Agents' Bidding Strategies in a Combinatorial Auction

Tim Stockheim[1], Michael Schwind[2], and Oleg Gujo[2]

[1] Business Information Systems and Operations Research, Technical University Kaiserslautern, Gottlieb-Daimler-Strasse 47, D-67663 Kaiserslautern, Germany
stockheim@wiwi.uni-kl.de
[2] Institute of Information Systems, Johann Wolfgang Goethe University, Mertonstrasse 17, D-60054 Frankfurt, Germany
schwind, gujo@is-frankfurt.de

Abstract. This paper presents an *agent-based simulation environment* for task scheduling in a grid. Resource allocation is performed by an *iterative combinatorial auction* in which *proxy-bidding agents* try to acquire their desired resource allocation profiles. To achieve an efficient bidding process, the auctioneer provides the bidding agents with approximated shadow prices from a linear programming formulation. The objective of this paper is to identify optimal bidding strategies in multi-agent settings with respect to varying preferences in terms of resource quantity and waiting time until bid acceptance. On the basis of a utility function we characterize two types of agents: a *quantity maximizing agent* with a low preference for fast bid acceptance and an *impatient bidding agent* with a high valuation of fast allocation of the requested resources. Bidding strategies with varying initial bid pricing and different price increments are evaluated. Quantity maximizing agents should submit initial bids with low and slowly increasing prices, whereas impatient agents should start slightly below market prices and avoid 'overbidding'.

1 Introduction

We present an agent-based simulation environment for resource allocation in a distributed computer system that employs a combinatorial task scheduler. Our environment enables the simulation of a mechanism for the simultaneous allocation of resources in the distributed computer system. In contrast to traditional grid allocation approaches, our allocation process considers production complementarities and substitutionalities for these resources thus raising the efficiency level of the resulting resource [1]. The central scheduling instance of our system is comparable to an auctioneer that performs an iterative *combinatorial auction (CA)*. Agents try to acquire the resources required in computational tasks for the provisioning of *information services and information production (ISIP)* by submitting package bids. We introduce a utility function that will allow us to represent different preferences of agents, i.e. a trade-off between quantity maximization and fast acceptance of bids. In a first setting we identify the strategy

K. Fischer et al. (Eds.): MATES 2006, LNAI 4196, pp. 37–48, 2006.

that maximizes the utility of four homogeneous, quantity maximizing bidders. Subsequently, we introduce an additional (competitive-bidding) agent that requires the requested resources as soon as possible. The earlier this agent receives the acceptance the higher is its gained utility. We compare the bidding strategies under changing resource capacity situations with respect to their allocation efficiency. Finally, we interpret the outcome in terms of gained utility for the various bidding strategies.

2 Combinatorial Auctions for Resource Allocation in Distributed Computer Systems

Various auction protocols have been proposed for resource allocation in distributed computer systems. This may derive from the fact that auctions have been thoroughly investigated by economists and have proved to be efficient allocation mechanisms [2]. The transfer of economic principles to resource attribution in grid systems, such as *price controlled resource allocation (PCRA)*, allows the *flexible* implementation of control mechanisms in decentralized systems [3,4].

CAs are a suitable tool to allocate interdependent resources according to the willingness-to-pay *(W2P)* of the participants. The production process for information services in distributed systems comprises an allocation problem with strong complementarities. An example of such an information service is the provisioning of a video conference service via the web or the off-line calculation of distributed database jobs that have to be processed on different computers and acquires CPU time. Without obtaining communication network capacity between the computers, the acquired CPU time is useless. The application of CAs for resource allocation in distributed computer systems is still in its infancy despite its excellent applicability to grid computing. In a recent approach, Chun et al. [5] present a CA based mechanism for resource allocation in a *SensorNet* testbed where the devices have 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 continuation of this work is the grid computing environment *Bellagio* [6]. The approach relies on Berkeley's CA based allocation scheme *SHARE* [7]. Each bidder has a budget of a virtual currency available for task payment purposes. The required resources are allocated to the particular tasks through a combinatorial second-price auction, which can be regarded as a strategy proof mechanism [7]. In several experiments, the system is tested with respect to scalability, efficiency, and fairness. A simple greedy algorithm guarantees the system's scalability, however, the resulting allocation does not reveal a satisfactory efficiency level. Most recent approaches in "Grid Economics" use double sided CAs for the exchange of resources [8]. However, there is no CA grid system that makes use of proxy-bidding agents to autonomously procure the resources required for ISIP provision.

3 An Agent-Based Simulation Environment for Combinatorial Resource Allocation

The presented CA environment is based on the JADE 3.3 agent workbench and goes beyond the recent research approaches in several points:

- The system allows the usage of several winner determination algorithms such as greedy, simulated annealing, genetic programming, and integer programming methods according to the users' requirements in terms of allocation quality and computation time.[1]
- The simulator provides tools to investigate various bidding behaviors of the proxy agents in the resources acquisition process.
- The framework can simulate changing resource capacities to test allocation efficiency and system stability.
- The ontology-based bidding protocol opens the system to additional agents, e.g. to test strategies based on machine learning or reasoning.

The results presented in this paper concentrate on the second aspect.

3.1 Scenario for a Price Controlled Resource Allocation

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 required for the processing of the data, *volatile memory capacity (MEM)* which is necessary to store short-term processing data and program codes for the central processing units, *non-volatile storage capacity (DSK)* which is necessary to keep mass data on databases, and *network bandwidth (NET)* that is required for data interchange between different computer units.[2]

The task agents submit several bids as exclusively eligible bundles *(OR-of-XOR)*. For the formal representation of the bids, a two-dimensional *bid-matrix (BM)* is used. One dimension describes the time $t \in \{1, \ldots, T\}$ at which the resource is required within the request period. The other dimension $r \in \{1, \ldots, R\}$ denotes the resource types MEM, CPU, NET, and DSK. The request for a certain quantity of an individual resource r at time t is then denoted by a matrix element $q_{i,j}(r,t)$. A price $p_{i,j}$ is assigned to each BM expressing the agent's W2P for the resource bundle. In both cases the corresponding bid bundle is identified by the index i and the single BM by j.

The value q^{bmax} denotes the maximum resource load that can be requested by a bidder for a single BM element $q_{i,j}(r,t)$. These elements are occupied with *time slot occupation probability* wk^{tso}. The value q^{bmax} denotes the maximum

[1] Schwind et al. provides a description of the algorithms [9].

[2] Network connections themselves exhibit complementarities due to their peering character. For simplicity we assume that NET capacity can be managed as one single system resource. To consider the individual connections explicitly in our model, they should be treated as additional resources, one for each connection type.

resource load that can be requested by a bidder for a single BM element $q_{i,j}(r,t)$. These elements are occupied with *time slot occupation probability* wk^{tso} and have a maximum allocable resource quantity q^{max}. We use a structured bid matrix, i.e. up to t^{max} time slots are occupied in a row [9].

The agents used within the combinatorial grid simulator comply with three different *roles*. *Task agents* bid for the required resource combination via the mediating agent. This auctioneer receives the resource bids and calculates an allocation profile for the available resources managed by the *resource agents* according to the allocation mechanism. *Resource agents* manage available resources on their particular IT systems and offer them to the task agents via the mediating agent. If the auctioneer accepts a bid, he reserves the resources via the resource agents.

3.2 The Combinatorial Auction

Following the description of the scenario, Figure 1 illustrates the course of action of the system. The AUML sequence diagram depicts the message flow based on the FIPA definition of the English auction protocol.[3]

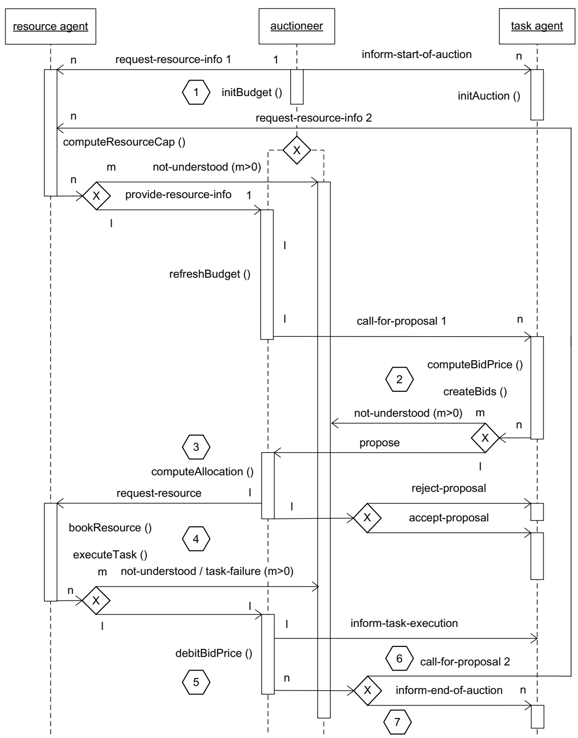

Fig. 1. FIPA AUML diagram for the iterative combinatorial scheduling auction

[3] `www.fipa.org/specifications/fipa00031D`.

While in our closed economy each bidder has two roles (as provider and user of resources), Figure 1 separately depicts both roles (*resource agent* and *task agent*) to provide a higher generalization and better readability. Each step is marked by a ○-symbol and detailed in the corresponding paragraph:

1. The auctioneer requests the resource agents to evaluate the available resource capacities and informs the bidders about the bidding terms. He also awards an initial budget to the task agents. Subsequently, he announces the start of the auction.
2. Following the auctioneer's call for proposal, the task agents create their bids according to the desired resource combination.
3. The auctioneer receives the bids and calculates the return-maximizing combinatorial allocation. He informs the task agents about 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 task status information to the task agents and debits the bid price for the awarded bids from their accounts, followed by a call for proposal for the next round.
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.

3.3 The System's Budget Management Mechanism

We assume that the ownership of the resources is distributed among a group of companies and that each proxy-agent represents a company. In order to avoid an expiration of the agents' budgets during the iterative auctioning process, the agents are integrated into a monetary circuit. The system simulates an economy with a constant circulating budget, i.e. a *closed-loop* grid.

Each agent is initialized with a monetary budget BG^{ini}. At the beginning of each round k, the task agents' budgets are refreshed (see Figure 1 - ○1) such that each agent is able to acquire *'computational capacity'* proportional to the amount of resources provided to the system. Task and resource agents act as a unit of consumer and producer both owning the resources of their peer system. The resource agent does the reporting of resource usage and provisioning for the task agent owning the peer computer resources (see Figure 1 - ○1, and ○4). All task agents receive the same budget in each round of the simulation. The accounting of the agents' budgets in the grid system is done by the combinatorial auctioneer (see Figure 1 - ○1, and ○5).

3.4 The Combinatorial Auctioneer

The combinatorial auctioneer controls the iterative allocation process of the grid system. For this purpose, the auctioneer awaits the bids that have been submitted by the task agents for the current round. The bids that are submitted in the form of j XOR-bundled BMs in bid i and represent the task agents' requested capacity $q_{i,j}(r,t)$ of the resources r at a particular point of time t. After having received all alternative BMs submitted by the task agents, the auctioneer has to solve the *combinatorial auction problem (CAP)* which is NP-hard [10,11]. 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 is formulated as:

$$
\begin{aligned}
&\max \sum\nolimits_{i=1}^{I} \sum\nolimits_{j=1}^{J_i} p_{i,j}\ x_{i,j} \\
\text{s. t.}\ \ & q(r,t) = \sum\nolimits_{i=1}^{I} \sum\nolimits_{j=1}^{J_i} q_{i,j}(r,t)\ x_{i,j} \le q^{max}(r,t),\ \ \forall_{r \in \{1,\dots,R\}, t \in \{1,\dots,T\}} \\
& \sum\nolimits_{j=1}^{J_i} x_{i,j} \le 1, \quad \forall_{i \in \{1,\dots,I\}}.
\end{aligned} \tag{1}
$$

The following variables are used: number of resources $R \in \mathbb{N}$, number of time slots: $T \in \mathbb{N}$, number of bid bundles $I \in \mathbb{N}$, number of bids in bundles $J_i \in \mathbb{N}$, W2P of bundle j in bid i as $p_{i,j} \in \mathbb{R}^+$, and the acceptance variable $x_{i,j} \in \{0;1\}$.

The auctioneer's primary goal is to maximize the received income under the limitation of the available resources and a maximum of one accepted bid per XOR bundle (equation 1). In order to accelerate the price-finding process, the auctioneer provides feedback on resource availability to the bidders. As mentioned above, it is not always possible to calculate *unambiguous prices* (anonymous prices) for the individual resources in a CA. In many cases, explicit resource prices can only be calculated for each individual bid. Kwasnica et al. [12] describe a pricing scheme for all individual goods in a CA by approximating the prices in a divisible case based on a linear programming *(LP)* approach first proposed by Rassenti et al. [1]. Like in a similar approach by Bjørndal and Jørnsten [13], they employ the *dual solution* of the relaxed WDP to calculate the shadow prices. In our simulation model, the approach of Kwasnica et al. [12] is adopted:[4]

$$
\begin{aligned}
&\min\ z = \sum\nolimits_{r=1}^{R} \sum\nolimits_{t=1}^{T} q^{max}(r,t) \cdot sp_{r,t} \\
s.t.\quad & \sum\nolimits_{r=1}^{R} \sum\nolimits_{t=1}^{T} q_{i,j}(r,t) \cdot sp_{r,t} + (1 - x_{i,j}) \cdot \delta_{i,j} = p_{i,j}
\end{aligned} \tag{2}
$$

Reduced cost: $\delta_{i,j} \in \mathbb{R}_0^+$
Shadow price of one element: $sp_{r,t} \in \mathbb{R}_0^+$

[4] The result of the following formula is denoted as reduced SPs. Omitting the rejected bids in the calculation of dual prices yields a higher result [13].

The proposed SP calculation uses the *primal solution* for the LP problem delivered from open source LP solver *LPSOLVE 5.5*[5] for the determination of accepted bids. Now the *market value of a resource unit* can be calculated while using the weighted shadow prices and summarizing the utilized capacity of each resource r for all accepted bids as follows:

$$\text{Market value:} \quad v_r = \frac{\sum_{t=1}^{T} sp_{r,t} \cdot q(r,t)}{\sum_{t=1}^{T} q(r,t)} \quad \forall_{r \in \{1,\dots,R\}} \tag{3}$$

Due to the fact that bid prices are non-linear in this framework, shadow prices $sp_{r,t}$ cannot be calculated in each round, i.e. there is no solution to the LP problem [12]. In such cases the auctioneer relies on an approximation of the market values $\hat{v}_r$ as the averaged market values calculated in the last n rounds.[6]

3.5 The Task Agents' Bidding Model

Except ΔP and P^{ini} all agents show the same behavior. Based on the market values of resources v_r, the task agents of the combinatorial simulation model try to acquire the resources needed for ISIP provision. Besides the market values of resources, their bidding behavior is determined by their budgets and by a *bidding strategy*. In each round, a task agent generates M new bids. In the first round, a market value of the resources is not provided to the bidders. Therefore, bidder agents formulate the W2P for their *initial bids* by dividing the budget by $L \cdot M \cdot J$ to calculate a mean bid price that guarantees the task agents' budget to last for the next L rounds. In the following rounds, if a bid is initialized, the capacities required are multiplied by the corresponding market value v_r and summarized. To control the price adaption process, an additional price acceleration factor P_i^{inc} is introduced.

$$\text{W2P of bundle } j \text{ in bid } i\text{:} \quad p_{i,j} = P_i^{inc} \cdot \sum_{r=1}^{R} \sum_{t=1}^{T} v_r \cdot q_{i,j}(r,t) \tag{4}$$

If a bid is rejected, the task agent repeats bidding for this rejected bid in the following rounds. Its W2P is adapted by $P_i^{inc} = P^{ini} + (l_i \cdot \Delta P)$, resulting in the value of P^{ini} in round $l_i = 0$. Rejected bids are *repeated* with an updated W2P up to a maximum of L rounds or until the bid is accepted. When an agent's budget is exhausted, it formulates no new bids until the budget is refreshed.

4 The Preferences of the Task Agents

We will now have a closer look at the bidders' different *preferences*. Two proxy-agent types are used in the context of this paper to represent these preferences:

[5] `http://www.geocities.com/lpsolve/`

[6] If the market value is approximated, the value $\hat{v}_r$ is not saved in the history.

– A *quantity maximizer* that requires high resource capacities but has weak preferences regarding the timing. However, the time of execution and the complementarities of the resources within the bundle have to be satisfied. The hypothesis is that a *smooth bidding strategy*, i.e. to slowly increase the bid prices, maximizes the utility of this agent. The economic rationale for this 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 is the generation of reports based on large databases on a distributed system that have to be done in a relaxed time window.
– An *impatient bidder* that benefits from the possibility to instantaneously use the resources will apply an *aggressive bidding strategy* to maximize his utility. This agent has to submit high initial prices, but overpaying will reduce the quantity he can acquire. We analyze whether a fast inclining pricing strategy combined with lower initial bids can help to further increase the utility of this agent. The economic motivation of this utility function can be a proxy agent that bids for the execution of time-critical tasks. A good example of this is the performance of a video conference in the distributed computer system which is scheduled for a narrow time window.

Clearly, the amount of acquired ISIP resources has a positive but diminishing marginal impact on the agent's utility. The strength of this impact will be defined by α. Opposing the positive impact of the amount of acquired resources, the number of periods an agent has to wait before its bids are accepted has a negative impact. To calculate the decreasing impact of the waiting time, we use a time index $\bar{l_a}$ which is defined as the averaged number of periods an agent bids until it has placed a successful bid and β to adjust the influence of the waiting time. The utility of an agent is calculated by the following function:

$$U_a = \frac{\left(\sum_{(i,j)\in B_a} x_{i,j} \cdot \sum_{r=1}^{R} \sum_{t=1}^{T} \ q_{i,j}(r,t)\right)^{\alpha}}{\left(\bar{l_a}\right)^{\beta}} \tag{5}$$

Utility function of agent a: $U_a \in \mathbb{R}^+$
Bids of agent a: $B_a \in \{(i,j) \mid i,j \in \mathbb{N}\}$

Using the utility function the two different agent types are: (1) the *quantity maximizer* with $\alpha = 0.5$ and $\beta = 0.01$ and (2) the *impatient bidder* with $\alpha = 0.5$ and $\beta = 1.0$.

5 Results

The primary objective of the experiments is to find out the test agent's optimal bidding strategy in competition with the remaining default bidding agents, given the two types of utility function (quantity maximizer, impatient bidder) as defined above.

In all simulations an identical basic setting is used: Beginning with one bundle containing three XOR-bids in the first round, four agents generated three additional bid bundles for each further round k. The task agents increase the W2P of rejected bids by Δp over a maximum of $L = 5$ rounds. The pattern of newly generated BMs is defined by $q^{bmax} = 3$, $wk^{tso} = 0.333$, and $t^{max} = 4$. The auctioneer was able to allocate a maximum load of $q_{max} = 8$ per resource while T was 8 units for the CM. For the evaluation of our model, we set the number of bids per agent (M) to 3, which for four agents results in $I = 12$ bids per round. Resource 1 is reduced to an amount of 4 units in the 25th round.

In a first setting, all agents use a *default bidding* behavior with a constant value of $\Delta P = 0.2$. This supports the price adaptation process in case of resource failures. Figure 2 shows the results of 50 simulations for varying values of $P^{ini} = 0.3 \ldots 1.0$ to identify the optimal bid introduction level.

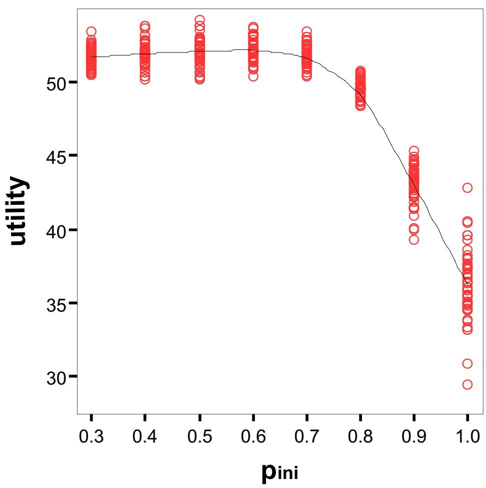

Fig. 2. Averaged utility values of the four homogeneous bidders with varying initial price P^{ini} and static price increment $\Delta p = 0.2$

Assuming quantity maximizing agents, it turns out that setting $P^{ini} = 0.6$ maximizes the average utility of the agents ($\bar{U_a} = 52.21$, capacity $= 2762.78$, time index $= 1.96$). The stochasticity of the bidding process in connection with the combinatorial complexity of the CAP leads to an occasional acceptance of low-priced bids. Therefore, a bidding strategy that tries to procure resources below market prices turns out to be successful. In contrary, agents that initialize bids at the market price (which is an averaged value) risk to overpay the required resources. The cumulated capacity used by the agents was 11,051 units, which translates in a utilization rate of 73.7 % (max. capacity: 15,000). While in round 1 to 25 the rate was 82.5 %, it dropped to 67.2 % percent in round 26 to 50 (after reducing the capacity of resource 1 from 8 to 4 units per time slot).

The next setting introduces a competitive bidder that differs in his strategy from the other agents. In compliance with the results from the first setting, we set the default bidding strategy to $P_{ini} = 0.6$ and $\Delta P = 0.2$. Figure 3 shows the resource units acquired by the test agent and the averaged bid acceptance time of 50 simulation runs for each ΔP, Pp^{ini} combination (steps of 0.1).

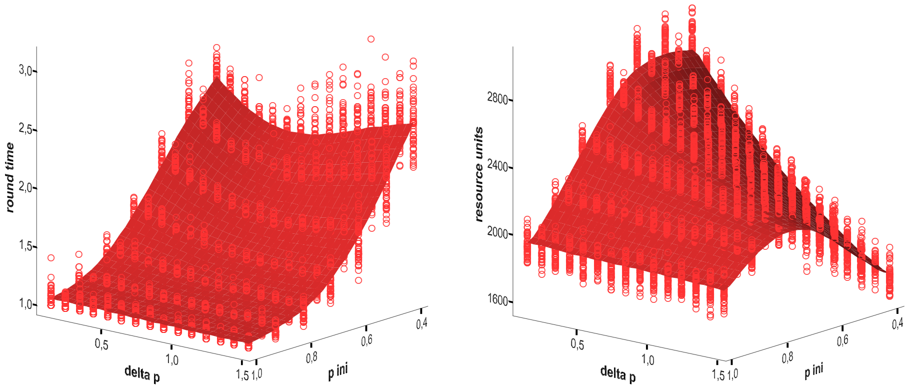

Fig. 3. Mean acceptance time and quantity of resource units for competitive bidder with varying price increment Δp and initial price P^{ini}

In case of small ΔP and P^{ini}, the highest amount of resource units can be acquired by the task (test) agents. An (too) aggressive strategy with high ΔP and P^{ini} leads to a declining amount of acquired resources. While a reduction of acceptance time is mainly achieved by high P^{ini}, increasing ΔP has only impact on average acceptance time if P^{ini} is low.

In Figure 5, the utility (cp. equation 5) resulting from varying price increment ΔP and initial pricing P^{ini} is depicted. It pays off for the quantity maximizer to wait if his bids fit into the current allocation at a relative low price (low increment and initial price). In contrary, the impatient bidder gains low utility from such a strategy (Fig. 5 right side). The impatient agent receives the highest utilities by using an initial bid price close to the market value of the resources ($p^{ini} = 0.9$). Interestingly, the price increment in the following round does not have much impact on the acceptance time and therewith on the utility of the impatient test bidder.[7] However, for bids exactly at market value utility declines sharply, signaling the peril of *'overbidding'* or simply paying too much for the required ISIP resources. This underlines the importance of accurate market value information to achieve allocations that maximize the benefits of the bidders. The shadow price-controlled combinatorial grid enables agents to implement efficient

[7] If the impatient bidder follows its optimal strategy ($p_{ini} = 0.9$ - default value of $\Delta p = 0.2$), the cumulated capacity used by the agents was 11,051 units, which translates in a utilization rate of 65.0 % (max. capacity: 15,000). While in round 1 to 25 the rate was 74.34 %, it dropped to 54.31 % in round 26 to 50 (after reducing the capacity of resource 1 from 8 to 4 units per time slot.

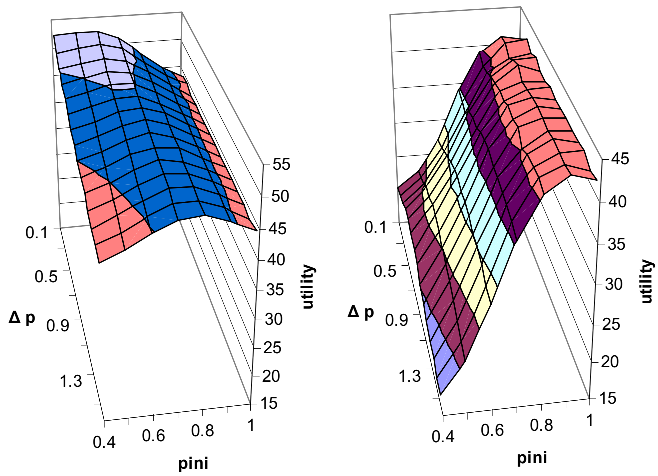

Fig. 4. Utility of the test bidder for quantity maximizing preference $\beta = 0.01$ (left) and impatient bidding behavior $\beta = 1.0$ (right) for determination of the optimal bidding strategy under varying price increment Δp and initial pricing behavior P^{ini}

bidding strategies according to the user's utility functions. Clearly, a high impact of the competitors' behavior remains as challenge.

6 Conclusion

This paper presents an agent-based simulation environment that enables the simultaneous allocation of resources in a grid-like computer system. In this economically inspired approach where proxy-agents try to acquire optimal resource bundles with respect to limited budgets, the allocation is done by a CA. The auctioneer provides price information that is calculated as shadow prices in connection with solving the $\mathcal{NP}$-hard winner determination problem by an integer programing approach. Based on these settings, bidding strategies are evaluated with respect to utility functions that incorporate different levels of time preferences of the bidders. We introduce two characteristic bidders: A *quantity maximizing agent* with low preference for fast bid acceptance and an *impatient bidding agent* with a high valuation of fast allocation of the requested resources. While searching the strategy space by varying the bidding behaviors in terms of initial bid price and price increment strategy for rejected bids, we identified optimal bidding strategies in terms of achieved utility. For the quantity maximizing agent, patience at low initial bids pays off, whereas the impatient agent should avoid 'overbidding'.

Although we used a small number of agents, our simulations turned out to be very time consuming. A way to reduce the volatility of our market prices is the use of a larger population of agents. Therefore, a future objective is to improve the allocation algorithm and analyze heuristic approaches in order to reliable handle larger settings. A second objective might be the introduction of agents that learn their optimal strategy to find a stable equilibrium.

References

1. Rassenti, J.S., Smith, V.L., Bulfin, R.L.: A combinatorial auction mechanism for airport time slot allocation. The Bell Journal of Economics **13**(2) (1982) 402–417
2. Milgrom, P.: Putting Auction Theory to Work. Cambridge University Press (2004)
3. Buyya, R., Stockinger, H., Giddy, J., Abramson, D.: 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 (2001)
4. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: Why grid and agents need each other. In: Proceedings of the 3rd Int. Conference on Autonomous Agents and Multi-Agent Systems (AAMAS04), New York, NY (2004) 8–15
5. Chun, B.N., Buonadonna, P., AuYoung, A., Ng, C., Parkes, D.C., Shneiderman, J., Snoeren, A.C., Vahdat, A.: Mirage: A microeconomic resource allocation system for sensornet testbeds. In: Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II); Sidney, Australia. (2004)
6. AuYoung, A., Chun, B.N., Snoeren, A.C., Vahdat, A.: Resource allocation in federated distributed computing infrastructures. In: Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT InfraStructure, San Francisco, USA. (2004)
7. Ng, C., Parkes, D.C., Seltzer, M.: Virtual worlds: Fast and strategyproof auctions for dynamic resource allocation. In: Proceedings of the third ACM Conference on Electronic Commerce (EC-2003), San Diego, CA, ACM (2003) 238–239
8. Schnizler, B., Neumann, D., Veit, D., Weinhardt, C.: A multiattribute combinatorial exchange for trading grid resources. In: Proceedings of the 12th Research Symposium on Emerging Electronic Markets (RSEEM), Amsterdam, Netherlands, 2005. (2005)
9. Schwind, M., Stockheim, T., Rothlauf, F.: Optimization heuristics for the combinatorial auction problem. In: Proceedings of the Congress on Evolutionary Computation CEC 2003. (2003) 1588–1595
10. Parkes, D.C., Ungar, L.H.: Iterative combinatorial auctions: Theory and practice. In: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). (2000) 74–81
11. Fujishima, Y., Leyton-Brown, K., Shoham, Y.: 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. (1999) 548 – 553
12. Kwasnica, A.M., Ledyard, J., Porter, D., DeMartini, C.: A new and improved design for multi-objective iterative auctions. Management Science **51**(3) (2005) 419–434
13. Bjørndal, M., Jørnsten, K.: 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 (2001)