Q-Strategy: Automated Bidding and Convergence in Computational Markets

Nikolay Borissov
University of Karlsruhe
Institute of Information Systems and Managament
Englerstrasse 14, 76131 Karlsruhe, Germany
borissov@iism.uni-karlsruhe.de

Abstract

Agents and market mechanisms are widely elaborated and applied to automate interaction and decision processes among others in robotics, for decentralized control in sensor networks and by algorithmic traders in financial markets. Currently there is a high demand of efficient mechanisms for the provisioning, usage and allocation of distributed services in the Cloud. Such mechanisms and processes are not manually manageable and require decisions made in quasi real-time. Thus agent decisions should automatically adapt to changing conditions and converge to optimal values.

This paper presents a bidding strategy, which is capable of automating the bid generation and utility maximization processes of consumers and providers by the interaction with markets as well as to converge to optimal values. The bidding strategy is applied to the consumer side against benchmark bidding strategies and its behavior and convergence are evaluated in two market mechanisms, a centralized and a decentralized one.

Introduction

Technology trends like Grid and Cloud computing foster research and business to investigate mechanisms and business models that organize and utilize their computing infrastructures more efficiently in order to reduce costs and identify free capacities. Such free resource capacities can be offered to external consumers on-demand by deriving additional income to their providers. Researchers e.g. in physics, biology and chemistry already share and utilize distributed computing resources using well-known Grid middleware. Prominent business like Amazon, Google, Sun and Salesforce already offer Grid-based services on-demand by applying static pricing models like pay-per-use or subscription for given static resource configurations.

Although technologies emerge, there is still a demand for convincing mechanisms and business models that drive consumers to use external resources without any objections. One of these objections is the required effort to execute a job or an application on an external provider resource. In research, Grid middleware offer tools, standardized interfaces and message protocols to bundle and provide computing resources as well as to submit and execute demanding jobs. However, the allocation of jobs to computing resources is performed by schedulers, which only consider the technical descriptions of provider resources and consumer preferences. In a Cloud scenario, computing services can be utilized by any consumer, who is willing to pay for the provided resource configuration and price model. Similar to the Grid, there is a demand for mechanisms, tools, standardized interfaces and message protocols, which allow an efficient market-based allocation of Cloud services and support their provisioning and usage processes. For providers, such processes are the identification of idle resource capacities, calculation of corresponding reserve prices and achievement of a higher payoff from the market. Consumers want to execute their jobs on-demand or scale their applications by achieving higher utility from the market than from a local job execution. All these processes are not manually manageable and should be automated with minimal human intervention.

In a real market scenario, it is assumed that consumers and providers will act rationally by maximizing their own utilities. Since there is no mechanism that satisfies all mechanism-design desiderata (Myerson and Satterthwaite 1983), existing market mechanisms and bidding strategies are investigated that can enable market-based allocation of Cloud-based services. Our main focus is to explore bidding strategies, which can automate the bidding processes for consumers and providers and are capable to converge and adapt in dynamic market environments, where demand and supply change over time. This paper presents such a bidding strategy, which can adapt in dynamic market environments by converging to optimal values (bids). The presented bidding strategy is applied on consumer side against benchmark bidding strategies and its behavior and convergence is evaluated in centralized and decentralized market mechanisms.

Market-Based Resource Allocation

Economic models for resource scheduling are widely explored in the literature (Wolski et al. 2001; Parkes, Singh, and Yanovsky 2004; Lai et al. 2005; Nassif, Nogueira, and de Andrade 2007). According to the coordination modes, scheduling mechanisms can be categorized into “centralized mechanisms”, where the allocation decision of bids and offers is taken by a central unit, and “decentralized mechanisms", where the allocation decision is decentralized, i.e. made by the requesters based on all of the responses they receive from the environment. According to the allocation modes, scheduling mechanisms can be divided into mechanisms that execute periodically (also called "off-line mechanisms"), and mechanisms that execute continuously (also called "on-line mechanisms"). Examples for off-line and centralized mechanisms are the Call Double Auction (Grosu and Das 2006) and combinatorial mechanisms like proposed in (Bapna et al. 2007). Examples for centralized on-line mechanisms are the Continuous Double Auction (CDA) and Tycoon (Lai et al. 2005). Decentralized market mechanisms for machine scheduling have been explored to a lesser extent in the current literature. An example for on-line machine scheduling is the Decentralized Local Greedy Mechanism (DLGM) (Heydenreich, Müller, and Uetz 2006).

Auction and strategy selection are closely connected in the sense that a given choice of an auction mechanism will affect the choice of the target bidding strategy, and vice versa. For example, some bidding strategies perform well in a CDA, but not in a Dutch auction. This also implies that the agent success in a particular auction depends on the selected bidding strategy. Past and current research explores trading agents and bidding strategies in various fields like financial markets (Das et al. 2001; Sherstov and Stone 2005; Vytelingum, Cliff, and Jennings 2008), supply chain management (Pardoe and Stone 2007) and market-based Grid scheduling (Li and Yahyapour 2006; Reeves et al. 2005). Wellman, Greenwald, and Stone gave an overview of the various agents and their used strategies in the trading agent competition. Phelps investigated an evolutionary approach for learning the space of bidding strategies.

This section presents two on-line scheduling mechanisms, CDA as centralized and DLGM as decentralized one, adopted for market-based scheduling of computational services as well as two state-of-the-art bidding strategies as benchmarks for the selected market mechanisms.

Continuous Double Auction

The continuous double action (CDA) is one of the well-studied market mechanisms, commonly employed in commodity and financial markets. In this market, the consumers and providers bids are matched "continuously" in the sense that the market clears instantaneously on receipt of a bid. If there is no match, the bid is stored into an order book until a match or the specified valid time ("time to leave") expires.

The CDA market has been widely employed in experimental economic studies, where different agent strategies are investigated for applying automated bidding behavior by the provisioning and usage of resources (Das et al. 2001; Vytelingum, Cliff, and Jennings 2008). The matching in a CDA is mostly based on a single value – price, which incorporates in a computing resource scenario the consumer's preferences for a job i.e. his value v_j per time unit. Based on a preferred bidding strategy a consumer generates and submits a bid price ($b_j \leq v_j$) to the CDA market. Respectively, based on a preferred bidding strategy, a resource provider also sets an offer price ($o_i \geq v_j$) for its resource. In case of a match the consumer can immediately execute his job on the provider's machine with a payment π to the provider, calculated by the winning bid and offer prices. The market price of a match, in our case, is calculated using the k-pricing schema (Satterthwaite and Williams 1989), $\pi_m = kb_j + (1-k)o_i$, with $k = 0.5$. The choice of the parameter $k \in [0, 1]$ influences the price for which the trade occurs. If $k = 0$, the provider sets the price, at $k = 1$, the price is set by the consumer.

Decentralized On-line Machine Scheduling

In the case of the Decentralized Local Greedy Mechanism (DLGM) (Heydenreich, Müller, and Uetz 2006), each time a job j arrives on the consumer side, his bidding agent creates a request in the form $t_j = \{r_j, d_j, v_j\}$ with its release date r_j, duration d_j and valuation v_j, and reports this to all known providers. The valuation v_j expresses the costs of the job for waiting one additional time unit in the provider machine's queue.

Based on the received bids and the local scheduling policy, each machine performs real-time planning - if job j has a higher priority value than $k \Leftrightarrow \frac{v_j}{d_j} \geq \frac{v_k}{d_k}$, then j is scheduled before job k in the waiting queue. Depending on the current local waiting queue, the machine i reports a tentative (ex-ante) completion time $\hat{C}_{i,j}$ and a tentative payment $\hat{\pi}_{i,j}$ to the agent of job j. The payment $\hat{\pi}_{i,j}$ contains the aggregated compensation payments $\pi_{i,j} = \sum_{k, \frac{v_j}{d_j} \geq \frac{v_k}{d_k}} d_j * v_k$ to all job-agents whose jobs are currently waiting at machine i and are delayed due to allocation of j.

Upon receiving information about its tentative completion time and required payments, the job-agent makes a binding decision to queue at a certain machine i, and pays $\hat{\pi}_{i,j}$ to the delayed jobs. The decision on which machine to submit the job is taken based on the consumer's utility function, $u_j = -v_j * \hat{C}_{i,j} - \hat{\pi}_{j,i}$, which selects a provider machine's offer i with the shortest weighted tentative completion time $v_j * \hat{C}_{i,j}$ and tentative compensation payments $\hat{\pi}_{i,j}$. The providers applying DLGM do not behave strategically and do not request compensation for the use of their services. The payments are divided only among the consumers for compensating the delayed jobs. Heydenreich et al. showed that DLGM achieves a performance ratio of 3.281 against an optimal off-line scheduling mechanism.

Bidding Strategies

Following sections introduce benchmark strategies for the selected market mechanisms used to evaluate Q-Strategy.

Truth-Telling Strategy In the model of Heydenreich, Müller, and Uetz agents do not remember the outcomes of earlier market interactions, but are somewhat "myopic" in the sense that they only consider the current situation. As shown for the model in Heydenreich, Müller, and Uetz, without knowledge about the future, at time r_j it is a utility maximizing strategy (myopic best response) s_j for an agent $j \in J$ to report truthfully $t_j = \{r_j, d_j, v_j\}$ instead of $\hat{t}_j = \{\hat{r}_j, \hat{d}_j, b_j\}$ to the system and to choose the machine i which maximizes $\hat{u}_j \ (i|s_{-j}, \hat{t}_j, t_j)$ whatever other agents' strategies are. Although a simple strategy, truth-telling is essential in strategy-proof mechanisms, because it guarantees optimal payoffs, no matter what strategies are adopted by other agents. However, in budget-balanced double-auction mechanisms, this strategy is not dominant (Phelps 2007).

Zero Intelligence Plus Strategy Zero-Intelligence Plus (ZIP) agents are widely explored and become a popular benchmark for agents trading in CDA (Das et al. 2001). In Cliff, the author shows that zero intelligence (ZI) agent strategy is not enough, since the bids are uniformly generated between a given interval and do not depend on current or past market information (bids, offers, clearing prices). They introduced a new type of agent, the ZIP agent, which uses the public market information to adapt the bid price. Central to the ZIP agent is the rule for updating the profit margin μ. For a consumer agent, this is the difference between its valuation v (i.e. maximum willingness to pay) and the generated bid price b, for the provider respectively, the difference between his valuation v (reservation price) and offer price b. The ZIP's relationship between profit margin, generated bid and valuation is represented through the rule $b_i(t) = (1 + \mu_i(t))v_i$. The rules for raising and lowering consumers' and providers' profit margins are based on whether the last signal was a bid or an offer and whether the agent has selling or buying intention. Das et al. show that ZIP agents perform better than (non-expert) human traders on CDA markets. Simulations show that adopting a ZIP strategy in markets dominated by other kinds of agents results in an increased profit when the dominating agents are ZI, Kaplan or GD.

This strategy is mainly evaluated and designed for CDA. A drawback of the ZIP strategy is that it requires public information of the providers and consumers price signals. This bidding strategy is selected as one of the benchmark strategies for CDAs, which perform well by high demand and supply as well as quickly converge to equilibrium price.

Q-Strategy

Q-Strategy, first proposed in Borissov and Wirström, adopts a reinforcement learning approach - Q-Learning (Watkins and Dayan 1992) with an e-greedy selection policy (Kaelbling, Littman, and Moore 1996). Using Q-Strategy, for each job-type $t_j = \{r_j, d_j, v_j\}$, an agent explores the environment with a probability of ϵ, e.g. available provider machines and the rewards from executing jobs on them. With a probability of $1 - \epsilon$ the strategy exploits the collected market information in order to generate bids more intelligently (Algorithm 1). In the following the strategy is described in more detail.

Q-Strategy is executed in two phases, *exploration* and *exploitation*. In the *exploration phase*, for each incoming job type $t_j = \{R_j, d_j, v_j\}$, the agent is generating a random bid price $b_j \in [s * v_j, v_j]$, $s \in [0, 1]$, where R_j represents technical resource requirements for job j, d_j its estimated (upper-bound) duration (Medernach 2005) and v_j its valuation. The bid $\hat{t}_j = \{R_j, d_j, b_j\}$ for job j is submitted to the market and after a match - allocation of a provider - the job is executed on the provider machine. When the job finishes, the agent calculates the outcome (reward) of its execution. The agents maintain a history of the executed jobs, generated bids and received payoffs in a so called Q-Table. The Q-Table is composed by the Q-Learning dimensions state s, action a and payoff ρ, where the state-dimension represents the job type t_j, action-dimension represents the learned bids b_j and the payoff dimension aggregates the payoffs for similar jobs and generated bids in a so called *Q-Value*, $Q(s, a)$. The payoff function is in our case a utility maximizing function $u(j)$, which is configurable for each job type. The payoff is aggregated according to the Q-Rule:
$Q(s_t, a_t) = Q(s_t, a_t) + \alpha_t[\rho_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$

In the *exploitation phase*, with a probability of $1 - \epsilon$, for each new or *similar* job type $\tilde{t}_j$, $\tilde{t}_j \approx t_j$, the agent exploits the aggregated knowledge from his Q-Table and for job j it selects the bid price $\tilde{b}_j$, which achieved the highest aggregated payoff in the past. The ex-post payoff of the job execution is again aggregated in the Q-Table according to the *Q-Rule*. Currently a job-type $\tilde{t}_j$ is defined to be *similar* to t_j, $\tilde{t}_j \approx t_j$, if $\tilde{R}_j = R_j \wedge \tilde{d}_j = d_j \wedge \tilde{v}_j = v_j$. This implies that at this stage, Q-Strategy is suitable for repeated jobs with similar job requirements, upper-bound durations and valuations.

Algorithm 1 Q-Strategy: Bid Generation Rule

```
$s \in (0, 1)$
$val_j := job.getValuation()$
**if** $\epsilon < Stochastic.random(0, 1)$ **then**
    //Explore:
    $bidPrice_j := Stochastic.random(s * val_j, val_j)$
**else**
    //Exploit:
    $state := State.getState(job)$
    $action := qLearner.bestAction$(state)
    **if** $action! = nil$ **then**
        $bidPrice_j := action.getBidPrice()$
    **else**
        //Q-Table is empty for job-type j $\in J$
        $bidPrice_j := Stochastic.random(s * val_j, val_j)$
    **end if**
**end if**
```

In case of a provider agent, the state-dimension of Q-Strategy represents the resource-type $t_i = \{R_i, d_i, v_i\}$, where R_i is the technical description of a provider resource, d_i is the duration for which it is offered to the consumers and v_i is its reserve price.

The main advantage of Q-Strategy is that agents can bid and adapt also in markets, where market information is not available or incomplete. Maintaining classes of similar jobs $\tilde{t}_j \approx t_j$ will accelerate the learning process and thus the reaction of market dynamics – fluctuating prices, new providers and computing resources with changing qualities. A common drawback of reinforcement learning algorithms like Q-Learning is that learning the optimal bid price needs "training time". In the worst case, Q-Strategy will perform worse at the beginning, but will converge towards optimal values by and by (Watkins and Dayan 1992).

Evaluation

The convergence and behavior of Q-Strategy is evaluated in an agent-based simulation with selected market mechanisms against benchmark bidding strategies. The following sections present the evaluation design and results.

Evaluation Methodology

Results in Borissov and Wirström showed that market mechanisms like DLGM improve the overall welfare in comparison to mechanisms like CDA and *first-in-first-out*. Moreover, the paper presented first results on the convergence of Q-Strategy, showing generated bids over time for selected job classes. This work extends the analysis by looking at the overall convergence and behavior of Q-Strategy against benchmark strategies in two market mechanisms – DLGM and CDA. To better understand the behavior of Q-Strategy, it was decided to vary the strategies only on the consumer side.

Table 1: Simulation Scenario

Provider	1000 machine agents
Consumer	10 consumer agents
Scenarios of Markets & Strategy Configurations	Ratios of [9:1], [8:2], [5:5] and [0:10] CDA consumer strategies [ZIP:Q-Strategy] DLGM consumer strategies [TruthTelling:Q-Strategy]
Real Job Workloads $d \in [1h, 72h]$	W1:HPC2N with 130,770 jobs W2:LPCEGEE (Medernach 2005) with 155,669 jobs W3:LLNLATLAS with 16,897 jobs
Valuations	Job valuations N(50,5); Resource reserve prices N(8,1)
Metric	Consumer utility $U_c = \sum_c \sum_j u_c(j)$;

Table 1 depicts the overall simulation scenario. In order to evaluate the behavior and convergence of the Q-Strategy, settings with 10 consumers and 1000 machine agents were defined, with one agent per machine. For each market and job workload four simulation runs were executed with a varying number of benchmarks (DLGM:TruthTelling, CDA:ZIP) and Q-Strategy agents in ratios of [9:1], [8:2], [5:5] and [0:10]. The providers behaved strategically as ZIP agents only in the CDA. In DLGM, they offered their resources without reward. Each simulation run was executed subsequently with real cluster usage workloads, taken from Feitelson's workload archive[1]. Exploring strategies and markets for interactive and quasi real-time applications (e.g. demand forecasting, information aggregation, video processing), a filter was applied to extract jobs lasting d one hour or more but less than three days. The resulting number of the workload jobs was filtered to 130,770, 155,669 and 16,897, respectively. The workloads were chosen based on the variety of run-times, numbers of used CPUs and start times. As the most common assumption (Sandholm, Lai, and Clearwater 2008), the valuations of jobs and resource reserve prices (by CDA) were drawn from a normal distribution. The results were aggregated with the consumer utility metric U_c.

In this evaluation setting it was not aimed to compare the DLGM and CDA market mechanisms themselves, but to analyze the behavior and convergence of Q-Strategy in both markets. In the case of the CDA, there was no waiting queue on the provider side for the allocated jobs. Thus the jobs spurred high competition for computing resources, whereby the execution of the job started immediately as long as there was a match. To reduce the simulation runtime in the CDA, the job durations were scaled down by a factor of 100.

[1]www.cs.huji.ac.il/labs/parallel/workload

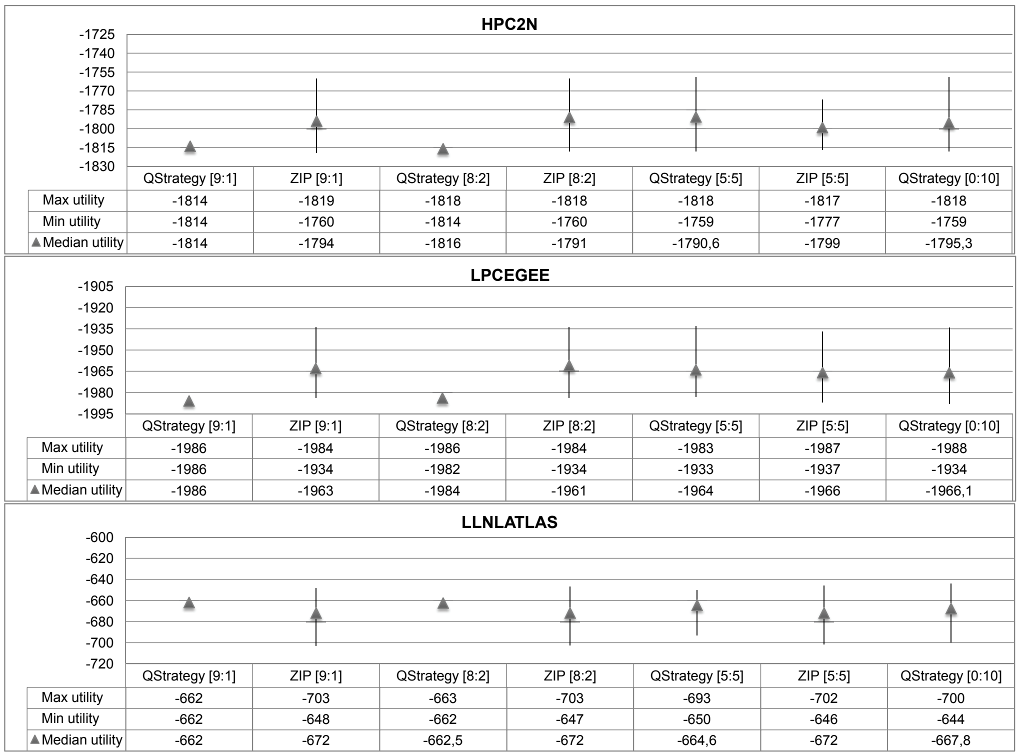
Evaluation Results

Figures 1 shows the evaluation results of the ZIP and Q-Strategy agents for the settings [9:1] through [0:10] executed in the CDA market. Based on the information gleaned from the executed jobs, for each of the 10 agent types, an aggregated ex-post utility was calculated using the utility function in Heydenreich, Müller, and Uetz, $u_j = -v_j * \hat{C}_{i,j} - \hat{\pi}_{j,i}$, which aims to minimize the weighted completion time and payments. In our scenario, a "higher importance" was assigned to the completion time C than to the payments π. According to the simulation settings, the generated values for the valuations were $v \in [23, 76]$ for the job durations in the DLGM simulations $d_{dlgm} \in [3600000, 259200000]$, and the CDA simulations $d_{cda} \in [36000, 2592000]$. During the simulations, the following payments were transacted: $\pi_{cda} \in [4, 57]$ and $\pi_{dlgm} \in [-185416, 152352]$.

The results show that in the workloads W1 and W2 in settings [9:1] and [8:2], the Q-Strategy achieved lower utilities on average than the ZIP strategy. On the other hand, given a rising number of Q-Strategy agents in setting [5:5], the agents that applied the Q-Strategy received higher utilities on average than the ZIP agents. In contrast, in the case of W3, the Q-Strategy agents generally achieved better average utilities than the ZIP agents. Overall, it was observed that with a rising number of Q-Strategy agents, competing in the CDA against ZIP agents, the Q-Strategy agents learned to maximize their configured utility better than the ZIP agents.

Figure 2 shows the evaluation results of the DLGM market, where the consumer agents applied either the TruthTelling strategy or the Q-Strategy and the providers offered their resources for free. The utilities were aggregated per agent type the same way as the CDA simulations. As observed in the DLGM simulations, the agents that applied the TruthTelling strategy achieved higher average utilities than the Q-Strategy agents. This was expected due to the fact that the Q-Strategy agents tried to maximize their utilities by underbidding from their true valuation, whereas in the DLGM scenario bidding truthfully is in the myopic best response equilibrium. With a rising number of Q-Strategy agents, their overall utilities show higher improvements compared to the utilities of the TruthTelling agents. In setting [0:10], where all agents applied the Q-Strategy, their average utilities were nearest to those of the TruthTelling agents in the other settings. One can observe that in the case of increased number of agents that apply the Q-Strategy, the aggregated agent utilities tended to converge to better values.

Table 2 shows the *standard deviation* σ of the generated bids. Based on σ it was observed a stable behavior of the Q-Strategy agents in both markets. In the case of the DLGM, the σ of TruthTelling in all three workload settings is constant, because the valuation is reported truthfully. With a ris-



HPC2N

	QStrategy [9:1]	ZIP [9:1]	QStrategy [8:2]	ZIP [8:2]	QStrategy [5:5]	ZIP [5:5]	QStrategy [0:10]
Max utility	-1814	-1819	-1818	-1818	-1818	-1817	-1818
Min utility	-1814	-1760	-1814	-1760	-1759	-1777	-1759
▲ Median utility	-1814	-1794	-1816	-1791	-1790,6	-1799	-1795,3

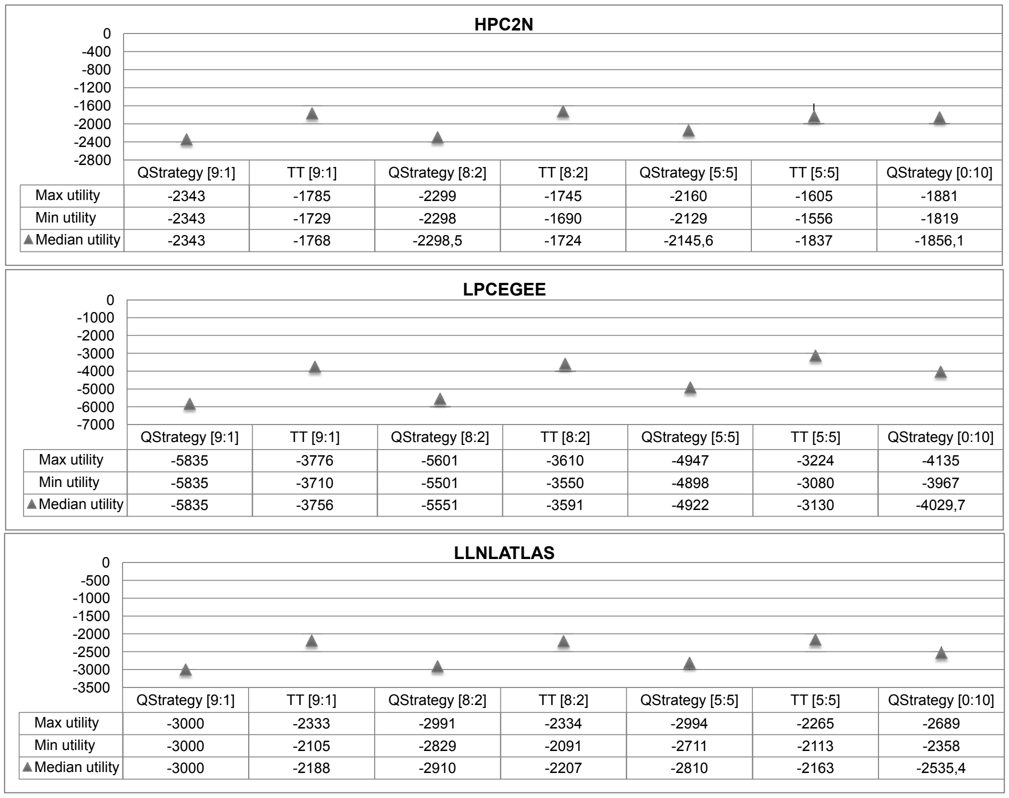
LPCEGEE

	QStrategy [9:1]	ZIP [9:1]	QStrategy [8:2]	ZIP [8:2]	QStrategy [5:5]	ZIP [5:5]	QStrategy [0:10]
Max utility	-1986	-1984	-1986	-1984	-1983	-1987	-1988
Min utility	-1986	-1934	-1982	-1934	-1933	-1937	-1934
▲ Median utility	-1986	-1963	-1984	-1961	-1964	-1966	-1966,1

LLNLATLAS

	QStrategy [9:1]	ZIP [9:1]	QStrategy [8:2]	ZIP [8:2]	QStrategy [5:5]	ZIP [5:5]	QStrategy [0:10]
Max utility	-662	-703	-663	-703	-693	-702	-700
Min utility	-662	-648	-662	-647	-650	-646	-644
▲ Median utility	-662	-672	-662,5	-672	-664,6	-672	-667,8

Figure 1: Aggregated consumer agent utilities in the CDA-Market



HPC2N

	QStrategy [9:1]	TT [9:1]	QStrategy [8:2]	TT [8:2]	QStrategy [5:5]	TT [5:5]	QStrategy [0:10]
Max utility	-2343	-1785	-2299	-1745	-2160	-1605	-1881
Min utility	-2343	-1729	-2298	-1690	-2129	-1556	-1819
▲ Median utility	-2343	-1768	-2298,5	-1724	-2145,6	-1837	-1856,1

LPCEGEE

	QStrategy [9:1]	TT [9:1]	QStrategy [8:2]	TT [8:2]	QStrategy [5:5]	TT [5:5]	QStrategy [0:10]
Max utility	-5835	-3776	-5601	-3610	-4947	-3224	-4135
Min utility	-5835	-3710	-5501	-3550	-4898	-3080	-3967
▲ Median utility	-5835	-3756	-5551	-3591	-4922	-3130	-4029,7

LLNLATLAS

	QStrategy [9:1]	TT [9:1]	QStrategy [8:2]	TT [8:2]	QStrategy [5:5]	TT [5:5]	QStrategy [0:10]
Max utility	-3000	-2333	-2991	-2334	-2994	-2265	-2689
Min utility	-3000	-2105	-2829	-2091	-2711	-2113	-2358
▲ Median utility	-3000	-2188	-2910	-2207	-2810	-2163	-2535,4

Figure 2: Aggregated consumer agent utilities in the DLGM-Market

Table 2: Standard deviation σ of generated bid prices

	[9 : 1]	[8 : 2]	[5 : 5]	[0 : 10]
DLGM_W_1_QS	8.39	8.37	8.41	8.41
DLGM_W_1_TT	5.00	5.00	5.00	-
CDA_W_1_QS	9.28	9.37	9.00	8.53
CDA_W_1_ZIP	2.57	2.69	3.60	-
DLGM_W_2_QS	8.38	8.45	8.35	8.34
DLGM_W_2_TT	5.00	5.00	5.00	-
CDA_W_2_QS	9.02	9.55	8.97	8.83
CDA_W_2_ZIP	3.72	3.90	7.04	-
DLGM_W_3_QS	8.31	8.20	8.30	8.39
DLGM_W_3_TT	4.98	4.98	4.98	-
CDA_W_3_QS	9.52	9.25	9.10	8.41
CDA_W_3_ZIP	1.82	1.92	2.23	-

ing number of Q-Strategy agents in settings [8:2] and [5:5], the σ of the ZIP agents rose in all settings, which reflects the increased competition between both agent strategies.

Conclusion

This paper proposed a novel bidding strategy, designed to bid, converge and adapt in many markets. The proposed strategy was applied on the consumer side and compared to benchmark strategies in two on-line market mechanisms. The evaluation shows that with a rising number of Q-Strategy agents, the strategy outperforms the ZIP agents in the CDA, in DLGM however the utilities converge towards those of the Truth-Telling agents.

Future work will elaborate on suitable business models and utility functions for providers. Identifying utility functions for Cloud-based services will allow automation of the provisioning and bidding processes on the provider side. Hence, the evaluation scenario will be extended to provider agents that apply the Q-Strategy with the identified utility functions for the particular resources. It is aimed to extend the DLGM mechanism and introduce strategic behavior on the provider side. This will change the payment scheme and will probably unseat Truth-Telling as the myopic best response, at least for the providers. Another point of research are the dynamic adjustments of the Q-Strategy α and ϵ parameters in relation to different markets and preferences.

References

Bapna, R.; Das, S.; Garfinkel, R.; and Stallaert, J. 2007. A Market Design for Grid Computing. *INFORMS Journal on Computing*.

Borissov, N., and Wirström, N. 2008. Q-Strategy: A bidding strategy for market-based allocation of grid services. In *OTM Conferences (1)*, 744–761.

Cliff, D. 1997. Minimal-intelligence agents for bargaining behaviors in market-based environments. *Technical Report, Hewlett Packard Labs*.

Das, R.; Hanson, J.; Kephart, J.; and Tesauro, G. 2001. Agent-human interactions in the continuous double auction. *Artificial Intelligence*.

Grosu, D., and Das, A. 2006. Auctioning resources in Grids: model and protocols. *Concurrency and Computation: Practice and Experience* 18(15).

Heydenreich, B.; Müller, R.; and Uetz, M. 2006. Decentralization and mechanism design for online machine scheduling. *Algorithm Theory SWAT 2006, DOI - 10.100711785293_15* 136–147.

Kaelbling, L.; Littman, M.; and Moore, A. 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* 4:237–285.

Lai, K.; Rasmusson, L.; Adar, E.; Zhang, L.; and Huberman, B. 2005. Tycoon: An implementation of a distributed, market-based resource allocation system. *Multiagent and Grid Systems* 1(3):169–182.

Li, J., and Yahyapour, R. 2006. Learning-based negotiation strategies for grid scheduling. *International Symposium on Cluster Computing and the Grid (CCGRID2006)* 576–583.

Medernach, E. 2005. Workload analysis of a cluster in a grid environment. *LNCS* 3834:36–61.

Myerson, R., and Satterthwaite, M. 1983. Efficient mechanisms for bilateral trading. *Journal of Economic Theory* 29(2):265–281.

Nassif, L.; Nogueira, J.; and de Andrade, F. 2007. Distributed resource selection in grid using decision theory. *7th IEEE Symposium on Cluster Computing and the Grid*.

Pardoe, D., and Stone, P. 2007. An autonomous agent for supply chain management. In Adomavicius, G., and Gupta, A., eds., *Handbooks in Information Systems Series: Business Computing*. Elsevier.

Parkes, D.; Singh, S.; and Yanovsky, D. 2004. Approximately efficient online mechanism design. *Proc. 18th Annual Conf. on Neural Information Processing Systems*.

Phelps, S. 2007. *Evolutionary mechanism design*. Ph.D. Dissertation, University of Liverpool, UK.

Reeves, D.; Wellman, M.; MacKie-Mason, J.; and Osepayshvili, A. 2005. Exploring bidding strategies for market-based scheduling. *Decision Support Systems* 39(1).

Sandholm, T.; Lai, K.; and Clearwater, S. 2008. Admission Control in a Computational Market. 277–286.

Satterthwaite, M., and Williams, S. 1989. Bilateral trade with the sealed bid k-double auction: Existence and efficiency. *Journal of Economic Theory*.

Sherstov, A., and Stone, P. 2005. Three automated stock-trading agents: A comparative study. *LNCS* 3435:173.

Vytelingum, P.; Cliff, D.; and Jennings, N. 2008. Strategic bidding in continuous double auctions. *Artificial Intelligence* 172(14):1700–1729.

Watkins, C., and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3):279–292.

Wellman, M.; Greenwald, A.; and Stone, P. 2007. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*. MIT Press.

Wolski, R.; Plank, J.; Brevik, J.; and Bryan, T. 2001. Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications*.