\in \mathbb{R}^+$

The goal is to maximize the auctioneers income.  $q_{max}(r, t)$  is the maximum capacity of resources at time  $t$  available to the auctioneer and  $B$  is the set of all bids  $b_{i,j}$ . Furthermore, we refer to the set of accepted bids as  $I^+$  (with  $I^+ \subseteq B$ ).

### 3.2.3 Shadow Price Calculation

For an efficient bidding process it is necessary to provide preferably exact information about the actual auctioneer's valuation of the resources to the proxy-agents. However, it is not possible to calculate *unambiguous prices* (anonymous prices) for the individual resources in a combinatorial auction due to the non-linearities in the bidders' valuations (Xia et al., 2004). In many cases, explicit resource prices can only be calculated for each individual bid. (Kwasnica et al., 2005) describes a pricing scheme for all individual goods in a combinatorial auction by approximating the prices in a divisible case based on a *linear programming (LP)* approach first proposed by (Rassenti et al., 1982). Like in a similar approach by (Bjørndal and Jørnsten, 2001), they use the *dual solution* of the relaxed WDP which is used to calculate the *shadow prices (SP)*. In our simulation model we adopt the

dual LP approach of (Kwasnica et al., 2005) including accepted bids as well as rejected bids<sup>8</sup>:

$$\min z = \sum_r \sum_t q_{max}(r, t) \cdot sp_{r,t} \quad (2)$$

subject to

$$\sum_r \sum_t q_{i,j}(r, t) \cdot sp_{r,t} = p_{i,j} \quad \forall b_{i,j} \in I^+ \quad (3)$$

$$\sum_r \sum_t q_{i,j}(r, t) \cdot sp_{r,t} + \delta_{i,j} \geq p_{i,j} \quad \forall b_{i,j} \in I^- \quad (4)$$

Accepted bid set:	$I^+$	$\subseteq$	$B$
Rejected bid set:	$I^-$	$\subseteq$	$B$
Reduced cost:	$\delta_{i,j}$	$\in$	$\mathbb{R}_0^+$
Shadow price acceptance:	$sp_{r,t}$	$\in$	$\mathbb{R}_0^+$

We use the primal solution of the LP problem delivered from open source LP solver *LPSOLVE 5.5*<sup>9</sup> to the appointment of sets  $I^+$  and  $I^-$ . As described above, our matrix has  $R \times T$  elements, i.e. for every resource  $r \in R$  there are  $T$  time slots. We group the results as follows:

$$SP_r(k) = \sum_{t=1}^T sp_{r,t} \quad \forall r \in R \quad (5)$$

$$\text{Shadow price: } SP_r(k) \in \mathbb{R}_0^+$$

In general bid prices are not assumed to be linear in our framework. This means that shadow prices  $SP$  cannot be calculated by the auctioneer for each round, i.e. there is no solution to the LP problem, or reduced costs equal to zero for a number of bids (Bjørndal and Jørnsten, 2001). In such cases we rely on an approximate shadow price calculation based on pricing history ( $H_{sp}$ )<sup>10</sup>:

$$SP_r(k) = \frac{\sum h_{sp}}{|H_{sp}|} \quad \forall h_{sp} \in H_{sp} \quad (6)$$

$$h_{sp} = \begin{cases} SP_r(k) & \text{if } SP_r(k) \neq 0 \wedge k \neq 1 \\ 0 & \text{if } k = 1 \end{cases} \quad (7)$$

Now we can investigate the market value of a resource unit while we use the shadow prices and form the sums of each resource  $r \in R$  and each time slot  $t \in T$  for all accepted bids:

<sup>8</sup>The result of the following formula is denoted as reduced shadow prices. By omitting the rejected bids in the calculation of dual prices, the result would be higher (Bjørndal and Jørnsten, 2001).

<sup>9</sup><http://www.geocities.com/lpsolve/>

<sup>10</sup>(Xia et al., 2004) propose an iterative price adaption process for those cases to achieve an approximate solution anyway. Due to time criticality of our system we stick to the pricing history solution.

$$v_r(k) = \begin{cases} \frac{SP_r(k)}{\sum_t \sum_i \sum_j q_{i,j}(r,t)} \quad \forall b_{i,j} \in I^+ & \text{if } I^+ \neq 0 \\ \wedge SP_r(k) \neq 0, \\ \frac{h_{sp}}{\sum_t \sum_i \sum_j q_{i,j}(r,t)} \quad \forall b_{i,j} \in I^+ & \text{if } I^+ \neq 0, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Market value of a resource unit:  $v_r(k) \in \mathbb{R}_0^+$

### 3.2.4 The Task Agents' Bidding Model

The task agents' bidding behavior is determined by two factors:

- At each round,  $o$  new bids  $b_{i,j}$  are generated for each task agent  $a_i$ . The structure of the *new generated BMs* varies according to the matrix types defined in section 3.1. The proxy-agents  $a_i$  have the possibility to submit bids  $b_{i,j}$  as exclusively eligible bundles. The eligibility is defined such that  $m$  *BM*s are treated as XOR bids<sup>11</sup>.
- Task agents repeat bidding for rejected bids in the following round while changing the W2P with respect to the actual resource supply/demand situation.

In our simulation environment we use different types of *new BMs generated* by the task agents<sup>12</sup>. The *BM*s used in this paper have a *structured pattern*: bidder agents require resources with the same intensity for a longer period of time (up to  $t_{max}$  slots). This results in continuous bids of varying lengths that are close to realistic demand structures in distributed ISIP systems.

Based on the resource occupancy  $q_{i,j}(r,t)$  in the *BM*s that are requested by the ISIP provision process, the proxy-agents have to formulate their associated W2P for the bids. Two cases have to be distinguished (see equation 10):

- In the first round, no estimation of the prices can be given to the bidders. For this reason, bidders formulate the W2P for their *first bids* with respect to the initial budget  $BG_i$  and their bidding strategy. This is done by calculating a mean bid price that guarantees the proxy bidder's budget to last for the next  $l$  rounds if  $o$  bids are added in each round.
- In the following rounds, the task agents' calculate their W2P for the *new submitted bids* employing market values for the resources given in equation 8. A factor  $P_i^{ini}$  is included into the calculation of

<sup>11</sup>For the definition of OR, XOR, and other variants of bid connections within bidding languages see (Nisan, 2005).

<sup>12</sup>See (Schwind et al., 2003) for the *BM* type definitions.

the initial bids. By setting  $P_i^{ini}$  to e.g. 0.8, task agents are prompted to submit initial bids slightly lower than resource market price or above market level for e.g.  $P_i^{ini} = 1.2$ . For the *rejected bids*, task agents show the following behavior. The actual price for a bid  $b_{i,j}$  of task agent  $a_i$  in the actual round of a bid is calculated by using the market values of the resources derived from the shadow prices of the preceding round. In order to control the price adaption process, a price acceleration factor  $\Delta P$  is introduced rising  $P_i^{inc}$  by multiplying  $\Delta P$  with the actual number of rounds of the particular bid  $b_{i,j}$ :

$$P_i^{inc} = P_i^{ini} + (\text{round of bid} \cdot \Delta P) \quad (9)$$

The bidding for rejected bids is *repeated* for the following production time span  $t_{pro}$  for the next  $l$  rounds until the bid is accepted, otherwise the bids are *discarded*. The agents bidding behavior is limited by the task agents' budget. If the agents budget is exhausted, no further bids are formulated until the budget is refreshed in the next round  $k$ .

$$p_{i,j}(k) = \begin{cases} \frac{BG_i}{l \cdot o} & \text{for } k = 1 \\ \sum_{r=1}^S \sum_{t=1}^N v_r(k-1) \cdot q_{i,j}(r,t) \cdot P_i^{inc} & \text{for } k > 1 \end{cases} \quad (10)$$

The bidding behavior of the task agents can be modified by varying parameters like  $\Delta P$ ,  $o$ ,  $l$  and  $m$ .

## 4 TESTING DIFFERENT BIDDING STRATEGIES

In this section we will have a closer look at two different economically motivated bidding strategies defined by the task agents' parameters described above. The bidding strategies evaluated in this paper only differ in  $\Delta P$ :

- An *aggressive bidding agent* that submits combinatorial bids while trying to achieve the bid acceptance by using a fast inclining bid pricing strategy. The economic motivation of this behavior can be a proxy agent that bids for the execution of time critical tasks in an ISIP provisioning system. A good example of this is the performance of a video conference in the distributed system. The conference is scheduled for a narrow time window. The proxy agents have to bid for a prompt fulfillment of the resource usage tasks. Therefore, it is useful that proxy agents quickly raise their bids to market level.

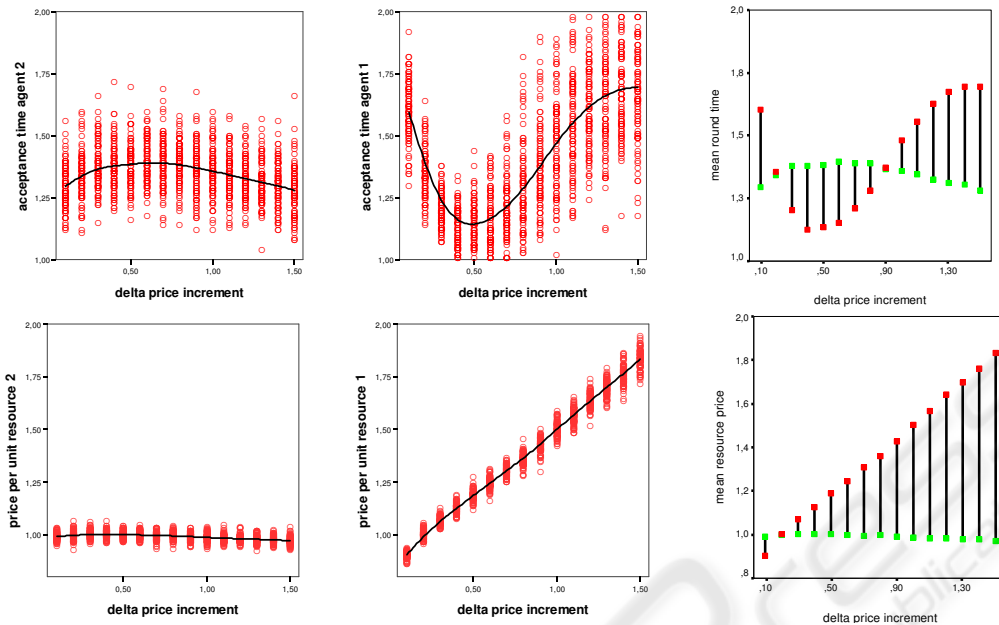


Figure 3: Mean round time (up) and price per resource unit (below) for smooth (agent 2) and aggressive bidder (agent 1) in dependency of price increment  $\Delta P$ .

- A *smooth bidding agent* that submits multiple bid bundles to the auctioneer waiting for bid acceptance of some of the alternative bids while increasing the bid prices slowly. The economic rationale for this type of proxy agent strategy can be the fact that it bids for resources required for the fulfillment of an ISIP task that is not time-critical. An example of this may be the computation of large time-consuming database jobs on a distributed system that have to be done in a very relaxed time window. A plausible strategy for the proxy bidding task agent is then to try to acquire the required resource capacity bundles at low market values with bids with slightly increasing W2P.

For construction of the *closed-loop* grid economy in our experiments we assumed the same production function for all task agents leading to equal payoff  $Inc^{acc}/I$  of the auctioneer's income  $Inc^{acc}$  ( $BG^{ini} = 200MU_s$ ). The applied strategy was either increasing the  $\Delta P$  for the rejected bids by a constant 0.2 as described in equation (9) for the *smooth bidding* agents or varying the bidding strategy in a range from  $\Delta P = 0.1$  to 1.5 (see Table 2) for the *aggressive bidding* agents. Beginning with one bundle containing three XOR bids ( $m = 3$ ) in round one, both types of agents generated three additional bids ( $o = 3$ ) in each further round  $k$ . The bids were held and increased by  $\Delta P$  over a maximum of  $l = 5$  rounds in case of non-acceptance. The pattern of the new generated bids was identical to the structured *BM* type described in Table 1 ( $q_{bmax} = 3, p_{tso} = 0.333, t_{max} =$

4). The auctioneer could allocate a maximum load of  $q_{max} = 8$  per resource while  $T$  was ten time units for the length of the *AM*. Figure 3 shows the results of the strategy simulations, 100 runs for each  $\Delta P$  in 0.1 steps. The aggressive bidder competes fiercely against three smooth bidders. In the upper part of Figure 3, the mean round time until bid acceptance can be seen, whereas the lower part depicts a mean of budget spending per acquired resource unit. For an increasing aggressive agent the mean acceptance time  $\bar{k}_{aggr}^{acc}$  is reduced by 0.25 for  $\Delta P = 0.4, 0.5$  and 0.6 compared to, the average acceptance time  $\bar{k}_{smoo}^{acc}$  of the smooth bidder (See Table 2.). While rising  $\Delta P$ , the average acquisition price  $\Delta \bar{p}$  per resource unit (over all resource types) increases linearly for the aggressive bidder (See Figure 3 lower part.). As illustrated in Table 2, the optimal strategy for the aggressive bidder is a price increment of  $\Delta P = 0.4$  resulting in  $\bar{k}_{aggr}^{acc}$  reduced by  $\Delta \bar{k}^{acc} = 0.25$  with an average resource price increment of  $\Delta \bar{p} = 0.12$  if the agents' utility is defined by  $U_{aggr} = -\bar{k}^{acc} - 0.1 \cdot \Delta \bar{p}$  yielding 0.23.  $U_{aggr}$  is a trade-off between fast bid acceptance and high resource costs in favor of short acceptance time.

## 5 CONCLUSION

We presented an agent-based simulation environment for a grid scheduler that enables the simultaneous allocation of resources in a grid-like computer system.

Table 2: Efficiency of two competing bidding strategies (*smooth, aggressive*) in terms of mean price per acquired resource unit  $\bar{p}$  and average round time  $k^{acc}$  until bid acceptance.

	price increment $\Delta P$														
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
$\bar{p}_{smoo}$	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.98	0.98	0.98	0.97
$\bar{p}_{aggr}$	0.90	1.00	1.07	1.13	1.19	1.25	1.31	1.36	1.43	1.51	1.57	1.64	1.70	1.76	1.83
$\bar{k}_{smoo}^{acc}$	1.30	1.34	1.38	1.38	1.38	1.40	1.39	1.39	1.37	1.36	1.35	1.33	1.31	1.30	1.28
$\bar{k}_{aggr}^{acc}$	1.60	1.36	1.20	1.13	1.13	1.15	1.21	1.28	1.37	1.48	1.56	1.63	1.67	1.69	1.69
$\Delta\bar{p}$	-0.09	0.0	0.07	0.12	0.19	0.25	0.31	0.36	0.43	0.52	0.58	0.66	0.72	0.79	0.86
$\Delta\bar{k}^{acc}$	0.31	0.01	-0.18	-0.25	-0.25	-0.25	-0.18	-0.11	0.0	0.12	0.21	0.30	0.36	0.39	0.41
$U_{aggr}$	-0.29	-0.01	0.16	0.23	0.21	0.20	0.12	0.04	-0.09	-0.22	-0.33	-0.43	-0.51	-0.55	-0.59

Allocation is done by a combinatorial auction in our economically inspired approach where proxy-agents try to acquire optimal resource bundles with respect to limited budgets. The system allows the provision of price information for the resources required to perform various information services and information production tasks in the grid. This is done by calculating shadow prices in connection with solving the  $\mathcal{NP}$ -hard winner determination problem of the combinatorial auction by an integer programming approach. The efficiency of the shadow price-based allocation was tested in a *closed loop* grid system where the agents can use monetary units rewarded for the resources they provide to the system for the acquisition of complementary capacity. Two types of bidding agents have been compared in terms of efficiency (average resource price paid and waiting time until bid acceptance): An *aggressive bidding agent* with strongly rising bids and a *smooth bidding agent* using low bid increments. While searching the strategy space by varying the bidding behavior of the aggressive agent from smooth to very aggressive in a competitive environment with multiple smooth bidders, it turns out that there is a bidding strategy where trade-off between bid acceptance time and average resource price paid is optimal. Future research will address system behavior in resource failure situations and the question of incentive compatible bidding. Additionally, the question of alternative definitions of the utility function for the different agent types should be discussed.

## REFERENCES

- AuYoung, A., Chun, B. N., Snoeren, A. C., and Vahdat, A. (2004). Resource allocation in federated distributed computing infrastructures. In *Proceedings of the 1st Workshop on Operating System and Architectural Support, San Francisco, USA*.
- Bjørndal, M. and Jørnsten, K. (2001). An analysis of a combinatorial auction. Technical Report 2001-11, Department of Finance and Management Science, Norwegian School of Economics and Business Administration, Bergen, Norway.
- Buyya, R., Stockinger, H., Giddy, J., and Abramson, D. (2001). Economic models for management of resources in peer-to-peer and grid computing. In *Proceedings of the SPIE International Conference on Commercial Applications for High-Performance Computing*, Denver, USA.
- Chun, B. N., Buonadonna, P., AuYoung, A., Ng, C., Parkes, D. C., Shneiderman, J., Snoeren, A. C., and Vahdat, A. (2004). Mirage: A microeconomic resource allocation system for sensor network testbeds. In *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*; Sydney, Australia.
- Fujishima, Y., Leyton-Brown, K., and Shoham, Y. (1999). Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence 1999 (IJCAI-99)*, Stockholm, Sweden, pages 548 – 553.
- Kwasnica, A. M., Ledyard, J., Porter, D., and DeMartini, C. (2005). A new and improved design for multi-objective iterative auctions. *Management Science*, 51(3):419–434.
- Nisan, N. (2005). Bidding languages. In Steinberg, R., Shoham, Y., and Cramton, P., editors, *Combinatorial Auctions*. MIT-Press.
- Parkes, D. C. and Ungar, L. H. (2000). Iterative combinatorial auctions: Theory and practice. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 74–81.
- Rassenti, J. S., Smith, V. L., and Bulfin, R. L. (1982). A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13(2):402–417.
- Schwind, M., Stockheim, T., and Rothlauf, F. (2003). Optimization heuristics for the combinatorial auction problem. In *Proceedings of the Congress on Evolutionary Computation CEC 2003*, pages 1588–1595.
- Vries, S. D. and Vohra, R. (2001). Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309.
- Xia, M., Koehler, G. J., and Whinston, A. B. (2004). Pricing combinatorial auctions. *European Journal of Operational Research*, 154(1):251–270.