8-2010

Decentralized Resource Allocation and Scheduling via Walrasian Auctions with Negotiable Agents

Huaxing CHEN
Singapore Management University, hxchen@smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

DOI: https://doi.org/10.1109/WI-IAT.2010.113

Published in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology WI-IAT 2010: 31 August-3 September 2010, Toronto: Proceedings. Los Alamitos: IEEE Computer Society. pp. 356-360.
http://doi.ieeecomputersociety.org/10.1109/WI-IAT.2010.113

Decentralized Resource Allocation and Scheduling via Walrasian Auctions with Negotiable Agents

HuaXing CHEN Hoong Chuin LAU
School of Information Systems, Singapore Management University
{hxchen,hclau}@smu.edu.sg

Extended Abstract

1. Introduction

Auctions are well-studied mechanisms to solve decentralized resource allocation problems. In situations where complementarity exists among resources, combinatorial auctions may be utilized. Unfortunately, combinatorial auctions often incur hefty computational requirements, in that the underlying winner determination problem is known to be NP-hard. Although a number of efficient implementations are proposed (e.g., see Sandholm [11]), combinatorial auctions are still not very computationally efficient.

To avoid solving winner determination problems, assuming that clearing is already feasible (i.e. aggregate resource demands could be met by supply), an alternative method for market clearing is the General Equilibrium or Walrasian model, first proposed by Walras [14]. Under the GE model, the auctioneer would not clear the auction and finalize the allocation. Instead, the auctioneer would adjust the prevailing resource prices based on the current aggregate demands induced by bidders against the supply, and announce the adjusted prices back to all agents. Reacting on the adjusted resource prices, the bidding agents generate the next set of bids and this process is iterated until either the supply and demand matches (i.e. general equilibrium is reached), or that the computational budget is exceeded. In a conventional Walrasian auction, bidders are assumed to be price takers meaning that they would view market prices as exogenous and not something that would change because of their own actions. General equilibrium theory states that the market will converge to an equilibrium state under very restrictive conditions (such as gross substitutability). For arbitrary resource allocation problems however, the existence of such an equilibrium and effective means to find an equilibrium state is often a challenging research question.

In this paper, we are concerned with solving decentralized resource allocation and scheduling problems; and assuming feasible solutions exist, we ask how we might achieve convergence and good quality solutions under a tight computational budget.

Motivated by recent works on adaptive bidding strategies in a winner determination context (Sui and Leung [13]) and the analysis of bidding strategies for simultaneous ascending auctions in separate markets (Wellman et al. [16]), we propose an approach that departs from the standard Walrasian auction, in that the bidders are not merely price takers reacting to price signals; in addition, their bidding strategies can also be influenced by a negotiation mechanism, where they may adopt different bidding strategies from their respective strategy sets from one iteration to the next during the auction process. In this sense, we say that the agents are "negotiable". We will show how a mechanism embedded in the auctioneer side detects a trigger state and persuades bidders to change their bidding strategies during the auction process, as well as how bidders respond to the solicitation of the auctioneer while simultaneously adjusting their bids based on resource prices, in such a way that a better system wide performance could be achieved.

In this paper, we deal specifically with the resource allocation and scheduling problem associated with container terminals, but believe that our approach can be customized to solve other problems of similar structures. The key result of this paper is to show, perhaps counter-intuitively (by experiments), that a better system-wide performance is not necessarily always achieved with each bidder adopting its best bidding strategy throughout the auction process. Instead, under various market conditions, convergence and better quality solutions can be achieved when agents take a negotiable stance in allowing the next-best bidding strategies to be adopted while still reacting to price signals.

2. Literature review

In recent years, there are a number of applications of auctions in solving distributed resource allocation and scheduling problems in a variety of domains. Kutanoglu and Wu [4] demonstrated the link between combinatorial auction and Lagrangean relaxation and use combinatorial auction to solve small-scale job shop scheduling problems. Other works include Wellman et al. [15], Goldberg et al. [3], Attanasio et al [1], Lau et al. [6], Liu and Zhao [7], Stokely et al. [12].

On strategic bidding, Wellman et al. [16] investigated a straightforward bidding policy and its variants to indicate that the efficacy of particular strategies depends critically on preferences and strategies of other agents, and that the strategy space is far too complex to yield to general game-theoretic analysis. Sui and Leung [13] proposed an adaptive bidding strategy in a first price seal-bid combinatorial auction where the bidders can adjust the profit margin constantly accordingly to the bidding history.

Our work has leveraged on the existing research findings of Lau et al. [6] by exploring the bidding strategies via an auction mechanism which solicits bidders' change of bidding strategies in the hope that system wide performance will be enhanced.

3. Notations and problem formulation

The problem we deal with in this paper can be described as a multi-agent flowshop problem with special sequencing constraints between jobs in a given sequence, and is described as follows. There is a pool of limited renewable resource types (Prime Movers or PM, Yard Blocks and Yard Buckets) and there are multiple units of resources for each resource type. There is a set of job agents, each endowed with a list of jobs to be serviced in a sequence, where each job is performed as a series of operations and needs a combination of a single unit of each resource type for each operation. Each job corresponds to either loading a container from a yard bucket within a yard block to a quay crane (QC), or vice versa (called a discharge job). For simplicity, we take QCs to be the proxy for job agents (which is the same approach taken by Lau et al. [6]). The goal is to allocate resource timeslots to the job agents such that the sum of makespans of the job agents is minimized.

To solve this problem by a Walrasian auction mechanism, the auctioneer will iteratively adjust the prices for each resource type in each time slot of the entire planning horizon; and in each iteration, the auction will begin with the announcement of the price vector to all bidders simultaneously. In a conventional scheme, each bidder then generates a bid (i.e. a demand for each resource type in each time slot over the entire planning horizon) in response to the resource price vector. This bid is generated by a bid-generation algorithm which seeks to find the bid that minimizes the respective agent total cost function, which is defined as the sum of the makespan cost and the total cost of the bidded resource slots at the prevailing prices. Note that there is a tradeoff between these two cost components, since more resources will yield a shorter makespan but incur a higher resource cost, and hence the bid-generation algorithm is one that seeks the optimal balance. A solution is called feasible if there are no resource conflicts.

In this paper, we depart from the conventional auction scheme, and allow each bidder to have a finite and discrete set of bidding strategies; each strategy is associated with a bid generation algorithm. Some bid generation algorithms may perform better than others in terms of the agent total objective function. (The reason for having inferior bid generation algorithms will become clear later, but suffice here to say that they are required to break resource conflicts among agents, seen as compromises in negotiation terminology.)

As noted earlier, there are 3 types of resources namely, PM, Yard Block and Bucket. A bidder may request a number of PMs in each time period in order to service its jobs. In addition, it also needs to bid for the utilization of Yard Blocks and Buckets. (Interested readers familiar with this problem may like to note that the yard block resource acts as the proxy for the utilization of yard cranes, which we do not explicit bid for in this paper, unlike Lau et al. [6] for the reason that yard crane slots are consumed by multiple bidders simultaneously). Instead in our approach, a helper method performs the scheduling of yard cranes based on the yard block bids, and this schedule will provide a feedback to the auctioneer for the update of the prices of the yard block resources.

We denote k to be the resource type index (PM, Block, Bucket) and the entire planning horizon is divided into T discrete time periods (aka time slots).
The notations used in this paper are given as follows:

- X^r_{ikt} denotes the demand quantity bidded by bidder i for resource k in time period t during iteration r.
- C^r_{kt} denotes the total supply (capacity) for resource k, in time period t during iteration r.
- D^r_{kt} denotes the aggregate demand (of all agents) for resource k, in time period t during iteration r.
- ω^r_i denotes the bid generation strategy used by bidder i during iteration r.

4. Solution approach

In our proposed approach, the strategy space for each bidder is denoted by Ω comprising of a number of bid generation strategies. Each strategy ω is associated with an algorithm which enables the bidder to generate bids and the corresponding *total cost* is denoted $U(\omega)$. Note that we drop the agent index for notational simplicity. The algorithms are designed in a way that one algorithm is cascaded with another so that the top algorithm always yields the best *total cost*. We term a strategy ω' as *smart* and another strategy ω'' as *less smart* if $U(\omega') < U(\omega'')$. Let ω^* denote the smartest bid generation strategy in Ω (where $U(\omega^*) < U(\omega)$ for any ω except itself). Note that a smart bid generation strategy tends to exploit cheap resources thereby potentially leading to resource contention with other agents assuming the other agents are also using smart strategies. Hence if all bidders adopt their respective smartest bid generation strategy throughout the auction process, it could be possible that those bidders will contend for resources with one another iteratively with possibly no feasible solution generated at the end of the auction process. Our intuition is that a less smart bid generation strategy may foster more cooperation and less competition, thereby leading to conflict resolution. Again it does not come without a cost. In the other extreme scenario where all bidders decide to use their less smart bid generation strategies throughout the auction process, we may obtain an undesirably large number of solutions, but none of them are good in quality as the resources utilization may not be high.

Hence, the challenge is to be able to derive a **negotiation mechanism** embedded into the auctioneer logic that aims to persuade some bidders to concede and modify their bid generation strategies to inferior ones whenever necessary, so that a better system-wide solution can be obtained. We assume therefore that all agents are cooperative, i.e. willing to do so with the proper incentives set in place. Our proposed negotiation approach is based on a similar reward scheme proposed by Lau et al. [5] and Ramchurn et al. [9] and is outlined as follows. All bidders start with their respective optimal strategy ω^*; the negotiation mechanism in the auctioneer side will solicit the change to an inferior strategy ω based on a certain criteria (details to be discussed later) and compensate those bidders who are willing to concede by an amount of credits which, if accumulated, could be used to reject future solicitations by surrendering some credits.

In the interest of space, we will skip over the discussion of bid generation and price adjustment for our problem. Interested readers may refer to [6] for details. In the following, we focus on presenting the key contribution of this work – the embedded negotiation mechanism. All bidders are assumed to start with the respective smartest bid generation strategy. A negotiation mechanism will be enabled if resource conflicts exist during the auction that cannot be resolved in the first λ (constant) number of iterations. Once the negotiation mechanism has been enabled, it will be constantly monitoring the ratio of the aggregate resource demands to the supply (i.e. D^{r}_{kt}/C^{r}_{kt}) – a large value that exceeds a threshold $f_{trigger}$ suggests that the gap is caused by aggressive bidding behavior by some agents on resources for certain time periods. We measure bidder *aggressiveness* by the relative ratio of the amount of resource of type k at time t requested by a bidder i to the average demand (i.e. $X^{r}_{ikt}/Ave\_D^{r}_{kt}$). Note that this seemingly simplistic definition of *aggressiveness* may overkill those bidders who have persistent preference for resources in a particular time period (this persistent behavior will occur if some jobs can only be performed in a certain time period). However, such a stringent time constraint on the jobs largely reduces the effort required for the search algorithm as these jobs can only be performed in a certain time period. Even with a less smart bidding strategy, those bidders will still be able to retain desirable resource bids.

$f_{\omega'}$ (≥1) specifies the strategic-specific threshold to classify a bidder with its current strategy ω' as an aggressive bidder. Hence, a large $f_{\omega'}$ value implies that fewer bidders will be selected and requested to use their next-best bidding strategy. Note that $f_{\omega'}$ is ∞ if ω' is the least smart strategy in the strategy space, so that the bidder with the least smart strategy is never regarded as an aggressive bidder. The Negotiation Unit is responsible for identifying aggressive bidders. A contract $O_{(\omega',\omega'')}$ comprises a request for conceding from current strategy ω' and adopting the next-best strategy ω'' together with an associated reward $M_{(\omega',\omega'')}$. This reward will be offered in subsequent iterations to sustain agent's interest to continue to bid with strategy ω''. The number of credits given in each subsequent iteration is computed from the estimated performance gap between bidding strategies ω' and ω'', normalized by the average performance gap of all such (ω', ω'') pairs.

On the bidder side, the credits C^{r}_{i} owned by a bidder are subject to a depreciation over time (similar to the concept of time value of money), which serves to discourage a bidder to use its credits at the later iterations in order to stabilize the overall bidding behavior. Bidders may choose to either accept the contract and receive $M_{(\omega',\omega'')}$ credits in each subsequent iteration by adopting the next-best strategy *ω''* or reject the contract at the expense of surrendering a number of $J_{(\omega',\omega'')}$ credits. In our approach, we set $J_{(\omega',\omega'')} < M_{(\omega',\omega'')}$ due to currency depreciation over time. If a bidder has enough credits, it may even choose to break the contract at any iteration to go back to the previous smart bidding strategy *ω'* by surrendering $E_{(\omega',\omega'')}$ credits, and in our approach, we set $E_{(\omega',\omega'')} > M_{(\omega',\omega'')}$ since a number of iterations would have elapsed before the bidder takes this course of action. The contracts can be cascaded on a single bidder in the sense that the rewards will be cumulatively added. Hence, a bidder on a contract $O_{(\omega',\omega'')}$ can be offered another contract $O_{(\omega'',\omega''')}$ if it is identified again as an aggressive bidder in a subsequent iteration, in which case the credits given to the bidder becomes $M_{(\omega',\omega'')} + M_{(\omega'',\omega''')}$ if the new contract $O_{(\omega'',\omega''')}$ is accepted. In our approach, a bidder with multiple contracts should always break the latest contract first if there are sufficient credits.

If at least one bidder changes its bid generation strategy, then the auctioneer will backtrack with a probability *R* by broadcasting the resource prices of current iteration to all the bidders *without* applying price adjustment. *R* is function of the frequency of demand/supply violations across all time periods so far, such that a low frequency leads a higher possibility to backtrack. The whole cycle (from submission of bids by bidders to the announcement of resource price by auctioneer) will be repeated iteratively until the stopping criteria are reached.

Consider bidder's current strategy to be *ω''*.

- Action 1: Bidder is not identified as an aggressive bidder so no new contract is offered, and bidder keeps the current strategy.
- Action 2: Bidder is identified as an aggressive bidder and a contract $O_{(\omega'',\omega''')}$ is offered; bidder surrenders credits $J_{(\omega'',\omega''')}$ to reject the contract and continue with existing strategy.
- Action 3: Bidder is identified as an aggressive bidder and a contract $O_{(\omega'',\omega''')}$ is offered; however bidder possesses fewer number of credits than $J_{(\omega'',\omega'')}$. Hence, bidder accepts the contract and move to next-best strategy.
- Action 4: Bidder have more than $E_{(\omega',\omega'')}$ credits, by surrendering $E_{(\omega',\omega'')}$ credits, it breaks the contract $O_{(\omega',\omega'')}$ and moves back to previous strategy *ω'*.

5. Experimental results

For simplicity, we consider only **two** bid generation strategies for each bidder (Simulated Annealing or SA, which is the smart strategy, and Relax-and-Repair or R&R, the next-best or inferior one).

Recall that each agent bids for the right to use various resource types in different time periods. The length of each time period is set to half an hour in our experiment. To model the scheduling of jobs more precisely, we further divide each time period into 60 *time units*, each with length of 0.5 minutes. We model the QC, PM, YC processing times as well as the YC gantry time with real data obtained from a container terminal operator. The experiment is conducted with a variety of test cases: under normal resource supply conditions as well as those under tight resource conditions by scaling down resource capacity from those normal test cases. For the negotiation mechanism, we set the value of λ to 5, $f_{trigger}$ to 1.05, f_{SA} to 1, $f_{R\&R}$ to ∞. All experiments were performed on four Pentium-4 3.0GHz processors each with 2 GB RAM.

First, we present results on system wide performance. We compare our approach with three others – a centralized exact approach (adapted from Pritsker et al. [8]) solved using the CPLEX 9.0 C++ callable library, as well as two standard auction approaches where agents adopt specific bidding strategies. In the tables below, the respective approach is labeled with the bidding strategies used (being R&R or SA). Our approach is labeled as "Negotiated".

As the centralized approach is computationally intensive, we first experimented on test cases of small problem size (4 bidders, each with 20 jobs). Our results show that for small problem size, in all 5 test scenarios, our approach on average gives the best system-wide performance (i.e. sum of agent makespans) among various auction approaches. The exact mathematical model provides the optimal solution, which on average is about 20% better than results obtained using auctions (the loss of efficiency is to be expected, as shown in other papers in the literature). However, the computational time taken is prohibitively large and has big variance among the test cases. Even for a small instance, the computation time on average can take up to 7 hours for one test run, whereas all auction approaches on average take less than 20 seconds to complete (where each agent runs on one of the four PCs).

For larger problems, we compare the performance of the 3 auction approaches. To ensure fair comparison of performance against run time, we let each auction process run for 200 iterations. These test cases have up to 30 bidders, each of which has a number of load and discharge jobs ranging from 90 and 120. From the experiment results, we may observe our approach yields the best system-wide performance on average within a reasonable computational budget, compared to the other standard auction approaches that adopted a single bidding strategy throughout the auction.

What is perhaps more interesting is the study on the convergence of various auction schemes on two specific large tight resource cases (with 20% reduction of the normal resource capacity). Experimental results show that the SA approach produces *no* feasible solution in both test cases, whereas the R&R and Negotiated approaches are still able to give solutions in both test cases. The best solution and number of feasible solutions of both the R&R and Negotiated approaches are summarized in Tables 1 and 2 below. For Test Case 2, the feasible solution is given by the Negotiated approach at iteration 53 and by R&R at iteration 72. The feasible solutions for Test Case 1 are traced against the iterations in Figure 1. Note that each point in the diagram indicates the time when a feasible solution is found. We observe that in both test cases, our approach finds a feasible solution in an earlier iteration compared with the R&R approach, and the quality of the feasible solutions are better than those obtained by R&R, even though R&R yields a greater number of feasible solutions.

Table 1. System-wide performance (makespan) comparison of R&R and SA

	R&R	Negotiated
Test Case 1	279.5	274.5
Test Case 2	240.5	223.9

Note: all makespans measured in terms of number of minutes

Table 2. Number of feasible solutions obtained by various approaches

	R&R	SA	Negotiated
Test Case 1	16	0	6
Test Case 2	1	0	1

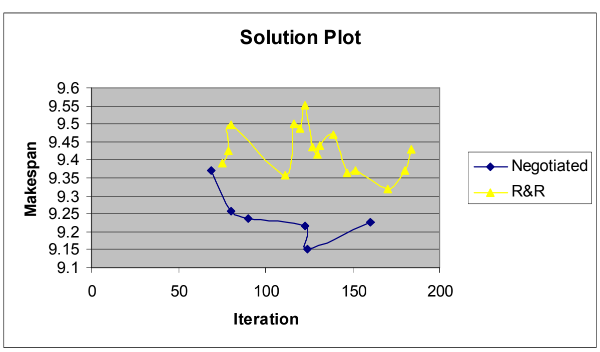

Figure 1. Solution plot for test case 1

6. Conclusion

It is perhaps apt to end the paper with a philosophic note from Robert Axelrod's remarkable book on the Iterated Prisoner's Dilemma – that "in a non-zero sum setting, it does not always pay to be so clever… you benefit from the other player's cooperation. The trick is to encourage that cooperation."[1]

References

[1] A. Attanasio, G. Ghiani, L. Grandinetti, and F. Guerriero, Auction algorithms for decentralized parallel machine scheduling. *Parallel Computing*. Vol. 32, 701-709, 2006.

[2] M. L. Fisher. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 27:1, 1-18, 1981.

[3] D. Goldberg, V. Cicirello, and M.B. Dias, Task allocation using a market-based planning mechanism. *Proceedings AAMAS*, 996-997, 2003.

[4] E. Kutanoglu and S. D. Wu. On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. *IIE Transactions* 31(9): 813-826, 1999.

[5] H. C. Lau et al. Dynamic multi-linked negotiations in multi-echelon production scheduling networks. *Proceedings* IAT, 2006.

[6] H. C. Lau et al. Multi-Period Combinatorial Auction mechanism for distributed resource allocation and scheduling. *Proceedings* IAT, 2007.

[7] X. Liu and L. Zhao. A Method based on Iterative Combinatorial Auction Mechanism for Resource Allocation in Grid Multi-agent systems. *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics, 2009*.

[8] A. Pritsker, L. Watters, and Ph. Wolfe. Multiproject scheduling with limited resources: a zero-one programming approach. *Management Science,* 16(1): 93-108, 1969.

[9] D.Sarvapali Ramchurn et al. Negotiation using rewards. *Artifical intelligence* Vol.171:805-837, 2007.

[10]R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review* 35(2):183-238, 1993.

[11]T. Sandholm. Algorithm for optimal winner determination in combinatorial auction. Artificial Intelligence, 135(1–2):1–54, 2002.

[12]M. Stokely et al. Using a Market Economy to Provision Compute Resources Across Planet-wide Clusters. *Proceedings of the International In Parallel and Distributed Processing Symposium,* 2009.

[13]X. Sui and H. F. Leung. An Adaptive Bidding Strategy in Multi-Round Combinatorial Auctions For Resource Allocation. *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence*, 2008.

[14]L. Walras. Elements of Pure Economics. 1874. English translation by William Jeff˜e, 1954.

[15]M. P. Wellman et al. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35: 271-303, 2001.

[16]M. P. Wellman et al. Exploring Bidding Strategies for Market-Based Scheduling. *Proceedings of the 4th ACM conference on Electronic commerce,* pp. 115-124, 2003.

[1] R. Axelrod, The Evolution of Cooperation (Revised Edition). Basic Books, 2006, p. 123.