11-2007

# Multi-Period Combinatorial Auction Mechanism for Distributed Resource Allocation and Scheduling

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

Shih-Fen CHENG
*Singapore Management University*, sfcheng@smu.edu.sg

Thin Yin LEONG
*Singapore Management University*, tyleong@smu.edu.sg

Jong Han PARK

Zhengyi ZHAO
*Singapore Management University*

**DOI:** https://doi.org/10.1109/IAT.2007.41

## Citation

LAU, Hoong Chuin; CHENG, Shih-Fen; LEONG, Thin Yin; PARK, Jong Han; and ZHAO, Zhengyi. Multi-Period Combinatorial Auction Mechanism for Distributed Resource Allocation and Scheduling. (2007). *IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2007: IAT '07: 2-5 November 2007, Fremont, CA: Proceedings*. 407-411. Research Collection School Of Information Systems.
**Available at:** https://ink.library.smu.edu.sg/sis_research/208

# Multi-Period Combinatorial Auction Mechanism for Distributed Resource Allocation and Scheduling

Hoong Chuin Lau    Shih Fen Cheng    Thin Yin Leong
Jong Han Park    Zhengyi Zhao
*School of Information Systems*
*Singapore Management University*
*{hclau, sfcheng, tyleong, jhpark, zyzhao}@smu.edu.sg*

**Abstract**

*We consider the problem of resource allocation and scheduling where information and decisions are decentralized, and our goal is to propose a market mechanism that allows resources from a central resource pool to be allocated to distributed decision makers (agents) that seek to optimize their respective scheduling goals. We propose a generic combinatorial auction mechanism that allows agents to competitively bid for the resources needed in a multi-period setting, regardless of the respective scheduling problem faced by the agent, and show how agents can design optimal bidding strategies to respond to price adjustment strategies from the auctioneer. We apply our approach to handle real-time large-scale dynamic resource coordination in a mega-scale container terminal.*

## 1. Introduction

In a classical scheduling problem (such as the job-shop scheduling problem), the goal is to schedule a set of jobs on a set of machines (resources), subject to resource capacity and other scheduling-specific constraints. In this paper, we consider the problem of resource allocation and scheduling where information and optimization decisions are decentralized, and our goal is to propose a coordination mechanism that allows resources from a central resource pool to be allocated to distributed decision makers that seek to optimize their respective optimization goals. More precisely, we are concerned with the following decentralized scheduling scenario:

a. there is a central pool of limited renewable resources that comprises multiple units of resources for each machine type;
b. there are multiple selfish agents, each having a job list and is responsible to service its job list by solving its respective scheduling problem; without loss of generality, we will assume that each agent is interested in minimizing a *performance function* (such as makespan, tardiness or combinations thereof).

One example of such problem can be found in the allocation of resources for loading and discharging vessels in a container terminal. Each job, comprising a container that needs to be moved from/to a vessel, requires three different types of machines – a Quay Crane (QC), a Prime Mover (PM) and a Yard Crane (YC). In practice, each QC is endowed with a job list and it has to acquire multiple units of different resources (PMs and YCs) over time to complete its given job list. All PMs have a common processing time, and so do YCs and QCs. The goal is to decide an optimal resource allocation for time-overlapping job lists in a distributed manner (since each QC works independently without considering other QCs' job lists).

To coordinate the resource usage by all selfish QC agents over time, we propose to design a market mechanism that allows agents to act as price takers and acquire the rights to utilize resources from the pool to solve their respective scheduling problems. In other words, each agent is expected to pay for the utilization of resources from the central pool defined by a *price function*. Hence, in our approach, each agent seeks to minimize its *total cost function*, which is the sum of the performance and price functions.

Solving the above decentralized scheduling problem with market mechanisms can provide several advantages (as described in Wellman et al., 2001):

a. Market mechanisms are naturally decentralized. Agents compute bids independently and submit them to the market mechanism.
b. Communication overhead for the market mechanisms is low. The only information required to be exchanged between markets and agents are bids and prices.

Previous works on market-based decentralized scheduling [Wellman et al. (2001), Thomas et al. (2002)] have provided encouraging empirical results. As a result, over the past decade, significant amount of efforts have been devoted to the building of a general framework for using market-based approaches in various resource allocation problems (including scheduling).

The primary contribution of this paper is to propose a market-based approach that can be used in solving large-scale decentralized scheduling problems where each agent is given a pre-determined job-list. The market mechanism we propose is a combinatorial auction. While there have been similar proposals in the literature (see Section 2 below), most of them are targeted at solving small and specific scheduling problems, and computational efficiency is usually not a primary concern.

## 2. Literature Review

Distributed scheduling has attracted much attention in recent years. One line of research is market-based distributed problem solving. In the market-based paradigm, auctions are introduced to coordinate resource usage among selfish agents who seek to maximize their respective objectives. Wellman et al (2001) introduced auction mechanisms which used prices derived through distributed bidding protocols to determine schedules. The existence of equilibrium prices for some general classes of scheduling problems was investigated. Not only ascending auction but also combinatorial auction's bidding behaviors and protocols were investigated. Dewan and Joshi (2002) used Lagrangean relaxation to decompose and solve the distributed scheduling in dynamic job shop environment. The jobs constructed their bids from subproblems obtained by Lagrangean relaxation which relaxes the capacity constraint. A central auctioneer was used for coordinating among job agents by adjusting prices for all the resources. Attanasio et al. (2006) applied a similar approach for decentralized parallel machine scheduling problem. In these studies, the price adjustment process was based on tatonnement process which was originally proposed by Walras (1954) and improved by many other studies (e.g. Joyce (1984), Cheng and Wellman (1998)). The tatonnement process resolves resource conflicts by updating the price based on excess demand iteratively. Several price adjustment processes has been proposed in different studies. Fisher (1985) proposed price adjustment process based on subgradient search method. Joyce (1984) set the step parameter based on the level of difference between demand and supply. He set higher value on step parameter when the difference between demand and supply is large and vice versa. Dewan and Joshi (2002) and Attanasio et al. (2006) used Fisher (1985)'s approach.

In general scheduling problems, a job agent may demand a combination of time slots from multiple machines to process its operations. Hence, we must consider auctions in which agents bid for multiple items that have inter-dependent valuations. This property of scheduling problem motivates us to adopt a combinatorial auction mechanism that allows bidders to submit a combination of items with a single bid. Kutanoglu and Wu (1999) noted the equivalence of the combinatorial auction approach to the Lagrangean relaxation and use the shadow prices as transfer pricing to link decomposed subproblems. They also used tatonnement process to resolve conflicts among agents. Jung and Kim (2006) considered the problem of load scheduling for multiple quay cranes in port container terminals.

## 3. Notations and Problem Formulation

We assume that a central pool of resources is owned by a central server (who is the auctioneer). Each job list is represented by a bidder agent who bids for resources to service all the jobs in its job list. Clearly, since each agent chooses to minimize total cost, it will be encouraged to bid for more resources when the prices are low and vice versa. We therefore seek to achieve a conflict-free resource allocation through a market mechanism of demand, supply and price adjustments.

Note that as opposed to conventional auctions, our auction mechanism seeks to achieve a minimum cost resource conflict-free resolution rather than revenue maximization. Our approach is based on the tatonnement process which resolves resource conflicts by iteratively updating the prices of resources based on excess demand. This approach has been applied similarly in different scheduling contexts such as [Dewan and Joshi 2002, Attanasio et al. 2006]. In our approach, the auctioneer iteratively updates the prices of resources starting from initial price. Based on current price of items, each bidder needs to find the best bid in order to maximize its own utility function. The auctioneer then evaluates bids from all bidders and updates the prices in response to excess demand. The new prices will be used in the next round of bidding, and this process repeats until a conflict-free allocation is found.

In our problem, there are $K$ resources, and the entire time horizon is divided into $T$ discrete time periods. Since each job list may span across multiple time periods, this gives rise to a multi-period combinatorial auction problem. A bid is hence composed of a combination of resources over multiple periods. Note that the bid may specify varying units of resources for different periods.

- Let $X^r_{ikt}$ denote the demand quantity bidded by agent $i$ for resource $k$ in time period $t$ during iteration $r$.
- Let $P^r_{kt}$ denote the price for resource $k$ in time period $t$ during iteration $r$.
- Let $C_k$ denote the total supply (capacity) for resource $k$ across all time periods.
- Let $D^r_{kt}$ denote the aggregate demand (of all agents) for resource $k$ in time period $t$ in iteration $r$.

## 4. Solution Approach

There are two key components that need to be designed: (a) bid generation strategies for individual agents, and (b) price adjustment strategy for the auctioneer. The bid generation strategies are based on the prices, while the prices are computed based on the submitted bids. In such a system where the decisions of two individual components depend on each other, what constitutes a solution is the pair of bids and prices such that the new bids computed from the current prices are identical to the current bids, and vice versa. In economics, this solution is called *general equilibrium* since it achieves equilibrium (supply equals demand) for all resources simultaneously. An intuitive way to approach such solution is through an iterative process, where in each iteration every agent generates a combinatorial bid using the bid generation strategy and upon receiving all the bids, the auctioneer adjusts the prices according to the excess demand (non-zero excess demand implies that the equilibrium is not reached yet). The process is repeated until we reach the general equilibrium (or the stopping criterion is satisfied). In the reminder of the section, we will discuss the detail of these two critical components.

### 4.1. Bid Generation Strategy

The bidder's objective is to maximize its utility. In our context, utility is the net value from job completions minus the payment for additional resources sold by the auctioneer. It can be shown that maximizing this utility function is equivalent to minimizing the total cost function. For job shop scheduling for instance, this is the sum of total makespan cost, tardiness penalty cost, and resource cost. The total resource usage cost is simply the sum of current price of each resource in each period multiplied by the amount requested for that period.

Hence, each bidder agent is confronted with an optimization problem that is concerned with the tradeoff between makespan/tardiness and resource usage. This optimization problem can be formulated as follows: for each agent $i$ in any iteration $r$, given the current price defined by $[P^r_{kt}]$ and $i$'s job list, find a bid $[X^r_{ikt}]$ that minimizes agent $i$'s total cost.

Although this optimization is smaller compared to the centralized optimization problem, it is still an extremely challenging problem that cannot be solved efficiently. Therefore, we propose the following 2-step heuristic bid generation strategy which we term as Relax and Repair:

Step 1. Relax the multi-period problem to a single period problem, i.e. decide on an initial common resource level for each resource across all periods. By eliminating the time dimension, the problem can be reduced to a classical resource allocation problem.

Step 2. (Repair) Improve the quality of the solution by local search – by increasing or decreasing the resource level in each period. Note that for each period, we need to consider either an increase or decrease in resource level but not both. Hence, the neighborhood of each local search iteration is $O(T)$. Note that while the resulting bid might not be optimal, our experimental study (see Section 5) shows that the performance difference is reasonably small, with considerable savings in execution time.

### 4.2. Price Adjustment Strategy

Within each iteration, the auctioneer collects bids from all agents and updates prices in order to resolve resource conflicts. A proper design of the price adjustment strategy is essential for fast convergence, since the tatonnement process is known to converge slowly especially in a market with large number of resources that violates competitive assumptions [Cheng & Wellman (1998)].

The price adjustment from one iteration to the next is essentially based on excess demand. For example, from iteration $r$ to $r+1$, the updated price for a resource $k$ at period $t$ is defined by $P^{r+1}_{kt} = P^r_{kt} + S\,(D^r_{kt} - C_k)/C_k$. If the step parameter $S$ is constant between iterations, we have a non-adaptive tatonnement. In contrast, adaptive tatonnement means that larger price adjustment is permitted in early iterations to allow quick move to promising region, and as the step size diminishes, the auctioneer could fine-tune the prices to find the equilibrium. More involved discussion on the price adjustment can be seen in [Fisher 1985, Joyce 1984].

From the economic standpoint, the price adjustment process will proceed until the general equilibrium is reached. However, this may not be possible from the resource allocation point of view. Based on the stopping rules defined in [Rockafellar 1993], we propose the following stopping criteria:

1. *Equilibrium solution*: All the following conditions must be satisfied:
   - The minimum number of iterations has been performed.
   - $X$ number of feasible solutions (i.e. conflict-free allocations where supplies meet demands for all resources in all time periods) have been found. Notice that in these solutions, supplies may exceed demands.
   - The change in the performance function or price vector does not exceed some prescribed thresholds in the last $Y$ (predefined constant) number of iterations subsequent to finding a feasible solution. This condition essentially allows the auction process to continue after the feasible solution has been found, with the hope to improve the performance function further, but capped at a limit when the improvement becomes trivially small.
2. *Best solution among feasible solutions*: If an equilibrium solution cannot be found after the maximum number of iterations, stop and return the best feasible solution.
3. *Manually adjusted feasible solution*: If no feasible solution has been found, repair the solution from the last iteration by manually resolving resource conflicts.

## 5. Application to Vessel Loading/Unloading

In this section, we discuss how we apply our approach to solve the resource coordination problem for vessel loading and unloading described in Section 1. Recall that each agent represents a QC with an endowed job list, and it wants to complete the jobs on the list with minimal cost, which is defined as the sum of makespan and total tardiness penalty. Essentially, this entails solving a job scheduling problem where each job comprises a sequence of 3 tasks executed contiguously: for a load job, the tasks require the use of 1 YC, 1 PM and the designated QC; and for a discharge job, the sequence is reversed.

It is assumed that each agent is assigned a fixed number of PM and YC initially (denoted $N_0$ and $M_0$ respectively). Unassigned PM and YC are managed by an auctioneer, and agents can acquire additional PM and YC from the auctioneer through bidding.

## 5.1. Bid Generation Strategy

To generate a bid, each QC agent solves a scheduling problem that, given the current unit prices of PM and YC, seeks the cost-optimal resource vector:

$$(\overline{N},\overline{M}) = \{(N_1, M_1), (N_2, M_2), ...(N_T, M_T)\},$$

where $N_t$ and $M_t$ respectively denote the number of PM and YC required for period $t$.

The bid generation strategy is a relax-and-repair approach (as presented in Section 4):

Step 1. Relax the multi-period problem to a single period problem. Search for the common resource level ($N^*$, $M^*$) that minimizes total cost (i.e. among different resource combinations ranging from the baseload ($N_0$, $M_0$) to the upper bound ($N_{\max}$, $M_{\max}$)[1]; if we set ($N^*$, $M^*$) as the resource capacity across all periods, we will obtain a schedule with minimal total cost). This search procedure is simply hill-climbing in the landscape bordered by ($N_0$, $M_0$) to ($N_{\max}$, $M_{\max}$). Note that a by-product of this hill-climbing search is a makespan matrix that contains the makespan value for each ($N$, $M$) resource combination, regardless of resource prices. Each entry in the matrix need only to be computed once, and looked up subsequently when prices change between iterations.

Step 2. Improve the quality of the solution by local search – by changing the resource level in each period.

As noted earlier, this bid generation process does not optimally solve the bidder's problem. In fact, for the problem instances we studied in this section, our bid generation heuristics on average generates bids that are worse than the optimal bids (in the sense of the objective function values) by about 14%. However, while the heuristics on average takes less than 2 seconds to finish, solving for the optimal bids needs nearly one hour. This huge difference in the execution time is the primary motivation for choosing this bid generation heuristics over the optimal bid generation procedure.

## 5.2. Price Adjustment Strategies

In this paper, we propose to adopt Fisher's subgradient search method [Fisher 1985] for our problem, which has proven effective in practice (Kutanoglu and Wu, 1999).

Given the price in iteration $r$ as $p^r_{kt}$, price adjustment is to calculate the new price regarding to the resource demand and resource capacity.

[1] One such bound can be obtained by assuming infinite resources, which can be computed via a greedy algorithm.

$$s^r = \alpha\left(\frac{1}{\sum_{k=1}^{K}\sqrt{\sum_{t=1}^{T}(D_{kt}-C_{kt})^2 \Big/ T}}\right)\cdot\left(\frac{\sum_k\sum_t p^r_{kt}C_{kt}}{\sum_k\sum_t C_{kt}}\right)$$

$$p^{r+1}_{kt} = \max\{0, p^r_{kt} + s^r(D_{kt}-C_{kt})\},$$

where α is constant over iterations and is in [0 ,2].

Compared to the original formula of Fisher (1985), we removed difference between upper bound and lower bound in a numerator term. Furthermore, we multiplied with the average resource price term, which has been shown to produce better convergence. The reason why it would accelerate convergence is intuitive: when the average price is high, the price step will be higher, thus allows the auctioneer to balance excess demand faster, and when the price is low, smaller step will let auctioneer to adjust price in a finer manner.

## 5.3. Experiments

### 5.3.1. Experiment Setup

In this section, we evaluate the performance of our proposed combinatorial auction approach. All experiments are performed on a Pentium-4 1.5GHz processor with 1 GB RAM, and implemented in Matlab.

Recall that each agent can bid for the rights to use certain resource in *time periods* it desires. The length of each time period is set to one hour in our experiment. To model the scheduling of jobs more precisely, we further divide each time period into 12 *time units*, each with length of 5 minutes. In our experiment, we assume that the processing times of QC and YC are 1 and 2 time-units respectively for all agents. The processing time for PM follows a uniform distribution where both upper and lower bounds are uniform random variables within [7, 15] and [15, 23] respectively. These numbers are based on real operations scenarios obtained from an industry partner. In all instances we study, there are 4 QC agents, each of them initially endowed with 1 PM and 1 YC. The due time to complete all the jobs is 5 hours and the tardiness penalty cost is 300 per hour. The initial unit price for PM and YC are set as 30 and 50 per hour respectively. The number of resource units in the common pool is identical across all time periods. The price adjustment parameter, α, is set to be 1.5. For the stopping criteria, we set X to 6, Y to 3, and δ to 10. These values are proved effective through many experiments.

### 5.3.2. Decentralized v.s. Centralized Approaches

To determine the loss in optimality due to decentralization, we compare our market-based approach against the state-of-the-art integer programming (IP) solver (we use CPLEX in our experiments). For each instance we solve, a corresponding IP model is also created and solved by CPLEX. The difference in performance is not what we focus on here, since solving the problem with an IP model ignores the decentralization consideration, instead, what we are interested in from these comparisons, is to obtain an upper bound on the performance loss due to the decentralized formulation.

The IP formulation of our problem is an extension of the classical scheduling formation proposed by Pritsker et al. (1969). In the interest of space, the model is not presented here. Interested readers can refer to Pritsker et al. (1969) for detail.

Although our approach is quite scalable to large examples, the sizes of IP models in those cases are quite intractable. Therefore, in order to make meaningful comparisons, we have to limit the size of our test instances. The test instances used here all have 40 discharge jobs and 40 load jobs, equally divided among four QC agents (in other words, each agent is given a list of 10 discharge jobs and 10 load jobs). The number of available PM and YC is 24 and 16 respectively. We compare the performance of our solution approach (loaded with different configurations) against the one obtained by CPLEX after running for one hour (we have to limit the execution time to one hour otherwise CPLEX would run for days before termination).

To estimate the performance of these two solution methodologies realistically, we have carefully crafted 11 test cases that reflect typical container terminal operations. In these 11 test cases, the solutions obtained by CPLEX on average perform 22% better than the ones obtained by the market-based approach (in terms of the schedule quality, which is defined as the sum of tardiness cost and makespan cost). However, the market-based approach takes only roughly 9 minutes to finish. Please refer to Figure 1 for the performance comparison in all instances.

For larger instances we study, e.g., the cases with 200 discharge jobs and 200 load jobs, solving the IP model is intractable. Even the linear relaxation of the problem cannot be solved within 48 hours. However, the market-based approach we used can solve each of these instances in 80 minutes.

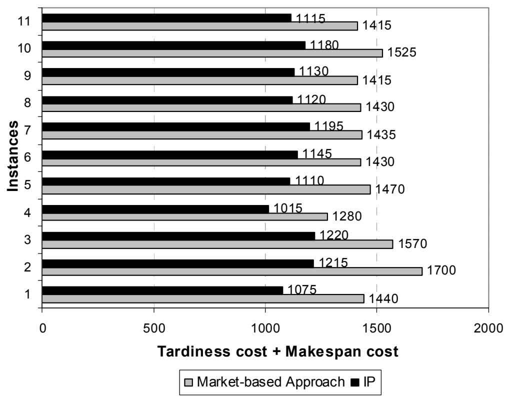

**Figure 1: The performance breakdown.**

Note that we do not intend to claim that the market-based approach we used is superior to the centralized approach in large problem instances, since there are many approximate centralized approaches in the scheduling literatures that can solve very large-scale scheduling problems efficiently. What we really want to emphasize is the capability of our approach in handling decentralized decision making paradigm in a large-scale problem, which is overlooked by many past research. This does not mean advances in large-scale scheduling are irrelevant to the further development of our solution approach; on the contrary, any such advance will be extremely useful in solving single-agent problem, and help deliver better solution quality and efficiency.

## 6. Conclusion

Future works arising from this research are wide-ranging. They include analytical works such as how different bidding behaviors of agents affect the rate of convergence of the tatonnement process and applied research of utilizing our approach to derive efficient solutions to other decentralized, self-interested planning and scheduling problems.

## References

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

[2] J. Q. Cheng and M. P. Wellmam. The WALAS algorithm: A convergent distributed implementation or general equilibrium outcomes. *Computational Econ.* 12: 1-24, 1998.

[3] P. Dewan and S. Joshi. Auction-based distributed scheduling in a dynamic job shop environment. *International Journal of Production Research* 40(5):1173-1191, 2002.

[4] M. L. Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces* 15(2): 10-21, 1985.

[5] P. Joyce. The Walrasian tatonnement mechanism and information. *RAND Journal of Economics* 15(3): 416-425, 1984.

[6] S. H. Jung and K. H. Kim. Load scheduling for multiple quay cranes in port container terminals. *Journal of Intelligent Manufacturing* 17 (4): 479-492, 2006.

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

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

[10] P. Thomas, D. Teneketzis, and J. K. Mackie-Mason. Market-based approach to optimal resource allocation in integrated-services connection-oriented networks. *Operations Research* 50(4): 603-616, 2002.

[11] L. Walras. Elements of pure economics. Allen and Unwin. English translation by William Jaffe 1954 (originally published in 1874).

[12] M. P. Wellman, W. E. Walsh, P. R. Wurman and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Econ Behavior* 35: 271-303, 2001.