

On the Tradeoff between Privacy and Efficiency: A Bidding Mechanism for Scheduling Non-Commercial Services

Chun Wang, Farnaz Dargahi, Mohammad Fozlul Haque Bhuiyan

Concordia Institute for Information Systems Engineering

Concordia University, Montreal, Quebec, Canada

cwang@ciise.concordia.ca

Abstract *Services providers, such as public healthcare systems and government agencies, are under tremendous pressure to reduce costs and improve service quality. Scheduling is an important managerial component which has considerable impact on both the costs and quality of services. Service providers need customers' availability information to improve resource utilization. On the other hand, customers may be of "two minds" about communicating their private information. While communicating certain amount of availability might be necessary in order to obtain preferred schedules, too much communication place a potential cost due to privacy loss. In this paper, we present a bidding-based mechanism which aims at generating high quality schedules and, at the same time, protecting customers' privacy. We show that, under the proposed bidding procedure, myopic bidding is the dominant strategy for customers. We also evaluate the privacy and efficiency performance of the proposed mechanism through a computational study.*

Keyword: Non-commercial services, distributed scheduling, privacy, efficiency, iterative bidding, auction

1. Introduction

With increasing levels of globalization, organizations no longer compete directly, but rather compete based on their respective supply chains. The competitiveness of supply chains largely depends on effective information-sharing among supply chain partners. Having accurate information about inventory levels, orders, production, and delivery status provides a tremendous opportunity to improve the way the supply chain is designed and managed. However, information-sharing in a supply chain has proved difficult to implement [1]. The barriers are significant due to a lack of trust between trading partners. Companies fear that information voluntarily shared will be used against them or will leak to a competitor. Huge potential benefits of supply chain collaborations go unrealized due to a reluctance to share private information. Privacy becomes an obstacle on the road to achieving greater supply chain management efficiency.

Various approaches have been proposed to enable supply chain information-sharing under the constraint that trading partners have the desire to protect their privacy. One stream of them focuses on the design of incentive mechanisms. As the most successful application of mechanism design, auction theory has been used to facilitate supply chain collaboration in distributed settings [2][3][4]. To optimize the supply chain, auction mechanisms typically provide incentives which motivate trading partners to reveal their private information truthfully. In addition to auctions, cooperative game theory has also find applications in supply chain collaboration [5]. Another line of research looks at the problem from cryptographic perspective. Typically, secure multiparty computation techniques [6] are applied in supply chain collaboration settings [7][8]. There is also an emerging trend in developing intellectual property protection mechanisms against information leakage and reverse engineering in supply chain collaboration [9]. While most of the existing supply chain collaboration approaches have been developed for manufacturing sector, in this paper, we focus our attention on services supply chain. In particular, we study the privacy and efficiency issues arising in services collaboration.

Unlike more traditional manufacturing supply chains, service capacity in a supply chain is usually time dependent and typically cannot be held in inventory. For example, if a certain amount of seats on a flight are not booked, the transportation capacity of those seats on this particular flight will be wasted and cannot be stored in inventory for future usage. Without inventory as a buffer, matching capacity with demand is frequently more challenging in service supply chains [10]. In many business services, such as airline tickets booking, hotel room reservation, and vacation package booking, companies encourage potential customers to get a service time and price “quote” prior to ordering. By quoting different prices on service times, service providers can balance the demands throughout a week, a month, or seasons of a year, therefore, optimize their resource utilization. However, in non-commercial services industries, such as scientific facility services, government services and healthcare services, for social economic and political reasons, service providers cannot use dynamic pricing strategies to balance the demands along the service timelines. In this paper, we study the scheduling aspect of non-commercial service supply chain management. In particular, we are interested in knowing how to design effective mechanisms for services scheduling and how privacy and efficiency interplay under such mechanisms. We design an auction-based (iterative bidding, in particular) scheduling framework under two constraints (1) service providers are restrained from using any

price mechanisms to balance the demand and (2) customers are reluctant to share their complete availability information due to privacy concerns. The rest of the paper is organized as follows. Section 2 introduces the non-commercial services scheduling problem and its privacy implication. Section 3 formulates the service provider's and customers' decision problems. Section 4 presents a bidding-based scheduling framework and provides theoretical analysis on its properties. Section 5 evaluates the privacy and efficiency performance of the proposed scheduling framework. Section 6 concludes the paper and discusses future improvements.

2. Non-Commercial Services Scheduling and its Privacy Implication

The non-commercial service scheduling (NCSS) problem concerns the allocation of limited resources to the service activities at specific times. This allocation must obey a set of rules or constraints that reflect the temporal relationships between activities and the capacity limitations of a set of shared service resources. To motivate the research from a practical perspective, in this section, we first describe an example application domain where the proposed approach is needed. We then discuss the privacy implication of NCSS.

2.1. Synchrotron Facility Scheduling

Canadian Light Sources (CLS: <http://www.lightsource.ca/>), is a national science research laboratory for the production of high intensity synchrotron light from the infrared, visible, and ultraviolet to x-ray region of the electromagnetic spectrum and is accessible to scientists and researchers from the academic, government and private sectors. Currently, the CLS has about 3,000 researchers in Canada and other parts of the world as its user community. The CLS has two calls for proposals each year resulting in a scheduling cycle of 6 months. Proposals are evaluated by a scientific committee composed of researchers from universities and industries across the country. Each application is assigned a weight based on its potential contribution to the advancement of knowledge and impact on the scientific community. The proposals that are approved by the peer review procedure need to be scheduled in the next scheduling cycle. CLS needs to improve the utilization of expensive synchrotron resources and, at the same time, maximize the overall scientific contributions of the experiments. CLS knows the weight (scientific contribution) of each application. However, they do not have direct access to customers' availability information, which is privately held by customers. To compute a schedule, CLS relies on the availability information reported by its customers. Within each cycle,

CLS first schedule facility development and maintenance shifts which are not available for customers' experiments. After the shifts open to customers are determined, CLS will contact the researchers and allocate the open shifts to their experiments. The scheduling of experiment shifts is a multilateral negotiation process mainly done through phone calls and emails.

2.2. Privacy vs. Efficiency

NCSS can be modeled as an optimization problem in which private availability information of customers constraints the solution space. If all customers report their full availability, the service provider can obtain an optimal schedule by solving the optimization problem. However, if, for privacy reasons, the customers only reveal partial availability to the service provider, the quality of the solution will be compromised.

The lack of complete availability information can be a major constraint that limits the quality of the schedules. High quality schedules maybe determined infeasible given the partial availability information from customers. The scheduling problem facing the service providers is a decentralized scheduling problem [11] in the sense that the true availability of the customers is their private information and may not be known to the service provider. Customers are reluctant to reveal their complete availability because a complete revelation exposes too much privacy and increases the possibility of being assigned an undesirable time slot. Consider, for example, the synchrotron facility scheduling environment mentioned previously. Bob needs to conduct his experiment in the facility. He can be available any time from January to August. However, he prefers the experiment to be scheduled as earlier as possible because there is a possibility that he may go vacation sometime during the summer. Based on his previous experience and his knowledge of the profile of current year's applications, he believes that experiments with similar weight of his are likely to be offered a service time slot two months after the originally requested dates. Therefore, statistically, if he reports January to April as his available time window, he will have much higher chance ending up in June or even sometime earlier. After his calculation, Bob may report January to April which is not his complete availability.

Generating high quality schedules and, at the same time, accommodating customers' preference and privacy concerns is challenging. In addition to dealing with strategic behaviors from customers, the administrative workload of collecting customers' availability information and negotiate with them for possible changes can be very heavy due to the large number of

customers and a manually managed process. The proposed approach provides the possibility of automating the NCSS procedure and improving the quality of schedules. In the next section, we formulate the service provider's and customers' decision problems in NCSS.

3. Formulation of Service provider's and Customers' Decision Problems

Services scheduling is multilateral decision making with the service provider and customers as independent decision makers. The service provider needs to decide how to schedule service requests to achieve its objectives and, at the same time, respect the customer's availability constraints. The decision facing a customer is how much availability information she needs to reveal in order to maximize her benefit. In this section, we formulate the decision problems facing the service provider and customers.

3.1. Service Provider's Decision Problem

Consider a NCSS problem consists of a service provider and a group of customers. The provider receives a set of n service requests from customers. Each request is assigned a weight which reflects its contribution to the provider's objective. The provider has limited service capacity and knows the time required for processing a request. The provider's objective is to maximize the sum of the weights of a schedule. An important type of constraints of NCSS is the customers' availability. Since customers need to be present for the service, the provider cannot schedule a customer to a time slot during which she is not available. We describe a customer's availability by a set of available time intervals along the scheduling timeline. As we will later develop an iterative bidding framework for NCSS, we represent an available time interval as a bid from a customer using the bundle bidding language [12] developed for combinatorial auctions. To apply the language, we need to first discretize the provider's service timeline into fixed-size time units. In this way, without loss of generality, an available time interval can be defined by a bundle of adjacent time units contained in the interval. Different from the general combinatorial auctions, customers do not attach prices to their bids in NCSS. In our case, bids are used by the customers to indicate their availabilities. If a customer submit a bid (available time interval), she informs the provider that she is available to be scheduled during that interval. The set of intervals that contains a customer's complete availability is referred to as the customer's set of Feasible Time Intervals (FTI).

Let E_j be the set of availability intervals revealed by customer j . It is clear that E_j is a subset of customer j 's FTI. The service provider will not schedule customer j 's request outside her E_j . Let w_j be the weight scale of customer j assigned by the service provider and p_j the processing time of customer j 's request. Let Ω be the set of time units available for allocation and J be the set of customers who have service requests to be scheduled; let $x_j(B) = 1$ if the time unit bundle $B \subseteq \Omega$ is allocated to customer j and zero otherwise. The *provider's decision problem* is to determine the allocation of limited service time to the requests in a way that the sum of the weights of the awarded requests is maximized. The problem can be formulated as the following integer programming.

$$\max \sum_{B \subseteq \Omega} \sum_{j \in J} x_j(B) w_j$$

subject to

$$\sum_{B \subseteq \Omega} x_j(B) \leq 1, \quad \forall j \in J \quad (1)$$

$$\sum_{B \ni i} \sum_{j=1}^n x_j(B) \leq 1, \quad \forall i \in \Omega \quad (2)$$

$$\sum_{B \subseteq \Omega} x_j(B) = \sum_{B \subseteq E_j} x_j(B), \quad \forall j \in J, \forall B \in \Omega \quad (3)$$

$$|B| + Hx_j(B) \leq p_j + H, \quad \forall j \in J, \forall B \in \Omega \quad (4)$$

$$|B| + H \geq p_j + Hx_j(B), \quad \forall j \in J, \forall B \in \Omega \quad (5)$$

$$x_j(B) = \{0,1\}, \quad \forall B \in \Omega, \quad \forall j \in J \quad (6)$$

The set of constraints (1) ensures that any customer can only obtain one bundle of time units. The set of constraints (2) ensures that a time unit is not included into two bundles which have been assigned to the customers. The set of constraints (3) ensure that if a bundle is assigned to a customer, it must belong to the set of available intervals submitted by the customer. These constraints prevent service provider from assigning customers time bundles which they are not willing to accept. Constraints (4) (5) ensure that if a bundle is assigned to a customer, the length of the bundle is equal to the processing time of the customer's request, where H is a large positive constant, which is used for the linearization of the logical constraint "if." The minimum value of H depends on the problem instance. In general, a H that is greater than the number of available time units of the service provider is large enough to enforce the logical "if" constraint. Constraints (6) are integer constraints. The provider's decision problem is NP-hard as stated in the following theorem.

Theorem 1 *The service provider’s decision problem in NCSS is NP-hard.*

Proof: Consider a special case of the provider’s decision problem, in which a set of n service requests from customers need to be scheduled. A request may be scheduled on one of the l intervals on a discrete time scale on a single resource. The decision version of this special case of provider’s decision problem is identical to the *job interval selection problem*, which is NP-complete [13]. Therefore, the decision version of provider’s decision problem is NP-complete. It follows that provider’s decision problem is NP-hard ■.

3.2. Customers’ Decision Problem

To model customers’ decision problem, we first introduce their preference structure over the time intervals in their FTIs. A customer’s FTI is her private information not known to the service provider. She may behave strategically, for example hide part of her FTI, to maximize her benefits. To reflect this self-interest property of the customers, we call them agents¹. We assume that an agent prefers some time intervals over others within its FTI and the preferences can be quantified by associating a *preference violation cost* to each time interval. Preference violation cost reflects the level of the preference violations to the agent. It is essentially a subjective measure adopted by an agent. For example, it can be a function of the number and severity of preference violations that a time interval may cause to the agent. In many cases, it is reasonable to assume that an agent can order the time intervals in its FTI according to the increasing order of their *preference violation costs*. That is, given an ordered FTI, $c_1 \leq c_2 \leq c_3 \leq \dots c_k \dots \leq c_{|FTL|} < c_0$ is known to the agent, where c_k denotes the *preference violation cost* of the k th time interval in FTI and c_0 denotes the preference violation cost of not being allocated any time intervals (for the sake of simplicity, we drop the agent subscript in this subsection). Note that an agent may have identical preference violation costs for more than one time intervals. In a FTI, the highest preference violation cost is that of not being awarded in the service schedule. Anything beyond that is not included since it is not relevant to our NCSS.

An agent would prefer to be assigned a time interval with lowest preference violation cost. However, the final schedule is computed based on the submitted time intervals from all agents. Because of the potential time conflicts among agents’ requests, it is difficult for them to decide how much availability information it should reveal in order to obtain a preferred assignment. If

¹ 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”.

an agent only submits a few low cost time intervals, it can control the upper bound of its preference violation cost as the awarded bundle must be within the set of submitted intervals. However, by doing so, it runs the risk of not being allocated anything if the submitted time intervals are also demanded by other agents with higher weights. On the other hand, if an agent submits its complete FTI, it maximizes its probability of getting an assignment. However, reporting complete FTI increases the possibility of ending up with an interval with high preference violation cost. In fact, there is not a clear strategy for agents to minimize their expected preference violation costs. The effectiveness of an agent's bidding strategy depends on how heavy the competition is for its desired time intervals and other agents' bidding strategies. This uncertainty leads to speculation during bidding, which will increase agents' computation cost and may render final schedule arbitrarily far from optimal. Our goal, therefore, is to design a mechanism which systematically evolves the solution towards an optimal one given the constraint that agents try to avoid high cost assignments by not revealing their complete availability. Since no payment is allowed in the NCSS setting, the possibility of applying standard one-shot VCG mechanisms [14][15][16] and even its iterative implementations [17] is eliminated. In the following section, we will propose a non-price bidding approach to the NCSS problem. We will also evaluate the performance of the approach in Section V.

4. The Iterative Bidding Framework

The bidding framework proposed in this paper is an iterative bundle auction. It contains two major components: an iterative bidding procedure and an integer programming model for winner determination. The winner determination model computes provisional schedules which maximize the sum of the weights of winning bids at each round. The iterative bidding procedure provides a structure for the agents and the service provider (auctioneer) to interact in a systematic way and eventually evolve the provisional solutions towards an 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 single-attribute auction, which allows negotiation over a non-price attribute: the level of availability of agents revealed to the auctioneer. The framework has good privacy preserving properties. It requires agents' revelation of their availability only on a necessary to reveal basis.

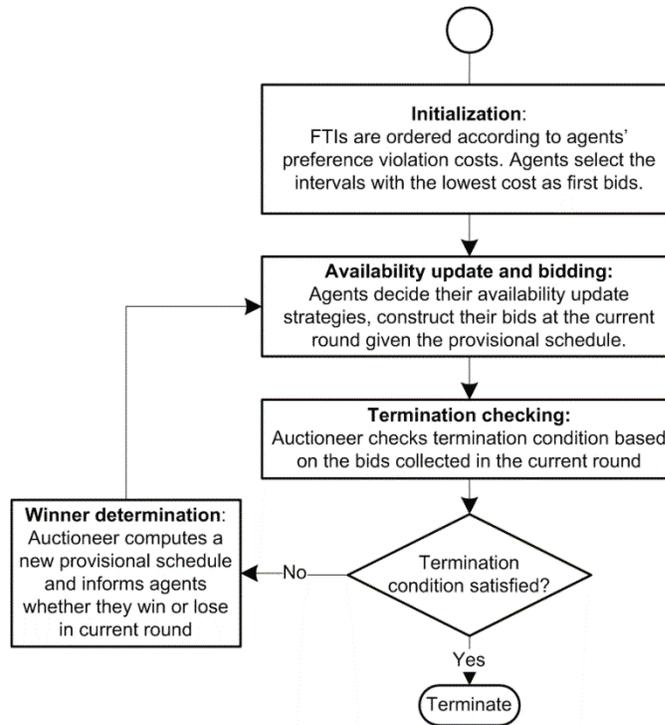


Figure 1. Flow chart of the iterative bidding procedure for NCSS problems

4.1. Iterative Bidding

The iterative bidding procedure is depicted as a flow chart in Figure 1. The bidding procedure consists of four components, namely initialization, availability update and bidding, termination checking and winner determination. The auctioneer first publishes its open time units to agents and subsequently invites agents to submit their preferred time intervals. The iterative bidding is a collaborative interaction procedure between the auctioneer and the agents.

4.1.1. Initialization

Initially, an agent has a service request and knows its FTI. The agent constructs its initial bid by selecting the available time intervals with lowest preference violation cost and sends them to the auctioneer. If a bid contains more than one time intervals, they are connected by XOR logical connector meaning that the agent is willing to be scheduled in either of the intervals and the agent is indifferent from the intervals in terms of the preference violation cost.

4.1.2. Availability Update and Bidding

After the provisional schedule which resulted from the winner determination at round $t - 1$ is presented to the agents, at the beginning of round t ($t > 1$), an agent needs to decide whether it submits additional time intervals to the auctioneer at round t . If an agent was not awarded in the

provisional schedule at round $t - 1$, it has two availability update options at round t : (1) it can submit additional time intervals; (2) it can also keep the set of submitted time intervals unchanged by submitting an *empty bid* (a bid does not contain any time intervals). However, if an agent does so, the auctioneer will consider the agent has entered into the final bid status and the agent is forbidden from updating its availability in future rounds. On the other hand, if an agent won in the last round, it also has the option of submitting additional time intervals or submitting an empty bid. However, in this winning case, the agent does not enter into the final bid status.

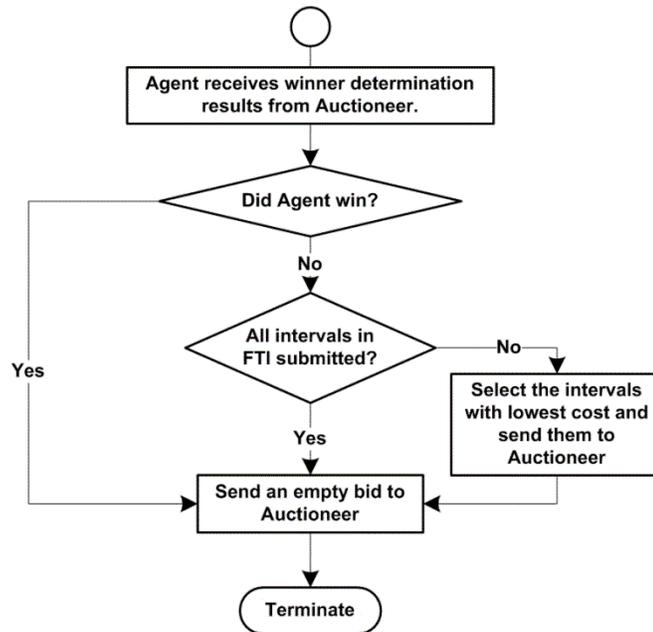


Figure 2. Agents' myopic bidding strategy at a specific round

Although the auction rule allows an agent to bid differently in both the losing and winning cases, we will show in the next section that, since agents are assumed to be rational in minimizing their preference violation costs, they will always follow the *myopic bidding strategy* which is described in Figure 2. The myopic bidding strategy says if an agent won during the last round, it will submit an empty bid at the current round; if the agent lost, it will check whether all the intervals in its FTI have been submitted. If yes, the agent will still submit an empty bid because there are no more available time intervals to be added; if no, the agent will select the ones with the lowest cost from its unrevealed part of the FTI and submit them to the auctioneer.

4.2. Bids Screening and Termination Checking

Once bids are received from the agents, the auctioneer first screens out invalid ones. Those bids will not be considered in the following winner determination procedure. Invalid bids are defined as containing (1) time intervals which have been submitted in the previous rounds; (2) new time intervals from agents who have already declared their final bidding status in previous rounds; and (3) a time interval which is shorter than the processing time of the agent's request.

The auctioneer then checks the termination condition against the valid bids. The auction terminates if there is no new availability added. That is, each agent that bid in the last round has either submitted an empty bid or withdrawn from the bidding process. After the auction terminates, the auctioneer implements the final schedule. If the termination condition is not satisfied, the auctioneer will update agents' availability information by adding the newly submitted time intervals to those already submitted in previous rounds and solve the winner determination model using the updated availability information as input.

4.3. 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 provider's decision problem model with updated availability from all bidding agents. It is possible that there exist multiple schedules with the same optimal overall weight. Which optimal schedule the auctioneer will find is determined by a combination of many factors, such as the design and configuration of the winner determination algorithm and the organization pattern of the input data. After winner determination, the auctioneer will inform all bidding agents with the results regarding whether they win or lose at round t . After receiving the results, the agents will decide their strategy on availability updating and start a new round of bidding. It is important to note that the winner determination model here is different from that of many other combinatorial auctions, in which the losing bids will not be considered in future rounds [18]. In our model, the bid from an agent is just a new addition to its already submitted availability. When computing the provisional schedule, winner determination algorithm will consider all time intervals submitted from an agent during current and previous rounds. In addition, the provisional schedule is determined by the updated availability at current round. It is not affected by the sequence of bidding in previous rounds.

4.4. Properties of the Iterative Bidding Framework

In the design of the iterative bidding framework, we have assumed that agents bid according to the myopic bidding strategy described in Figure 2. As agents are self-interested, a question arises naturally: will the agents really follow the myopic strategy? We now study the iterative bidding framework from incentive compatibility perspective. We prove that the myopic bidding strategy we have assumed is the dominant strategy for agents as stated in the following proposition.

Proposition 1 *Given the proposed iterative bidding mechanism, myopic bidding is the dominant strategy for agents.*

Proof: It's clear that if an agent has already been awarded in the previous round, there is no reason for it to add new time intervals in the current round because more availability increase the upper bound of its preference violation cost. Therefore, it will follow the myopic strategy by reporting an empty bid. Let's now consider the situation where the agent is not awarded in the previous round. Assume that the agent has reported first $k - 1$ time intervals in its FTI during the previous rounds. If the agent follows myopic strategy, it should add the k th time interval at the current round and update its availability to first k time intervals. To compare with the myopic strategy, we construct an *alternative strategy*, in which the agent reports first $(k + 1)$ time intervals. In the following we first prove that the myopic strategy weakly dominates the alternative strategy. We consider three cases:

Case#1: The agent is not awarded in the current round, no matter it submits first k or $k + 1$ time intervals. In this case, both first k and first $k + 1$ time intervals end up with the same preference violation cost which is c_0 . There is no difference between the myopic and the alternative strategies.

Case#2: The agent is awarded by submitting first k time intervals. In this case, the agent must be awarded by reporting first $k + 1$ time intervals because first k is a subset of first $k + 1$. Since the awarded time bundle can fall into any one of the submitted time intervals, we compare the expected preference violation cost of the myopic strategy and the alternative strategy. Let a_k denotes the number of available time intervals, each of which costs c_k . Since $c_1 \leq c_2 \leq c_3 \leq \dots \leq c_k \leq c_{k+1}$, we know that $c_{k+1} \sum_{i=1}^k a_i \geq \sum_{i=1}^k a_i c_i$. Since $a_{k+1} > 0$, it follows that $a_{k+1} c_{k+1} \sum_{i=1}^k a_i \geq a_{k+1} \sum_{i=1}^k a_i c_i$. Adding $\sum_{i=1}^k a_i \sum_{i=1}^k a_i c_i$ to both sides of the inequality we have $\sum_{i=1}^k a_i \sum_{i=1}^k a_i c_i + a_{k+1} c_{k+1} \sum_{i=1}^k a_i \geq \sum_{i=1}^k a_i \sum_{i=1}^k a_i c_i + a_{k+1} \sum_{i=1}^k a_i c_i$, that is

$$\sum_{i=1}^k a_i \sum_{i=1}^{k+1} a_i c_i \geq \sum_{i=1}^{k+1} a_i \sum_{i=1}^k a_i c_i, \text{ which is equivalent to } \sum_{i=1}^{k+1} a_i c_i / \sum_{i=1}^{k+1} a_i \geq \sum_{i=1}^k a_i c_i / \sum_{i=1}^k a_i.$$

Since adding the time interval $k + 1$ will increase the feasible schedule space of the winner determination, therefore, will not change the value of a_1, a_2, \dots, a_k , the left hand side of the last inequality can be interpreted as the expected cost of reporting first $k + 1$ intervals and the right hand side can be interpreted as the expected cost of reporting only first k intervals. It is clear that when the agent can be awarded by just reporting first k intervals. The myopic strategy always leads to less (or equal) expected preference violation cost than the alternative strategy does.

Case#3: The agent is not awarded by reporting first k intervals, but awarded by reporting first $k + 1$ intervals. In this case, although by the myopic strategy, the agent is not awarded at the current round, it always has the option to report first $k + 1$ by repeatedly applying the myopic strategy in the next round. Given that the sequence of bidding does not affect the winner determination result, that is, the same set of availability intervals will result in the same provisional schedule, the agent will not lose any opportunity by adopting the myopic strategy.

It follows that the myopic strategy weakly dominates the alternative strategy with first $k + 1$ time intervals. This conclusion also applies to the initial round of bidding. Since the provisional schedule before the initial round is empty, which can be interpreted as no agent is allocated a bundle. Therefore the best strategy for agents' initial round bidding is myopic strategy. That is, at the first round, an agent should bid with its lowest cost time intervals in its FTI. By mathematical induction, it follows that, myopic bidding is the dominant strategy for agents given the proposed iterative bidding mechanism ■.

4.5. Iterative Bidding with Partial Allocation during Each Round

The iterative bidding procedure we have proposed computes provisional allocation during each round. It does not permanently award time bundles to customers until the termination condition is reached. The procedure may reach higher solution quality since it collects more agents' availability along the process of bidding. However, as the bidding proceeds, the size of the winner determination problem will increase continuously. Since we have proved that the winner determination problem is NP-hard, for a service scheduling problem with larger amount of customers, winner determination will be slowed down considerably as more availability

information is added. As a variant of the proposed iterative bidding procedure, we can actually award the provisional allocation to the customers during each round. In the subsequent round, those awarded time intervals will be removed from the service time inventory, the awarded customers will withdraw from the bidding process, and the customers who are not awarded in the current round will construct their bids based on the updated inventory. The service provider will solve the winner determination problem formulated by the updated inventory and the bids submitted in the current round. In this case, the size of winner determination problem decreases along the iterations as both the number of provider's inventory and the number of bidding agents decrease. The bidding terminates in lesser rounds than the original procedure does. In the next section, we will study the relationship between the efficiency of a solution and the privacy loss of customers under the iterative bidding procedures through a computational study.

5. Privacy and Efficiency Analysis: A Computational Study

By designing an iterative bidding framework, agents reveal their availability information as necessary. Also, higher system transparency makes the adoption of the framework easier. However, these benefits are obtained with a cost of efficiency. If at the termination of bidding all agents have revealed their full availability, the winner determination algorithm will compute an optimal schedule which maximizes the sum of the weights of awarded agents. However, when bidding terminates before all feasibility information becomes known to the auctioneer, the optimality of the solution is not guaranteed. In this section, we evaluate the privacy and efficiency performance of the proposed approach through a computational study. Given a solution schedule, the measure of its efficiency is defined as the ratio between its overall weight and that of an optimal solution for the same problem instance. The measure of privacy loss is the ratio of the revealed availability of all agents when the solution is reached and their complete availability. Intuitively, submitting more availability incurs high privacy loss, which increases the expected preference violation cost. We use ILOG CPLEX 12.1 (<http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>) as optimization engine for solving the winner determination model given the set of bids from agents as the input. The iterative bidding control logic is coded using the OPL Script language [19]. The control module and the optimization engine are integrated using the ILOG OPL environment ([14](http://www-</p></div><div data-bbox=)

01.ibm.com/software/integration/optimization/cplex-optimization-studio). All experiments were conducted on a PC with a 2.4 GHz CPU and 4 GB memory.

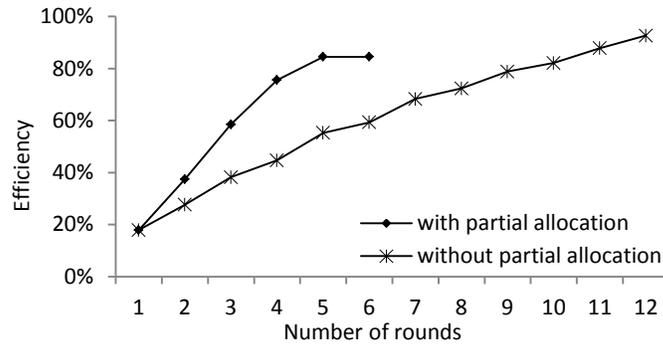


Figure 2. Efficiency increment during iterative bidding

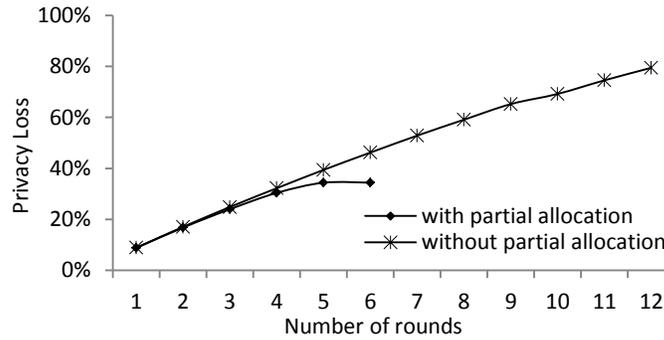


Figure 3. Privacy loss increment during iterative bidding

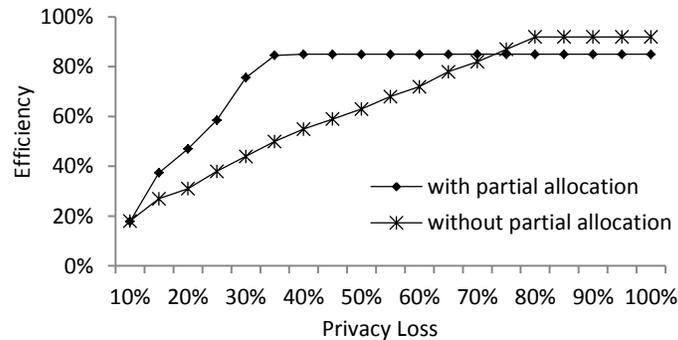


Figure 4. Tradeoff between efficiency and privacy loss

We generate a set of test problem instances by fixing the service provider’s time units inventory to 20 and the number of customers to 50. Customers’ weights are drawn from a uniform distribution in the range of 1 to 3. The processing times for agents’ requests are identical and restricted to one time unit. For each agent, we randomly select the set of time intervals from

the service provider's available time inventory to form its FTI. The sizes of agents' FTIs are drawn from a uniform distribution in the range of 8 to 16 with a mean of 12. The length of the time intervals in FTIs is also restricted to one. The time intervals in FTIs are randomly ordered. We solve the set of problem instances using the proposed iterative bidding framework and compute the average efficiency and privacy loss at each round of bidding. The bidding processes without partial allocation usually terminate within 12 rounds which is the mean of the size of FTIs. The bidding processes with partial allocation usually terminate within 6 round, which are, as expected, much faster than the bidding without partial allocation case.

Figures 3 and 4 show the efficiency and privacy loss increment during the bidding process. At round 6, the modified bidding procedure with partial allocation achieves on average 84% efficiency, whereas the original bidding procedure without partial allocation achieves on average 93% efficiency at round 12. The bidding procedure with partial allocation is essentially a greedy distributed search algorithm which can find a solution quickly. However, the solution quality can be compromised. The bidding procedure without partial allocation involves backtracking. It normally reaches higher quality solution, however, with more rounds of bidding. From Figure 4, we see that the privacy loss of bidding with partial allocation is always lower than that of bidding without partial allocation and the difference increases along the bidding process. Compared with bidding without partial allocation, it seems that bidding with partial allocation can find a reasonably good solution with much less computation costs and privacy loss. Figure 5 shows the tradeoff between efficiency and privacy loss. We see that high efficiency demands more privacy, which is natural. It is observed that increasing privacy loss has a diminishing return in efficiency. Bidding with partial allocation can reach 84% efficiency with only 34% privacy loss, whereas bidding without partial allocation needs to double the privacy loss (70%) in order to reach the same efficiency level. For bidding without partial allocation, a solution with 93% efficiency demands 79% privacy loss. Since agents decide when to stop submitting more availability information to the auctioneer, the bidding procedure actually provides agents with the option of setting their respective privacy loss limits based on their own calculation of the costs caused by privacy loss. In this experiment, we did not consider the situation where agents have privacy loss limits. However, Figure 5 gives an indication regarding the efficiency we can reach given various levels' loss of privacy.

Table 1 configuration of scalability testing problems and computational Results

Group	Num. of agents	Num. of available time units for bidding	Run time (seconds)
1	100	20	0.36
2	200	20	0.42
3	300	20	0.68
4	300	40	1.08
5	400	20	1.52
6	500	40	2.16
7	800	40	2.67
8	1000	40	3.48

The responsiveness and scalability of the proposed bidding framework are mainly determined by the computation time needed to solve the winner determination problem at each round of bidding (we assume that the computation needed for an agent to order its time intervals in FTI is trivial). To evaluate the effectiveness of using ILOG CPLEX 12.1 for solving the winner determination problems, we have randomly generated eight groups of problem instances with different sizes and structures. Weights of agents and sizes of FTIs are drawn from the same distributions as previously generated problem sets. The configuration of the test problem sets and the corresponding solving time by CPLEX 12.1 are summarized in Table 1. Based on the run time results, it is clear that the proposed iterative bidding framework can be practically applied to large scale NCSS problems. For example, CPLEX 12.1 can solve a winner determination problem with 40 available time units and 1000 agents in 3.48 seconds. In this case, if the mean of the size of agents' FTIs is 12, which is quite reasonable in many application domains, the overall bidding procedure will likely to terminate within 45 seconds. This level of responsiveness is sufficient for most of the NCSS applications.

6. Related work

There has been growing research efforts on services supply chains. Many of them extrapolate lessons learned in the manufacturing sector to the service sector [20]. For decentralized scheduling problems, economic based mechanisms have been proposed to facilitate information sharing and achieving high efficiency given the self-interested nature of participants [21]. These approaches usually model processing times of resources as goods to be sold in the market, e.g. landing timeslots of airport runways [22], machine processing times of a factory [11], computation and network accessing times of internet resources [23], and the right to use railroad tracks [24]. The market, then balance the demand and supply using various price mechanisms.

Different types of auctions have been proposed for decentralized scheduling in the literature. While giving a comprehensive review of these models is beyond the scope of this paper, we summarize three typical auctions which are of importance to decentralized scheduling. The core issue here is how to accommodate the complementarity of agents' preferences over bundles of time units.

Sequential and simultaneous auctions price bundles as the sum price of the individual items. They do not allow bidders to bid on bundles. 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 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 [25]. In [24], 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 [26]. Sequential and simultaneous 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 worthy of further study.

Combinatorial auctions (CAs) allow bidders to place bids on bundles of items. It addresses complementary preference issue explicitly. However, the computation required for solving 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 [18]. Some sophisticated algorithms have produced promising results [27].

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 only 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 [28], iterative auctions are applied to the job shop scheduling problem. The focus in [28] is to investigate the links between combinatorial auctions and Lagrangean relaxation, and to design auctions based on the Lagrangean based decomposition. In [11], the properties of several iterative auction protocols are investigated in the context of decentralized scheduling. In [29] and [30] 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 scheduling 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 required in many real-world scheduling applications. The sequential, simultaneous, combinatorial and iterative bundle auctions are important models for accommodating complementary preferences of agents. However, they require price or payment mechanisms, therefore cannot be directly applied to our NCSS setting.

Secure multiparty computation (SMC) protocols enable a group of mutually distrustful parties to perform a joint computation with private inputs. In theory, trading partners could benefit from strong privacy protection by using SMC protocols in supply chain collaboration. However, the uptake of SMC in practical applications is still rare [31]. There could be three lines of explanations. Catrina and Kerschbaum argue that this is due to poor performance, functionality, and scalability, as well as architectures that do not meet the needs of the applications. At a more general level, Goldreich states in [32] that although the general secure multi-party computation problem is solvable in theory, using the solutions derived by these general results for special cases can be impractical. In other words, efficiency dictates that development of special solutions for special cases for efficiency reasons. In addition, secure multi-party protocols are a form of cooperative distributed computing. They preserve the privacy of the participants' data. However, they assume cooperative behavior of participants. In other words, participants are willing to supply true and complete information to the secure protocol. Game theoretic behavior of participants is not explicitly modeled in SMC, which restricts its application to supply chain collaborations. For example, secure protocols for discriminatory and non-discriminatory e-

auctions are proposed in [7]. These protocols allow buyers to reveal only partial information of their price-quantity pair bids to the supplier. However, incentives are still needed to motivate buyers from hiding or misreporting their true private information. In our services scheduling model, customers can behave strategically by hiding availability information. Furthermore, SMCs usually do not assume any trusted center, which is not the case in our services scheduling setting.

7. Conclusion

In recent years, the economy has evolved from manufacturing to services. Service supply chain management has become an important research area with significant practical implications. Scheduling non-commercial services for self-interested customers who behave strategically to protect their privacy is a challenging problem attributed to the different objectives of service provider and the customers. In non-commercial services scheduling environments, no payment transfers are allowed, which eliminates the possibility of designing price or payment based mechanisms to balance the supply and demand. We propose a bidding framework for scheduling non-commercial services and evaluate its efficiency and privacy performance through theoretical analysis and computational experiments. We show that, under the proposed auction mechanism, myopic bidding is the dominant strategy for customers. In terms of the efficiency and privacy performance, the computational study shows that bidding with partial allocation can find a reasonably good solution with much less computation costs and privacy loss. For both cases of bidding with and without partial allocation, increasing privacy loss has a diminishing return in efficiency. Experimental results also show that the proposed framework scales well to large size problems.

For future research, we will study the applicability of the proposed bidding framework to various types of services scheduling domains. We plan to build an integrated simulation environment with interactive user interfaces for both customers and the service provider. We believe that such a simulation environment can provides more insights in terms of the role that privacy plays in service scheduling.

Reference

- [1] R. E. Sabath and J. Fontanella, "The unfulfilled promise of supply chain collaboration," *Supply Chain Management Review*, vol. 6, pp. 24-29, 2002.

- [2] R. R. Chen, R. O. Roundy, R. Q. Zhang, and G. Janakiraman, "Efficient auction mechanisms for supply chain procurement," *Management Science*, pp. 467-482, 2005.
- [3] M. Babaioff and W. E. Walsh, "Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation," *Decision Support Systems*, vol. 39, pp. 123-149, 2005.
- [4] M. Fan, J. Stallaert, and A. B. Whinston, "Decentralized mechanism design for supply chain organizations using an auction market," *Information Systems Research*, vol. 14, pp. 1-22, 2003.
- [5] M. Nagarajan and G. Susic, "Game-theoretic analysis of cooperation among supply chain agents: Review and extensions," *European Journal of Operational Research*, vol. 187, pp. 719-745, 2008.
- [6] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982, pp. 160-164.
- [7] M. J. Atallah, H. G. Elmongui, V. Deshpande, and L. B. Schwarz, "Secure supply-chain protocols," in *Proceedings of the IEEE International Conference on E-Commerce*, 2003, pp. 293-302.
- [8] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara, "Private collaborative forecasting and benchmarking," in *Proceedings of the 2004 ACM workshop on Privacy in the electronic society 2004*, pp. 103-114.
- [9] D. Y. Zhang, Y. Zeng, L. Wang, H. Li, and Y. Geng, "Modeling and evaluating information leakage caused by inferences in supply chains," *Computers in Industry*, vol. 62 pp. 351–363, 2011.
- [10] E. Simchi-Levi and P. Kaminsky, *Designing and managing the supply chain: concepts, strategies, and case studies*: Irwin/McGraw-Hill, 2003.
- [11] 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.
- [12] N. Nisan, "Bidding languages for combinatorial auctions" in *Combinatorial Auctions*, Cramton, Shoham, and Steinberg, Eds.: MIT Press, 2006.
- [13] J. M. Keil, "On the complexity of scheduling tasks with discrete starting times" *Operations Research Letters*, vol. 12, pp. 293-295, 1992.
- [14] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, 11(1):17-33, 1971.
- [15] T. Groves, "Incentives in Teams," *Econometrica*, 41(4):617-631, 1973.
- [16] W. S. Vickrey, Counterspeculation, "auctions, and competitive sealed tenders." *Journal of Finance*, 16(1):8-37, 1961.
- [17] D. Parkes, "Iterative combinatorial auctions," in *Combinatorial Auctions*, P. Cramton, Y. Shoham, and R. Steinberg, Eds.: Cambridge, MA: MIT Press, 2006, pp. 41–77.
- [18] S. de Vries and R. V. Vohra, "Combinatorial auctions: a survey," *INFORMS Journal on Computing*, vol. 15, pp. 284-309, 2003.
- [19] P. Van Hentenryck and L. Michel, "OPL script: Composing and controlling models," in *Lecture Note in Artificial Intelligence (LNAI1865)*: Springer Verlag, 2000, pp. 75-90.
- [20] K. Sengupta, D. R. Heiser, and L. S. Cook, "Manufacturing and service supply chain performance: a comparative analysis," *Journal of Supply Chain Management*, vol. 42, pp. 4-15, 2006.
- [21] B. Heydenreich, R. Müller, and M. Uetz, "Games and Mechanism Design in Machine Scheduling—An Introduction," *Production and Operations Management*, vol. 16, pp. 437-454, 2007.
- [22] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A Combinatorial Auction Mechanism for Airport Time Slot Allocation," *The Bell Journal of Economics*, vol. 13, pp. 402-417, 1982.
- [23] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in Grid computing," *Concurrency and Computation: Practice and Experience*, vol. 14, pp. 1507-1542, 2002.
- [24] D. C. Parkes and L. H. Ungar, "An auction-based method for decentralized train scheduling," *Proceedings of the fifth international conference on Autonomous agents*, 2001, pp. 43-50.
- [25] R. P. McAfee and J. McMillan, "Analyzing the airwaves auction," *The Journal of Economic Perspectives*, vol. 10, pp. 159-175, 1996.

- [26] P. Milgrom, "Putting auction theory to work: the simultaneous ascending auction," *Journal of Political Economy*, vol. 108, pp. 245-272, 2000.
- [27] 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.
- [28] E. Kutanoglu and S. D. Wu, "On combinatorial auction and Lagrangean relaxation for distributed resource scheduling," *IIE Transactions*, vol. 31, pp. 813-826, 1999.
- [29] J. K. Mackie-Mason, A. Osepashvili, 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.
- [30] D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepashvili, "Exploring bidding strategies for market-based scheduling," *Decision Support Systems*, vol. 39, pp. 67-85, 2005.
- [31] O. Catrina and F. Kerschbaum, "Fostering the Uptake of Secure Multiparty Computation in E-Commerce," in *Availability, Reliability and Security*, 2008. ARES 08. Third International Conference on, 2008, pp. 693-700.
- [32] O. Goldreich, "Secure multi-party computation" (working draft). Available from, <http://www.wisdom.weizmann.ac.il/~oded/pp.html>, 2002.