

Research Article

Auction-Based Capacity Allocation in Two Parallel Machines with Inclusive Processing Set Restrictions

Qianqian Zhu  and **Xiuli Wang** 

School of Economics and Management, Nanjing University of Science and Technology, Nanjing 210094, China

Correspondence should be addressed to Xiuli Wang; wangdu0816@163.com

Received 2 December 2021; Revised 24 December 2021; Accepted 5 January 2022; Published 28 January 2022

Academic Editor: Guoqiang Wang

Copyright © 2022 Qianqian Zhu and Xiuli Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider the scarce capacity allocation problem in two parallel machines with inclusive processing set restrictions. Our focus is to design an auction mechanism to allocate the production capacity among several selfish customers effectively and efficiently. In our iterative ascending auction mechanism, we need to design jointly two things: who wins what and who pays what. First, we propose an adaptive ascending pricing policy to ensure that bidding by truthful processing requirements and keeping on bidding until the ask prices reach his real revenue is a dominant strategy for each customer. Second, we establish a facility owner's profit maximization model to determine the capacity allocation in two parallel machines with inclusive processing set restrictions; it is an NP hard problem; we also develop a heuristic based on the Lagrangian relaxation technology to obtain the suboptimal solutions. Our computational analysis shows that the auction mechanism can achieve more than 87% of the global system value.

1. Introduction

In current customer-oriented market, the decentralized operating mode has become very popular in the manufacturing industries. This mode can provide services such as product design, manufacturing, and testing for small and medium-sized enterprises by the establishment of a service platform, which is a good solution to the problem of the insufficient funds and talents for small- and medium-sized enterprises, such as the United Microelectronics Corporation (UMC) which uses the online customer information portal MyUMC to provide transparent processing pricing, processing policies, and real-time processing capacity information to customers and allows the selfish customers to make decentralized decisions on booking processing capacity. However, due to the customers' selfish behaviours, the efficiency of the overall system would be reduced to some extent. Therefore, it is important to design an appropriate mechanism to guide individual competition results towards global optimization for the scarce capacity allocation problem in the decentralized operating mode.

To the best of our knowledge, the auction-based method was a suitable approach to solve the production capacity allocation problem, and it has been applied in a single machine and job shop environment [1, 2]. In practice, there existed a scenario that the facility owner possesses parallel machines and they may have the same speed but differ in their functionality, which could be called inclusive processing set (IPS) restrictions. Our study is focused on this case.

There are many applications for capacity allocation problem with IPS restriction. A classical application is assigning several cloud computing tasks with different memory requirements to processors with the same speed but different memory capacities. Each task can be processed by a processor with a memory capacity no less than its memory requirement. Another application exists in testing enterprises. Assuming there are several customers, each with a set of workpieces that needs to be tested, the testing enterprise has two pieces of equipment with the same speed but they differ in test accuracy. High precision testing equipment can provide high and low precision testing services, but low precision testing equipment can only provide low precision

testing services. Here, the testing organization can be described as the “facility owner,” the testing equipment can be described as the “machine,” the set of workpieces can be described as the “order,” and the problem that the facility owner needs to do is allocating the parallel machine capacity to several competing customers, so as to optimize one or more objectives.

In this study, we propose an iterative ascending auction mechanism to solve the capacity allocation problem in two parallel machines with IPS restrictions. The main research work is as follows. First, we design the auction mechanism that the selfish customers can use private information to make effective decisions without knowing the information and strategies of other customers. The pricing policy can make the customers bid by their truthful processing requirements and keep on bidding until the ask prices reach their real revenue. In this way, the customers’ local decisions can prompt the facility owner to make decisions that promote the achievement of the collective goal. Second, the winner determination problem is a bid selecting and scheduling problem in two parallel machines with IPS restrictions, and it is NP hard; we develop a heuristic using the Lagrangian relaxation technique to obtain approximate optimal solutions for the large-sized instances in a reasonable time.

The remaining part of the paper is structured as follows. Section 2 reviews the related work. Then, in Section 3, a detailed description of our problem setting is described. In Section 4, a global optimization problem is studied for mainly benchmarking purposes. In Section 5, an auction mechanism is proposed for the capacity allocation in two parallel machines with IPS restrictions. The pricing and winner determination problems, as the main components of our auction mechanism, are discussed in Sections 5.2 and 5.3, respectively. In Section 6, a computational analysis of the performance for the auction mechanism is presented. Finally, Section 7 provides concluding remarks.

2. Literature Review

In this section, we provide a brief review of the relevant literature in the fields of decentralized scheduling, auctions for resource allocation problems, and parallel machine scheduling with IPS restrictions.

2.1. Decentralized Scheduling. In a decentralized scheduling problem, the facility owner and several customers make decisions rationally for their own interest, and their selfish behaviours result in a situation that can be characterized as system equilibrium. However, such a system equilibrium may lead to suboptimal system performance from a global perspective. The literature on this subject mainly has the following two perspectives: (a) assuming all information is public, characterize and analyse the quality of the resulting system equilibria from the perspective of the global system performance, which can be quantitatively characterized by the price of anarchy (POA) [3–8]; (b) assuming the private information is not public, design and analyse mechanisms to

guide individual competition results towards global optimization. Our study is belonging to the second case.

When the private information is not public, it is necessary to provide a mechanism that all customers commit their truthful information, because the facility owner requires the customers’ private information for a global optimal decision. This research field is known as mechanism design, which began with the work of Leonid Hurwicz [9]. The mechanism design problem can be regarded as the process of solving the optimization problem of incomplete expression. In this case, the designer should first induce this expression and then solve the optimization problem [10]. There are two methods to induce the truthful information from agents: one class of mechanisms called direct revelation mechanisms (DRMs), which directly obtain truthful information by letting agents report their true types; the second called indirect mechanisms, the basic idea is to provide each agent with a choice of action and then assign a result to each action group; the strategy chosen by each agent will indirectly reflect their true type. Auctions are common examples of indirect mechanisms, and they provide several advantages over DRMs for the machine capacity allocation problems, such as the universal and anonymous auctions [11]. In the following subsection, we will specifically introduce the application of the auction mechanisms.

2.2. Auctions for Resource Allocation Problems. Auctions are market-based methods with an explicit set of rules determining prices and allocation of resource according to bids from the market participants [12]. The common auction formats include the Dutch auction, the English auction, the first price sealed bid auction, and the second price sealed bid auction (or Groves-Clarke-Vickery mechanism). These auction formats have been used to sell a wide range of objects, assets, and commodities [13]. With the development of auction theory, many new auctions formats have been designed for resource allocation problems in field of telecommunication system, traffic system, electric system, and so on. In telecommunication system, Cramton [14] presented a new combinatorial clock auction for governments to assign and price licenses for wireless communications. Cramton and Ockenfels [15] analysed and discussed the simultaneous ascending multiband auction for spectrum. In electric markets, Tang and Jain [16] designed auction mechanisms for the aggregators to procure stochastic resources. In traffic systems, Rassenti et al. [17] proposed a combinatorial auction mechanism for the allocation of airport time slots. Yang et al. [18] proposed an auction-based unified approach for prebooking urban logistics facilities.

A few papers also propose auction mechanisms for machine capacity allocation problems. For example, Kutanoğlu and Wu [19] proposed a general auction mechanism for the job shop scheduling problem which uses the notion of multi-item combinatorial auction. They also concluded that there are strong links between Lagrangian-based decomposition and combinatorial auction. Wellman et al. [1] first developed a broad framework for using markets to solve decentralized scheduling problems. They presented an

ascending auction mechanism for capacity allocation problem in a single machine environment with multiple customers each having an order. They also proved that the equilibrium prices may not exist when using time slots as market goods. Dewan and Joshi [2] considered the distributed scheduling problem with the objective of minimizing earliness-tardiness penalties in a job shop environment. They presented an auction mechanism to schedule orders and proposed the iterative price adjustment method to reduce resource contention for the job. Hall and Liu [11] designed an auction mechanism to allocate the limited single machine production capacity among several competing customers, each having an order. Their market goods were time blocks and then proved that the equilibrium solution always exists but is not unique. Karabat and Yalcin [20] considered the pricing and production capacity allocation problem in the single machine environment and proposed an auction mechanism that uses the finished products as market goods for the private information case. They also developed alternative pricing strategies and conducted research towards the impact of the proposed auction mechanism on allocation efficiency. From the above, we see that all these studies present auction mechanisms for the capacity allocation problems in a single machine or job shop environment. In this paper, we aim to design an auction mechanism for the capacity allocation problem in two parallel machines with IPS restrictions.

2.3. Parallel Machine Scheduling with IPS Restrictions.

There are many studies in the literature on parallel machine scheduling with IPS restrictions. Most of those focus on the development of the approximation algorithm (Polynomial Time Approximation Scheme, PTAS). For example, Kafura and Shen [21] considered a computing system model with several independent but identical processors, each with a limited amount of private memory. They proposed a largest memory time first (LMTF) algorithm to minimize the total completion time. Hwang et al. [22] considered the parallel machines scheduling problem in which the process service requests from various customers who are entitled to many different grades of service levels. They proposed a lowest grade and longest processing time (LG-LPT) algorithm for the problem. Ou et al. [23] developed a polynomial time approximation algorithm based on the LG-LPT algorithm. Li and Wang [24] extended the research problem of Kafura and Shen [21]; they studied the parallel machine scheduling problem with both release time and IPS constraints and developed a polynomial time approximation algorithm; Li et al. [25] considered parallel machine scheduling problems with batch processing and IPS restrictions and developed the polynomial time approximation algorithms for the same and different release times, respectively.

In this study, the winner determination problem has the following differences from the above researches. First, our winner determination problem features two concepts, the

bid selection and two parallel machines' scheduling with IPS restrictions. Second, our winner determination problem aims to maximize the profit, whereas the above studies all adopt the traditional objective functions. In addition, the efficiency of the auction mechanism is very important, so we develop a fast heuristic to solve the winner determination problem in reasonable time.

3. Preliminaries

We formally describe the problem under study as follows: We have a set N of n competing customer orders interested in using the facility owner's production capacity. The facility owner possesses two parallel machines M_1 and M_2 which differ in their functionality but not in their processing speeds. We assume that M_2 can process all those orders that M_1 can process. The planning horizon spans a time period $t = 1, 2, \dots, T$, and the facility owner sets per unit reserve values v_1 and v_2 for machines M_1 and M_2 , respectively.

Associated with each customer order i is a processing time p_i , a deadline d_i , a revenue u_i , and a machine index $a_i = 1$ or 2 . We assume that p_i , d_i , and u_i are integer valued. Let $N_1 = \{i | a_i = 1, i \in N\}$, and $N_2 = N/N_1$. Thus, M_1 can process those customer orders in N_1 , and M_2 can process those customer orders in N .

In the decentralized decision-making environment, both the facility owner and the customers hold information privately. The facility owner's objective is to maximize its profit from selling the capacity to customers and holding any unallocated capacity at their reserve values; each customer's objective is to maximize its profit from getting its revenue through purchasing the facility owner production capacity to produce order.

4. Global Optimization Problem

In this section, we assume the information of the customer orders' processing requirements and revenues is public. First, we model the global optimization problem. Then we identify some properties of the optimal solution that can help us to develop heuristics for the winner determination problem.

The objective of the global optimization problem is to maximize the system profit. Without loss of generality, we assume that the customer orders are numbered in nondecreasing sequence of the deadlines, and the shortest processing times break the ties. Set three binary decision variables x_{it}^1 , x_{it}^2 , and y_{it} , where $x_{it}^1 = 1$ denotes that customer order $i \in N_1$ is accepted and completed at time t on machine M_1 ; otherwise $x_{it}^1 = 0$; $x_{it}^2 = 1$ denotes that customer order $i \in N_1$ is accepted and completed at time t on machine M_2 ; otherwise $x_{it}^2 = 0$; $y_{it} = 1$ denotes that customer order $i \in N_2$ is accepted and completed at time t on machine M_2 ; otherwise $y_{it} = 0$. We formulate the global optimization problem as the following integer linear programming model:

$$(GOP) \max \sum_{i \in N_1} \sum_{t=p_i}^T (u_i - v_1 p_i) x_{it}^1 + \sum_{i \in N_1} \sum_{t=p_i}^T (u_i - v_2 p_i) x_{it}^2 + \sum_{j \in N_2} \sum_{t=p_j}^T (u_j - v_2 p_j) y_{jt} + (v_1 + v_2)T \quad (1)$$

$$s.t. \sum_{t=p_i}^T x_{it}^1 + \sum_{t=p_i}^T x_{it}^2 \leq 1, \quad \forall i \in N_1, \quad (2)$$

$$\sum_{t'=p_j}^T y_{jt'} \leq 1, \quad \forall j \in N_2, \quad (3)$$

$$\sum_{i \in N_1} \sum_{s=\max\{p_i, t\}}^{\min\{t+p_i-1, T\}} x_{it}^2 + \sum_{j \in N_2} \sum_{s'=\max\{p_j, t\}}^{\min\{t+p_j-1, T\}} y_{js'} \leq 1, \quad 1 \leq t \leq T, \quad (4)$$

$$\sum_{i \in N_1} \sum_{s=\max\{p_i, t\}}^{\min\{t+p_i-1, T\}} x_{it}^1 \leq 1, \quad 1 \leq t \leq T, \quad (5)$$

$$\sum_{t=1}^T t x_{it}^1 \leq d_i, \quad \forall i \in N_1, \quad (6)$$

$$\sum_{t=1}^T t x_{it}^2 \leq d_i, \quad \forall i \in N_1, \quad (7)$$

$$\sum_{t=1}^T t y_{jt} \leq d_j, \quad \forall j \in N_2. \quad (8)$$

In the above model GOP, Constraints (2) and (3) state that the accepted customer orders should be processed exactly once. Constraints (4) and (5) state that, in each time slot, at most one customer order can be processed on machines M_1 and M_2 , respectively. Constraints (6), (7), and (8) make sure that the accepted customer orders are completed before their deadlines.

Now, we present some optimal properties of the solution for the global optimization problem.

Lemma 1. *For the global optimization problem, there is an optimal schedule in which the accepted customer orders are processed in the earliest due date (EDD) order on both machines.*

Proof. Assuming that π^* is an optimal solution for the global optimization problem, in π^* , there are two adjacent customer orders i and j , and j is processed after i , and $d_i > d_j$. Let t be the start time of i . Perform a neighbor-pair exchange on order i and order j , and get a new solution π' .

In π^* , the completion times of i and j are $t + p_i \leq d_i$ and $t + p_i + p_j \leq d_j$, respectively. In π' , the completion times of i and j are $t + p_i + p_j$ and $t + p_j$, respectively. As we know, $d_i > d_j$, so $t + p_i + p_j \leq d_j < d_i$, $t + p_j < d_j$. Customer orders i and j can also be completed before their deadlines; the new

solution π' is also an optimal solution. Therefore, we reach the conclusion. \square

Lemma 1. *It tells us that no matter which set the accepted customer orders come from, they should be scheduled according to the EDD rule on each machine.*

Lemma 2. *For the global optimization problem, there is an optimal schedule in which if at least one order of S_i is processed on one of the machines, then customer order i is also accepted for processing, where $S_i = \{k | p_k > p_i, d_k \leq d_i, u_k \leq u_i \text{ and } a_k \geq a_i\}$, $i \in N$.*

Proof. Assume that π^* is an optimal solution for the global optimization problem, and customer order i is not in π^* , but customer order k in the set S_i is accepted and completed at time t in π^* , where $S_i = \{k | p_k > p_i, d_k \leq d_i, u_k \leq u_i \text{ and } a_k \geq a_i\}$, $i \in N$ and $t \leq d_k$. Knowing that $a_k \geq a_i$, the machine that can process k can also process i , so we can replace k by i . Denote the new solution as π' .

In the new solution π' , the completion time of i is t , we see that $d_k \leq d_i$, so $t \leq d_i$, and customer order i can also be processed by its deadline in π' . As known $p_k > p_i$ and $u_k \leq u_i$, so the profit of π' is larger than that of π^* , a contradiction arises. Therefore, we reach the conclusion. \square

Lemma 2. *It is implied that if a solution only includes customer order k but not i , then it is not optimal.*

5. Private Information Problem

In this section, we consider the capacity allocation problem, in which the customers keep all the information private, except for the processing times and deadlines. We propose an iterative ascending auction mechanism to solve the private information case. In the following, we will elaborate our auction mechanism from auction protocol, pricing problem, and winner determination problem.

5.1. Auction Protocol. In our iterative ascending auction mechanism, the auctioneer is the facility owner who owns two parallel machines that have the same processing speed but differ in their functionality. The bidders are the customers; each of them has an order to process. The market goods are time blocks; each customer can bid for one single time block at once.

In the first round, the facility owner sets the initial ask price α_i for customers, and (a) each customer sends a bid for processing order so as to maximize his own profit; the bids of the customers are defined as $B_i = [p_i, d_i, a_i, \alpha_i]$, where p_i and d_i are the order's processing time and deadline. Once the processing time and deadlines of the customer orders are confirmed in the first round, it cannot be changed. (b) Then the facility owner collects all bids and determines which bids to accept in order to maximize his own profit and updates the ask price α_i for each customer.

Step (a) and (b) will be iterated; as the ask prices continue to rise gradually, customers gradually withdraw from competition. Once any customer withdraws from the competition, he shall not return to participate in the auction again. If none of the customers submits a new bid, then the auction is terminated.

The design of an optimal auction mechanism (i.e., pricing policy and winner determination algorithm) should fulfil the following properties:

- (1) Individual rationality means that if a customer participates in the auction mechanism, his profit is at least as high as the profit that he is not participating.
- (2) Incentive compatibility means that bidding by the truthful order information is a dominant strategy for each customer.
- (3) Global optimization means the winner determination problem should be a global optimization problem when all the customers bid by their true values.
- (4) Computational efficiency means the auction should reach closure in reasonable time and at reasonable computational expense.

Nisan et al. [26] analyse that the second price sealed auction satisfies the above properties (1) and (2) in combinatorial auctions. However, a sealed auction is less efficient and profitable than an ascending auction in production scheduling problems (Hall and Liu 2015). What is more, in

the private value model, an ascending auction is equivalent to a second price sealed auction. That is because, when the value of market goods for each customer is independent of the others, each customer's dominant strategy is to keep in bidding until the ask price reaches his value, regardless of whether he can observe the process of the auction. At last, the winner will be the one who has the highest value. So the ascending auction mechanism can also be designed to satisfy properties (1) and (2) in our production capacity allocation problem.

In our problem, the winner determination problem is NP hard, so the above properties (3) and (4) are incompatible. When the problem size is small, the optimal solution can be obtained in a limited time. However, when the problem size is large, the optimal solution cannot be obtained in a reasonable time. Therefore, we use approximate optimal solution in exchange for high computational efficiency.

For our auction mechanism, the purpose of the pricing problem is to encourage all the customers to reveal truthful processing requirements, such as machine indexes, processing times, and deadlines, and make the customers' profits nonnegative; the winner determination problem is NP hard; we aim to develop a fast heuristic to ensure that the auction reaches closure in reasonable time and at reasonable computational expense.

5.2. Pricing Problem. We define the ask price in round $k - 1$ for each customer i ($i \in N$) as α_i^k . It consists of two parts: (a) current price β_i^{k-1} , which reflects the current capacity scarcity under the generated schedule in round $k - 1$, and (b) price increment γ_i^k , which reflects the expected impact on capacity scarcity caused by potential bids.

First, we define the current price β_i^k in round k for customer i , based on the generated schedule in round $k - 1$:

- (1) If no time slots before d_i are allocated in round $k - 1$ on both machines, the $\beta_i^k = v_1 + v_2/2$.
- (2) If the time slots before d_i are allocated to orders $\{1, 2, \dots, n_i\}$ with prices $\alpha_1^{k-1}, \alpha_2^{k-1}, \dots, \alpha_{n_i}^{k-1}$ in round $k - 1$ on both machines, then β_i^k is equal to the weighted average of the bid prices of all allocated time slots and the total reserved value of all unallocated time slots. It is clear that the last orders on both machines may be partially processed in $[0, d_i]$. To simplify the expression of β_i^k , let \bar{p}_j^{k-1} be the length of the time slots within $[0, d_i]$ that were allocated to order j on machine a_i in round $k - 1$. If order j is the last order on one of the machines before d_i , then, $\bar{p}_j^{k-1} = p_j - \max\{0, C_j^{k-1} - d_j\}$; else $\bar{p}_j^{k-1} = p_j$, where C_j^{k-1} is the completion time of the order j . Let I_i^{k-1} be the idle times before d_i on the two machines at round $k - 1$. Then β_i^k is equal to a weighted average price:

$$\beta_i^k = \frac{2 \sum_{h=1}^{n_i} \alpha_h^{k-1} \bar{p}_h^{k-1} + I_i^{k-1} (v_1 + v_2)}{2d_i}. \quad (9)$$

Then, we define the price increment γ_i^k in round k for customer i ; it is based on (a) the flexibility of order processing related to its deadline and machine index, if the flexibility of order processing impact on resource scarcity is greater and, consequently, the price increment increases; (b) the number of bids received by this time block that contains each individual time slot. The definition is as follows:

- (1) If no time slots before d_i are allocated in round $k - 1$, then

$$\gamma_i^k = \rho \frac{p_i a_i}{d_i}. \quad (10)$$

- (2) If the time slots before d_i are allocated in round $k - 1$, then

$$\gamma_i^k = \rho \left(\frac{p_i a_i}{d_i} + \frac{\sum_{t=1}^{d_i} D_t^{k-1}}{N d_i} \right). \quad (11)$$

ρ are predetermined constants, called price adjustment factors; a_i , p_i , and d_i are order i 's machine index, processing time, and deadline; D_t^{k-1} is the number of bids that contain time slot t in round $k - 1$.

The purpose of the pricing problem is to encourage the customers to reveal truthful machine indexes, processing times, and deadlines. We assume that the customer would not pretend that he has a smaller machine index and processing time and a greater deadline. See, for example, a customer j having an order with a machine index $a_j = 2$, a processing time $p_j = 5$, and a deadline $d_j = 10$. It is obvious that he would not pretend $p_j < 5$, or $d_j > 10$. If he pretends $a_j = 1$, his order may be assigned to machine M_1 or M_2 ; however, machine M_1 cannot process a high index order, so it is impossible for him to bid by a false smaller machine index. So, in our pricing problem, we only need to ensure that the customer would not bid by a false larger machine index and processing time and a shorter deadline. From the

above definition of price, we can see that the initial ask price for customer i ($i \in N$) is $\alpha_i = (v_1 + v_2)/2 + \rho p_i a_i / d_i$; it is positively correlated with machine index and processing time and negatively correlated with deadline, so each customer will not bid by a larger machine index and processing time or a shorter deadline.

5.3. Winner Determination Problem. In this section, we formulate our winner determination problem as the facility owner's profit maximization problem by selecting the customer bids and scheduling the accepted bids to the two machines synchronously. This subsection is structured as follows: first, we formulate the winner determination problem, and the Lagrangian relaxation method is used to determine the subset of the accepted bids; then we use a heuristic to construct a feasible schedule with these accepted bids.

5.3.1. Lagrangian Relaxation of the Winner Determination Problem. The integer linear programming model of the winner determination problem is described as follows:

$$\begin{aligned} (WDP) \max \quad & \sum_{i \in N_1} \sum_{t=p_i}^T (\alpha_i - v_1 p_i) x_{it}^1 + \sum_{i \in N_1} \sum_{t=p_i}^T (\alpha_i - v_2 p_i) x_{it}^2 \\ & + \sum_{j \in N_2} \sum_{t=p_j}^T (\alpha_j - v_2 p_j) y_{jt} + (v_1 + v_2) T \\ \text{s.t.} \quad & (1), (2), (3), (4), (5), (6), (7). \end{aligned} \quad (12)$$

Relax the constraints (3) and (4), set $\lambda = \begin{pmatrix} \lambda_{11} & \lambda_{21} \\ \lambda_{12} & \lambda_{22} \\ \dots & \dots \\ \lambda_{1T} & \lambda_{2T} \end{pmatrix}$ to be a vector of corresponding nonpositive multipliers, and get the following Lagrangian problem (LR):

$$\begin{aligned} (LR) L(\lambda) = \max \quad & \sum_{i \in N_1} \sum_{t=p_i}^T (\alpha_i - v_1) p_i x_{it} + \sum_{i \in N_1} \sum_{t=p_i}^T (\alpha_i - v_2 p_i) y_{it} \\ & + \sum_{j \in N_2} \sum_{t=p_j}^T (\alpha_j - v_2 p_j) z_{jt} + (v_1 + v_2) T \\ & + \sum_{t=1}^T \lambda_{1t} \left(\sum_{i \in N_1} \sum_{s=\max\{p_i, t\}}^{\min\{t+p_i-1, T\}} x_{is} + \sum_{j \in N_2} \sum_{s'=\max\{p_j, t\}}^{\min\{t+p_j-1, T\}} z_{js} - 1 \right) \\ & + \sum_{t=1}^T \lambda_{2t} \left(\sum_{i \in N_1} \sum_{s=\max\{p_i, t\}}^{\min\{t+p_i-1, T\}} y_{is} - 1 \right) \\ \text{s.t.} \quad & (1), (2), (5), (6) \text{ and } (7). \end{aligned} \quad (13)$$

LR can be rewritten as

$$\begin{aligned}
 (LR) L(\lambda) = & \max \sum_{i \in N_1} \sum_{t=p_i}^T \left(\alpha_i - v_1 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{1s} \right) x_{it} \\
 & + \sum_{i \in N_1} \sum_{t=p_i}^T \left(\alpha_i - v_2 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{2s} \right) y_{it} \\
 & + \sum_{j \in N_2} \sum_{t=p_j}^T \left(\alpha_j - v_2 p_j + \sum_{s=\max\{1, t-p_j+1\}}^t \lambda_{1s} \right) z_{jt} \\
 & + (v_1 + v_2)T - \sum_{t=1}^T \lambda_{1t} - \sum_{t=1}^T \lambda_{2t} \\
 \text{s.t. } & (1), (2), (5), (6) \text{ and } (7).
 \end{aligned} \tag{14}$$

Note that $L(\lambda)$ can be decomposed into

$$\begin{aligned}
 L(\lambda) = & \sum_{i \in N_1} L_i^1(\lambda) + \sum_{i \in N_2} L_i^2(\lambda) \\
 & + \left((v_1 + v_2)T - \sum_{t=1}^T \lambda_{1t} - \sum_{t=1}^T \lambda_{2t} \right).
 \end{aligned} \tag{15}$$

Here,

$$\begin{aligned}
 L_i^1(\lambda) = & \max \sum_{t=p_i}^T \left(\left(\alpha_i - v_1 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{1s} \right) x_{it} \right. \\
 & \left. + \left(\alpha_i - v_2 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{2s} \right) y_{it} \right), \\
 \text{s.t. } & \sum_{t=p_i}^T x_{it} + \sum_{t=p_i}^T y_{it} \leq 1, \sum_{t=1}^T t x_{it} \leq d_i, \sum_{t=1}^T t y_{it} \leq d_i, \\
 L_i^2(\lambda) = & \max \sum_{t'=p_i}^T \left(\alpha_i - v_2 p_i + \sum_{s=\max\{1, t'-p_i+1\}}^t \lambda_{1s} \right) z_{it'}, \\
 \text{s.t. } & \sum_{t'=p_i}^T z_{it'} \leq 1, \sum_{t=1}^T t z_{it} \leq d_i.
 \end{aligned} \tag{16}$$

$L_i^1(\lambda)$ and $L_i^2(\lambda)$ are two subproblems of bid i ; it is easy to see that

$$\begin{aligned}
 L_i^1(\lambda) = & \max \left\{ 0, \max_{p_i \leq t \leq d_i} \left(\left(\alpha_i - v_1 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{1s} \right), \right. \right. \\
 & \left. \left. \left(\alpha_i - v_2 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{2s} \right) \right) \right\}, \\
 L_i^2(\lambda) = & \max \left\{ 0, \max_{p_i \leq t \leq d_i} \left(\alpha_i - v_2 p_i + \sum_{s=\max\{1, t-p_i+1\}}^t \lambda_{1s} \right) \right\}.
 \end{aligned} \tag{17}$$

The Lagrangian dual of (LR) is defined as follows:

$$(D) L_D = \min_{\lambda \geq 0} L(\lambda). \tag{18}$$

Then, the subgradient method is used to determine λ so as to find optimal or near-optimal solutions for LD. The subgradient method is presented as follows.

(1) *Subgradient Algorithm.*

Step 1: set K and σ . Let $k = 0$; set λ^0 and μ^0 .

Step 2: calculate the original problem's feasible solution, and record its objective value as \bar{Z} .

Step 3: calculate the Lagrangian problem $L^k(\lambda^k)$, and record its solution x^k , y^k , and z^k .

Step 4: let $\lambda_1^{k+1} = \lambda_1^k + \beta_1^k G_1(x^k, y^k, z^k) / \|G_1(x^k, y^k, z^k)\|$, where $G_1(x^k, y^k, z^k)$ is the subgradient at point (x^k, y^k, z^k) and

$$\begin{aligned}
 G_1(x^k, y^k, z^k) = & \sum_{i \in N_1} \sum_{s=\max\{p_i, t\}}^{\min\{t+p_i-1, T\}} x_{is} \\
 & + \sum_{j \in N_2} \sum_{s_r=\max\{p_j, t\}}^{\min\{t+p_j-1, T\}} z_{js'} - 1.
 \end{aligned} \tag{19}$$

Set $\mu_1^k = \sigma \mu_1^{k-1}$, and β_1^k is the step size, $\beta_1^k = \mu_1^k (\bar{Z} - L^k(\mu^k)) / \|G_1(x^k, y^k, z^k)\|$.

Let $\lambda_2^{k+1} = \lambda_2^k + \beta_2^k G_2(x^k, y^k, z^k) / \|G_2(x^k, y^k, z^k)\|$, where $G_2(x^k, y^k, z^k)$ is the subgradient at point (x^k, y^k, z^k) and

$$G_2(x^k, y^k, z^k) = \sum_{i \in N_1} \sum_{s=\max\{p_i, t\}}^{\min\{t+p_i-1, T\}} y_{is} - 1. \tag{20}$$

Set $\mu_2^k = \sigma\mu_2^{k-1}$ and $\beta_2^k = \mu_2^k(\bar{Z} - L^k(u^k))/\|G_2(x^k, y^k, z^k)\|$.

Step 5: if $k < K$, let $k \leftarrow k + 1$; go to Step 3; otherwise, stop.

Li and Sun [27] demonstrated that, when using the above expression to calculate the step size, the subgradient algorithm will be convergent and it can converge to an optimal solution for the dual problem with suitable parameters. In the subgradient algorithm, the parameters need to be pre-determined experimentally. The iterations K should ensure sufficient convergence of the algorithm; the step adjustment factors μ_1^k and μ_2^k character the relationship between the quality of solutions and the step size of iterations; if the values are too large, convergence oscillation will occur; if the values are too small, convergence speed will decrease. Set the initial parameter $\mu^0 = (1, 1)$, $\lambda^0 = \begin{pmatrix} -1, -1, \dots, -1 \\ -1, -1, \dots, -1 \end{pmatrix}^T$. By the algorithm, we can obtain an upper bound $L^K(\lambda^K)$ for the winner determination problem, while vectors x^K , y^K , and z^K indicate which bids are accepted for processing. Using the upper bound $L^K(\lambda^K)$, we can evaluate the performance of the following heuristic.

5.3.2. Heuristic of the Winner Determination Problem.

From the subgradient algorithm, we get a preliminary result of bids acceptance; however, the result may not be a festival solution for the winner determination problem, so we design a heuristic to construct a feasible schedule. The heuristic is mainly based on the Lagrangian relaxation technique, and it is described as follows.

(1) Heuristic.

Step 1: let N' denote the subset of accepted bids in N by the subgradient algorithm; let $N'' = N/N'$. Index the bids of N' in nondecreasing order of their due dates and break ties according to the shortest processing time.

Step 2: for bid $j = 1, 2, \dots, |N'|$, perform Steps 2-1 to 2-2.

2-1: if $a_j = 1$, perform Steps 2-1-1 to 2-1-2.

2-1-1: if bid j can be completed before its deadline on M_1 , then assign it to M_1 , next j .

2-1-2: if bid j cannot be completed before its deadline on M_1 , then discard bid j and let $N'' = N'' \cup \{j\}$, next j .

2-2: if $a_j = 2$, perform Steps 2-2-1 to 2-2-3.

2-2-1: if bid j can be completed before its deadline on M_1 and M_2 , then assign it to one machine on which bid j is finished as close to its due date as possible, next j .

2-2-2: if bid j can be completed before its deadline on M_1 or M_2 , then assign it to the machine on which bid j is finished on time, next j .

2-2-3: if bid j cannot be completed before its deadline on M_1 or M_2 , then discard bid j and let $N'' = N'' \cup \{j\}$, next j .

Step 3: select successively one bid l with $b_l/p_l = \max\{b_k/p_k | k \in N''\}$. Calculate $S_l = \{k | p_k > p_l, d_k \leq d_l, u_k \leq u_l \text{ and } a_k \geq a_l, k \in N''\}$; if $S_l \neq \emptyset$, then perform Steps 3-1 to 3-2; else let $N'' = N''/l$.

3-1: insert bid l in all the positions on M_1 and record their objective values. Let $N'' = N''/\{l\}$ and if the maximum objective value among the generated schedules is larger than the current schedule, then take the generated schedule as the current schedule.

3-2: insert bid l in all the positions on M_1 or M_2 ; refer to Step 3-1.

Step 4: if $N'' \neq \emptyset$, go to Step 3; else, stop the heuristic.

In the heuristic of the winner determination problem, we construct a feasible solution in Step 2. In order to make the solution converge to an optimal solution for the winner determination problem, we continuously improve the solution by applying greedily some searching techniques: First, we use Lemma 2 to check whether the bids are likely to be inserted into the machines. Then we add the unaccepted bids to the parallel machines greedily. The computing cost of the winner determination problem is dominated by the solution of the Lagrangian dual problem, so the running time of this problem is $O(nT)$.

6. Computational Experiments

In this section, we conducted computational experiments to analyse the computational efficiency and effectiveness of the ascending auction mechanism. All experiments were run on an Intel 2.9 GHz octa-core processor with 32G RAM. In the following, we present the experimental schemes and the discussion of the results.

6.1. Data Generation. To test the performance of the auction mechanism extensively, we randomly generate instances of the problem by varying the problem size and parameters that may affect the analysis of the auction mechanism. First, we select $n \in \{25, 50, 100\}$ and randomly generate the processing time p_i ($i \in N$) from the integer uniform distribution in the interval $[1, 100]$. Let the total processing time $P = \sum_{i \in N} p_i$, for each instance with P , the facility owner's capacity $T = \lfloor \eta P/2 \rfloor$, where $\eta \in \{0.5, 0.7, 0.9\}$, and the reserve values v_1 and v_2 are randomly generated from the integer uniform distribution in the interval $[1, 10]$, respectively. For each customer order i , we randomly generate its deadline d_i from the integer uniform distribution in the interval $[\max\{p_i, \omega T\}, T]$, where $\omega \in \{0.2, 0.5, 0.8\}$, revenue u_i from the integer uniform distribution in the interval $[p_i \max\{v_1, v_2\}, 1000]$, and machine index a_i from $\{1, 2\}$. Then, for the pricing problem in Subsection 5.2, we define the price adjustment factors $\rho \in \{0.5, 1.0, 2.0\}$. For each $3 \times 3 \times 3 = 81$ possible combinations of parameters, we randomly generate 10 problem instances, for a total of 810.

6.2. Analysis of Heuristic of the Winner Determination Problem. To ensure the auction reach closure in reasonable time and at reasonable computational expense, we use a

TABLE 1: The performance of heuristic of the winner determination problem.

η	ω	R (%)		
		$n = 25$	$n = 50$	$n = 100$
0.5	0.2	97.48	94.46	90.12
	0.5	95.58	92.29	89.53
	0.8	96.53	91.73	87.38
0.7	0.2	95.34	89.67	87.69
	0.5	96.53	92.48	89.73
	0.8	94.82	90.53	90.45
0.9	0.2	93.91	88.42	90.10
	0.5	96.12	93.58	89.24
	0.8	97.73	91.67	87.30
Average		96.00	91.65	89.06

TABLE 2: The performance of the auction mechanism with $n = 25, 50$.

η	ω	ρ	POA (%)			Round		
			$n = 25$	$n = 50$	$n = 100$	$n = 25$	$n = 50$	$n = 100$
0.5	0.2	0.5	94.09	89.40	85.13	15.0	24.5	45.7
		1.0	93.03	87.14	85.92	12.0	20.2	28.3
		2.0	93.02	85.18	83.59	10.3	19.6	24.9
	0.5	0.5	91.68	87.60	83.63	14.0	21.6	39.9
		1.0	92.18	88.08	84.51	12.0	19.3	26.8
		2.0	90.78	86.28	81.29	11.3	19.4	21.9
	0.8	0.5	85.46	85.10	84.33	17.6	30.7	41.3
		1.0	92.49	84.35	83.58	18.3	26.4	32.6
		2.0	93.22	82.70	80.49	10.0	19.8	28.5
0.7	0.2	0.5	92.94	89.36	82.60	15.7	20.5	35.9
		1.0	94.86	90.05	83.78	13.3	20.2	27.5
		2.0	93.85	87.11	82.79	11.7	19.0	25.4
	0.5	0.5	90.58	86.19	83.55	18.8	30.1	39.3
		1.0	93.62	87.21	83.69	16.7	26.4	30.5
		2.0	93.24	84.62	82.22	14.6	24.0	31.6
	0.8	0.5	93.45	83.22	79.88	23.0	31.1	44.6
		1.0	93.34	85.58	81.36	21.7	31.8	39.5
		2.0	96.35	88.31	84.91	15.3	24.4	30.6
0.9	0.2	0.5	94.76	91.14	86.28	18.7	33.9	42.7
		1.0	96.52	86.47	85.81	15.0	24.8	37.8
		2.0	98.79	85.19	83.36	14.7	25.0	29.1
	0.5	0.5	96.61	87.06	85.67	18.3	30.8	38.4
		1.0	97.67	83.59	82.02	15.7	23.4	32.3
		2.0	98.47	89.04	80.07	16.7	25.4	28.2
	0.8	0.5	92.96	83.17	82.55	34.7	40.3	48.5
		1.0	93.26	82.32	81.84	28.0	35.6	39.9
		2.0	95.89	85.08	81.67	22.7	31.8	37.5
Average		93.82	86.24	83.20	16.9	25.9	34.4	

heuristic solving the winner determination problem in exchange for high computational efficiency. In this subsection, we evaluate the performance of this heuristic. First, we use the Lagrangian relaxation technique (Subsection 5.3.1) to establish an upper bound on the facility owner’s maximum profit, which is denoted as UB_f . Let OPT_f and H_f denote the optimal solution value and the heuristic solution, respectively. For the small-sized instances with $n = 25$, we compare the heuristic solution value to the optimal solution value, where $R = H_f/OPT_f$; for the large-sized instances with $n = 50, 100$, we compare the heuristic solution value to the upper bound, where $R = H_f/UB_f$.

The computational results are summarized in Table 1. First, the overall mean values of R for the cases $n = 25, 50$, and 100 are 96.00%, 91.65%, and 89.06%, respectively. The heuristic of the winner determination problem performs well across different problem sizes and parameters. Second, we notice that the mean value of R for the large-sized problem is significantly smaller than that of the small-sized problem. This is because the value of R for the large-sized problems is underestimated due to the gaps between the optimal solutions and the upper bounds. Third, when the value of the facility owner’s capacity parameter η is bigger, different values of the customer order’s deadline parameter

ω will have a more significant impact on the performance of the heuristic. This may indicate that the performance of the heuristic is mainly determined by the scarcity of resource. From the above, we can conclude that the heuristic of the winner determination problem is effective in finding good solutions rapidly.

6.3. Analysis of the Auction Mechanism. To evaluate the performance of the auction mechanism, we define the POA in our study as the percentage deviation of the system value of the auction outcome from that of the global system optimum. For the large-sized instances, it is impossible to get the global optimal solution in a reasonable time with CPLEX, so we use an upper bound instead of the global optimal solution.

The computational results with different parameters are summarized in Table 2. First, the mean values of POA for the case where $n = 25, 50,$ and 100 are 93.82%, 86.24%, and 83.2%, respectively. As the number of customers decreases, the auction performs better. Second, as η increases from 0.5 to 0.9, ω decreases from 0.8 to 0.2, and the auction mechanism performs better. This shows that the fewer the conflicts in resource requirements are, the better the auction performs. Third, the mean number of rounds for the auction to reach closure is 25.7. When the number of customers is larger, and the price adjustment factor ρ is smaller, the auction needs more rounds to reach closure. However, the auction does not perform better when the number of rounds is increased. This may result from the combinatorial optimization characteristics of the machine capacity's allocation problem.

7. Conclusion

In this study, we present an iterative ascending auction mechanism for the scarce production capacity allocation problem in two parallel machines with IPS restrictions. In our auction, the customers do not have to make decisions with all the information; they only need to decide whether to bid or not. The ask prices are updated adaptively by the facility owner which can make the customers bid by truthful processing requirements and keep on bidding until the ask price reaches their real revenues. The winner determination problem integrates the capacity allocation and scheduling decisions in two parallel machines with IPS restrictions. The proposed heuristic of winner determination problem is efficient in finding good solutions. This ensures our auction reaches closure in a reasonable time. Overall, our auction mechanism is an effective method to allocation the capacity in two parallel machines with IPS restrictions.

Data Availability

The data supporting the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported in part by the National Natural Science Foundation of China under Grant 71871118.

References

- [1] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason, "Auction protocols for decentralized scheduling," *Games and Economic Behavior*, vol. 35, no. 1-2, pp. 271-303, 2001.
- [2] P. Dewan and S. Joshi, "Auction-based distributed scheduling in a dynamic job shop environment," *International Journal of Production Research*, vol. 40, no. 5, pp. 1173-1191, 2002.
- [3] E. Koutsoupias and C. Papadimitriou, C. Meinel, "Worst-case equilibria, stacs 99," in *STACS 1999, 16th Annual Symposium on Theoretical Aspects of Computer Science*, S. Tison, Ed., vol. 1563, Berlin, Germany, Springer, 1999.
- [4] B. Heydenreich, R. Müller, and M. Uetz, "Games and mechanism design in machine scheduling--an introduction," *Production and Operations Management*, vol. 16, no. 4, pp. 437-454, 2007.
- [5] G. Christodoulou, E. Koutsoupias, and A. Nanavati, "Coordination mechanisms," in *Automata, Languages and Programming*, vol. 3142, Berlin, Germany, Springer, 2004.
- [6] N. Immorlica, L. Li, V. S. Mirrokni, and S. S. Andreas, "Coordination mechanisms for selfish scheduling," *Theoretical Computer Science*, vol. 410, pp. 1589-1598, 2009.
- [7] K. Lee, J. Y.-T. Leung, and L. M. Pinedo, "Coordination mechanisms for parallel machine scheduling," *European Journal of Operational Research*, vol. 220, pp. 305-313, 2012.
- [8] Q. Chen, L. Lin, Z. Tan, and Y. Yan, "Coordination mechanisms for scheduling games with proportional deterioration," *European Journal of Operational Research*, vol. 263, pp. 380-389, 2017.
- [9] L. Hurwicz, "Optimality and information efficiency in resource allocation processes," in *Mathematical Methods in the Social Science*, K. J. Arrow and S. Karlin, Eds., Stanford University Press, California, CL, USA, 1960.
- [10] D. Condorelli, "Market and non-market mechanisms for the optimal allocation of scarce resources," *Games and Economic Behavior*, vol. 82, pp. 582-591, 2013.
- [11] G. N. Hall and Z. X. Liu, "Market good flexibility in capacity auction," *Production and Operations Management*, vol. 22, no. 2, pp. 459-472, 2013.
- [12] R. P. McAfee and J. McMillan, "Auctions and bidding," *Journal of Economic Literature*, vol. 25, no. 2, pp. 699-738, 1987.
- [13] V. Krishna, *Auction Theory*, Academic Press, Amsterdam, Netherlands, 2nd edn. edition, 2010.
- [14] P. Cramton, "Spectrum auction design," *Review of Industrial Organization*, vol. 42, no. 2, pp. 161-190, 2013.
- [15] P. Cramton and A. Ockenfels, "The German 4G spectrum auction: design and behaviour," *Economic Journal*, vol. 127, no. 605, pp. 305-324, 2017.
- [16] W. Tang and R. Jain, "Stochastic resource auctions for renewable energy integration," in *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 345-352, Monticello, IL, USA, September 2011.
- [17] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A combinatorial auction mechanism for airport time slot allocation," *The Bell Journal of Economics*, vol. 13, no. 2, pp. 402-417, 1982.

- [18] K. Yang, R.-R. Mireia, and M. Menndez, "An auction-based approach for prebooked urban logistics facilities," *Omega, The International Journal of Management Science*, vol. 89, pp. 193–211, 2019.
- [19] E. Kutanoglu and S. D. Wu, "On combinatorial auction and lagrangean relaxation for distributed resource scheduling," *IIE Transactions*, vol. 31, no. 9, pp. 813–826, 1999.
- [20] S. Karabat and Z. B. Yalcın, "An auction mechanism for pricing and capacity allocation with multiple products," *Production and Operations Management*, vol. 23, no. 1, pp. 81–94, 2014.
- [21] D. G. Kafura and V. Y. Shen, "Task scheduling on a multi-processor system with independent memories," *SIAM Journal on Computing*, vol. 6, pp. 167–187, 1977.
- [22] H. C. Hwang, S. Y. Chang, and K. Lee, "Parallel machine scheduling under a grade of service provision," *Computers & Operations Research*, vol. 31, no. 12, pp. 2055–2061, 2004.
- [23] J. Ou, J. Y. T. Leung, and C. L. Li, "Scheduling parallel machines with inclusive processing set restrictions," *Naval Research Logistics*, vol. 55, no. 4, pp. 328–338, 2008.
- [24] C. L. Li and X. L. Wang, "Scheduling parallel machines with inclusive processing set restrictions and job release times," *European Journal of Operational Research*, vol. 200, pp. 702–710, 2010.
- [25] S. G. Li, "Parallel batch scheduling with inclusive processing set restrictions and non-identical capacities to minimize makespan," *European Journal of Operational Research*, vol. 260, pp. 12–20, 2017.
- [26] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, New York, NY, USA, 2007.
- [27] D. Li and X. L. Sun, *Nonlinear Integer Programming*, Springer, New York, NY, USA, 2006.