Integrated Due Date Management through Iterative Bidding

Chun Wang, *Member, IEEE*, Zhiguo Wang
Concordia Institute for Information Systems Engineering, Concordia University
1515 Ste Catherine West, EV 006.223, Montreal, Quebec, Canada
+1 (514) 848-2424 ext. 5628 | Fax: (514) 848-3171
cwang@ciise.concordia.ca

Hamada H. Ghenniwa
Department of Electrical and Computer Engineering,
The University of Western Ontario
London, Ontario, Canada, N6A5B9
+1 519 6612111-88262
hghenniwa@eng.uwo.ca

Weiming Shen, *Senior Member, IEEE*
National Research Council Canada
800 Collip Circle, London, Ontario, Canada
N6G4X8
Tel: +1 519 4307134
Weiming.shen@nrc.ca

*** An earlier version of this paper was presented in IEEE SMC 2009, October 14-18, 2009, San Antonio, Texas.

Abstract—This paper proposes an iterative bidding framework for integrated due date management decision making. We focus on a type of make-to-order environment, in which a firm needs to quote due dates and prices and to schedule the production of a variety of job orders required by a large group of customers. In most cases, customers prefer shorter due dates. However, given limited production capacity and various cost constraints, the firm has to balance the attractiveness of their due date quotations and the reliability in terms of scheduling and delivering accepted job orders. The key issue here is how to integrate due date management decisions such that high quality solutions which benefit both the firm and the customers can be obtained. We study the integrated due date management in an economic setting where customers are modeled as self-interested agents and the objective of the firm is to maximize social welfare. We present an iterative bidding framework as a decentralized decision support tool which enables the integration of key due date management decisions. Effective solutions are achieved through the automated negotiation between the firm and its customers. We provide analytical results on the application of the proposed framework to two special cases of the integrated due date management. We also evaluate the performance of the framework on general due date management problems through a computational study.

Keywords: Due date management; Scheduling; Make to order; Auctions; Iterative bidding; Bidding languages; Multilateral negotiation

I. INTRODUCTION

In supply chain management, the due date of an order is the promised date that the supplier will deliver the product(s). The task of due date management (DDM) is to determine, in a timely manner, if and when an order can be fulfilled profitably. As in other management processes, DDM involves different types of decisions, namely pricing, order acceptance (or demand management), due date setting, and scheduling[1]. These decisions are interrelated. In general, demand can be modeled as a function of market price and delivery time. Customer demand usually increases with lower delivery times as well as with lower prices [2, 3]. In the case of make-to-order manufacturing, before a firm agrees to accept an order, it evaluates the 'profitability" of that order given the resources (e.g. manufacturing capacity) required to satisfy that order and other potential orders that could demand those resources. In addition, the customer and the firm need to agree on the terms of the transactions, in particular, on the price and the due date. If the price or the due date quoted by the firm is too high compared to what the customer is willing to accept, the customer may choose not to place the order. Alternatively, if the price the customer is willing to pay is low or the due date requested is too short to make it a profitable transaction for the firm, the firm might decide not to accept the order. As the DDM decisions are tightly coupled, it is desirable to model the interrelations among them and consider them simultaneously. However, given the complexities of these decisions, in practice, they are often made sequentially. The way that a firm makes DDM decisions is reflected by the time-based competition strategy it adopts.

There are three time-based competition strategies used by firms [3], (1) quick service with minimal wait; (2) "uniform" short lead time[1] guarantee; (3) due date quotation. The first strategy does not involve order acceptance and due date quotation decisions. The focus here is how to schedule job orders such that they can be served as fast as possible. The second strategy promises a uniform lead time guarantee to all customers. Although, a firm can influence the demand rate by adjusting the length of the guaranteed lead time, there is no direct integration between the decisions of order acceptance and scheduling. In fact, under this strategy, there is a risk that demand may exceed the firms' capacity to respond. Uniform lead time guarantee is widely adopted in the service and make-to-order manufacturing sectors [3, 4]. While the strategy may be easy to implement and effective in terms of attracting more customers, it has certain negative impacts, especially in make-to-order manufacturing. As pointed in[1], since the lead times are set without considering the characteristics of the order and the current status of the production, they may be unrealistic in terms of production scheduling, thereby worsening lead time performance, leading to disappointed customers, and/or inflicting higher costs due to expediting. On the other hand, the lead times will be overstated when the demand is low and some customers may choose to go elsewhere. Furthermore, adding additional capacity may be inevitable to maintain the reliability of on-time delivery. The added capacity increases the total production costs and affects the price of the products provided. Although industry practice suggests that customers may be willing to pay a price premium for shorter delivery times [5], in the cases where the premium that a customer is willing to pay cannot compensate the cost of expediting, accepting short-lead-time orders becomes un-profitable.

The third strategy encourages potential customers to get a due date "quote" prior to ordering. As

[1] In the context of DDM, the lead time is defined as the number of working days between the release date of the order and its due date.

the quoted due dates can be calculated based on the schedule of already accepted orders, this strategy has the potential of integrating due date quotation and scheduling decisions. In addition, by quoting prices and due dates, the firm makes an order selection/acceptance decision by influencing which orders finally end up the system. This strategy is more aggressive than the first one because a “quoted” lead time is considered as an irrevocable offer and, once accepted by the customer, the firm needs to deliver as promised. Otherwise, delay penalties may occur. Given limited capacities of a firm, the due date quotation strategy has the potential of effectively coordinating the DDM decisions and achieving optimal solutions in terms of resource utilization and profit. However, this strategy is difficult to implement. In addition to that more decisions need to be considered concurrently, if counter offers from customers are allowed, the implementation of this strategy also requires a multilateral negotiation mechanism between the firm and its customers.

The purpose of this research is to develop an iterative bidding based multilateral negotiation framework to support the integration of DDM decisions under the due date quotation strategy. Unlike some on-line dynamic bidding systems [6, 7], we focus on an off-line setting, in which all the information about the problem, such as the job arrival and processing times, are available at the beginning of the scheduling horizon. In this setting, the resource requirements of multiple job orders need to be considered concurrently during the decision making process. Our main contribution is the design of the multi-lateral negotiation framework for DDM. The framework is implemented by an iterative bidding procedure. It incorporates all key DDM decisions. It also provides DDM process automation, which allows the firm and its customers to construct efficient production schedules through automated multilateral negotiation. The rest of the paper is organized as follows. Section 2 describes and formulates the integrated DDM problem. Section 3 presents the structure and components of the proposed iterative bidding framework. Section 4 provides theoretical analysis on the properties of the framework and evaluates its performance through a computational study. Section 5 compares the proposed framework with existing DDM approaches. Section 6 concludes the paper and discusses future improvements of the framework.

II. The Integrated Due Date Management Problem

Integrated DDM is a decentralized multilateral decision making process. From the perspective of the firm, it combines pricing, order selection, due date setting, and scheduling decisions. The decisions facing the customer are whether she should place the order given the price and due date offered by the firm and how to assign prices to the due dates in a counter offer to maximize her benefit. We assume that the firm has limited manufacturing capacity that can be used to process job orders from customers and the objective of the firm is to maximize the market efficiency, which is the sum of the values on a solution across all customers, rather than its revenue[2]. Each customer has one job order to be processed by the firm. An order has a release time, a preferred due date, and a deadline. The customer's value on an order (the price that she is willing to pay) declines with the delay of the delivery date. The customers' value functions are their private information. The Integrated DDM problem involves the selection of customer orders and allocation of the manufacturing resources of the firm to the orders such that the deadline requirements of all selected orders are met and the sum of customers' values is maximized.

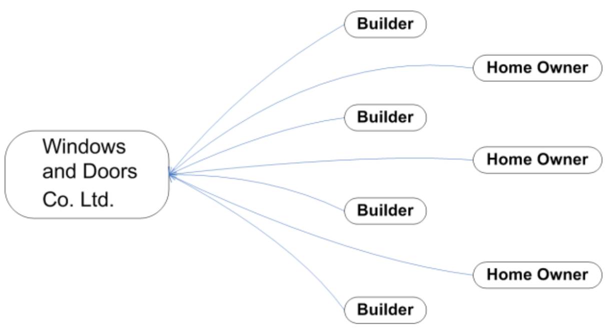

Figure 1 The DDM problem setting in a windows and doors company

As an example, we present a typical integrated DDM problem based on a case study from[9] as follows[3]. As shown in Figure 1, a firm manufactures windows and doors for home builders as well as individual home owners. The products are customized based on the requirements from the customers, which may include different types, sizes and quantities, preferred due dates, and deadlines. In this setting, the integrated DDM problem facing the firm is to coordinate the decisions regarding which order to accept, at what price, and with what delivery date. For

[2] We approach the integrated DDM from a social-welfare perspective. The objective of DDM in this context is to achieve efficient solutions in decentralized environments, which maximize the social welfare of all participants in the supply chain. In[8] D. C. Parkes and J. Kalagnanam, "Models for iterative multiattribute procurement auctions," *Management Science,* vol. 51, pp. 435-451, 2005.Parkes and Kalagnanam suggest that market efficiency is well suited for the design of stable long-term markets that will form the basis for repeated trade. As in most make-to-order cases, customers and the firm will expect repeated trade, it is appropriate to focus on market efficiency in our DDM formulation.

[3] In [9] W. Li, X. Luo, Y. Tu, and D. Xue, " Adaptive production scheduling for one-of-a-kind production with mass customization," *Transactions of the North American Manufacturing Research Institution of SME,* vol. 35, pp. 41-48, 2007., a case study, based on data collected from Gienow Windows and Doors Co. Ltd. (Calgary, Alberta, Canada), is presented to verify the effectiveness of an adaptive scheduling algorithm at shop floor level. While our scope is at supply chain level, we use this manufacturing setting to demonstrate the integrated DDM problem in make-to-order environments.

customers they need to decide how to adjust their orders in terms of requested due dates and prices offered if their original orders are turned down. Again we consider the off-line DDM problems, in which the firm needs to coordinate its DDM decisions across a larger group of customers for a specific production time window (say a week or a month) and the information about customers' job orders are available at the beginning of the decision making process.

A. Centralized Formulation

As previously mentioned we consider integrated DDM as a decentralized decision making problem in the sense that the actual valuation of a customer on due dates is private information to the customer, which is not known to the firm. However, to clearly demonstrate the combinatorial optimization nature of the problem, we first assume a centralized environment, i.e., customers' valuations are known to the firm. With this assumption, we can conveniently model the problem as a mixed integer program. The decentralized characteristic of the problem will be considered when we develop the game theoretic modeling and iterative bidding framework.

Consider a type of the DDM problem which consists of a set of n customers, denoted N, and a firm. Each customer $j(j = 1, \ldots n)$ has a job to be processed by the firm. A job requires the processing of a sequence of operations $o_{j,k}(k = 1, \ldots, n_j)$. An operation $o_{j,k}(j = 1, \ldots n, k = 1, \ldots, n_j)$ has a specified processing time $t_{j,k} \in R^+$ and its execution requires the exclusive use of a designated resource for the duration of its processing. There are precedence constraints among operations of a job, that is, $o_{j,k}$ must precede $o_{j,k+1}$. $q_{j,k,j',k'} = 1$, if $o_{j,k}$ and $o_{j',k'}$ need to be processed on the same resource, otherwise $q_{j,k,j',k'} = 0$. A job j has a release time r_j and a hard deadline $\overline{d}_j$. r_j is the earliest time that job j can be available for processing. $\overline{d}_j$ is the latest completion time of job j in a schedule. For a schedule S which contains an allocation of the firm's production resources to customer orders, a customer will have a valuation on S. In this paper we follow the definition of valuation as described in the private value model, introduced by Vickrey [10]. In integrated DDM, each customer has a value for each schedule and these values do not depend on the private information of the other customers. Each customer knows her values, but not the values of the others. A customer will not accept any schedule S if its job to be completed after the job's hard deadline $\overline{d}_j$ or before its release time r_j. In these cases, the customer's valuation on S is zero or, in terms of DDM, the customer's job is not accepted. In our model, we also allow customers to request *preferred due dates*. For customer j, her *preferred due* *date* is denoted as d_j. $v_j(d_j)$ is the valuation customer j assigns to a schedule in which her job is completed before d_j. Completion of a job after its *preferred due date* is allowed. However, for delayed jobs, there are extra costs incurred to the customer. For a schedule S, if customer j has her job completed at c_j, her valuation on S is defined as $v_j(S) = v_j(d_j) - u_j(c_j)$, where c_j is the completion time of job j in S; $u_j(c_j)$ is a non-decreasing function gives the cost incurred for a delayed c_j within the *acceptable delay window* $d_j < c_j \leq \overline{d}_j$. For the time window $r_j < c_j \leq d_j$ in which the job is not delayed, $u_j(c_j) = 0$. The DDM involves the selection of a set of job orders $N^* \subseteq N$ such that the scheduling constraints for all selected jobs are satisfied and, at the same time, the sum of customer values is maximized. Let $S_{j,k}$ be the starting time of the operation k of the job j; let $Z_j = 1$ if job j is selected and $Z_j = 0$ otherwise; also let $Y_{j,k,j',k'} = 1$ if $o_{j,k}$ is performed before $o_{j',k'}$ and $Y_{j,k,j',k'} = 0$ otherwise ($j \neq j'$). The optimization problem, denoted CDM, is to solve

$$max \sum_{j=1}^{n} Z_j \left(v_j(d_j) - u_j(c_j)\right), \tag{1}$$

subject to

$$c_j = S_{j,n_j} + t_{j,n_j}, \tag{2}$$

$$S_{j,1} \geq r_j Z_j, \tag{3}$$

$$\left(S_{j,n_j} + t_{j,n_j}\right) Z_j \leq \overline{d}_j, \tag{4}$$

$$S_{j,k-1} + t_{j,k-1} \leq S_{j,k}, \tag{5}$$

$$S_{j,k} + t_{j,k} - S_{j',k'} + \mathrm{H}q_{j,k,j',k'} + \mathrm{H}Z_j + \mathrm{H}Z_{j'} + \mathrm{H}Y_{j,k,j',k'} \leq 4\mathrm{H}, \tag{6}$$

$$Y_{j,k,j',k'} + Y_{j',k',j,k} + 3H \geq 1 + \mathrm{H}Z_j + \mathrm{H}Z_{j'} + \mathrm{H}q_{j,k,j',k'}, \tag{7}$$

$$Y_{j,k,j',k'} + Y_{j',k',j,k} + \mathrm{H}Z_{j'} + \mathrm{H}Z_{j'} + \mathrm{H}q_{j,k,j',k'} \leq 1 + 3H, \tag{8}$$

$$Y_{j,k,j',k'} \in \{0,1\}, \tag{9}$$

$$Z_j \in \{0,1\}, \tag{10}$$

$$S_{j,k} \geq 0, \tag{11}$$

where $j, j' = 1, \ldots, n$, $j \neq j'$, $k = 1, \ldots, n_j$ ($k = 2, \ldots, n_j$ in (5)) and $k' = 1, \ldots, n_{j'}$. The set of constraints (3) and (4) ensures that the operations of a job do not start before its release time and finish after its deadline. The set of constraints (5) ensure that an operation does not start before the previous operation of the same job is completed. The set of constraints (6) is a set of logical constraints which states the following: if two jobs j and j' are selected in the schedule, and operations $o_{j,k}$ and $o_{j',k'}$ are to be processed on the same resource ($q_{j,k,j',k'} = 1$), and $o_{j,k}$ precedes $o_{j',k'}$ ($Y_{j,k,j',k'} = 1$), then $S_{j,k} + p_{j,k} \leq S_{j',k'}$. These constraints ensure that, at most, one operation can be processed by a particular resource at a time, where H is a large positive constant, which is used for the linearization of the logical constraint "if". Explanations on how this "large positive constant technique" is used in scheduling problem formulation can be found in [11]. The minimum value of H depends on the problem instance. In general, a H $= max(\overline{d_j}) + max(t_{j,k})$, where $j = 1, \dots, n$ and $k = 1, \dots, n_j$, is large enough to enforce the logical "if" constraint. Constraints (7) and (8) ensure the values assigned to the two related variables $Y_{j,k,j',k'}$ and $Y_{j',k',j,k}$ are consistent, that is, if $S_{j,k}$ and $S_{j',k'}$ are to be processed on the same resource, then $Y_{j,k,j',k'} + Y_{j',k',j,k} = 1$. Constraints (9), (10), (11) are non-negative and integer constraints.

Having modeled the integrated DDM in a job shop environment, we gain insights to the complexity of the problem in terms of number of constraints and variables. We also know that CDM is a nonlinear model as the objective function of CDM is nonlinear. Now we turn our attention to the game theoretical modeling of the problem by considering customer valuations as private information not known to the firm. As the computational complexities inherited from the combinatorial nature of the scheduling problem are not related to the game theoretical modeling, we ignore the scheduling details and focus only on strategic interactions. We first model the integrated DDM as a game. We then construct a Vickrey-Clarke-Groves (VCG) auction that solves the game with an economically efficient outcome.

B. Game Theoretic Modeling and an Auction Construction

In the *centralized formulation*, we have assumed that customers' valuations are known to the firm. In the game theoretic modeling, we remove this assumption and consider customers' valuations are private information and they will behave strategically to maximize their own benefits. To reflect this self-interested property of the customers, we call them agents[4]. Let N denotes a set of n job agents. Each represents a job order from a customer. Job orders need to be scheduled in the firm. Let Γ be the set of all feasible schedules[5]. An agent j needs to pay the firm

[4] In this paper, agents also refer to the trading software entities that represent the customer. From this point forward, when we mention customers in the context of system modeling and design, we will use the term "agent".

[5] Γ can be obtained by solving constraints of CDM as a constraint satisfaction problem. Note that, unlike in classical scheduling theory, a feasible schedule for a DDM problem does not have to include all job orders. If a job order is not included in a schedule, the customer's valuation on the schedule is zero.

$p_j(S)$ in exchange for producing its job order as scheduled in S. The agents must collectively choose a schedule $S \in \Gamma$, and a vector of payments $(p_1(S), p_2(S), p_3(S), \ldots, p_n(S))$.

Our goal is to design a mechanism which enables the collective selection of schedules. As we have assumed that the firm maximizes market efficiency, the chosen schedules need to maximizes the sum of agent valuations. We use Vickrey's private value model. Therefore, an agent's payoff is linear in the agent's valuation of the schedule and the price paid for the schedule, that is $v_j(S) - p_j(S)$. Agents maximize their payoffs. So an agent is willing to pay up to its valuation $v_j(S)$ to obtain the schedule S. In the following, we construct a VCG auction for the DDM problem. Let $V(N) = max_{S\in\Gamma} \sum_{j\in N} v_j(S)$ and $V(N\backslash j) = max_{S\in\Gamma} \sum_{j\in N\backslash j} v_j(S)$. The auction proceeds as follows. Each agent submits its valuations on each element of the set of all feasible schedules Γ. The auctioneer chooses S^* from Γ as the final schedule, such that S^* maximizes $\sum_{j\in N} v_j(S)$, that is, S^* solves $V(N)$. In addition, the auctioneer also computes a schedule for each $j \in N$, such that the schedule solves $V(N \setminus j)$. After the schedules are computed, agent j pays

$$p_j(S^*) = V(N \setminus j) - \sum_{i\neq j} v_i(S^*) \tag{12}$$

and agent j's payoff from participating is

$$\begin{aligned} v_j(S^*) - \left[V(N \setminus j) - \sum_{i\neq j} v_i(S^*)\right] &= v_j(S^*) + \sum_{i\neq j} v_i(S^*) - V(N \setminus j) \\ &= V(N) - V(N \setminus j). \end{aligned} \tag{13}$$

It is clearly that $V(N) - V(N \setminus j)$ is non-negative, which means agents always get non-negative payoffs when participating in the auction. In addition to providing agents with the incentive to participate, the auction is also strategic proof meaning that submitting truth valuations to the auctioneer is a dominant strategy. Suppose agent j reports $w_j \neq v_j$ instead. The auctioneer then chooses a $\tilde{S} \in \Gamma$ that solves $max_{S\in\Gamma}\left[\sum_{i\neq j} v_i(S) + w_j(S)\right]$. Agent j's payoff becomes

$$\sum_{i\neq j} v_i(\tilde{S}) + v_j(\tilde{S}) - V(N \setminus j) \leq V(N) - V(N \setminus j). \tag{14}$$

It is clear that no agent can benefit from misreporting its valuation function.

Given that the CDM can be used to obtain Γ and we have constructed the VCG auction that finds the optimal schedule in Γ, it seems that we have everything needed to solve the DDM game. However, the reality is, despite VCG's theoretical elegance, its limitations in terms of implementation restrict its application to DDM problems. From the auctioneer's side, the implementation of the VCG auction requires the solution of a $V(N)$ and a $V(N \setminus j)$ for all $j \in N$, that is $n+1$ NP-hard optimization problems. The computation cost can be prohibitively expensive if the auction is applied to non-trivial size problems. In our case, the underlying scheduling model CDM is nonlinear, which usually demands more computation than a linear one. From the agents' side, the VCG auction requires an exponential number of schedules in Γ to be valued by each agent, which presents hard valuation problems to agents. In addition to computation, communicating the large number of schedules to agents can also be a huge burden to the system. Most importantly, VCG requires complete valuation on alternative schedules to be revealed to the auctioneer. In DDM, customers are often reluctant to do so when that information might leak out and adversely affect other decisions or negotiations. Transparency is another practical concern in VCG auctions. It can be difficult to explain to the customers why a certain schedule is chosen. In the following section, we propose an iterative bidding framework aimed at addressing some of the limitations arising in the application of VCG to DDM.

III. THE ITERATIVE BIDDING FRAMEWORK

The iterative bidding framework proposed in this paper is an auction-based approach to the integrated DDM problem. The framework contains three major components, a requirement-based bidding language, a linear integer programming model for winner determination, and an iterative bidding procedure. The requirement-based bidding language allows an agent's bid to be expressed by a requirement of processing a job, which naturally represents scheduling constraints and objectives. The winner determination model takes bids expressed in the requirement-based language as input and computes feasible schedules which maximize the auctioneer's revenue. The iterative bidding procedure provides a structure for agents and the auctioneer to interact in a systematic way and eventually evolve the provisional solutions towards an optimal or near optimal one. Iterative bidding also reduces agents' information revelation and adds the potential of accommodating dynamic changes during the bidding process. The iterative bidding framework is a multi-attribute auction, which allows negotiation over price and a non-price attribute: the due date of an agent's schedule. In addition, the framework has good privacy preserving properties. For example, unlike VCG auctions, it does not require agents' knowledge about the resources, such as their capabilities, availabilities and configurations. Also, it does not require complete revelation of agents' valuations.

A. Requirement-Based Bidding Languages

In integrated DDM, customers derive values based on how their jobs are scheduled according to their objectives. From a scheduling perspective, the quality of a schedule can be measured by time related parameters, e.g. completion times, tardiness. During the due date negotiation with a firm, a customer can often express her preferences using a conditional statement. For example, a customer may say she is willing to pay a specific price if her job is completed within a time window, e.g. $r_j < c_j \leq d_j$. There are three components in this conditional statement, the job, the time window and the price. In this section, we propose a requirement-based language for the representation of customers' preferences in terms of these three elements. We first define the atomic bid (C-Bid) of this language.

C-Bid is a 4-tuple $\langle R, eft, lft, p\rangle$ where $\boldsymbol{R}$ is the requirement of processing a job consisting of a set of operations to be performed, the precedence constraints among them and resource requirements. p is the price that the agent is willing to pay for R to be completed within the time window $eft < c \leq lft$ where c denotes the completion time of $\boldsymbol{R}$, eft stands for earliest finishing time, and lft stands for latest finishing time, which is the required due date by the customer. C-Bids can be connected by XOR connective as a XOR-C-Bid to represent values that a customer has on different time windows. For example, $\langle R_j, eft_{j,1}, lft_{j,1}, p_1\rangle XOR\langle R_j, eft_{j,2}, lft_{j,2}, p_2\rangle$ indicates that the customer j is willing to pay $p_1$if R is completed with $eft_{j,1} < c_j \leq lft_{j,1}$ and p_2 if R_j is completed with $eft_{j,2} < c_j \leq lft_{j,2}$. Implicitly here, the customer only wants R_j to be processed once and there is no overlap between the two time windows. If we restrict the value of c to integers, the requirement based language has *full expressiveness* in terms of representing customers' valuations using an XOR-C-Bid with finite number of C-Bids.

Proposition 1 *If the value of* c_j *is restricted to integers for a customer* $j \in N$*, any valuation of customer* j *in integrated DDM can be represented by an XOR-C-Bid with finite C-Bids.*

Proof: See the Appendix.

In Proposition 1, we have proved that if c_j is restricted to integers, a customer j can express her full preferences by assigning a value to each possible c_j with $r_j < c_j \leq \overline{d_j}$. This restriction is reasonable as customers usually define their due dates in terms of the number of certain time units such as hours or days from the time when a job is released. In addition, by restricting the values of the completion times of all jobs to integers, we will have a finite set of $lfts$, which provides us with the possibility of formulating a linear winner determination model as shown in the next section.

B. Linear Winner Determination Model

Given the set of XOR-C-Bids from customers, the task of winner determination is to select a subset of the bids such that all scheduling constraints are satisfied and, at the same time, the sum of customer's value is maximized. A C-Bid can represent a customer's value over a time window defined by the eft and lft. This is natural because, very often, a customer could be indifferent between the completion times within a certain time interval (a block of adjacent time units). Suppose a customer has m_j indifferent time intervals within the *acceptable delayed window* $d_j < c_j \leq \overline{d_j}$. Accordingly an XOR-C-Bid with m_j C-Bids can be constructed to represent the customer's valuations within the window. With the non-delayed $r_j < c_j \leq d_j$ interval included, the full valuation of a customer can be represented by $\langle R_j, eft_{j,0}, lft_{j,0}, p_0\rangle XOR\langle R_j, eft_{j,1}, lft_{j,1}, p_1\rangle XOR\langle R_j, eft_{j,2}, lft_{j,2}, p_2\rangle XOR$ ... $XOR \langle R_j, eft_{j,m_j}, lft_{j,m_j}, p_{m_j}\rangle$ (or in short, $XOR_{0\leq i\leq m_j}\langle R_j, eft_{j,i}, lft_{j,i}, p_i\rangle$), where $eft_{j,0} = r_j$, $lft_{j,0} = d_j$, $p_0 = v_j(d_j)$. We assume that the $m_j + 1$time intervals are adjacent, that is $lft_{j,i-1} = eft_{j,i}$ for $1 \leq i \leq m_j$. Given that $eft_{j,0}$ can be obtained from R_j, it is sufficient to represent the customer's valuations through a *simplified version* of the XOR-C-Bid, $XOR_{0\leq i\leq m_j}\langle R_j, lft_{j,i}, p_i\rangle$. The simplified version does not use eft, however, it contains all the information needed to uniquely construct a corresponding full version of the XOR-C-Bid.

In fact, in most due date quotation scenarios, customers usually use the format of the simplified XOR-C-Bid to express their preferences. For example, they might say "if you promise to complete the job by Thursday, I will pay you $1000; however, if the completion time is Friday, I can only pay you $900". In many cases, an XOR-C-Bid without eft can be a natural format for expressing customers' preferences. In the following, we formulate a winner determination model, denoted LDM, which takes the *simplified version* of XOR-C-Bids from customers as input. By doing this, we make the format of the inputs more intuitive for customers. Note that, as stated in Proposition 1, an XOR-C-Bid has the capability to represent a customer's full valuation. However, this does not mean a customer will reveal her valuation in the XOR-C-Bid submitted to the firm. Iterative bidding is essentially a price system, not a direct revelation mechanism. The bidding prices do not necessarily correspond to a customer's valuations. In LDM, we denote the bidding price from customer j on $lft_{j,i}$ as $p_j(lft_{j,i})$. We also need to define several variables. Let $Z_{j,i} = 1$ if job j is completed before $lft_{j,i}$ and $Z_{j,i} = 0$ otherwise; let $S_{j,k}$ be the starting time of the operation k of the job j; also let $Y_{j,k,j',k'} = 1$ if $o_{j,k}$ is performed before $o_{j',k'}$ and $Y_{j,k,j',k'} = 0$ otherwise ($j \neq j'$). Let $q_{j,k,j',k'} = 1$, if $o_{j,k}$ and $o_{j',k'}$ need to be processed on the same resource, otherwise $q_{j,k,j',k'} = 0$. The winner determination model LDM can be formulated as follows.

$$max \sum_{j=1}^{n} \sum_{i=1}^{m_j} Z_{j,i} p_j(lft_{j,i}), \tag{15}$$

$$\sum_{i=1}^{m_j} Z_{j,i} \leq 1, \tag{16}$$

$$S_{j,1} \geq r_j \sum_{i=1}^{m_j} Z_{j,i} , \tag{17}$$

$$\left(S_{j,n_j} + t_{j,n_j}\right) Z_{j,i} \leq lft_{j,i} , \tag{18}$$

$$S_{j,k-1} + t_{j,k-1} \leq S_{j,k} , \tag{19}$$

$$S_{j,k} + t_{j,k} - S_{j',k'} + \mathrm{H}q_{j,k,j',k'} + \mathrm{H}Z_{j,i} + \mathrm{H}Z_{j',i',} + \mathrm{H}Y_{j,k,j',k'} \leq 4\mathrm{H}, \tag{20}$$

$$Y_{j,k,j',k'} + Y_{j',k',j,k} + 3H \geq 1 + \mathrm{H}Z_{j,i} + \mathrm{H}Z_{j',i'} + \mathrm{H}q_{j,k,j',k'} , \tag{21}$$

$$Y_{j,k,j',k'} + Y_{j',k',j,k} + \mathrm{H}Z_{j,i} + \mathrm{H}Z_{j',i'} + \mathrm{H}q_{j,k,j',k'} \leq 1 + 3H, \tag{22}$$

$$Y_{j,k,j',k'} \in \{0,1\} , \tag{23}$$

$$Z_{j,i} \in \{0,1\}, \tag{24}$$

$$S_{j,k} \geq 0 , \tag{25}$$

where $j, j' = 1, \dots, n,\ j \neq j',\ k = 1, \dots, n_j$ ($k = 2, \dots, n_j$ in (14)), $i = 0, \dots, m_j$ and $k' = 1, \dots, n_{j'}$. Unlike that of CDM, the objective function of LDM is linear. Constraints (16) ensure that only one C-Bid of an XOR-C-Bid is selected in the schedule. Constraints (17) to (25) are essentially constraints (3) to (11) from CDM, except that here variable Z has a two dimension index. For the sake of completeness and readability, we reproduce the constraints here.

LDM takes simplified XOR-C-Bids as input. Constraints (18) show that LDM only requires a job to be finished before the lft. Job completion after the eft is not required. Adding constraints (17), the actual semantic meaning of a simplified C-Bid $\langle R_j, lft_{j,i}, p_i \rangle$ in LDM is interpreted as $\langle R_j, r_j, lft_{j,i}, p_i \rangle$ (*LDM interpretation*). However, the original meaning of $\langle R_j, lft_{j,i}, p_i \rangle$ as it is constructed should be interpreted as $\langle R_j, eft_{j,i}, lft_{j,i}, p_i \rangle$(*EFT interpretation*, as it considers eft).

Would this "misinterpretation" make any difference in winner determination? The answer depends on whether the agents submit their full valuation in XOR-C-Bids. In the case that agents submit their full valuation (this happens in a direct revelation mechanism), as stated in Proposition 2, interpreting $\langle R_j, lft_{j,i}, p_i\rangle$ as $\langle R_j, r_j, lft_{j,i}, p_i\rangle$ does not lead to different optimal solutions. Before presenting Proposition 2, it is useful to go through a small example which demonstrates the basic idea of the proposition.

Example1: Suppose that agent j's full valuation can be described by a simplified XOR-C-Bid, $\langle R_j,\ day\ 5, \$200\rangle XOR\langle R_j, day\ 7, \$150\rangle$ and from R_j we know $r_j = day\ 0$. In LDM the XOR-C-Bid is interpreted as $\langle R_j, day\ 0, day\ 5, \$200\rangle XOR\langle R_j, day\ 0, day\ 7, \$150\rangle$. Since $\langle R_j, day\ 0, day\ 7, \$150\rangle$is equal to $\langle R_j, day\ 0, day\ 5, \$150\rangle XOR\langle R_j, day\ 5, day\ 7, \$150\rangle$, agent j's full valuation can be written as $\langle R_j, day\ 0, day\ 5, \$200\rangle XOR\langle R_j, day\ 0, day\ 5, \$150\rangle$ $XOR\langle R_j, day\ 5, day\ 7, \$150\rangle$. Note that, in these three C-Bids, the first one and second one have identical R_j, eft and lft. That is they can be processed by the same production resources. Whenever a schedule can accommodate the second C-Bid, it must be able to accommodate the first one. Because the two C-Bids are connected by XOR, only one of them can be included in a schedule. Given that the two C-Bids require the same production resources and the price of the first C-Bid is \$200 is greater than that of the second one, the second C-Bid will never be selected in a final schedule because LDM maximizes the sum of prices. Therefore removing the second C-Bid from the valuation does not change the optimal solutions. Customer$j$'s valuation now can be represented by $\langle R_j, day\ 0, day\ 5, \$200\rangle XOR\langle R_j, day\ 5, day\ 7, \$150\rangle$, which is the EFT interpretation of the simplified XOR-C-Bid.

Proposition 2 *If customers submit their full valuations in the format of simplified XOR-C-Bids, for the winner determination model LDM, LDM interpretation and EFT interpretation of the bids do not lead to different optimal solutions.*

Proof: See the Appendix.

As described in the next section, LDM is used for winner determination for the iterative bidding procedure. In each round of the bidding, agents do not submit a complete valuation. In fact, partial revelation of customers' valuations is one of the main benefits of iterative bidding. Without complete revelation, neither full version nor simplified version of C-Bids can guarantee optimal solutions. However, the simplified C-Bids provide potential gains in terms of accommodating more customers in the provisional schedule. This is because constraints (18) in LDM do not require jobs to be completed after ft . The simplified C-Bids in LDM can result in a larger solution space.

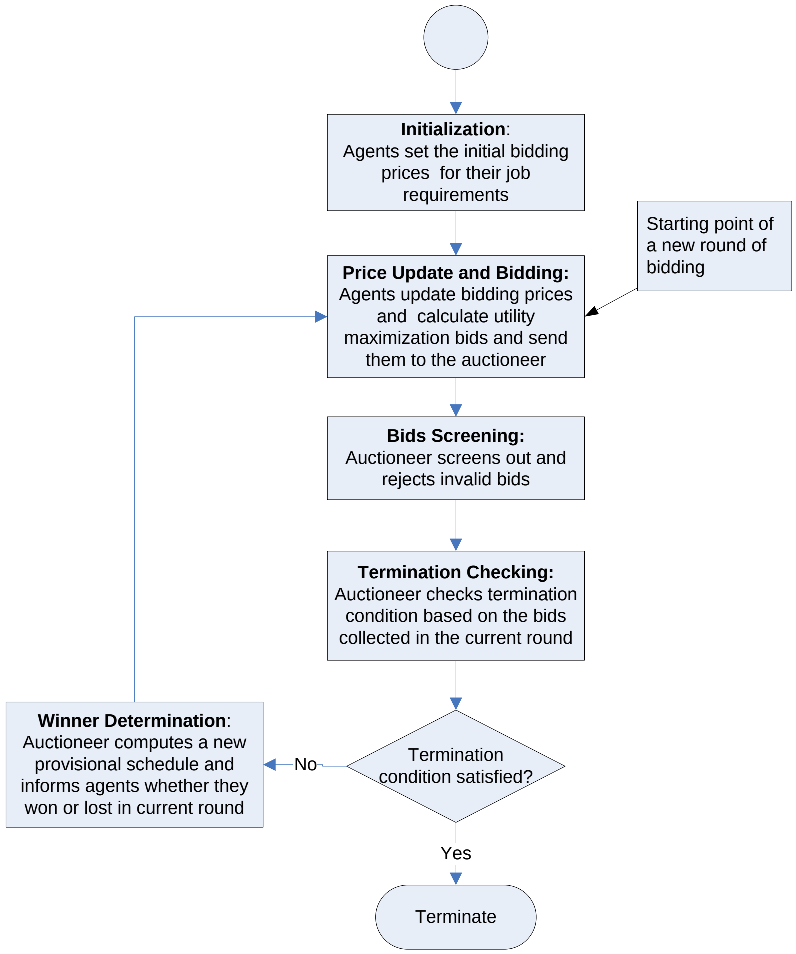

Figure 2 Overview of the iterative bidding and pricing process

C. *The Iterative Bidding*

The iterative bidding procedure is depicted as a flow chart in Figure 2. Initially, an agent has a job to be processed. Before submitting the first bid, the agent needs to initialize a reserve price for the job to be completed between its preferred due date and any other delayed due dates, i.e. $lfts$. The reserve price reflects the basic cost of processing a job. Usually a firm will not go below it for a loss. If an agent has no estimation about the reserve price, it can set the initial reserve prices as zero. However, appropriate setting-up of initial bidding prices can speed up the overall bidding process and, at the same time, maintain the solution quality. In our iterative bidding framework, agents have the incentive to obtain the right reserve prices. It is irrational to submit bids below the reserve prices because those bids will be rejected by the auctioneer. An alternative way is to acquire reserve prices from the auctioneer before the bidding starts. After setting up the reserve prices, agents use them as the first round bidding prices.

1) Price Update and Bidding

At the beginning of round t $(t > 1)$, agents need to update their bidding prices for each of their due dates. This is based on the provisional schedule which resulted from the winner determination at round $t-1$. If an agent was not included in the provisional schedule at round $t-1$, it has three price updating options at round t:

(1) It can increase its bidding prices by ε on due dates it bid for at round $t-1$ or rounds before $t-1$, where ε is the minimum increment imposed by the auctioneer. Since agents are assumed to be rational in maximizing their utilities, they, in general, do not bid with an increment more than ε . However, an agent is allowed to bid aggressively with higher bidding prices than the minimum increment. This may happen when an agent believes that the competition is heavy and bidding with minimum increment is just a waste of time and communication cost and the minimum increment will not get her into a provisional schedule.

(2) It can also keep the bidding prices unchanged (taking a ε discount). However, if an agent takes this ε discount, the auctioneer will consider the agent has entered into *final bid status* and the agent is forbidden from increasing the bidding prices at any of its due dates in future rounds;

(3)Alternatively, it can of course withdraw from the bidding process

If an agent is included in the provisional schedule at round $t-1$, it can keep its bidding price unchanged at round t. That is, it is allowed to repeat its bid at round -1 . However, the bidding procedure does not prevent them from bidding higher.

After updating its bidding prices, an agent needs to compute its set of utility maximizing C-Bids based on the updated bidding prices and its valuation on indifferent time intervals. In computing such a set, an agent *j* solves a maximization problem $max_{\, i\in\{0,1,\dots,m_j\}}\left[v_j\left(lft_{j,i}\right) - p_j^t\left(lft_{j,i}\right)\right]$ and obtains the set of C-Bids which equally maximize its utility, where $p_j^t\left(lft_{j,i}\right)$ is the bidding price for $lft_{j,i}$ at round t. That is, for any two due dates i and i' in the utility maximizing set, $v_j\left(lft_{j,i}\right) - p_j^t\left(lft_{j,i}\right)=v_j\left(lft_{j,i'}\right) - p_j^t\left(lft_{j,i'}\right)$. After obtaining the set of utility maximizing C-Bids, the agent joins them together as an XOR-C-Bid and submits it to the auctioneer. If an agent has entered into *final bid status*, it is no longer allowed to increase it bidding price. However, the auctioneer can choose to allow the agent to repeat its final bid in future rounds until termination. The purpose for this *final bid repeating* arrangement is to boost auctioneer's revenue. During the iterative bidding process, some bids can be temporarily "excluded" from the provisional schedule by a particular combination of scheduling constraints and resource requirements from other bids with higher combined values. After several rounds, that particular combination may have changed and this change may allow the space for the previously excluded bids to be included in the schedule. However, without *final bid repeating*, those bids will not be submitted again if their valuations have been reached during the "excluded" periods. Therefore, they cannot be included anymore, even though there are spaces for them in provisional schedules later on.

2) Bids Screening and Termination

Once receiving XOR-C-Bids from the agents, the auctioneer first screens out invalid bids. Those bids will not be considered in the following winner determination procedure. Invalid bids are defined as follows:

- Any bidding price for a due date, which is below the highest bidding price for that due date received in previous rounds.
- Bids with increased prices from agents who already declared their *final bidding status* in previous rounds.
- Bids with bidding prices which are below the reserve prices.

The auctioneer then checks the termination condition against the valid bids. The auction terminates if there are no price updates for all valid bids in this round. That is, all agents that bid in the last round have repeated their bids. After the auction terminates, the auctioneer implements the final schedule and the agents pay their bidding prices. If the termination condition is not satisfied, the auctioneer will take the set of valid bids as input and solve the winner determination model.

3) Winner Determination

The auctioneer needs to compute a new provisional schedule in each round as long as the auction is not terminated. At round t, the new provisional schedule S_t solves:

$$max_{S_t \in \Gamma_t} \sum_{lft_{j,i} \in S_t} p_j^t(lft_{j,i}) \quad (26)$$

where Γ_t is the set of all feasible schedules given the valid bids submitted at round t. By $lft_{j,i} \in S_t$ we mean the due date $lft_{j,i}$ of agent j is satisfied in the provisional schedule S_t. As the input for winner determination is a collection of XOR-C-Bids consisting of simplified C-Bids, the LDM model can be used for winner determination.

LDM can be solved using standard integer programming optimization packages or dedicated winner determination algorithms. In [12], we have developed a constraint-based winner determination algorithm which allows only one single C-Bid from an agent. The algorithm was designed for the single attribute (price) negotiation and did not take XOR-C-Bids. For the Multi-attribute negotiation model LDM, we expand the capability of the algorithm allowing agents to negotiate over both prices and due dates. LDM can take an agent's preference over these two attributes in the format of an XOR-C-Bid. However the constraint is at most one C-Bid of an XOR-C-Bid can be awarded. To handle this restriction, we have added a checking mechanism to the constraint-based winner determination algorithm to prevent the algorithm from selecting more than one C-Bids from the same XOR-C-Bid into a provisional schedule. The checking mechanism is implemented in the Select-Unassigned-Bid (AV) method of Algorithm 1 in [12] . When the method selects an unassigned C-Bid, it first checks the current schedule. If there is a C-Bid from the same XOR-C-Bid has already included, the unassigned C-Bid will be excluded from the selection. For the details of the constraint-based winner determination algorithm, readers are referred to [12].

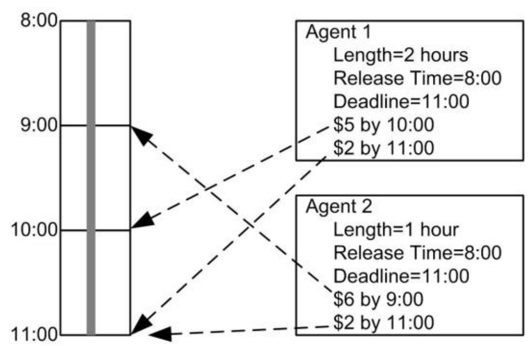

Figure 3 Example of a DDM problem, in which two agents with multiple-due-date-valuations compete for the processing of their one-operation jobs.

D. An Example

This section presents a worked example of the iterative bidding procedure. As shown in Figure

3, Agent1's valuation can be expressed by XOR-C-Bid: $\langle R_1, 8, 10, \$5 \rangle XOR \langle R_1, 10, 11, \$2 \rangle$, where R_1 is the job requirement of Agent1; Agent2's valuation can be expressed by XOR-C-Bid: $\langle R_2, 8, 9, \$6 \rangle XOR \langle R_2, 9, 11, \$2 \rangle$, where R_2 is the job requirement of Agent2. Assume that the resource has the reserve price of 1 dollar an hour and the price increment $\varepsilon = \$2$. The bidding prices and allocation of each round of the iterative bidding are shown in Table 1.

Table 1 Bidding process of an iterative bidding example

Round #	Utility Maximizing Bids				Bidding Prices				Allocation		Auctioneer Revenue	Sum of values
	Agent-1		Agent-2		Agent-1		Agent-2					
	(8,10] \$5	(10,11] \$2	(8,9] \$6	(9,11] \$2	(8,10] \$5	(10,11] \$2	(8,9] \$6	(9,11] \$2	Agent-1	Agent-2		
1	*		*		2	1	1	2	(8,10]		2	5
2	*		*		2	1	3	2		(8,9]	3	6
3	*	*	*		4	1	3	2	(10,11]	(8,9]	4	8
4	*	*	*		4	1	3	2			4	8

The iterative bidding proceeds as follows:

1) Round #1: the agents use the reserve prices for their jobs as the bidding prices. Agent1 uses a simplified XOR-C-Bid and bids on due date 10:00, $\langle R_1, 10, \$2 \rangle$, which requires the time interval (8, 10], and Agent2 bids on due date 9:00, $\langle R_2, 9, \$1 \rangle$, which requires the time interval (8, 9], because, given the current bidding prices, these two due dates maximize agents' utilities. The auctioneer includes only Agent1 into the provisional schedule because the two bids cannot coexist in a schedule and Agent1's bid maximizes auctioneer's revenue.

2) Round #2: While Agent1 repeats its bid at Round#1 $\langle R_1, 10, \$2 \rangle$, Agent2 increases its bidding price on (8, 9] to \$3, $\langle R_2, 9, \$3 \rangle$. After price update, (8, 10] from Agent 1 and (8, 9] from Agent 2 become utility maximizing bids. The auctioneer selects only (8, 9] from Agent 2.

3) Round #3: Agent2 repeats its bid because it was included in the provisional schedule in Round #2. Agent1 increases its bidding price on (8, 10] to \$4. After bidding price update, both (8, 10] and (10, 11] becomes utility maximizing bids from Agent1. So Agent1 sends $\langle R_1, 10, \$4 \rangle XOR \langle R_1, 11, \$1 \rangle$ to the auctioneer. Given the bids submitted by Agent 1 & 2, it is easy to see that there are two solution schedules with the same revenue \$4 for the auctioneer, $\langle (8, 10], \text{Agent1} \rangle$ or $\langle (8, 9], \text{Agent2} + (10, 11], \text{Agent 1} \rangle$. According to the winner determination rule, if there are more than one solutions with identical revenue, winner determination prefers the one that includes

more agents. Therefore, $\langle (8,9], \text{Agent2} + (10,11], \text{Agent 1} \rangle$ is selected.

4) Round #4: Both Agent 1 & 2 repeat their bids. The iterative bidding terminates with an optimal schedule.

IV. Properties of the iterative bidding framework

Compared to one-shot auctions, such as VCG, iterative bidding promises reduced computation at auctioneer side and partial revelation of the private information at agents' side. Also, higher system transparency makes its adoption easier. However, in general, these benefits are obtained with a cost of efficiency. This section evaluates the proposed iterative bidding framework in terms of the trade-offs among four properties, namely efficiency, computation, revenue and information revelation. The evaluation is conducted in the context of integrated DDM. We first develop efficiency and revenue analysis on the application of the iterative bidding framework to two special cases of DDM. We then evaluate the performance of the framework by comparing it with VCG auction through a computational study.

A. Theoretical Results of Two Special Cases

We have proposed an iterative bidding framework for integrated DDM. It provides a platform for customers and the firm to negotiate on both prices and due dates concurrently. However, in some cases, negotiation along multiple attributes is not always needed. For example, a customer might have a firm *single due date* (deadline). She would not consider placing an order if the *single due date* is not satisfied. In addition, she is indifferent between the actual completion times as long as they are within the *single due date*. We refer to this type of valuation functions as *single-due-date-valuation*. If all customers' preferences are *single-due-date-valuation,* negotiation is conducted only along the price dimension because the *single due dates* are not negotiable. On the other hand, prices in certain industries are largely dictated by the market or industrial standards (i.e. the case of vehicle maintenance and repair industry). In these industries, the manufacture or service provider may not have much flexibility in setting the prices. Therefore, negotiation is mainly along the due date dimension. We refer to this type of scenarios as *fixed-price* scenario. We first provide the efficiency result of applying iterative bidding to DDM with *single-due-date-valuation* customers.

Proposition 3 *In integrated DDM problems, if all customers' preferences are single-due-date-valuation and their values on the single due date are congruent to the reserve prices modulo ε ,the iterative bidding procedure with final bid repeating always maximizes the* *sum of customers' valuations at its termination.*

Proof: See the Appendix.

Proposition 3 states, in the case of *single-due-date-valuation*, if customers' values are congruent to the reserve prices modulo the minimum increment, the iterative bidding procedure can always maximize the social welfare of customers without revealing complete valuation information. The purpose of the hypothesis, *agents' valuations are congruent to the reserve prices modulo* ε, is to make sure that an agent can bid exactly at its valuation when necessary given the minimum increment requirement. If we relax the minimum increment requirement at least once during the bidding, such that, when an agent is approaching its valuation, it can always adjust the bidding increment as needed and hit the valuation exactly. In this case, the hypothesis is not necessary.

Let's now turn our attention to the *fixed-price* scenario. In *fixed-price*, the price of processing a particular type of jobs is dictated by a commonly known market price or industrial standard. Agents have different valuations on different $lfts$, however, unlike in the multi-attribute case, they cannot signal the auctioneer about their preferences using price mechanism as prices are fixed and known up front. The only attribute that they can negotiate with the auctioneer and other agents is the lft. As previously assumed, an agent will strictly prefer a shorter lft. Therefore, there is no reason for an agent to submit an XOR-C-Bid consisting of multiple C-Bids with different $lfts$. An agent will not submit a longer lft during the iterative bidding unless they are excluded from the provisional schedule. Due to *fixed-price* restriction, agents cannot indicate their preferences by setting bidding prices. Without the guidance of bidding prices, the iterative bidding procedure cannot guarantee to converge to the schedule that maximizes agents' social welfare as it does in the *single-due-date-valuation* case. However, as stated in Proposition 4, the iterative bidding procedure with *final bid repeating* can achieve Pareto optimality, which means, at termination, no agent can improve its schedule without hurting at least one agent.

Proposition 4 *For the fixed-price cases of integrated DDM, the iterative bidding procedure with final bid repeating terminates with a Pareto optimal schedule.*

Proof: See the Appendix.

We have established some theoretical results on applying the iterative bidding framework to the two special cases of DDM. For general DDM problems, we evaluate the performance of our framework through a computational study. We start with defining the evaluation metrics.

B. Experimental Evaluation Metrics

As mentioned at the beginning of this section, we evaluate the iterative bidding framework in terms of efficiency, computation (running time), revenue and information revelation. These metrics were developed in[13] for testing the performance of *i*Bundle, an iterative combinatorial auction for general combinatorial auction problems. We redefine them in the context of integrated DDM:

Efficiency of Scheduling, $eff(S)$, is measured as the ratio of the value of the final schedule S to the value of the optimal schedule that maximizes total value across the agents:

$$eff(S) = \frac{\sum_{lft_{j,i}\in S} v_j(lft_{j,i})}{\sum_{lft_{j,i}\in S^*} v_j(lft_{j,i})} \ , \tag{27}$$

where S^* is the optimal schedule given customers' valuations.

Running Time of Auction refers to the computation time needed to terminate the auction on a DDM problem instance.

Revenue of Auction, $rev(S)$, is measured as the ratio of auctioneer's income to the value of the optimal solution:

$$rev(S) = \frac{\sum_{lft_{j,i}\in S} \overline{p}_j(lft_{j,i})}{\sum_{lft_{j,i}\in S^*} v_j(lft_{j,i})} \ , \tag{28}$$

where $\overline{p}_j(lft_{j,i})$ is the maximum bid from customer $j$for the due date $lft_{j,i}$ during the auction.

Information Revelation for customer j, $inf(j)$, is measured as the sum of the final price bid by the customer for all due dates in its valuation function, as a fraction of the sum of the true values of each due date.

$$inf(j) = \frac{\sum_{i=0}^{m_j} \overline{p}_j(lft_{j,i})}{\sum_{i=0}^{m_j} v_j(lft_{j,i})} \ , \tag{29}$$

The overall auction information revelation, , is computed as the average information revelation over all agents. The auction often terminates before agents have revealed complete information about their values for due dates. The information revelation metric is designed to measure the extent to which an agent has revealed its value for each due dates to the auctioneer during the auction.

C. Problem Sets

We construct our DDM testing problem sets using a two-step procedure. We first generate *single-due-date-valuation* problems. The design of the *single-due-date-valuation* problems is based on a suite of job shop CSP benchmark problems developed in [14]. Two parameters were adjusted to cover different scheduling conditions. The first one is a range parameter, RG, which controls the distribution of job due dates and release times. The second is a bottleneck parameter, BK, which controls the number of major bottleneck resources. Due dates are randomly drawn from a uniform distribution $M * U(1 - RG, 1)$, where $U(a, b)$ represents a uniform probability distribution between a and b, and M is an estimate of the minimum makespan of the problem, which is determined by the average duration of all operations and the average duration of the operations requiring bottleneck resources. This estimate was first suggested in [15]. Similarly, release times are randomly drawn from a uniform distribution of the form: $M * U(0, RG)$. The price of bid j is randomly drawn from a uniform distribution on $U(du_j, du + du_j)$, where du is the average duration of all bids, and du_j is the duration of bid j. By considering different combinations of RG, BK, and problem sizes (number of operations and number of bids), problems with various configurations can be randomly generated. For testing the iterative bidding framework, we have generated 15 groups of *single-due-date-valuation* problems (detailed configurations are summarized in Table 2).

Table 2 Configuration of the DDM Test Problem Sets

Group	Due Dates	Operations	Bids	BK	RG	Instances
1	1	5	5	2	0.4	5
2	1	5	6	2	0.5	5
3	1	3	7	1	0.5	5
4	1	2	8	1	0.5	5
5	1	2	9	1	0.5	5
6	1	2	10	1	0.5	5
7	3	5	6	1	0.4	10
8	3	5	7	1	0.4	10
19	3	5	8	1	0.4	10
10	3	5	6	1	0.5	10
11	3	5	7	1	0.5	10
12	3	5	8	1	0.5	10
13	3	5	6	2	0.5	10
14	3	5	7	2	0.5	10
15	3	5	8	2	0.5	10

The next step is to generate *multiple-due-date* problem sets by adding two more due date valuations to problem instances of the *single-due-date-valuation* problem sets. The first due date added represents a delay up to 20%. Accordingly, the agent’s value on the delayed due date decreases 20%. The second due date added represents a delay up to 40% and the agent's value decreases 40%. For example, if the *single-due-date-valuation* of an agent's valuation can be represented as a C-Bid $\langle R, 10, \$10 \rangle$, the *multiple-due-date-valuation* of the agent can be represented as an XOR-C-Bid $\langle R, 10, \$10 \rangle XOR \langle R, 12, \$8 \rangle XOR \langle R, 14, \$6 \rangle$. As shown in the Table 2 we generated 9 *multiple-due-date-valuation* problem sets (Group7-15) all with 5 operations.

D. Comparison Results

We compare the iterative bidding framework with VCG in which agents report their complete valuations over different due dates at the beginning of the auction and the auctioneer computes the optimal schedule to maximize the summation of agent values. We have coded LDM into ILOG Optimization Programming Languages (http://www-01.ibm.com/software/websphere /products/optimization/) and solved the *single-due-date-valuation* problems (Group 1-6) using ILOG CPLEX. The reason for using ILOG for computation is to validate the correctness of LDM and test the performance of ILOG CPLEX on the model. For the *multiple-due-date-valuation* problem sets, we have applied the modified version of our previously developed constraint-based winner determination algorithm [12] because CPLEX is relatively slow on these *multiple-due-date-valuation* problem sets.

Figure 4, 5 &6 go around here

We tested the efficiency and the revenue performance of the iterative bidding framework on both *single-due-date-valuation* problem sets and *multiple-due-date-valuation* problem sets. For *single-due-date-valuation* problems, we tested two price updating options, *final bid repeating* and *non final bid repeating*. The optimality result for *single-due-date-valuation* with *final bid repeating* stated in Proposition 3 is validated by the experiments. As we see from the experiment data, when $\varepsilon = 1$, which makes every valuation congruent to every reserve prices, the iterative bidding procedure always finds optimal solutions. Figure 4 shows the efficiency and revenue performance of the iterative bidding framework over the 6 groups of *single-due-date-valuation* problem sets. It is demonstrated in all 6 problem sets, in general, bidding with *final bid repeating* has higher efficiency and revenue than bidding without it. However, the cost is increased computation time. As shown in Figure 5, for the *single-due-date-valuation* problems, bidding with *final bid repeating* significantly increases computation time, especially when the minimum increment is small. Figure 5 also shows that computation times for both *single and multi-due-date valuation* problems are reduced by increasing the minimum increment.

It is interesting to see that for the problem sets with small numbers of bids, such as group 1, when bidding without *final bid repeating*, increasing minimum increment can sometime increase the efficiency. This is due to the "temporary exclusion" we mentioned previously. With small increments, there will be larger number of rounds before termination, which increases the possibility for a bid to be "excluded". For a problem with a small number of agents, if one is "mistakenly excluded" the efficiency cost could be high in terms of the percentage of values across a small number of agents. As shown in Figure 6, a larger number of agents help mitigate the problem to some extent as we see efficiency goes higher with larger number of bids under *non final bid repeating*. To completely avoid this "temporary exclusion" problem, we have designed the *final bid repeating* price updating rule. From Figure 4, it is clear that *final bid repeating* is very effective in terms of boosting the efficiency. For all 6 groups, bidding with *final bid repeating* has close to 100% efficiency for different values of increments. The same reasoning applies to revenue as well.

Figure 7, 8 & 9 go around here

For the *multi-due-date-valuation* problem sets (Group 7-15), Figure 7 plots the efficiency of the iterative bidding over the 9 problem sets with bid increment $\varepsilon = 4$. Compared to VGA (100% efficiency), on average, the iterative bidding without *final bid repeating* can achieve more than 90% efficiency. Figure 8 shows the Information Revelation performance of the iterative bidding procedure. Compared to VCG which requires 100% Information Revelation, the auction requires less than 50% at increment=2 and 4. Bigger increment value requires slightly more Information Revelation. This makes sense because bigger increments may pass some low price equilibrium points which smaller increments may find. Figure 9 compares the run time between the iterative bidding procedure and VCG over 9 *multi-due-date-valuation* problem sets. On average, the iterative auction is more than 10 times faster than VCG with the cost of losing 6%-10% efficiency as shown in Figure 7.

V. RELATED WORK

DDM involves four types of decisions, namely pricing, order acceptance, due date setting, and scheduling. In this paper, we have proposed a framework which allows the integration of these decisions. Compared with existing DDM approaches, the main contribution of this work is the multi-lateral negotiation framework implemented by iterative bidding, which allows decentralized DDM decision making between the firm and its customers. In this section, we discuss this contribution in the context of the DDM literature. Since the proposed framework is an application of iterative auctions to DDM, we will also compare the applicability of several economic-based software systems to the DDM problems and position our bidding framework in the literature.

DDM policies proposed in the literature integrate DDM decisions at different levels. To facilitate the comparison of the proposed framework with the literature, we group existing DDM policies into four categories, namely DS, DSO, DSOP, and BB. We first describe these categories. We then summarize them and provide exemplary references in Table 3.

DS policies only consider due-date setting and scheduling decisions. They ignore the impact of quoted due dates on customers' decisions to place the orders and usually assume that customers are indifferent as to when an order is completed (i.e., due date indifferent) as long as it is within the specified deadline. DSO policies add order acceptance decisions to the DS by modeling the probability of a customer placing an order as a decreasing function of the quoted due date. DSOP policies extend the DSO by modeling the probability of a customer placing an order as a function of both quoted price and quoted due date. Negotiation between the firm and its customers is an important aspect of the due date quotation strategy. BB policies incorporate a bargaining process into bilateral due date decision making. In the model, both the customer and the firm have a reservation tradeoff curve between price and due date, which is private information. BB provides a negotiation mechanism between the firm and its customers. However, it is a bilateral bargaining model, which is not directly applicable to the off-line situations, where the firm needs to optimize the DDM decisions across a group of customers concurrently. In the case of a large number of customers involved in the negotiation, a multilateral model is required.

Table 3 Summary of the four categories of DDM policies

DDM approaches	DDM decisions integrated	Decentralized decision making	References
DS	Due-date setting Scheduling	No support	[16], [17], [18], [19]
DSO	Due-date setting Scheduling Order acceptance	No support	[20], [21], [22], [23], [24]
DSOP	Due-date setting Scheduling Order acceptance Pricing	No support	[25], [4], [3], [26], [27],[28]
BB	Due-date setting Scheduling Pricing	Bilateral bargaining	[29], [30]

Compared with existing work in the DDM literature, the proposed framework integrates all DDM decisions and it also supports decentralized decision making through a multi-lateral negotiation mechanism. Specifically, the framework extends DSOP by providing decentralized decision making through a multilateral negotiation mechanism. It is also more applicable than BB in the DDM situations, where the firm needs to deal with multiple customers concurrently because it supports multi-lateral concurrent negotiation.

In this paper, we have modeled the customers as agents who compete with each other for the firm's production resources to schedule their own jobs according to their respective objectives, the integrated DDM can be seen as a subclass of scheduling problems in decentralized settings. This type of scheduling problems is known as decentralized scheduling problems [31]. In decentralized scheduling problems, agents exhibit complementary preferences over discrete goods. As a subclass, DDM problems also exhibit complementary preferences in agents. For example, a customer usually needs a specific combination of time units on different production resources to complete his/her job. Part of the combination may have no value to the customer because the job cannot be completed without obtaining the combination as a whole. In the rest of this section, we first review economic models that are relevant to DDM; we then analyze their applicability to DDM problems and position our approach in the literature.

Table 4 Summary of the economic models and their applicability to DDM

Economic models	Key Characteristics	Applicability to DDM	Exemplary References
General equilibrium mechanisms	Solve resource allocation or scheduling problems by constructing computational markets based on general equilibrium theory.	DDM problems exhibit indivisibility of goods and complementary preferences of agents, which violate the ideal conditions of the general equilibrium theory. Performance cannot be guaranteed.	[32], [33], [34]
Sequential and simultaneous auctions	These auctions do not allow bids on bundles of items. Sequential auctions sell multiple items in sequence. Simultaneous auctions sell multiple items in separate markets simultaneously.	In DDM, customers have complementary preferences over the firm's resources. These auctions fail when there are no prices that support an efficient solution and also when agents bid cautiously to avoid purchasing an incomplete bundle.	[35],[36],[37],[38],[39]
Combinatorial auctions (CAs)	Allow bidders to submit valuations on bundles of items.	Computation demanded to solve hard valuation problems and winner determination problems can be prohibitive, especially for large size DDM problems.	[40],[41],[42],[43]
Iterative bundle auctions	Allow bidders to submit multiple bids during and auction and provides information feedback to support adaptive and focused eliciation.	Compared with CAs, iterative bundle auctions have smaller sizes of bids and winner determination problems, resulting in lower computational costs. For DDM problems, they are more practical in terms of computation than CAs.	[44],[8],[45],[46],[31]

Many economic models that have been studied in the literature can be applied to decentralized scheduling and DDM to some extent. While giving a comprehensive review of these models is beyond the scope of this paper, in Table 4, we summarize four of them which are of importance to DDM. In economics, the concept of a set of interrelated goods in balance is called general equilibrium. General equilibrium theory provides a distributed method for efficiently allocating goods and resources among agents based on market prices. In applying general equilibrium based mechanism to DDM problems, the goods in the markets need to be specified by imposing a discretization on the continuous timeline to be scheduled on the firm's production resources. These goods are discrete ones, which violate the *infinite divisibility* goods condition of general equilibrium theory. Markets with discrete goods and complementary preferences of agents can lack equilibria [33]. The performance of general equilibrium based market mechanisms on DDM problems is not guaranteed.

Sequential and simultaneous auctions price bundles as the sum price of the individual items. However, they do not allow bidders to bid on bundles of items. Sequential auctions suppose that the set of items are auctioned in sequence. Bidders bid for items in a specific, known order, and can choose how much (and whether) to bid for an item depending on past successes, failures, prices, and so on. Sequential auctions are particularly useful in situations where setting up a combinatorial or simultaneous auctions are infeasible. Simultaneous auctions sell multiple items in separate markets simultaneously. Bidders have to interact with simultaneous but distinct markets in order to obtain a combination of items sufficient to accomplish their task. Real-world markets quite typically operate separately and concurrently despite significant interactions in preferences. A typical example is the series of FCC spectrum auctions [37] . In [44]simultaneous auctions are designed for decentralized train scheduling problems. A review of the uses of economic theory in simultaneous auction design can be found in [47]. Sequential and simultaneous auctions tackle the complementarities over resources in the same spirit of general equilibrium theory. These auctions fail when there are no prices that support an efficient solution (the existence problem) and also when agents bid cautiously to avoid purchasing an incomplete bundle (the exposure problem). However, given that these auctions are more practical in terms of computation, they are two important models worth further studying.

Combinatorial auctions (CAs) allow bidders to place bids on bundles of items. It addresses complementary preference issue explicitly. However, computation demanded to solve hard valuation problems and winner determination problems can be prohibitive. In general, CAs are likely to be practical for smaller size problems. The computational complexities of CAs have been studied by various researchers [43]. Some sophisticated algorithms have produced promising results [42] . In terms of applying CAs to DDM, if general bundle languages, such as LG or LB [48], are used, the timeline of the firm’s production resources needs to be discretized into small time units. This timeline discretization usually results in large amount of items to be sold in the auction, which lead to bigger size problems. Applying CAs to a big size DDM can inflict heavy computation burdens on both customer side and the firm side. Another limitation with VCG is the so called “lying auctioneer” problem [49], which partially explains why Vickery auction is not widely used in practice, even though it has been proposed since 1960’s.

Iterative bundle auctions are iterative implementations of CAs. This class of auction has practical significance because it addresses the computational and informational complexities of

CAs by allowing bidders to reveal their preference information as necessary as the auction proceeds, and bidders are not required to submit (and compute) complete and exact information about their private valuations. With careful design of the structure and components, iterative bundle auctions have the potential of significantly reducing computational costs in CAs. In addition, iterative auctions specially designed for scheduling problems have also been proposed in the literature. In [46] iterative auctions are applied to the job shop scheduling problem. The focus in [46] is to investigate the links between combinatorial auctions and Lagrangean relaxation, and to design auctions based on the Lagrangean based decomposition. In [31], the properties of several iterative auction protocols are investigated in the context of decentralized scheduling. In [50] [38], price prediction and bidding strategies for simultaneous auctions are studied in the setting of market-based scheduling. The proposed framework in this paper is an iterative bundle auction specially designed for DDM problems. In many cases, iterative auctions present better computational and privacy properties than those of CAs. In addition, iterative auctions have the potential of accommodating dynamic events, which is a common requirement in real-world DDM applications. Compared with existing iterative bundle auctions, the novelty of our design is that it uses a requirement-based bidding language to represent DDM domain specific due date, pricing, and job requirements. Unlike general iterative auctions which use bundle languages, the requirement-based language avoids imposing timeline discretization, which causes large amount of items sold in the auction; the adoption of this language also enables the design of more efficient winner determination algorithms which take advantage of the domain specific information to improve the search efficiency. Our previous study [51] has shown that, in auction-based decentralized scheduling, requirement-based language results in more efficient winner determination models than bundle languages do.

In agent-based manufacturing control literature, the contract net [52] and its later variants have been used in DDM as a class of distributed decision making protocols. Unlike auctions, which usually require a mediator, contract nets are purely distributed models, in which any agent can act as a manager and subcontract tasks to other agents. Most of the agent-based control systems were designed for the coordination of production processes within the boundary of an enterprise focusing only on the planning and scheduling part of the DDM. The integration with due date quotation and order selection decisions is usually not considered. References and reviews of this line of research can be found in [53] and [54].

VI. Conclusion

One of the major challenges facing organizations today is the demand for ever-greater levels of responsiveness and shorter defined lead times for deliveries of high-quality goods and services. In order to gain an edge over competitors, firms need to gear their management toward time-based competition, i.e. providing competitive and reliable lead times. However, shorter lead times are not always translated into profits. Given a firm's existing production and supply chain management processes, shorter lead times usually incur higher costs due to expediting. The proposed iterative bidding framework aims at striking the balance between shorter lead times, reliable delivery and anticipated profits.

The uniqueness of the proposed approach is that it integrates the exploration of customers' due date flexibility and the support of the firm's *due date management* decisions within an iterative bidding framework, which has the potential to coordinate the behaviours of self-interested parties in decentralized supply chain environments. For combinatorial (or combinational) auction problems[41], linear programming formulations have been developed[55], which enable the construction of incentive compatible iterative bidding auctions based on the primal-dual design paradigm[44, 56]. For our *due date management* problem model, the decentralized procedure proposed in this paper does not approaches the pricing equilibrium corresponding to the social opportunity cost. As our iterative bidding procedure does not terminate with VCG payments, it is not incentive compatible under the game theoretic assumption of agent behaviour. However, we are designing the system for the type of make-to-order environment in which a firm supplies a large group of customers, such as the case of the Windows & Doors, Co, Ltd example. In this context, it is reasonable to take the market (price-taking) assumption, that is, agents will bid myopically given that each individual agent will have very little impact on the market prices. Despite this game theoretic vs. market argument, designing an incentive compatible iterative bidding auctions for the integrated *due date management* problems is a very important research task on our agenda.

Appendix Proofs of propositions

Proposition 1 *If the value of c_j is restricted to integers for a customer $j \in N$, any valuation of customer j in integrated DDM can be represented by an XOR-C-Bid with finite C-Bids.*

Proof. By the definition of customers' value function, we know that for any $c_j \leq r_j$ and $c_j > \bar{d}_j$, $v_j(c_j) = 0$. For $r_j < c_j \leq \overline{d_j}$, since c_j is restricted to integers, there are finite number of c_j between r_j and $\overline{d_j}$. For each of them, we can construct a unique C-Bid $\langle R_j, c_j - 1, c_j, p\rangle$ where $p = v_j(c_j)$. By joining these C-Bids, we can construct a XOR-C-Bid which expresses the valuation of the customer on $r_j < c_j \leq \overline{d_j}$ and this XOR-C-Bid contains finite number of C-Bids. Implicitly here, for any $c_j \leq r_j$ and $c_j > \bar{d}_j$, $v_j(c_j) = 0$. ∎

Proposition 2 *If customers submit their full valuations in the format of simplified XOR-C-Bids, for the winner determination model LDM, LDM interpretation and EFT interpretation of the bids do not lead to different optimal solutions.*

Proof. Suppose that the complete valuation of a customer is represented by a simplified XOR-C-Bid, $XOR_{0\leq i\leq m_j}\langle R_j, lft_{j,i}, p_i\rangle$. According to the definition, its LDM interpretation is

$$XOR_{0\leq i\leq m_j}\langle R_j, r_j, lft_{j,i}, p_i\rangle \ , \tag{30}$$

For a C-Bid $\langle R_j, r_j, lft_{j,k}, p_k\rangle$ $(0 \leq k \leq m_j)$ within (30), it can be written as an XOR-C-Bid,

$$XOR_{0\leq i\leq k}\langle R_j, eft_{j,i}, lft_{j,i}, p_i\rangle \ , \tag{31}$$

where, $eft_{j,0} = r_j$. We call the C-Bids in (31) *sub-bids* of a C-Bid in (30). Each of the sub-bids represents an indifferent time interval of the customer. Note that for a C-Bid in (30), say the l th C-Bid $\langle R_j, r_j, lft_{j,l}, p_l\rangle$, its last sub-bid is $\langle R_j, eft_{j,l}, lft_{j,l}, p_l\rangle$, which is actually the EFT interpretation of the corresponding simplified C-Bid $\langle R_j, lft_{j,l}, p_l\rangle$. We, therefore, call this last sub-bid *EFT sub-bid*. Given that the EFT interpretation is a sub-bid of the corresponding LDM interpretation, if we prove only the EFT sub-bids are effective in the LDM model, other sub-bids will be dominated (as seen in Example 1), we can conclude that LDM and EFT interpretations lead to same optimal solutions. In the following, we prove that only the EFT sub-bids are effective using the format of mathematical induction.

<u>Basis step</u>:

Let $m_j = 1$. The customer's valuation becomes $\langle R_j, d_j, p_0\rangle XOR\ \langle R_j, lft_{j,1}, p_1\rangle$. Since, under

LDM interpretation, $\langle R_j, lft_{j,1}, p_1\rangle$ will be interpreted as $\langle R_j, r_j, lft_{j,1}, p_1\rangle$, which can be written as $\langle R_j, r_j, d_j, p_1\rangle XOR\langle R_j, eft_{j,1}, lft_{j,1}, p_1\rangle$, the LDM interpretation of the customer's valuation becomes $\langle R_j, r_j, d_j, p_0\rangle XOR\langle R_j, r_j, d_j, p_1\rangle XOR\langle R_j, eft_{j,1}, lft_{j,1}, p_1\rangle$. In our model, we have assumed that customers prefer shorter due dates, that is $p_0 > p_1$. As $\langle R_j, r_j, d_j, p_1\rangle$ requires the same resources, release time and due date as those required by $\langle R_j, r_j, d_j, p_0\rangle$, however, with a lower price, $\langle R_j, r_j, d_j, p_1\rangle$ will be dominated by $\langle R_j, r_j, d_j, p_0\rangle$ in the optimization process. That is, given the presence of $\langle R_j, r_j, d_j, p_0\rangle$, $\langle R_j, r_j, d_j, p_1\rangle$ will never be selected in the final schedule. Therefore, for $\langle R_j, r_j, lft_{j,1}, p_1\rangle$, only the EFT sub-bid $\langle R_j, eft_{j,1}, lft_{j,1}, p_1\rangle$ is effective.

Inductive step:

Let $m_j = k$. Assume that, for all C-Bids in $XOR_{0\le i\le k}\langle R_j, r_j, lft_{j,i}, p_i\rangle$, only the EFT sub-bids are effective and other sub-bids will be dominated in the optimization process. We need to prove that for $m_j = k+1$, for all C-Bids in $XOR_{0\le i\le k+1}\langle R_j, r_j, lft_{j,i}, p_i\rangle$, only the EFT sub-bids are effective.

The LDM interpretation of a customer's valuation (with $m_j = k+1$) is $XOR_{0\le i\le k+1}\langle R_j, r_j, lft_{j,i}, p_i\rangle$, which can be written as

$$XOR_{0\le i\le k-1}\langle R_j, r_j, lft_{j,i}, p_i\rangle XOR\langle R_j, r_j, lft_{j,k}, p_k\rangle XOR\langle R_j, r_j, lft_{j,k+1}, p_{k+1}\rangle \quad (32)$$

The last C-Bid in (32) (with $k+1$ indifferent time intervals) can be represented by XOR-C-Bid:

$$\langle R_j, r_j, lft_{j,k}, p_{k+1}\rangle\ XOR\ \langle R_j, eft_{j,k+1}, lft_{j,k+1}, p_{k+1}\rangle \quad (33)$$

The first sub-bid $\langle R_j, r_j, lft_{j,k}, p_{k+1}\rangle$ in (33) will be dominated by $\langle R_j, r_j, lft_{j,k}, p_k\rangle$ in (32) because $p_{k+1} < p_k$. Therefore, for the two sub-bids in (33) only the EFT sub-bid (last one) is effective. Since we have assumed for all C-Bids in $XOR_{0\le i\le k}\langle R_j, r_j, lft_{j,i}, p_i\rangle$, only the EFT sub-bids are effective, it follows that for $m_j = k+1$, for all C-Bids in $XOR_{0\le i\le k+1}\langle R_j, r_j, lft_{j,i}, p_i\rangle$, only the EFT sub-bids are effective. Therefore, LDM and EFT interpretations of $XOR_{0\le i\le m_j}\langle R_j, lft_{j,i}, p_i\rangle$ lead to the same optimal solutions in LDM. █

Proposition 3 *In integrated DDM problems, if all customers' preferences are single-due-date-valuation and their values on the single due date are congruent to the reserve prices modulo ε ,the iterative bidding procedure with final bid repeating always maximizes the sum of customers' valuations at its termination.*

Proof. Since customers' preferences are single-due-date-valuation, they only need to send simple C-Bids (no XOR-C-Bids) to express their preferences. We assume *private value module* for all customers. Under this model, each customer has a value for her schedule. A customer's payoff for a schedule is the difference between her value on the schedule and the bidding price. To maintain positive payoff, the customer is willing to pay up to her value to get her job scheduled. Therefore, if a customer is not included in a provisional schedule, she will keep increasing her bidding prices in future rounds until she is included or she reach her valuation. Since we have assumed *final bid repeating*, customers repeat their previous bids at termination (round T). Therefore, all customers that are not included in the termination schedule (denoted S^T) have bid with their valuations and the customers that have room to increase their bidding prices at termination are all included in S^T. We prove the proposition by showing that S^T is identical to the optimal schedule S^* computed by solving the winner determination problem using all customers' valuations as inputs.

We construct the customers' bidding prices for an additional round (round $T+1$) as follows. Pick a customer $l \in S^T$ with bidding price at termination (denoted as p_l^T) smaller than her valuation. Let $p_l^{T+1} = p_l^T + n\varepsilon$. n is selected to make sure that p_l^{T+1} is the single-due-date-valuation of l . Since we have assumed that customers' single-due-date-valuations are congruent to the reserve prices modulo ε, n must be an integer. For any other customer $j \in S^T$ and $\neq l$, $p_j^{T+1} = p_j^T$. Let S^{T+1} be the resultant schedule generated by the winner determination for round $T+1$. We first proof that $S^{T+1} = S^T$ by contradiction. Suppose that $S^T \neq S^{T+1}$, we consider the following two cases.

Case #1: $l \in S^{T+1}$

Because S^T is the schedule that maximizes the auctioneer's revenue given the set of bidding prices at round T and we have assumed $S^T \neq S^{T+1}$, it follows that $\sum_{j\in S^T} p_j^T > \sum_{j\in S^{T+1}} p_j^T$. By adding $n\varepsilon$ to both sides, we have $\sum_{j\in S^T\backslash l} p_j^T + p_l^T + n\varepsilon > \sum_{j\in S^{T+1}\backslash l} p_j^T + p_l^T + \mathrm{n}\varepsilon$. That is $\sum_{j\in S^T\backslash l} p_j^T + p_l^{T+1} > \sum_{j\in S^{T+1}\backslash l} p_j^T + p_l^{T+1}$. Because$\sum_{j\in S^{T+1}} p_j^{T+1} = \sum_{j\in S^{T+1}\backslash l} p_j^T + p_l^{T+1}$, it follows that $\sum_{j\in S^T\backslash l} p_j^T + p_l^{T+1} > \sum_{j\in S^{T+1}} p_j^{T+1}$, which means S^{T+1} does not contain the set of customers whose bidding prices at round $T+1$ maximize the auctioneer's revenue. This is a contradiction to our assumption.

Case #2: $l \notin S^{T+1}$

Because S^T is the schedule that maximizes the auctioneer's revenue given the set of bidding prices at round T and we have assumed $S^T \neq S^{T+1}$, it follows that $\sum_{j\in S^T} p_j^T > \sum_{j\in S^{T+1}} p_j^T$. Since $\sum_{j\in S^T} p_j^T = \sum_{j\in S^T\setminus l} p_j^T + p_l^T$, it is clear that $\sum_{j\in S^T\setminus l} p_j^T + p_l^T + n\varepsilon > \sum_{j\in S^{T+1}} p_j^T$. Given the way that bidding prices at round $T+1$ are constructed and $l \in S^{T+1}$, we have $\sum_{j\in S^T\setminus l} p_j^{T+1} + p_l^{T+1} > \sum_{j\in S^{T+1}} p_j^{T+1}$, which means S^{T+1} does not contain the set of customers whose bidding prices at round $T+1$ maximize the auctioneer's revenue. This is also a contradiction to our assumption.

By deriving two contradictions in case #1 & #2, we can conclude that $S^T = S^{T+1}$.

We are now ready to prove that S^T is optimal, that is $S^T = S^*$. Note that S^* is a schedule computed using all customers' valuations as input. In S^{T+1}, customer l has bid with its valuation. Since l was a arbitrary pick, $S^T = S^{T+1}$ can be a general conclusion for all other customers included in S^T. By repeating the above process for other customers, we can reach a final round where all customers included in S^T bid with their valuations. Note that, by definition, the resultant schedule at this final round is S^*. Since the resultant schedules do not change during the bidding process after round T, it follows that $S^T = S^*$. Therefore, S^T maximizes the sum of customers' valuations. ∎

Proposition 4 *For the fixed-price cases of integrated DDM, the iterative bidding procedure with final bid repeating terminates with a Pareto optimal schedule.*

Proof. Under the *fixed-price* restriction, at termination, if an agent is not included in the final schedule, it must have submitted its deadline. This is because an agent will keep extending its lft in its C-Bids if it is not included during the bidding procedure until the deadline is reached. Given that agents also repeat their final bids, at termination, the schedule S^T is computed based on the deadlines from all agents that are not included and the $lfts$ from the agents that are included. To improve their individual schedules, for the agents that are excluded, they have to be scheduled into S^T; on the other hand, for the agents that are already included, they have to move to positions with shorter $lfts$. Since S^T is the optimal solution at termination given the inputs from all agents, in both cases, in order to improve an agent's schedule, at least one other agent will be excluded or pushed to a position with a larger lft. Therefore, S^T is a Pareto optimal schedule. ∎

Figures:

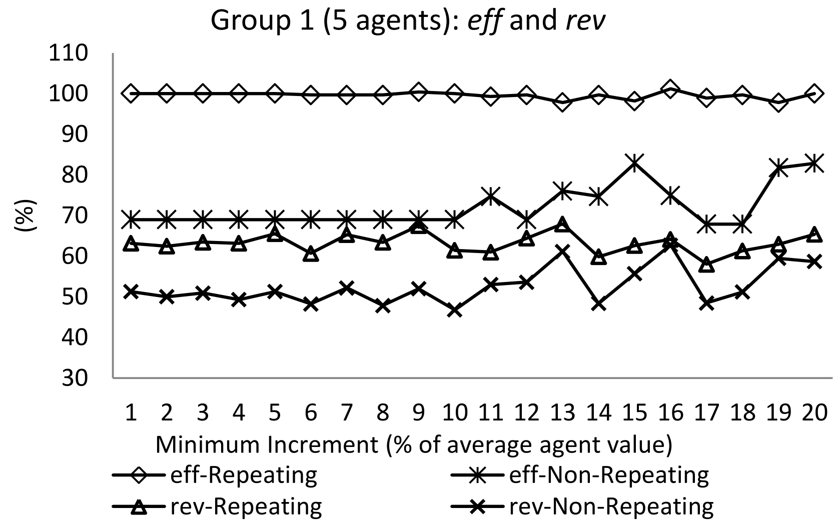

Figure 4-a

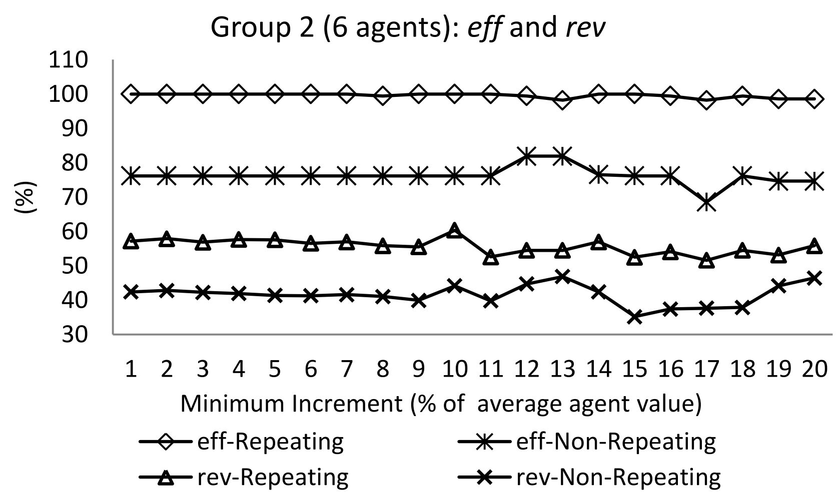

Figure 4-b

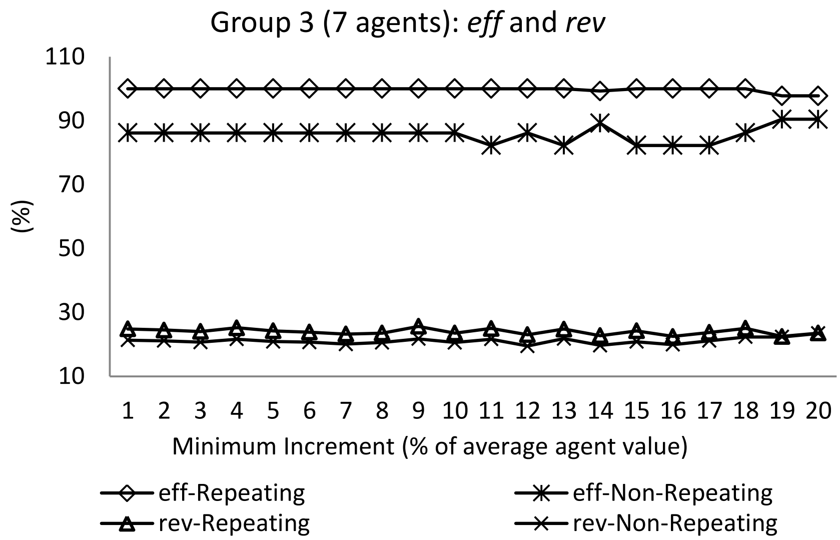

Figure 4-c

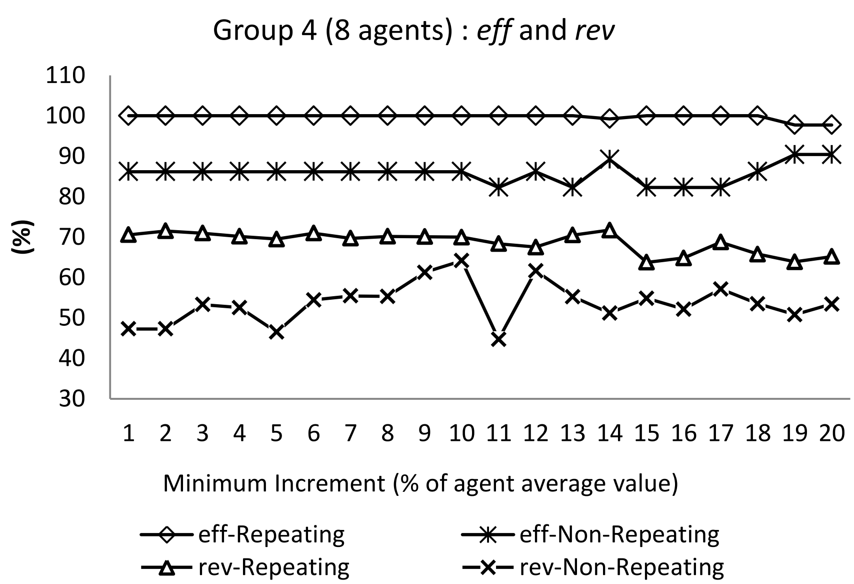

Figure 4-d

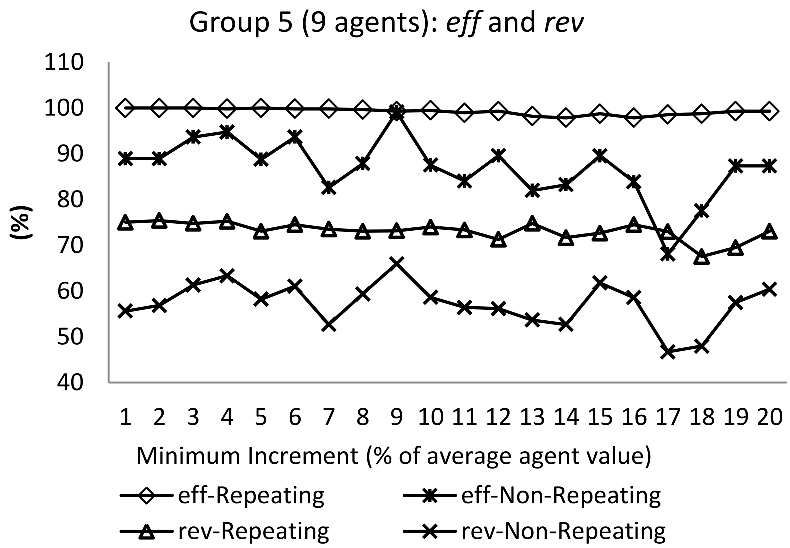

Figure 4-e

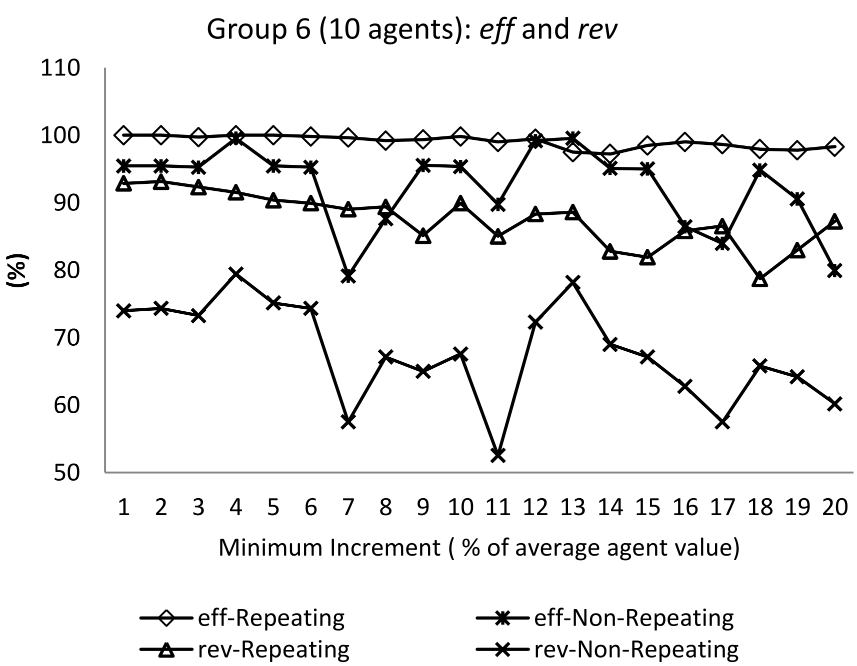

Figure 4 Efficiency and Revenue performance of the iterative bidding framework on 6 groups of *single-due-date-valuation* problems

Figure 4-f

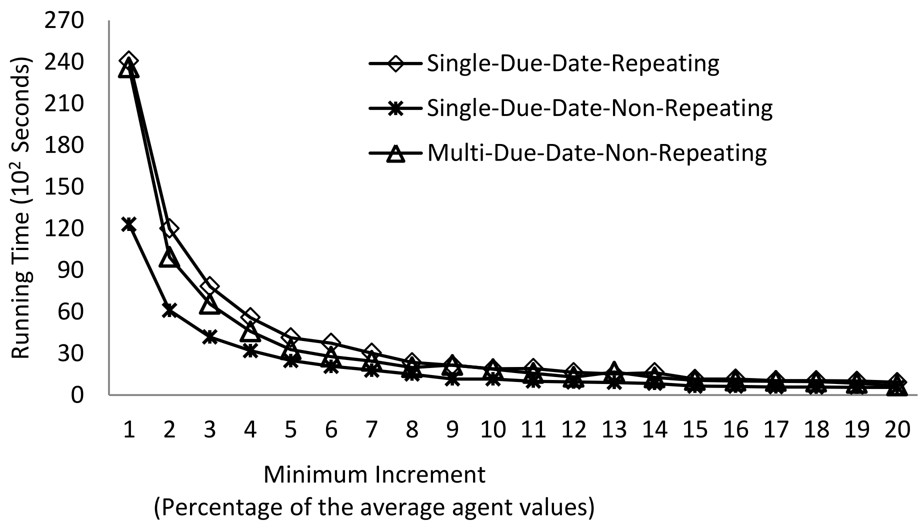

Figure 5 Running time of the iterative bidding on a single due date valuation problem set and a multiple due date problem set

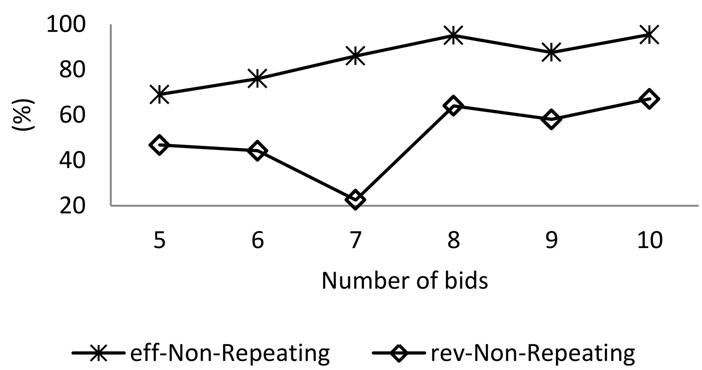

Figure 6 Larger number of bids help mitigate the "temporary exclusion" problem

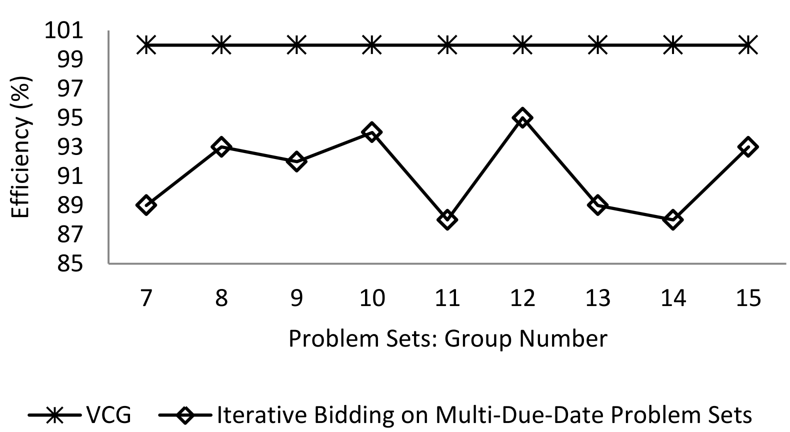

Figure 7 Efficiency comparison of VCG and the iterative bidding procedure on 9 multi-due-date-valuation problem sets.

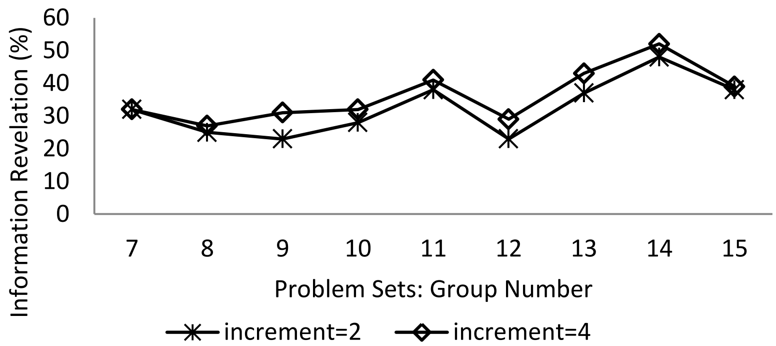

Figure 8 Information revelation performance of the iterative bidding procedure over 9 problem sets with bid increment $\varepsilon = 4$ and $\varepsilon = 2$

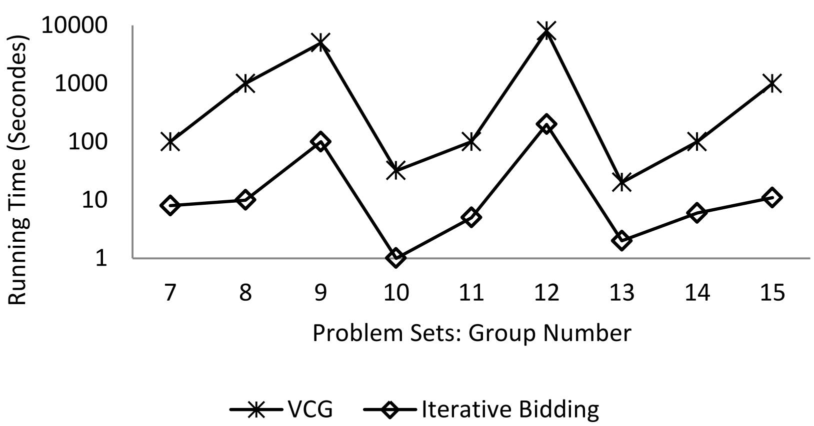

Figure 9 Run time comparison of VCG and iterative bidding over multiple-due-date-valuation sets

References

[1] P. Keskinocak and S. Tayur, "Due date management policies " in *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era, International Series in Operations Research and Management Science*, D. Simchi-Levi, S. D. Wu, and Z. M. Shen, Eds. Norwell, MA: Kluwer Academic Publishers, 2004, pp. 485-553.

[2] J. D. Blackburn, T. Elrod, W. B. Lindsley, and A. J. Zahorik, "The strategic value of response time and product variety," in *Manufacturing Strategy - Process and Content*, C. A. Voss, Ed. London: Chapman and Hall, 1992.

[3] K. C. So and J. S. Song, "Price, delivery time guarantees and capacity selection," *European Journal of Operational Research,* vol. 111, pp. 28-49, 1998.

[4] K. C. So, "Price and time competition for service delivery," *Manufacturing & Service Operations Management,* vol. 2, pp. 392-409, 2000.

[5] J. Magretta, "Fast, global, and entrepreneurial: supply chain management, Hong Kong style-an interview with Victor Fung," *Harvard Business Review,* vol. 76, pp. 102-115, 1998.

[6] S. Kwang Mong and W. Eric, "Toward market-driven agents for electronic auction," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* vol. 31, pp. 474-484, 2001.

[7] E. Campos-Naez, N. Fabra, and A. Garcia, "Dynamic Auctions for On-Demand Services," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 37, pp. 878-886, 2007.

[8] D. C. Parkes and J. Kalagnanam, "Models for iterative multiattribute procurement auctions," *Management Science,* vol. 51, pp. 435-451, 2005.

[9] W. Li, X. Luo, Y. Tu, and D. Xue, " Adaptive production scheduling for one-of-a-kind production with mass customization," *Transactions of the North American Manufacturing Research Institution of SME,* vol. 35, pp. 41-48, 2007.

[10] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *Journal of finance,* vol. 16, pp. 8-37, 1961.

[11] R. L. Rardin, *Optimization in Operations Research*. Upper Saddle River, New Jersey: Prentice Hall Publishers, 1998.

[12] C. Wang, H. H. Ghenniwa, and W. Shen, "Constraint-based winner determination for auction-based scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* vol. 39, pp. 609-618, 2009.

[13] D. C. Parkes, "iBundle: an efficient ascending price bundle auction," in *Proc. ACM Conference on Electronic Commerce (EC-99)*, Denver, 1999, pp. 148-157.

[14] N. Sadeh and M. S. Fox, "Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem," *Artificial Intelligence,* vol. 86, pp. 1-41, 1996.

[15] P. S. Ow, "Focused Scheduling in Proportionate Flowshops," *Management Science,* vol. 31, pp. 852-869, 1985.

[16] M. M. Vig and K. J. Dooley, "Dynamic rules for due-date assignment," *International Journal of Production Research,* vol. 29, pp. 1361-1377, 1991.

[17] V. S. Gordon, "A note on optimal assignment of slack due-dates in single-machine scheduling," *European Journal of Operational Research,* vol. 70, pp. 311-315, 1993.

[18] T. C. E. Cheng, "Optimal total-work-content-power due-date determination and sequencing," *Computers & Mathematics with Applications,* vol. 14, pp. 579-582, 1987.

[19] J. Chuzhoy, R. Ostrovsky, and Y. Rabani, "Approximation algorithms for the job interval selection problem and related scheduling problems," *Mathematics of Operations Research,* vol. 31, pp. 730-738, 2006.

[20] K. Charnsirisakskul, P. M. Griffin, and P. Keskinocak, "Order selection and scheduling with leadtime flexibility," *IIE Transactions,* vol. 36, pp. 697-707, 2004.

[21] N. P. Dellaert, "Due-date setting and production control," *International Journal of Production Economics,* vol. 23, pp. 59-67, 1991.

[22] I. Duenyas, "Single facility due date setting with multiple customer classes," *Management Science,* vol. 41, pp. 608-619, 1995.

[23] P. Keskinocak, R. Ravi, and S. Tayur, "Scheduling and reliable lead-time quotation for orders with availability intervals and lead-time sensitive revenues," *Management Science,* vol. 47, pp. 264-279, 2001.

[24] S. Chatterjee, S. A. Slotnick, and M. J. Sobel, "Delivery guarantees and the interdependence of marketing and operations," *Production and Operations Management,* vol. 11, pp. 393-410, 2002.

[25] K. Charnsirisakskul, P. M. Griffin, and P. Keskinocak, "Pricing and scheduling decisions with leadtime flexibility," *European Journal of Operational Research,* vol. 171, pp. 153-169, 2006.
[26] S. Ray and E. M. Jewkes, "Customer lead time management when both demand and price are lead time sensitive," *European Journal of Operational Research,* vol. 153, pp. 769-781, 2004.
[27] T. Boyaci and S. Ray, "Product differentiation and capacity cost interaction in time and price sensitive markets," *Manufacturing & Service Operations Management,* vol. 5, pp. 18-36, 2003.
[28] F. F. Easton and D. R. Moodie, "Pricing and lead time decisions for make-to-order firms with contingent orders," *European Journal of Operational Research,* vol. 116, pp. 305-318, 1999.
[29] D. R. Moodie, "Demand management: the evaluation of price and due date negotiation strategies using simulation," *Production and Operations Management,* vol. 8, pp. 151–162, 1999.
[30] D. R. Moodie and P. M. Bobrowski, "Due date demand management: negotiating the trade-off between price and delivery," *International Journal of Production Research,* vol. 37, pp. 997-1021, 1999.
[31] 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, pp. 271-303, 2001.
[32] F. Ygge and H. Akkermans, "Power load management as a computational market," in *Proc. 2nd International Conference on Multiagent Systems,*, Menlo Park, CA, 1996, pp. 393–400.
[33] W. E. Walsh and M. P. Wellman, "Efficiency and equilibrium in task allocation economies with hierarchical dependencies," in *Proc. 6th International Joint Conference on Artificial Intelligence,* Stockholm, 1999, pp. 520-526.
[34] M. P. Wellman, "A computational market model for distributed configuration design," *AI EDAM,* vol. 9, pp. 125-133, 1995.
[35] C. Boutilier, M. Goldszmidt, and B. Sabata, "Sequential auctions for the allocation of resources with complementarities," in *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99),* Stockholm, 1999, pp. 527-534.
[36] R. Engelbrecht-Wiggans and R. J. Weber, "A sequential auction involving asymmetrically-informed bidders," *International Journal of Game Theory,* vol. 12, pp. 123-127, 1983.
[37] R. P. McAfee and J. McMillan, "Analyzing the airwaves auction," *The Journal of Economic Perspectives,* vol. 10, pp. 159-175, 1996.
[38] D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepayshvili, "Exploring bidding strategies for market-based scheduling," *Decision Support Systems,* vol. 39, pp. 67-85, 2005.
[39] D. C. Parkes and L. H. Ungar, "An auction-based method for decentralized train scheduling," in *Proc. 5th International Conference on Autonomous Agents,* Montreal, Quebec, 2001, pp. 43-50.
[40] E. Kutanoglu and S. D. Wu, "Incentive compatible, collaborative production scheduling with simple communication among distributed agents," *International Journal of Production Research,* vol. 44, pp. 421-446, 2006.
[41] S. de Vries and R. V. Vohra, "Combinatorial auctions: a survey," *INFORMS Journal on Computing,* vol. 15, pp. 284-309, 2003.
[42] T. Sandholm, S. Suri, A. Gilpin, and D. Levine, "CABOB: a fast optimal algorithm for winner determination in combinatorial auctions," *Management Science,* vol. 51, pp. 374-390, 2005.
[43] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," *Artificial Intelligence,* vol. 135, pp. 1-54, 2002.
[44] D. C. Parkes and L. H. Ungar, "Iterative combinatorial auctions: theory and practice," in *Proc. 17th National Conference on Artificial Intelligence,* Austin, TX. , 2000, pp. 74–81.
[45] S. Bikhchandani and J. M. Ostroy, "Ascending price Vickrey auctions," *Games and Economic Behavior,* vol. 55, pp. 215-241, 2006.
[46] E. Kutanoglu and S. D. Wu, "On combinatorial auction and Lagrangean relaxation for distributed resource scheduling," *IIE Transactions,* vol. 31, pp. 813-826, 1999.
[47] P. Milgrom, "Putting auction theory to work: the simultaneous ascending auction," *Journal of Political Economy,* vol. 108, pp. 245-272, 2000.
[48] C. Boutilier and H. H. Hoos, "Bidding languages for combinatorial auctions," in *Proc. 17th International Joint Conference on Artificial Intelligence,* Seattle, Washington, 2001, pp. 1211–1217.
[49] T. W. Sandholm, "Distributed rational decision making," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,* Cambridge, MA: MIT Press, 1999.
[50] J. K. Mackie-Mason, A. Osepayshvili, D. M. Reeves, and M. P. Wellman, "Price prediction strategies for market-based scheduling," in *Proc. 14th International Conference on Automated Planning and Scheduling,* Whistler, BC, 2004.

[51] C. Wang, H. Ghenniwa, and W. Shen, "Bidding languages for auction-based distributed scheduling," in *Proc. 2009 IEEE International Conference on Systems, Man, and Cybernetics* San Antonio, Texas, 2009, pp. 4518-4523.

[52] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on computers,* vol. 100, pp. 1104-1113, 1980.

[53] W. Shen, L. Wang, and H. Qi, "Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 36, pp. 563-577, 2006.

[54] W. Shen, "Distributed manufacturing scheduling using intelligent agents," *Intelligent Systems, IEEE,* vol. 17, pp. 88-94, 2002.

[55] S. Bikhchandani and J. M. Ostroy, "The package assignment model," *Journal of Economic Theory,* vol. 107, pp. 377-406, 2002.

[56] L. M. Ausubel and P. Milgrom, "Ascending auctions with package bidding," *Frontiers of Theoretical Economics,* vol. 1, pp. 1-42, 2002.