A Dynamic Pricing and Bidding Strategy for Autonomous Agents in Grids

B. Pourebrahimi[1], K. Bertels[1], S. Vassiliadis[1], Luc Onana Alima[2]
[1]{behnaz, koen, stamatis} @ce.et.tudelft.nl , [2]luc.onana@umh.ac.be

[1]Computer Engineering Laboratory, EEMCS, Delft University of Technology, The Netherlands
[2]Service des Systémes Distribués, Institut d′Informatique, Université de Mons-Hainaut, Belgium

Abstract. In this paper, we propose a dynamic pricing strategy which is used for a market-based resource allocation mechanism in a local Grid. We implement an agent based Grid economy in which the decision-making process regarding task and resource allocation is distributed across all users and resource owners. The Continuous Double Auction is used as the platform for matchmaking where consumer and producer meet. In this paper, we analyze the parameter regime of this pricing mechanism considering different network conditions. Our experiments described in the paper show that using the pricing parameters, the consumers and producers agents can decide the price to influence the way they contribute resources to the Grid or complete the jobs for which they need resources. These agents are individually capable of changing the degree of their task usage and resource contribution to the Grid.

Key words: dynamic pricing strategy, active/lazy agents, aggressive/conservative biddig, system efficiency, network condition

1 Introduction

In recent years, the intensive computational applications are becoming more and more popular. In case of lack of computational resources in such applications, instead of consuming an extra budget to buy centralized resources, one solution is to use existing computing resources over the network that otherwise lie idle. These networks of distributed and shared resources are known as Grids.

The problem we are looking at is resource allocation and task distribution in a Grid-based environment, where the resources are heterogeneous in nature, they are owned by various individuals or organizations with different objectives and they have dynamically varying loads and availability. It could be deployed to any organization having a LAN with any number of computers, in which the tasks can be processed on any node that has idle resources. Managing resources in such dynamic and heterogeneous environment is a challenging task and need to be smart, adaptable to the changes in environment and user requirements. Conventional resource management schemes are often based on relatively static models with a centralized controller. They might work well where resources are known in advance, however this fails to work in heterogeneous and dynamic systems where jobs need to be executed by computing resources whose availability is difficult to predict. Market-based approaches can provide adaptability and self-organization in such dynamic networks. As in these approaches, individual decision making is achieved through a large amount of decentralized information which is condensed into a single, simple entity, namely the price.

In this paper, we look at a particular pricing strategy and study its parameter's regime given different Grid conditions. In this approach, the consumers and producers are considered as autonomous agents that make their own decisions according to their capabilities and their local knowledge. Continuous Double-Auction is used as the platform for implementing our model. In this model, the consumers and producers of resources put their requests or offers attached with a price into the market as bids/asks. Buy orders (requests) and sell orders (offers) may be submitted at anytime during the trading period. No global and single equilibrium price is computed in this strategy, rather at any time there are open requests and offers that match or are compatible in terms of price and requirements (e.g. quantity of resources), a trade is executed immediately.

The main contribution of the paper is to identify how an individual agent can take into account its own task loads and available resources as well as the Grid overall condition. We show using this pricing strategy how the agents can adapt to a dynamic network condition where the distribution of tasks and availability of resources may change at any time. In addition, based on this strategy each agent can decide on the contribution of its resources or demanding for its tasks at any time as its available resources or its workloads change. In our pricing strategy, the price proposed by consumer and producer agents changes based on the supply and demand in the system. Producers generate aggressive bids by raising the price when the demand is high and conservative bids by lowering the price when demand is low. On the other hand, consumers with a conservative and aggressive strategy respectively lower the price when supply is high and raise the price when supply is low.

The paper is structured as follows: In section 2, we give an overview of related works in economic and auction based resource allocation. Section 3 discusses the system architecture and pricing model is introduced in section 4. The experiments are presented in section 5 considering different network conditions and node's activities. Finally, conclusion and future work are discussed in section 6.

2 Related Work

Economic models have been used widely in resource allocation algorithms [12] [1]. Price-based economic models are classified into two main categories: Auctions and Commodity Markets. In Commodity Markets, allocations are done based on reaching an equilibrium price where demand equals the supply. For instance, Wolski et al [13] have used the commodity market approach to allocate two types of resources (CPU and disk storage) in Grid. The auction protocols are either one-to-many or many-to-many. In one-to-many auctions one agent initiates an auction and a number of other agents can make a bid. The English auction, Dutch auction, first-price auction, second-price (Vickrey auction) belong to this category. Popcorn [7] and Spawn [10] are examples that use these types of auctions. In many-to-many auctions, several agents initiate an auction and several other agents can bid in the auction. The double auction is the most widely used auction protocol for many-to-many auctions. In these auctions, buyers and sellers are treated symmetrically with buyers submitting requests and sellers submitting offers. There are two types of double auctions, continuous double auction (CDA) and periodic double auction. Continuous Double Auction matches buyers and sellers immediately on detection of compatible bids. A periodic version of the double auction instead collects bids over a specified interval of time, then clears the market at the expiration of the bidding interval [14]. JaWS [6], [9] and [8] are examples which use double auction model. The Proportional Share Protocol (PSP) is a similar protocol to Continuous Double Auction, as both use a centralized scheduling algorithm. In this approach, the amount of resources allocated to a task depends on its price bid in relation to the sum of price bids of all tasks executing on that server. Proportional Share Protocol is proposed for the scheduling of tasks in computational clusters [11].

In the literature, we can find several studies on auction based resource allocation. Gomoluch et al [3] investigate under which circumstances market-based resource allocation by CDA and Proportional Share Protocol, outperforms a conventional round-robin approach. It is concluded that CDA will perform best in most cases comparing two other approaches. [5] compares three different Double-Auction protocols from both resource's and user's perspective in terms of resource utilization and resource profit and spent budget. It concludes that CDA protocol performs best from both perspectives. In [4], three types of auction-based resource allocation protocols are investigated; First-Price Auction, Vickrey Auction and Double Auction. Resource utilization, resource profit and user payment is measured as the parameters for comparing these protocols. Their results show First-Price Auction is better from resource's perspective while Vickrey Auction is better from user's perspective. And Double Auction favors both resources and users. The work in [2] analyzes the different auction models in terms of communication demand for resource allocation in Grid computing environments. The investigation is done on First-Price sealed, English, Dutch and CDA. Their experiments show that English auction present higher communication requirements while CDA presents least demand of communications.

What distinguishes our work from the others is using a dynamic pricing strategy in which the consumer and producer agents using aggressive or conservative bids and with different level of activity can adapt to the current condition. The economic approach used in this work is not novel but the main novelty is applying the dynamic pricing algorithm to the Grid. Our experiments are performed in a local Grid with different network conditions regarding distribution of tasks and resources. We investigate the pricing behavior of the consumer and producer agents and study the influence of this behavior on the system efficiency in terms

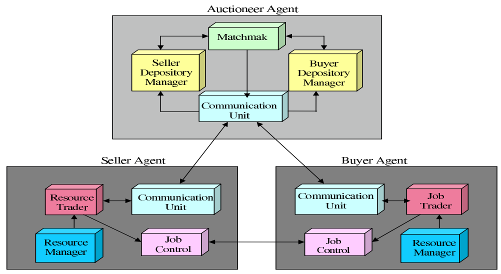


Fig. 1. System agents components.

of task/resource utilization and average matching time. Eagerness of the agents for contribution to the Grid has been also applied in this strategy. This work is part of the research in which we aim to investigate different economic models and pricing strategies to find out an efficient approach to obtain self-organization.

3 System Architecture

The system is composed of three entities: Consumer (buyer), Producer (seller) and Auctioneer. The market works in the following simple manner: the buyers and sellers announce their desire to buy or sell processing power to the market. The auctioneer finds the matches between buyers and sellers by matching offers (starting with lowest price and moving up) with requests (starting with highest price and moving down). When a task query arrives at the market place, the protocol searches all available resource offers and returns the best match which satisfies the task's constraints (such as resource quantity, time frame and price). If no match is found, the task query object is stored in a queue. The queries are kept in the queue till the time to live (TTL) has expired or a match has been found. When a resource becomes available and several tasks are waiting, the one with the highest price bid is processed first.
The system components can be summarized as follows(see Figure 1):

- **Consumer(Buyer)/Producer(Seller) Agent**: There is one consumer/ producer agent per node. A consumer/producer agent controls the process of buying/selling resources by estimating the execution time of the job or availability of the resource (*Resource Manager*), calculating the price (*Job/Resource Trader*) and generating and submitting a request/offer for corresponding job/resource (*Communication Unit*). Submitting/accepting the job to the matched consumer/producer (*Job Controller*) is also the task of these agents.
- **Auctioneer Agent**: The auctioneer agent controls the market using a double auction protocol. Based on this protocol, every consumer and producer sends its request and offer to the auctioneer. Auctioneer inserts each received request or offer in its sorted depositories (*Buyer/Seller Depository Manager*).

The requests are sorted from high price to low price and the offers are sorted from low price to high price. A request is matched with an offer if the resource offered by the producer meets the consumer requirements regarding the quantity, time and price (*Matchmaker*).

4 Pricing Algorithm

In a price based system, the resources are priced based on the demand, supply, and the wealth in the economic system. The prices vary with the demand and supply of the resources. In each market, the objective of a seller is to maximize its earning as much as possible and the objective of a buyer is to spend money as less as possible. Based on these objectives, the strategy of producers of resources is to raise the price when the demand for associated resource is high and lower the price when the demand is low. On the other hand, the strategy of consumers of the resources is to lower the price when supply is high and raise the price when the supply is low. Based on these strategies, we define a seller and buyer pricing functions as follows:
Sellers and buyers start with an initial price and update it over time:

$$p(t) = p(t-1) + \Delta p \tag{1}$$

The value of Δp determines whether the price is increasing or decreasing. To change the price according to the demand or supply in the system, Δp is defined based on the past resource or task utilization on this particular seller/buyer. Δp for seller and buyer is calculated as below:
for seller:

$$\Delta p = \alpha(u(t) - u_{thR})p(t-1) \tag{2}$$

for buyer:

$$\Delta p = \beta(u_{thT} - u(t))p(t-1) \tag{3}$$

where α and β are the coefficients that control the rate of price changing. u_{thT}/u_{thR} are the threshold values below which, the task/resource utilization should not go. u_{thT} and u_{thR} could be interpreted as the degree of the agent activity. If the activeness is very low, it implies that the agent is satisfied with a low usage of its resources or a low completion rate of its tasks. If it is high, the agent is more demanding for itself by imposing higher satisfaction thresholds. $u(t)$ is resource/task utilization at the individual node which is defined as:

$$u(t) = \sum_{i=t_0}^{t} x(i) / \sum_{i=t_0}^{t} N(i) \tag{4}$$

Where $\sum_{i=t_0}^{t} x(i)$ is the total numbers of sold/purchased resources in the time period $[t_0, t]$ and $\sum_{i=t_0}^{t} N(t)$ is the total numbers of offered/requested resources in the time period $[t_0, t]$.

Consumers and producers submit their bid/ask price along with the quantity of requested or offered resources to the auctioneer. Auctioneer finds the matched pairs and the trade between each pair is made at the average of the corresponding consumer's and producer's prices. This price is called **transaction price**.

5 Experiments in a Local Grid

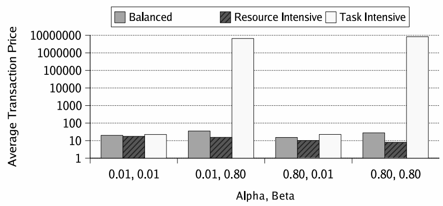
To perform our experiments, we set up a Grid like environment based on a local LAN in which our application test-bed is developed using J2EE and Enterprise Java Beans. A JBOSS application server is used to implement the auctioneer. Java Message Service (JMS) is used for the communication between clients and auctioneer. MySQL server is used as a database server to store the results.

CPU time is considered as the resource in our system. Whenever a consumer needs additional CPU time for running a job, it sends a request to the auctioneer and whenever a producer has some idle CPU time, it sends an offer. In the sent messages, in addition to resource quantity and price, a Time To Live (TTL) is also included. It represents the time during which the request or offer is valid or available. Each node creates a number of requests or offers during the simulation time. For each request the resource requirements are expressed in terms of job execution time that is generated randomly in a specific range. An offer which is the time during which the CPU is idle, is also generated randomly.

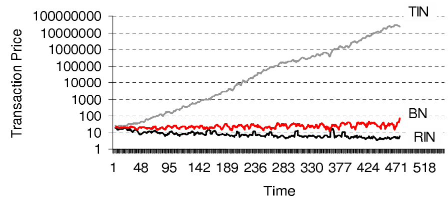
The simulation is done in an environment with 50 nodes with various CPU speeds. In these experiments consumer and producer agents start from a random price between 10 and 30 Grid Dollars and update their price according to supply and demand in the system. During the simulation, each node creates 50 tasks/resources. In each time, based on the node workloads and idle resources, a node either creates a task and activates a consumer agent or creates a resource and activates a producer agent. Unbalanced number of created consumer and producer agents can lead to a task or resource intensive network.

5.1 Adapting to Different Network Conditions

In this section, we perform the experiment on three different network conditions: **the balanced network** which is the type of the network where there is more or less an equal number of tasks and resources, **the task intensive network** where there are more tasks than resources and **the resource intensive network** where there are more resources than tasks. In this experiment we want to show how the agents decide on the price changing when updating their prices in each network condition. The behavior of the price is discussed in these networks and efficiency of the system is measured in the terms of task and resource utilization and average time of finding matches. In this experiment, we study the impact of α and β parameters in different network conditions. These parameters, as already discussed in section 4, determine the rate at which the price is changing. We consider the value of $u_{thT} = u_{thR} = 0.9$ for all consumers and producer agents, which means the agents who have task or resources are contributing with the same degree of activity in the grid(these parameters are studied in section 5.2).



(a)



(b)

Fig. 2. Transaction Price **(a)** Average Transaction Price with various values of α and β in different network conditions (Logarithmic Y-scale); **(b)** Transaction price evolution in task intensive network (TIN), resource intensive network (RIN) and a balanced network (BN) with the values of $\alpha = 0.8$ and $\beta = 0.8$ (Logarithmic Y-scale)

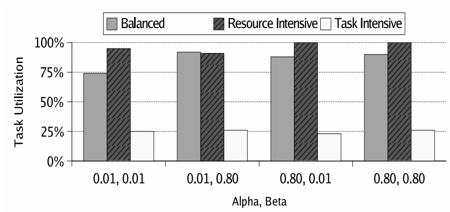
Transaction Price The average transaction price is studied in three network conditions with various values of α and β. As seen from figure 2(a), the lowest prices are observed in resource intensive networks. This type of network is similar to what is called a buyer market. In a buyer market, there are more sellers than buyers and low prices result from this supply excess. The average transaction price has the highest values in the task intensive networks comparing to the other networks. In task intensive network which has more buyers than sellers, high prices result from the excess of demand over supply. In such network which is similar to a seller market, buyers enter into competition with each other in order to obtain scarce resources. In a balanced network as the supply equals the demand, no very high or low prices are expected.

To see how the prices evolve during the time in each network condition, transaction price evolution is shown in figure 2(b) in the three networks with the values of $\alpha = 0.8$ and $\beta = 0.8$. Upward, downward and stable trend of transaction price curve in task intensive, resource intensive and balanced networks respectively is consequence of what we already discussed about these markets.

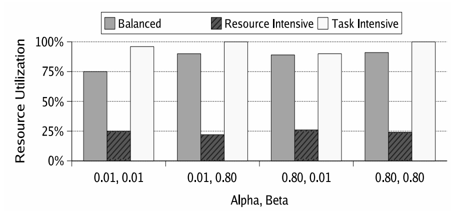
As seen from figure 2, the average transaction price in task intensive networks when the value of β is high, goes to the very big values. This is expected as in such network the resources are scarce and consumers increase their prices in a competitive way with high rate which leads to a high transaction price. However the high prices should be prevented by applying some restriction, such as budget.

System Efficiency In this experiment system efficiency is measured in three network conditions with varying the values of α and β.

- **Task/Resource Utilization**
  Task/resource utilization is defined as the ratio of allocated tasks/resources to all sent resource requests/offers. As figures 3(a) and 3(b) show, task and resource utilization in a balanced network for all values of α and β is around 90%, except when α and β are very low. In case of $\alpha = 0.01$ and $\beta = 0.01$,



(a)

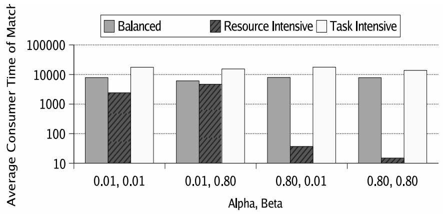


(b)

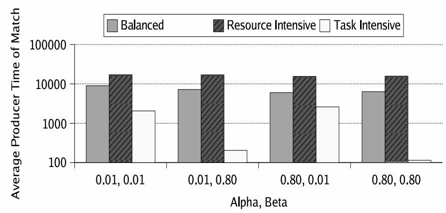
Fig. 3. Task/Resource Utilization with various values of α and β in different network conditions

task and resource utilization are around 75%. As with low values of α and β, producer and consumer update their prices in a slower rate that leads to a lower utilization of tasks and resources. In a resource intensive network, a global observation is apparent from the figures as the task completion is close to 100% in most of the cases and only around 25% of the available resources are used. This is expected as we are now looking at the Grid condition where there are abundant resources. In such network, as figures show, the highest task/resource utilization is obtained when $\alpha = 0.8$ ($\beta = 0.01$ or $\beta = 0.8$). As in case of high competition between producers, if producers update their prices in a higher rate, they will be more successful in selling their resources. A global view on the task/resource utilization in a task intensive network determines a usage of almost 100% for resources and 25% for allocated tasks. These results are consequence of higher number of the tasks than the resources. The highest task/resource utilization in this type of network is obtained when $\beta = 0.8$ ($\alpha = 0.01$ or $\alpha = 0.8$). Higher rate in updating the consumer price helps the competitive consumers to find more matches which leads to more task/resource utilization.

- **Average Time of Finding Matches**
  Figures 4(a) and 4(b) show the average time which it takes for consumers and producers to find their required matches. In a resource intensive network, a global observation is that the average time for producers to find a task for their available resources is at least 4 times higher than for consumers. This is evidently the consequence of the fact that there are more resources available than tasks to perform. In a task intensive network, the average time of finding match for consumers is at least 6 times higher than for producers. As in this kind of network there are more tasks than resources. However, the lowest consumer matching time in a resource intensive network is obtained when $\alpha = 0.8$ and the lowest producer matching time in a task intensive network is obtained when $\beta = 0.8$, which are corresponding to the highest task/resource utilization in respective networks.
  In a balanced network, we don't see much difference in the average matching time of consumers and producers. However the matching time is higher for both consumers and producers when $\alpha = 0.01$ and $\beta = 0.01$. This is because



(a)



(b)

Fig. 4. Average time of finding match with various values of α and β in different network conditions (Logarithmic Y-scale)

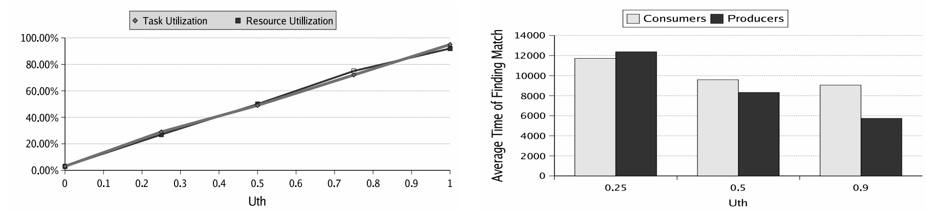
of slower rate of updating producer and consumer price that results a longer time for finding proper matches.

An overall study of system efficiency shows that the highest task/resource utilization and lowest matching time in different network conditions is provided when α and β have bigger values. The question is how the agents recognize the current network condition? The answer is the agents can realize the condition by the way their price is evolving. For instance, when the price is increasing consumer agent knows that resources become scare. Therefore it has to adapt his bidding strategy to become more aggressive by increasing the value of β. On the other hand when the price is decreasing, it shows the demand for the resources is low then producer agent bidding strategy has to be converted to a conservative strategy by increasing the value of α.

5.2 Adaptation at Node Level

In previous experiments presented in section 5.1, we studied the behavior of the price and efficiency of the system in different network conditions regarding distribution of tasks and resources generated in the network. We showed how the agents adapt to the current condition of the network. In current experiment, we want to show how the agents can adapt based on current condition of their own tasks and resources. For instance if a node has more resources than tasks, it should generate an active producer agent and a lazy consumer agent. we need a way to represent this information and to incorporate it into the agent's behavior. As already discussed (section 4), u_{thT}/u_{thR} could be interpreted as the degree of laziness of the agents in the Grid. If it is low, it implies that the agent is contributing with a low usage of its resources or is demanding a low completion rate of its tasks. If it is high, the agent is contributing to the Grid with offering more resources or is demanding more resources from the Grid. To study the impact of u_{thT} and u_{thR}, we consider the fixed value of 0.8 for α and β.

Impact of u_{thT}/u_{thR} on system efficiency: In the first set of experiment, we study the impact of varying u_{thT} and u_{thR} on the system efficiency. We



(a) **(b)**

Fig. 5. Task/Resource utilization (a) and average time of finding match (b) with different values of u_{th} in a Balanced network (u_{th}=u_{thT}=u_{thR})

consider the same value of utilization threshold for the consumers and producers ($u_{thT} = u_{thR} = u_{th}$) and experiment in a balanced network.

- Task/Resource Utilization: We measure the task and resource utilization considering different values for u_{th}. The result of these experiments shows that task and resource utilization is linearly proportional to this threshold value (see figure 5(a)). Agents with low value of u_{th} represent lazy agents and agent with high value of u_{th} show the agents which are active in Grid. Seen from the figure 5(a), as u_{th} increases, the Grid utilizes more from the agent's tasks or agent's resources.
- Average Time of Finding Matches: In the same experiments, the average time which it takes to find a match is measured for both producers and consumers. With increasing the degree of the activeness (u_{th}), the agents become more active in the Grid, so the time of finding a match for them is decreasing. As Figure 5(b) shows with increasing the value of u_{th}, producers and consumers spend less time to find their required matches.

Lazy/active agents: To show in a heterogeneous Grid, how the consumer and producer agents can decide individually on their task usage and resource contribution to the Grid by modifying u_{thT} and u_{thR} parameters, we present a new experiment. Assume in a Grid some nodes have heavy workloads and need extra resources. These kind of nodes would prefer to complete their tasks rather than offering their resources. So these nodes are lazy producers but active consumers. The rest of the nodes are more willing to contribute their resources as they have idle resources or low workloads. Which means this nodes are active producers but lazy consumers. We have considered 40 nodes in the Grid. The values of $u_{thR} = 0.9$ and $u_{thT} = 0.25$ are set for the nodes those have more resources than tasks and the values of $u_{thR} = 0.25$ and $u_{thT} = 0.9$ are set for half of the nodes having more tasks than resources. We studied the task and resource utilization of the individual nodes from these two categories. Figure 6 shows resource and task utilization of two typical nodes of each category. Node A is an instance of the first category of nodes with low workload that have more idle

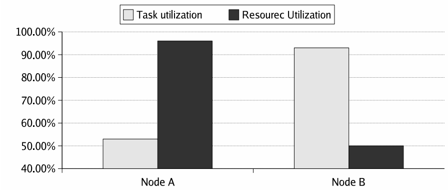


Fig. 6. Node A: $u_{thR} = 0.9$ and $u_{thT} = 0.25$; Node B: $u_{thR} = 0.25$ and $u_{thT} = 0.9$

resources and Node B shows an instance of the nodes with high workloads. As it is observed from the figure, the nodes of type A contribute more as producers than consumers and the Grid utilizes more resources (93%) from this group than tasks (56%). On the other side, more tasks are utilized from the nodes of type B than resources, which is 96% for tasks and 59% for resources. As these nodes, contribute to the Grid more as consumers than producers.

6 Conclusion and Future Work

In this paper, a dynamic pricing and bidding strategy is introduced where the consumer and producer agents determine the price of their tasks/resources to contribute to the Grid. In this strategy, the pricing function is adaptive to changing supply and demand of resources where the adaptation is achieved by increasing or decreasing the price when the supply or demand is low. For instance, if the demand for the resources is high, the agents apply an aggressive bidding strategy by increasing the price so as to discourage users from demanding this resource and to maintain equilibrium of supply and demand of resources. We study parameter regime of the pricing equations in three network conditions. There are four parameters that can be manipulated by the consumer and producer agents: α, β, u_{thT} and u_{thR}. The α and β are parameters using which the rate of changing ask and bid price is defined. The u_{thT} and u_{thR} determine the degree of activity of the agents in the Grid.

Our experiments show that a resource intensive network is more influenced by α parameter while a task intensive network is more influenced by β. In a resource intensive network, agents decrease their ask prices using a conservative bidding strategy by increasing the α-value. In a task intensive network, agents speed up their bid prices by increasing the β-value using an aggressive bidding strategy to make more use of the Grid. In balanced network both α and β parameters have the same affect. Besides, producers and consumers can change the degree of their activity in the Grid using u_{thR} and u_{thT} parameters. The producers and consumers can decide by how much they are contributing to the Grid considering their capabilities and their workloads. They become more/less lazy in the Grid by decreasing/increasing these values.

As the results show, in each condition higher values of α or β provide higher resource/task utilization and lower matching time. But in the other side, with higher values of α and β, higher fluctuation and higher valuses in the prices is observed which is not the case in real situation. In future work we aim to implement our model considering a given budget for each node which provides an upper boundary for prices. Different auction models with differnet pricing strategies are to be studied in future work.

References

1. R. Buyya, D. Abramson, Jonathan Giddy, and Heinz Stockinger. Economic models for resource management and scheduling in grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15):1507–1542, 2002.
2. M. Dias de Assuncao and R. Buyya. An evaluation of communication demand of auction protocols in grid environments. Technical report, Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2006.
3. J. Gomoluch and M. Schroeder. Market-based resource allocation for grid computing: A model and simulation, 2003.
4. D. Grosu and A. Das. Auction-based resource allocation protocols in grids. In *In Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 20–27, November 2004.
5. U. Kant and D. Grosu. Double auction protocols for resource allocation in grids. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, pages 366–371, 2005.
6. S. Lalis and A. Karipidis. Jaws: An open market-based framework for distributed computing over the internet. In *GRID*, pages 36–46, 2000.
7. N. Nisan, S. London, O. Regev, and N. Camiel. Globally distributed computation over the internet - the popcorn project. In *ICDCS '98: Proceedings of the The 18th International Conference on Distributed Computing Systems*, page 592. IEEE Computer Society, 1998.
8. E. Ogston and S. Vassiliadis. A peer-to-peer agent auction. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems Part I*, pages 151–159, July 2002.
9. M. Preist C., Van Tol. Adaptive agents in a persistent shout double auction. In *Proc. of 1st International Conference on the Internet Computing and Economics*, pages 11–17, 1998.
10. Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and W. Scott Stornetta. Spawn: A distributed computational economy. *Software Engineering*, 18(2):103–117, 1992.
11. Carl A. Waldspurger and William E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Operating Systems Design and Implementation*, pages 1–11, 1994.
12. R. Wolski, J. Brevik, J. S. Plank, and T. Bryan. Grid resource allocation and control using computational economies. In F. Berman, G. Fox, and A. Hey, editors, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley & Sons, 2003.
13. Rich Wolski, James Plank, John Brevik, and Todd Bryan. G-commerce: Market formulations controlling resource allocation on the computational grid. In *In Proc. International parallel and Distributed Processing Symposium (IPDPS)*, April 2001.
14. P. Wurman, W. Walsh, and M. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.