

DESIGNING REAL-TIME FEEDBACK FOR BIDDERS IN HOMOGENEOUS-ITEM CONTINUOUS COMBINATORIAL AUCTIONS¹

Gediminas Adomavicius and Alok Gupta

Carlson School of Management, University of Minnesota, Minneapolis, MN 55455 U.S.A.
{gedas@umn.edu} {gupta037@umn.edu}

Mochen Yang

Kelley School of Business, Indiana University, Bloomington, IN 47405 U.S.A.
{yangmo@iu.edu}

Although combinatorial auctions are important mechanisms for many specialized applications, their adoption in general-purpose marketplaces is still fairly limited, partly due to the inherent difficulty in evaluating the efficacy of bids without the availability of comprehensive bidder support. In this paper, we present both theoretical results and computational designs to support real-time feedback to bidders in continuous combinatorial auctions, where bidders are free to join and leave the auction at any time. In particular, we focus on the broad class of single-item multi-unit (SIMU) combinatorial auctions, where multiple identical units of one homogenous item are being auctioned. We also consider two common ways to express bidding preferences: OR bids and XOR bids. For SIMU auctions with each of the two bid types, we present comprehensive analyses of auction dynamics, which can determine winning bids that satisfy allocative fairness, and compute critical evaluative metrics needed to provide bidder support, including bid winning and deadness levels. We also design the data structures and algorithms needed to provide bidder support in real time for SIMU auctions of practically relevant sizes. The computational tools proposed in this paper can facilitate the efficient and more transparent implementation of SIMU combinatorial auctions in business- and consumer-oriented markets.

Keywords: Continuous combinatorial auctions, multi-unit auctions, electronic markets, real-time bidder support, allocative fairness, computational artifact design

Introduction and Motivation

Rapid advances in computing capabilities and the ubiquitous nature of the Internet have provided strong impetus for designing and deploying complex electronic markets. In particular, the emergence of “smart markets” (Bichler et al. 2010) with superior allocative efficiency are made possible by

theory-based *computational* tools that facilitate market participants to make *real-time* decisions. One prominent example of smart market mechanisms is *combinatorial auctions*, where bidders can bid on combinations of items (a.k.a. *packages* or *bundles*). Compared to traditional single-object auctions, combinatorial auctions enable bidders to express complex preferences, such as complementarities and substitutions, and therefore can achieve higher allocative efficiency (Banks et al. 1989; Cramton et al. 2006; Petrakis et al. 2012).

Despite the advantages of combinatorial auctions, its adoption in electronic markets (especially consumer-oriented markets)

¹Jeffrey Parsons was the accepting senior editor for this paper. Xiao Fang served as the associate editor.

The appendices for this paper are located in the “Online Supplements” section of *MIS Quarterly*’s website (<https://misq.org>).

has been limited so far (Adomavicius, Curley et al. 2012), partly due to its difficulty. First, the winner determination problem in such auctions is computationally intractable, that is, NP-hard (Rothkopf et al. 1998, Sandholm 1999). Second, bidders are usually not able to determine appropriate competitive information to evaluate and construct their bids, due to the prohibitive cognitive complexity of such auctions (Adomavicius, Curley et al. 2012; Bichler et al. 2017, Eso et al. 2005; Porter et al. 2003). While in single-unit auctions (e.g., English auctions) a bidder that is not the highest bidder must outbid the current highest bid to have a chance of winning the auction, this is not necessarily the case in combinatorial auctions (i.e., a bid that is not among the currently winning bids can be among the future winners simply based on the combinations of later bids). As a result, it is hard for bidders to know how much to bid on certain bundles in order to win the auction.

Over the years, the increasingly powerful computational resources as well as a large number of research advances have effectively mitigated the first obstacle, that is, determining winners within feasible time (e.g., Rothkopf et al. 1998; Sandholm 1999, 2002; Tennenholtz 2000; Xia et al. 2005). However, the second obstacle (i.e., providing critical information that supports bidders to make informed decisions) remains challenging. A notable stream of literature that aims to address this issue focuses on designing *iterative* combinatorial auctions, that is, combinatorial auctions that run iteratively for multiple rounds, with auctioneers announcing the intermediate results following each round (Ausubel et al. 2005; Bichler et al. 2009; Goeree and Holt 2010; Kelly and Steinberg 2000; Parkes 1999; Pekec and Rothkopf 2003; Scheffel et al. 2011). These iterative combinatorial auctions typically impose certain prespecified rules and restrictions to create well-defined rounds of bidding.

As a contrast, the primary goal of this paper is to address the issue of real-time bidder support for *continuous* combinatorial auctions (i.e., auctions that facilitate a completely free-flowing, asynchronous, and transparent participation of bidders). In other words, bidders in continuous combinatorial auctions are free to join, leave, or bid at any time with no restrictions. Adomavicius and his colleagues (Adomavicius Curley et al. 2012, 2013a; Adomavicius, Sanyal et al. 2007) have argued that continuous combinatorial auctions are desirable in modern marketplaces (e.g., see the VicForests timber supply auctions conducted on the TradeSlot platform for a real-world example; TradeSlot 2017).

Providing real-time bidder support in such auctions is of pivotal importance for several reasons. First, without bidder support, both bidders and auctioneers would have little incentive to *adopt* a continuous combinatorial auction

mechanism at all (Adomavicius, Curley et al. 2013b). Real-world bidders only have bounded rationality and limited cognitive and computational capacity (Simon 1982), and they cannot be automatically assumed to behave fully rationally. Formulating reasonable bidding strategies in combinatorial auctions is very complicated (Banks et al. 1989; Bichler et al. 2017; Pekec and Rothkopf 2003). Therefore, having access to up-to-date information about the status of their bids *during* the auctions is imperative for bidders to make informed bidding decisions and participate in the auctions in a continuous manner.² Second, providing real-time bidder support makes the auction environment more transparent and user-friendly, which can improve the economic outcomes of the auctions. The benefit of having greater transparency and support in complex markets has been demonstrated in several studies. For example, Adomavicius, Curley et al. (2013a) showed that providing bidder support in multi-item auctions can significantly increase allocative efficiency. Adomavicius, Gupta, and Sanyal (2012) found similar benefits for more complicated multi-attribute combinatorial auctions. More generally, information transparency (i.e., disclosure of important market information) is a key dimension in designing many electronic markets (Granados et al. 2010; Granados and Gupta 2013). In addition, providing bidder support in *real-time* can also benefit the convergence of the auctions (see Appendix D for an illustration). Third, while the literature on auction theories often focuses on the game-theoretical properties of certain auction mechanisms (e.g., equilibrium strategies), such direction has proven extremely hard for combinatorial auctions (e.g., Pekec and Rothkopf 2003). In these complex auctions, bidders must first have access to basic information reflecting the status of the auctions before any sophisticated bidding behaviors can arise. All of these considerations motivate us to design a real-time bidder support system to facilitate adoption, participation, and transparency in continuous combinatorial auctions.

It is important to distinguish different types of combinatorial auctions, as they exhibit different properties with respect to real-time bidder support. Two dimensions are of interest, based on (1) general type of items that are being sold and (2) bid types (or bidding language) supported by the auction.

²Bidder support is deemed a necessity even for professional or expert bidders. Many auction service companies (e.g., TradeSlot, Optimal Auctions, and Power Auctions LLC) serving high-stake industrial combinatorial auctions (e.g., multi-item spectrum auctions or multi-unit industrial procurement auctions) feature the provision of bidder support as a critical comparative advantage. Furthermore, bidder support as a key component of real-world combinatorial auction design has also been extensively discussed in major academic books on the topic (e.g., Bichler and Goeree 2017; Cramton et al. 2006).

First, in terms of the two general, canonical item types, while Multi-Item Single-Unit (MISU) combinatorial auctions involve auctioning a set of *heterogeneous* (i.e., distinct) items, one unit for each, Single-Item Multi-Unit (SIMU) auctions involve auctioning *homogeneous* items (i.e., multiple identical units of the same item) (Kothari et al. 2005; Maskin and Riley 1989). These two types of combinatorial auctions are basic, yet drastically opposite in terms of feasible sets of items that a bidder may bid on. While bidders in a MISU auction choose the bundle of items to bid on, bidders in a SIMU auction choose *the number of units* to bid on. Both MISU auctions and SIMU auctions have numerous real-world applications. The applications of MISU auctions include the allocation of spectrum licenses for wireless communications (Cramton 2013), industrial procurement (Bichler et al. 2006), truckload transportation (Caplice 2007), airport time slot allocation (Rassenti et al. 1982), and railroad track allocation (Brewer and Plott 1996). The application of SIMU auctions includes commodity auctions (a fixed number of identical units of a homogeneous commodity is being auctioned; Ausubel and Cramton 1995), procurement contract auctions (suppliers bidding for the right to sell multiple units of a particular material to the buyer; Milgrom 2000; Narahari and Dayama 2005), network capacity auctions (auctioning network bandwidth; Murillo et al. 2008), and pollution-rights auctions (e.g., selling sulfur dioxide or CO₂ emission quotas; Leyton-Brown et al. 2000). Combining these two basic auction types is the most general form of combinatorial auction—Multi-Item Multi-Unit (MIMU) auctions—where multiple heterogeneous items are being auctioned, and multiple identical units are available for each item.

Second, in terms of supported bid types, a number of possibilities have been discussed in the literature (Nisan 2000). The two most common types are OR bids and XOR bids. Under the *OR* bidding language, a bidder may win multiple bundle bids placed by him/her. In contrast, under the *XOR* bidding language, a bidder may win at most a single bundle bid (among multiple bundle bids) placed by him/her. The *XOR* bidding language is known to be fully expressive (Nisan 2000), that is, it can express any kind of bidding preference. In this study, we investigate the bidder support issues for SIMU auctions of both types (i.e., auctions with *OR* bidding as well as with *XOR* bidding).³

³The *OR* and *XOR* bidding languages differ on at least two aspects: *expressiveness* and *simplicity* (Nisan 2000). *XOR* is more expressive than *OR* (for example, it can directly express substitutability among bids), but is less simple (i.e., it may take fewer *OR* bid elements to express certain preference than *XOR*). Therefore, auctioneers should pick the bidding language that best fits their specific application needs (see Appendix E for a discussion).

To the best of our knowledge, theoretical results and computational tools for providing real-time bidder support in continuous combinatorial auctions are limited, with the notable exceptions of Adomavicius and Gupta (2005) and Petrakis et al. (2012). Adomavicius and Gupta studied the bidder support issue for MISU auctions with *OR* bids (MISU-*OR* in short). One of the fundamental contributions of their work is that they introduced the concepts of *winning levels* and *deadness levels* as two pieces of critical information for bidders to evaluate the efficacy of their bids. The winning level of an item bundle is the lowest possible value above which one needs to bid in order to win immediately. The deadness level of an item bundle is the lowest possible value above which one needs to bid in order to still have a theoretical chance of winning in the future. Furthermore, they introduced the concept of a *sub-auction*, which refers to all bids that are placed on a particular subset of items. Using sub-auctions, they derived theoretical methods to calculate winning and deadness levels, and designed a computational infrastructure to provide bidder support information in real-time. Petrakis et al. examined MISU auctions with *XOR* bids (MISU-*XOR* in short). While their primary focus was not on providing real-time bidder support in MISU-*XOR* auctions, they adopted the same concepts of sub-auctions, winning levels, and deadness levels to discuss some game-theoretic and computational properties of MISU-*XOR* auctions. Based on both theoretical results and computational evaluations, it is evident that *XOR* bids (in contrast to *OR* bids) add very substantial difficulty to auction dynamics. For example, in MISU-*XOR* auctions, the problem of calculating deadness levels is even harder than NP-complete (Petrakis et al. 2012).

The importance of bid evaluation in SIMU auctions has been acknowledged in the research literature (Eso et al. 2005; Katok and Roth 2004); however, the issue of providing comprehensive real-time information to bidders in such auctions has not been comprehensively explored. Following Adomavicius and Gupta and Petrakis et al., we close the gap in addressing the real-time bidder support issues for an important canonical type of combinatorial auctions: continuous SIMU auctions, with *OR* and *XOR* bidding (SIMU-*OR* and SIMU-*XOR* in short, respectively). Table 1 highlights the key elements of relevant work on providing real-time bidder support for the two basic combinatorial auction types (MISU and SIMU auctions) and two commonly used bidding languages (*OR* and *XOR* bidding).

First, we provide and characterize basic properties of each type of SIMU auctions. Importantly, even though there are only N possible bundles in a SIMU auction of N identical items (because it only matters how many units the bid is placed on), the winner determination problem is nonetheless NP-hard for both SIMU-*OR* and SIMU-*XOR* auctions, as will

Table 1. Key Types of Continuous Combinatorial Auctions and Relevant Work on Bidder Support

		Item Type	
		MISU (Distinct Item Auctions)	SIMU (Identical Item Auctions)
Bid Type (Bidding Language)	OR	MISU-OR Adomavicius and Gupta (2005) (Winning and deadness levels; allocative fairness; computational tools to provide bidder support; real-time capabilities).	SIMU-OR <i>This study</i> (Winning and deadness levels; comprehensive theoretical analysis of bidder support metrics; allocative fairness; computational tools to provide bidder support; real-time capabilities).
	XOR	MISU-XOR Petrakis et al. (2012) (Winning and deadness levels; game-theoretic properties of a specific auction mechanism design).	SIMU-XOR

be discussed below. Second, we theoretically derive the bid evaluation metrics (i.e., winning and deadness levels) for each type of SIMU auction. The difference in supported bid types results in key differences between SIMU-OR and SIMU-XOR auctions in providing comprehensive real-time bidder support. The sub-auction concept proves to be critical in understanding the dynamics of both types of SIMU auctions. Third, we design computational artifacts based on our theoretical results, which are able to provide real-time bidder support for both types of SIMU auctions of practically relevant sizes.

Our research follows the paradigm of design science (Hevner et al. 2004). The real-time bidder support issues in SIMU auctions have strong practical relevance (Bichler et al. 2010; Bichler et al. 2017). Industrial auction service providers (e.g., TradeSlot, Optimal Auctions, and Power Auctions LLC) also advocate the importance of bidder support and feedback. We design theoretical, algorithmic, and computational artifacts that benefit bidders' decision making and economic outcomes of the auctions. We also provide rigorous evaluation of our design artifacts for auctions with practically relevant sizes. In other words, our work builds the foundations of a *bidder support system* for both SIMU-OR and SIMU-XOR auctions. Depending on particular application needs, the auction designer can determine the best way to use the bidder support information. For instance, it can be provided to bidders *on-demand*, or it can be integrated in a bidder agent (i.e., a software agent that could bid on a bidder's behalf).

Throughout our investigation of SIMU auctions, we have three important considerations in mind. First, we always consider a strict version of SIMU auctions, where *partial allocation (or partial fulfillment) is not allowed*. In other words, we consider SIMU auctions where all bids are "all-or-nothing" bids: any bid on x units will either win all x units or none of them. Considering bids to be all-or-nothing allows

bidders to express synergies (e.g., complementarities and substitutions) in valuations and is what makes the auctions *combinatorial* (Pekec and Rothkopf 2003). Also, most theoretical and computational problems (including winner determination) in SIMU auctions with partial allocation become trivial. However, without partial allocation, the winner determination problem is computationally intractable. The SIMU-OR auction is a straightforward variant of KNAPSACK, and the SIMU-XOR auction is a variant of Multiple-Choice KNAPSACK, both of which are well-known NP-hard problems (Vazirani 2003; Kellerer et al. 2004). Second, we do not impose any external restrictions on bidders' valuations and bidding strategies, auctioneers' characteristics, or the auction rules.⁴ While they are important parameters to consider when implementing any specific form of a SIMU auction and affect its game-theoretic properties, they are not the focus of our study. Instead, our goal is to solve the bidder support problem for the general class of continuous SIMU-OR/XOR auctions, *regardless of the specific rules of the auctions or specific bidder/auctioneer behaviors*. This is because the feedback information (e.g., winning bids, winning and deadness levels) represents basic and fundamental understanding of the state of any revenue-maximizing SIMU auction. Without it, the auction mechanism may not even be adopted, and meaningful bidding behaviors or economic properties cannot be realized. Third, our proposed theoretical results and computational designs have the desired property of ensuring *strict allocative fairness*. As is common in the auction literature (e.g., Sandholm 2002), winner determination in combinatorial auctions follows the revenue maxi-

⁴In particular, bidders may have common values or private values toward the items being auctioned, their valuations may follow specific distributions, and may be independent or correlated. Bidders are free to adopt any strategies (e.g., honest bidding). Auctioneers may have specific reserve prices, and may choose to impose additional auction rules (e.g., clock auctions).

zation principle: the goal is to find the feasible allocation that maximizes auctioneer revenue. However, in reality, a number of feasible allocations can have the same auction revenue. In this case, it is important to have a well-defined tie-breaking mechanism that is able to uniquely, efficiently, and systematically identify the winners, that is, a single feasible allocation that *should* win the auction according to certain straightforward, yet important, fairness principles. The bidders should also be informed of the mechanism in advance, so that they can incorporate it into their bidding strategies. While the allocative fairness property is necessary for tie-breaking purposes, it has received limited attention in the combinatorial auction literature (Guler et al. 2016; Katok and Roth 2004). In this paper, we define the *allocative fairness principle* based on the general idea of “first come first served”: when multiple feasible allocations are tied by value, we favor the allocation that forms the earliest. In other words, allocations formed later cannot win the auction just by matching the value of the currently winning allocation. Precise operationalization of this principle will be discussed in later sections.

In addressing the real-time bidder support problem, our results also reveal several significant intricacies of SIMU auctions. First, despite its seemingly simple setup as compared with MISU auctions, SIMU auctions are still combinatorial in nature. This can be illustrated using a straightforward example. Consider a SIMU auction of three identical units and the following bids: (1) \$5 on one unit and (2) \$10 on three units. Although the \$5 bid is not currently winning, it can become winning if a new bid of \$6 on two units arrives. The winner determination problems of both SIMU and MISU auctions are NP-hard—the former represents a knapsack problem and the latter represents a set-packing problem—but their difficulties manifest in different ways. While the difficulty of MISU auctions of N distinct items stems from the exponential number of item bundles that one can bid on (i.e., $2^N - 1$ different subsets of N distinct items, excluding the empty set), in SIMU auctions of N identical items, there are only N possible ways to bid on different item subsets. However, the difficulty of SIMU auctions comes in large part from the exponential number of ways to partition N identical items among different bids (i.e., an integer partition problem). Second, auctions with XOR bidding represent considerable additional theoretical and computational complexity over the auctions with OR bidding. In SIMU-XOR auctions, it turns out that, in order to calculate bid evaluation metrics, one needs to keep track of *subsets of bidders*, the number of which grows exponentially with the number of bidders in an auction.

Our work contributes to the information systems and auction literature in several ways. First, providing comprehensive,

real-time bidder support for SIMU auctions has been an open problem. We design theoretically based computational artifacts that can support SIMU auctions of practically relevant sizes in real-time. Second, this study facilitates the efficient and more transparent implementation of bidder support in various SIMU auction applications, which is an important step toward a wider adoption of the SIMU auction mechanism in business-oriented and potentially even consumer-oriented markets. Third, we believe that the combination of insights from our work on multi-unit auctions and insights from prior work on multi-item auctions can inform the systematic development of real-time bidder support for the most general MIMU auctions, the importance of which has been discussed in prior work (for some initial attempts at calculating winning and deadness levels, see Sen and Bagchi 2012).

SIMU-OR: Homogeneous-Item Combinatorial Auction with OR Bids

In this section we present theoretical results regarding SIMU-OR auctions, that is, the type of SIMU auction where multiple bids of the same bidder can win the auction simultaneously.

Preliminaries of SIMU-OR Auctions

Basic Definitions and Notation

Let N be the *auction size*, that is, the number of (identical) items to be auctioned. An arbitrary bid b in SIMU-OR auctions can be represented by the tuple $b = (s, v, t)$, where s denotes the *span* of the bid; v denotes the *value* of the bid (i.e., the monetary amount specified in this bid); and t denotes the *time* the bid is placed. Because all items in SIMU-OR auctions are identical, span s denotes simply *the number of items* the bid is being placed on. In other words, auction participants can place bids on an item “lot” of any size s ($1 \leq s \leq N$). Given bid b , we use $s(b)$, $v(b)$, and $t(b)$ to denote the span, value, and time of the bid, respectively. We could also include $p(b)$ to denote the bidder of the bid for completeness, but we will omit this in the discussion of SIMU-OR auctions. Because multiple OR bids placed by the same bidder can win simultaneously, bidder information is not necessary to understand the auction dynamics.

We assume that there exists a strict chronological order of bids, that is, that no two bids arrive at the exact same time moment, or more formally, for any two different bids b_i and b_j , we always have $t(b_i) \neq t(b_j)$. Therefore, all bids in an

auction can be ordered according to their timestamp (i.e., b_1, b_2, \dots). This is a natural setup for a tie-breaking approach that would be considered fair in many contexts. In particular, if we have two bids with spans of N items (i.e., the whole auction set) and identical values, the earlier bid is typically preferred over the later one. Therefore, a given auction is “discretized” into auction states B_k ($k = 1, 2, \dots$) based on each incoming bid, that is, $B_k = \{b_1, b_2, \dots, b_k\}$. For notational completeness, we define $B_0 = \emptyset$. Note that most of our theoretical results would apply as long as certain strict ordering of bids (other than arrival-time-based ordering) can be established.

Given an arbitrary set of bids B in a SIMU-OR auction of size N , bid set C (where $C \subseteq B$) is called a *feasible allocation* in B if the total number of items across all bids in C does not exceed N , or, more formally, if $\sum_{b \in C} s(b) \leq N$. Let C_k denote the set of all feasible allocations possible at auction state k , that is, $C_k = \{C \subseteq B_k \mid \sum_{b \in C} s(b) \leq N\}$. We can straightforwardly extend span, value, and time notions from bids to feasible allocations as follows. Specifically, if C is a feasible allocation, then $s(C)$, $v(C)$, and $t(C)$ are defined as follows: $s(C) = \sum_{b \in C} s(b)$, $v(C) = \sum_{b \in C} v(b)$, and $t(C) = \max_{b \in C} t(b)$.⁵ Also, for notational completeness, we define the span, value, and time of an empty allocation \emptyset as follows: $s(\emptyset) = 0$, $v(\emptyset) = 0$, and $t(\emptyset) = 0$. Finally, we assume that the winners of the SIMU-OR auction are determined by maximizing the seller’s revenue, that is, $\max_{C \in C_k} v(C) = \max_{C \in C_k} \sum_{b \in C} v(b)$. The feasible

allocation that achieves maximum revenue is called the *winning allocation*, denoted as WIN_k for auction state k .

Allocative Fairness Principle for Determining Winners

As discussed earlier, it is crucial to have a well-defined tie-breaking mechanism in order to always be able to uniquely and systematically determine the winners at any state of an auction. For example, in the context of MISU-OR auctions, Adomavicius and Gupta (2005) defined a tie-breaking mechanism such that if two feasible allocations are tied by value then, after eliminating all overlapping bids between the two allocations, the one having earlier time is always preferred. In this paper we use the same tie-breaking mechanism and *strict total order* on feasible allocations as in Adomavicius and Gupta for SIMU-OR auctions, because it intuitively captures the fairness principle. Specifically, we will state that feasible allocation C'' is better than C' (denoted as $C' < C''$), if either of the following two conditions is true: (1) $v(C') <$

$v(C'')$; or (2) $v(C') = v(C'')$, and $t(C' \setminus C'') > t(C'' \setminus C')$. In other words, if there are several feasible allocations with equal value, the earliest feasible allocation is chosen over the later ones—that is, one cannot win the auction by coming in late and just matching the current highest revenue. More details about this strict total order can be found in Adomavicius and Gupta. Below we construct an example illustrating this tie-breaking mechanism for SIMU-OR auctions.

Illustration 1. Consider a four-item SIMU-OR auction ($N = 4$) with the following four OR bids already submitted, made in the exact chronological sequence shown here:

$$b_1 = (3, \$28), b_2 = (1, \$5), b_3 = (2, \$23), b_4 = (1, \$12)$$

Consider two feasible allocations $C_1 = \{b_1, b_4\}$ and $C_2 = \{b_2, b_3, b_4\}$. Both C_1 and C_2 have the maximum possible value of a feasible allocation in this example (i.e., $v(C_1) = \$40 = v(C_2)$). However, after removing overlapping bids (b_4 in this example), $C_1 \setminus \{b_4\}$ finished earlier than $C_2 \setminus \{b_4\}$ (i.e., $t(C_2 \setminus C_1) = t(\{b_2, b_3\}) = 3 > 1 = t(\{b_1\}) = t(C_1 \setminus C_2)$). Therefore, based on the tie-breaking mechanism, C_1 should be preferred over C_2 (i.e., $C_2 < C_1$). This mechanism ensures allocative fairness as later bids b_2 and b_3 cannot win the auction just by matching the value of an earlier bid b_1 .

Key Elements of Real-Time Bidder Support

Once the strict total order is defined on feasible allocations, we can state the auction winner determination problem simply as $WIN_k = \max_{C \in C_k} C_k$. Also, let REV_k denote the auction revenue (i.e., $REV_k = v(WIN_k)$).

In addition to knowing the current winning bids at any state of the auction, two types of evaluative information are particularly useful for a given bidder: (1) how much does the bidder need to place on a bid to make it winning in the next state; and (2) how much does the bidder need to place on a bid to make it possible to be winning in at least one future state. In order to provide such bidder-support information, we directly adopt from Adomavicius and Gupta the concept of winning, dead, and live bids to the context of SIMU auctions. Bid $b \in B_k$ is winning if $b \in WIN_k$. If bid b is not winning, and cannot possibly be a winning bid in *any* subsequent auction state, then b is called a *dead bid* in B_k . Formally, bid $b \in B_k$ is dead if $b \notin WIN_k$ and $(\forall B_l \supset B_k)(b \notin WIN_l)$. The set of all dead bids in B_k is denoted as $DEAD_k$. On the other hand, if $b \notin DEAD_k$, then bid b is called a *live bid* in B_k . The set of all live bids in B_k is denoted as $LIVE_k$. Based on the definitions of WIN_k , $DEAD_k$, and $LIVE_k$, note that $DEAD_k \cap LIVE_k = \emptyset$, $DEAD_k \cup LIVE_k = B_k$, $WIN_k \subseteq LIVE_k$, and $DEAD_k \subseteq DEAD_{k+1}$ for any k .

⁵The time, $t(C)$, of feasible allocation C is defined to be the time when this allocation is fully formed (i.e., the time of the last bid in the allocation).

At state k of the SIMU-OR auction, bidders typically want to know how much they need to bid on a particular span, in order for their *new* bid (i.e., bid b_{k+1} to be submitted at state $k + 1$) to be live or winning at state $k + 1$. Accordingly, we define deadness level and winning level metrics to reflect such information. The *deadness level* of span x , denoted as $DL_k(x)$, is theoretically defined as an amount that satisfies the following property: $b_{k+1} \in LIVE_{k+1} \Leftrightarrow v(b_{k+1}) > DL_k(x)$. The *winning level* of the same span, denoted as $WL_k(x)$, is theoretically defined as an amount that satisfies the following property: $b_{k+1} \in WIN_{k+1} \Leftrightarrow v(b_{k+1}) > WL_k(x)$. Note that the above definitions of winning and deadness levels do not provide the means to compute them. We discuss winning and deadness level calculations in a later section.

The Sub-Auction Concept for SIMU-OR Auctions

The notion of *sub-auction* is crucial to understanding the dynamics of MISU auctions, as demonstrated in Adomavicius and Gupta (2005) and Petrakis et al. (2013). It can be used to provide real-time feedback to bidders regarding which of the bids are winning, live, or dead. In SIMU-OR auctions, because all items are identical, we propose to define the sub-auction by *item count* x . More precisely, given auction state k and item count x ($1 \leq x \leq N$), define $B_k[x]$ to include only those bids from B_k that are placed on x or fewer items (i.e., $B_k[x] = \{b \in B_k | s(b) \leq x\}$). We can then say that each item count x defines a *sub-auction* of the overall auction of size N .

Furthermore, given auction state k and lot size x , we can define $C_k[x]$ as the set of all feasible allocations that “cover” up to x items (i.e., $C_k[x] = \{C \subseteq B_k[x] | \sum_{b \in C} s(b) \leq x\}$). Consequently, having defined the feasible allocations at any auction state k for any sub-auction x , we can straightforwardly define the sub-auction winning bids and revenue. Specifically, using the strict total order on feasible allocations, the winner determination problem for each sub-auction is formulated as $WIN_k[x] = \max C_k[x]$, where $WIN_k[x]$ represents the winning allocation for sub-auction x at auction state k . Again, for notational completeness, we define $WIN_k[\emptyset] = \emptyset$. Also, let REV_k denote the revenue of sub-auction x at auction state k or, more formally, $REV_k(x) = v(WIN_k[x])$. Based on this definition of sub-auction revenue, we can straightforwardly derive some properties of $REV_k(x)$, such as (1) $REV_k(x) \leq REV_{k+1}(x)$ and (2) $x \leq y \Rightarrow REV_k(x) \leq REV_k(y)$. In other words, sub-auction revenue $REV_k(x)$ is monotonically nondecreasing with respect to both auction state k and span size x .

More importantly, we want to be able to provide the bidder with real-time information about the auction at every instant (e.g., we would like to know winning and deadness levels for

each possible bid at each auction state). To do so, we show that there exist key relationships between sub-auctions and bid categories (i.e., winning, live, or dead), which can lead to the calculations of various bid evaluation metrics. Previous results on MISU-OR auctions cannot be easily generalized, because MISU-OR and SIMU-OR auctions have some significant differences in this regard. For instance, Adomavicius and Gupta showed that MISU-OR auctions of size N can possibly have up to $2^N - 1$ live bids (as well as an exponential number of sub-auctions), which was a major source of computational complexity; it meant that $O(2^N)$ amount of data was needed to be stored in order to provide bidders with constant-time access to important auction information (e.g., DL and WL for any bid). On the other hand, SIMU-OR auctions have only N different sub-auctions ($1 \leq x \leq N$).

Before we discuss the necessary and sufficient conditions for bid winning and deadness in SIMU-OR auctions, one of the important issues is to understand the iterative dynamics of winning allocations (i.e., how the current winning allocation changes when a new bid is submitted to the auction). Suppose that we are at auction state k (i.e., k bids have been submitted so far) and a new bid b_{k+1} is placed. The following theorem explains the dynamics of the winning allocation for each sub-auction x (i.e., the relationship between $WIN_k[x]$ and $WIN_{k+1}[x]$ for each sub-auction x ($1 \leq x \leq N$)).

Theorem 1. For a SIMU-OR auction, given auction state k and new bid b_{k+1} :

1. $(\forall x < s(b_{k+1}))(WIN_{k+1}[x] = WIN_k[x])$;
2. $(\forall x \geq s(b_{k+1}))(WIN_{k+1}[x] = \max_x[WIN_k[x], \{b_{k+1}\} \cup WIN_k[x - s(b_{k+1})]])$

Proofs of all theoretical results are included in Appendix A. Simply put, Theorem 1 says that, for all sub-auctions x to which b_{k+1} does not belong, the winning allocation does not change. In contrast, for all sub-auctions x to which b_{k+1} does belong, the winning allocation of sub-auction x can change only if the combination of new bid b_{k+1} and the winning bids in the $x - s(b_{k+1})$ sub-auction is better than the previous winning allocation of sub-auction x .

Calculating Winning and Deadness Levels of Bids in SIMU-OR Auctions

As mentioned earlier, we would like to know when exactly the submitted bids are winning, live, or dead in SIMU-OR auctions, in order to be able to provide comprehensive feedback to the bidders in real time. As was shown in the pre-

vious section, while there are significant inherent differences between MISU-OR and SIMU-OR auctions in terms of their sub-auction structure (e.g., exponential versus linear number of sub-auctions), the incremental dynamics of winning bids in each sub-auction follow almost the exact same pattern in both auction classes. As a result, one might expect significant similarities regarding the bid evaluation metrics (e.g., deadness and winning levels) as well. However, the notion of bid deadness presents another significant difference between MISU-OR and SIMU-OR auctions, as described below.

In MISU-OR auctions, bid deadness turned out to be a very intuitive concept, that is, if a bidder wants to bid on item set X at auction state k , she should bid more than the current revenue of sub-auction X (i.e., $DL_k(X) = REV_k(X)$), otherwise the existing winning bids from sub-auction X (i.e., $WIN_k[X]$) will clearly dominate her newly submitted bid. In SIMU-OR auctions, bid deadness is not as intuitive, as illustrated with the following example.

Illustration 2. Consider a four-unit SIMU-OR auction ($N = 4$) with the following four OR bids already submitted:

$$b_1 = (3, \$21), b_2 = (1, \$12), b_3 = (2, \$15), b_4 = (1, \$10)$$

Suppose a bidder plans to make a bid on one item. How much should the bidder bid, so that her bid is not dead, that is, it has a theoretical chance to end up as a winning bid, depending on what the subsequent bids are? In other words, what is $DL_4(1)$? Before exploring this question, note that bids b_2, b_3 , and b_4 are currently winning, that is, $WIN_4 = \{b_2, b_3, b_4\}$, and the revenue of the whole auction (i.e., $REV_4(4)$) is \$38.

- Unlike MISU-OR auctions, $DL_4(1)$ cannot be equal to $REV_4(1)$, where $REV_4(1) = \$12$ (i.e., the largest one-item bid). We can clearly bid a smaller amount. For example, if we make bid $b_5 = (1, \$11)$, this bid will not only be live, but even winning, by displacing b_4 from the winning allocation (i.e., $WIN_6 = \{b_2, b_3, b_5\}$).
- Alternatively, perhaps $DL_4(1) = \$10$, since any one-item bid that is $\leq \$10$ will not displace b_4 from the winning allocation in the above scenario. However, consider bids $b_5 = (1, \$8)$ and $b_6 = (1, \$9)$. Clearly, $WIN_6 = \{b_2, b_4, b_5, b_6\}$, because $REV_6(4) = \$39$. Then, by definition, it means that b_5 is live, which implies that $DL_4(1) < \$8$.
- The actual answer turns out to be $DL_4(1) = \$6$, as will be seen from the subsequent theoretical results and Illustration 3.

By exploring the idea of sub-auctions further and by analyzing how incoming bids affect the auction dynamics, we derive the relationships between the sub-auction revenues and the bid evaluation metrics in SIMU-OR auctions, which are presented in the rest of this section. First, we present the

theoretical results regarding the necessary and sufficient conditions for a newly submitted bid to be winning, as summarized in the following theorem and corollaries.

Theorem 2. For a SIMU-OR auction, given auction state k and new bid b_{k+1} ,

$$b_{k+1} \in WIN_{k+1} \Leftrightarrow v(b_{k+1}) > REV_k(N) - REV_k(N - s(b_{k+1}))$$

Corollary 2a. For a SIMU-OR auction, given auction state k , the winning level at span x is calculated as

$$WL_k(x) = REV_k(N) - REV_k(N - x)$$

The above theoretical results indicate that, in order for a newly submitted bid to be winning, it has to outbid the difference between the revenue of the entire auction and the revenue of the bids placed on the “complementary” item lot. Another insight is that, while $REV_k(N) - REV_k(N - x)$ represents the winning level for bids on x items in the *whole* auction of size N , $REV_k(i) - REV_k(i - x)$ represents the winning level for bids on x items in the sub-auction of size i , where $i \geq x$, as indicated by the following corollary.

Corollary 2b. For a SIMU-OR auction, given auction state k and new bid b_{k+1} :

$$b_{k+1} \in WIN_{k+1}[i] \Leftrightarrow v(b_{k+1}) > REV_k(i) - REV_k(i - s(b_{k+1}))$$

Now we present the theoretical results regarding the necessary and sufficient conditions for a newly submitted bid to be dead. Specifically, the next theorem states that any bid that is among the winning bids in at least one sub-auction must be live (with respect to the whole auction), and vice versa.

Theorem 3. For a SIMU-OR auction, given state k and any bid $b \in B_k$:

$$b \in LIVE_k \Leftrightarrow (\exists x \geq s(b))(b \in WIN_k[x])$$

Simply put, Theorem 3 states that, at auction state k , any bid on x items will be live if its value is greater than the winning level of *at least one* sub-auction of which x is part (i.e., if this bid is among the winning bids in at least one sub-auction $i \in \{x, \dots, N\}$). The following Corollary and example further illustrate the bid deadness level calculation in SIMU-OR auctions.

Corollary 3a. For a SIMU-OR auction, given auction state k , the deadness level at span x is calculated as

$$DL_k(x) = \min_{i \in \{x, \dots, N\}} [REV_k(i) - REV_k(i - x)]$$

Illustration 3. Consider the four-unit auction from Illustration 2, and the same four submitted bids:

$$b_1 = (3, \$21), b_2 = (1, \$12), b_3 = (2, \$16), b_4 = (1, \$10)$$

It is easy to see that, after the above four bids, the winning allocations for each sub-auction are $WIN_4[1] = \{b_2\}$; $WIN_4[2] = \{b_2, b_4\}$; $WIN_4[3] = \{b_2, b_3\}$; $WIN_4[4] = \{b_2, b_3, b_4\}$. Consequently, the revenues of the sub-auctions are $REV_4(1) = \$12$; $REV_4(2) = \$22$; $REV_4(3) = \$28$; $REV_4(4) = \$38$. Also, by definition, $REV_4(0) = \$0$. Now the question posed in Illustration 2 (what is $DL_4(1)$?) can be answered in a straightforward manner by applying the results from Corollary 3a. Specifically one has to calculate the minimum differences between revenues of adjacent sub-auctions:

$$DL_4(1) = \min_{i \in \{1, \dots, 4\}} [REV_4(i) - REV_4(i-1)] = \min \{\$22, \$10, \$6, \$10\} = \$6$$

Therefore, after the above four bids, one must bid above \$6 on a single item in order to have a chance to win. Similarly, one could calculate $DL_4(x)$ for all other spans x as

$$DL_4(2) = \min_{i \in \{2, \dots, 4\}} [REV_4(i) - REV_4(i-2)] = \min \{\$22, \$16, \$16\} = \$16$$

$$DL_4(3) = \min_{i \in \{3, \dots, 4\}} [REV_4(i) - REV_4(i-3)] = \min \{\$28, \$26\} = \$26$$

$$DL_4(4) = \min_{i \in \{4\}} [REV_4(i) - REV_4(i-4)] = \min \{\$38\} = \$38$$

In MISU-OR auctions, the set of all live bids is the union of the winning bids for all sub-auctions (Adomavicius and Gupta 2005). This turns out to also hold for SIMU-OR auctions that is, at any state of the SIMU-OR auction, *the set of live bids is precisely the same as the set of all the winning bids across all the sub-auctions*, or $LIVE_k = \cup_{1 \leq i \leq N} WIN_k[i]$. Finally, numerous properties of $DL_k(x)$ and $WL_k(x)$ can be derived from their definitions and the theoretical results presented above. Some interesting properties are stated in the following theorem.

Theorem 4. For a SIMU-OR auction, given any auction state k and sub-auctions x and y , the following statements are true:

1. $DL_k(x) \leq REV_k(x)$
2. $DL_k(x) \leq WL_k(x)$
3. $DL_k(N) = WL_k(N) = REV_k(N)$
4. $DL_k(x) \leq DL_{k+1}(x)$
5. $x \leq y \Rightarrow DL_k(x) \leq DL_k(y)$ and $WL_k(x) \leq WL_k(y)$
6. $b \in WIN_k \Rightarrow WL_k(s(b)) \leq v(b)$
7. $b \in LIVE_k \Rightarrow DL_k(s(b)) \leq v(b)$

The results presented in Theorem 4 provide additional insights about the dynamics of SIMU-OR auctions. Statements 1–3 describe the relationships among the three closely related

metrics (i.e., winning level, deadness level, and sub-auction revenue). At any state of the SIMU-OR auction, deadness level of a particular span cannot exceed the revenue or winning level of the same span. However, one special case where deadness level equals sub-auction revenue and winning level is when one considers the entire auction. This is intuitive because, if a bidder wants to bid on N items, she must overbid the current revenue of the entire auction in order to both win the auction and be live (otherwise her bid will always be dominated by the current winning bids and is therefore dead). Statements 4–5 describe the monotonicity of deadness level with respect to auction state, as well the monotonicity of both deadness and winning levels with respect to span. These relationships are consistent with the fact that one needs to bid higher to stay in the auction both as it progresses and as the number of items to bid increases. Statements 6–7 describe two interesting special cases of winning and deadness levels. The two statements suggest that, if a bid with span x is currently winning (live), its value cannot be lower than, but could be equal to, the winning (deadness) level at span x . Note that this is not a contradiction to the definitions of winning (deadness) level, because the definitions are made based on a *new* bid placed in the next auction state. The next section presents another interesting theoretical result of this paper regarding the *tight* upper bound on the number of live bids in SIMU-OR auctions.

Tight Upper Bound on the Number of Live Bids in SIMU-OR Auctions

Using the theoretical results from the previous section, we derive a tight upper bound on the number of live bids in a SIMU-OR auction, as stated in the following theorem.

Theorem 5. In a SIMU-OR auction of size n , for any auction state k , we have $|LIVE_k| \leq N$.

The above result provides an interesting insight about SIMU-OR auctions: while the winner determination problem of such auctions is generally NP-hard, at any state of the auction there are no more than N bids that “matter” (i.e., that are either currently winning or can become winning depending on subsequent bids). This makes SIMU-OR auctions substantially different from MISU-OR auctions, where the number of live bids at any auction state can reach $2^N - 1$ (Adomavicius and Gupta 2005). To show that the upper bound derived in Theorem 5 is tight, below we describe two simple example scenarios where the number of live bids is exactly N .

Illustration 4. Consider a SIMU-OR auction with N units and auction state k where WIN_k contains N single-unit bids. Generally, we have that $WIN_k \subseteq LIVE_k$ and $|LIVE_k| \leq N$; and in this specific case we also have that $|WIN_k| = N$. Therefore, $|LIVE_k| = N$.

Illustration 5. Consider a SIMU-OR auction with N units, where the first N bids were the following (where m is some standard unit cost): $b_1 = (N, \$Nm)$, $b_2 = (N-1, \$(N-1)m)$, ..., $b_{N-1} = (2, \$2m)$, and $b_N = (1, \$m)$.

It is easy to see that b_1 is currently a winning bid (and, therefore, live as well); we have many feasible allocations with the highest auction revenue ($REV_N = Nm$), but b_1 was the earliest (and that is our tie-breaking rule, based on allocative fairness).

It is also easy to see that all other bids were live; we can come up with a new bid that can make any of the above bids winning. For example, b_{N-1} “ $2m$ dollars on 2 units” is live, because the next bid b_{N+1} on $N-2$ units that is higher than $Nm-2m$ dollars would make this feasible allocation (i.e., b_{N-1} and b_{N+1}) winning, because $REV_{N+1} \geq REV_N$. Therefore, we have $|LIVE_N| = N$.

Implementing Bidder Support in SIMU-OR Auctions

The concept of sub-auction proved to be very useful in SIMU-OR auctions for calculating bid evaluation metrics. In this section, we develop data structures to store auction information and to provide real-time bidder support. In particular, throughout the auction, we maintain two arrays that contain information about each sub-auction:

- REV is an array where $REV[i]$ stores the current revenue of sub-auction i .
- $LastWinBid$ is a two-dimensional array where $LastWinBid[k, i]$ stores the information (i.e., value and span) of the temporally latest bid that belongs to the current $WIN_k[i]$ for sub-auction i at auction state k . In particular, $LastWinBid[k, i].value$ stores the value information and $LastWinBid[k, i].span$ stores the span information. We define $LastWinBid[k, 0].value = 0$ and $LastWinBid[k, 0].span = 0$ for any k .

The space complexity of the REV array is simply $O(N)$, because there are N sub-auctions in total for a SIMU-OR auction. The space complexity of the $LastWinBid$ array is $O(NK)$, where K is the total number of auction states. The space efficiency of $LastWinBid$ can be further improved if one discards dead bids as the auction progresses. Because there are at most N live bids in total (as shown in Theorem 5) and,

therefore, at most N auction states containing live bids, one needs no more than $O(N \cdot N) = O(N^2)$ space for the $LastWinBid$ array when $K > N$. The following algorithm incrementally updates REV and $LastWinBid$ of all sub-auctions after each new bid.

ALGORITHM 1.1: Given a new bid, update REV and $LastWinBid$ of all sub-sections.

```

Inputs:      s, v, t (span, value, and time of a newly submitted bid)
NewREV = REV // temporary array of sub-auction revenues
// initialize LastWinBid for latest state t
For x = 1 to N Do LastWinBid[t,x].value = 0; LastWinBid[t,x].span = 0
For SubAuction = s To N Do
    // if the new bid wins the sub-auction
    If v > REV[SubAuction] - REV[SubAuction - s] Then
        // update sub-auction revenue
        NewREV[SubAuction] = v + REV[SubAuction - s]
        // update the LastWinBid array
        LastWinBid[t, SubAuction].value = v
        LastWinBid[t, SubAuction].span = s
REV[s..N] = NewREV[s..N]
```

As mentioned above, the computational complexity of the above algorithm is $O(N-s)$ for each bid, where N is a size of the auction and s is the span of the newly submitted bid, or $O(N)$ in the worst case (i.e., for bids with small spans). Therefore, in order to process k bids, the cumulative computational complexity for incremental SIMU auction update is $O(kN)$. Note that this is not a contradiction to the fact that the winner determination problem for SIMU auctions is NP-hard. Since N is given just as a number in the problem specification (that can be encoded using bits), the problem input size then is $O(k \cdot \log N)$ and, therefore, $O(kN)$ complexity is still exponential in terms of the problem input. These types of algorithms for NP-hard problems (e.g., Knapsack) are called *pseudo-polynomial* algorithms (Hu 1969).

It is important to point out that the above implementation automatically satisfies the requirement of our tie-breaking mechanism and ensures allocative fairness. First, the implementation clearly maximizes auction revenue. Second, by only updating REV and $LastWinBid$ when the new bid's value (i.e., v) is *strictly* larger than $REV[SubAuction] - REV[SubAuction - s]$, we ensure that feasible allocations that finish earlier are preferred over feasible allocations that finish later with same revenue.

After the sub-auction revenues are updated, deadness and winning levels can be calculated by querying the REV array. In particular, winning level of span x is simply $REV[N] - REV[N-x]$, and deadness level of the same span is $\min_{s \in \{x, \dots, N\}} \{REV[N] - REV[N-s]\}$. Clearly, it takes $O(1)$ time to query the winning level of a particular span, and $O(N-x)$

time to query the deadness level of a particular span. In other words, it takes $O(N)$ time and $O(N^2)$ time to calculate winning and deadness levels for *all* spans respectively. After these bid evaluation metrics are calculated, the bidders can obtain answers to various questions about the state of the auction in real time, such as: How much should a bidder bid on x items in order for the bid to be winning? For it to be live? What is the revenue of the whole auction?

The up-to-date information stored in LastWinBid can be used to recover the winning allocation. Based on Theorem 1, the winning allocation consists of the last winning bid, b , and the winning allocation of remaining items (i.e., $N - s(b)$). Algorithm 1.2, with inputs CurrSpan = N and CurrState = K , can recover the winning allocation for the auction at state K within $O(N)$ time in the worst case.

ALGORITHM 1.2: Find Winning Allocation of a Sub-Auction

```

Inputs: CurrSpan, CurrState      // span of the sub-auction and the
                                current auction state
While (CurrSpan > 0) and (CurrState > 0) Do
    v = LastWinBid[CurrState, CurrSpan].value
    s = LastWinBid[CurrState, CurrSpan].span
    If (s ≠ 0) Then
        Print <v, s>
        CurrSpan = CurrSpan - s
    CurrState = CurrState - 1

```

Because the union of winning bids across all sub-auctions is precisely the set of all *live* bids, we can use the above algorithm to recover the set of all live bids. In particular, at any given auction state, we can run Algorithm 1.2 once for every possible span (i.e., initialize Algorithm 1.2 with CurrSpan = x where $x = 1, 2, \dots, N$ and CurrState = K). All bids that are returned in this process will comprise the set of all live bids.

Table 2 shows the amount of time it takes to update all sub-auction revenues, bid evaluation metrics, and the LastWinBid array for one incoming live bid. For each auction size configuration, we simulate 1,000 live bids (i.e., 1,000 auction states in total). Each bid has span randomly drawn uniformly from $\{1, 2, \dots, N\}$, and value assigned as the current *winning* level of its span plus a random integer drawn uniformly from $\{1, 2, 3\}$. In other words, each new bid always has higher value than the current winning level at its span, and therefore it is guaranteed to be winning (and certainly live); this was purposefully chosen in order to evaluate the proposed approach on the “worst case” scenario that maximizes the amount of computation needed for incremental update (dead bids are dealt with very efficiently, by comparing against current deadness levels and discarding). The full incremental

update is run *after each bid*. The reported time is the average running time over 1,000 live bids. The experiments were done on a 2.6GHz Intel Core i5 computer with 6G RAM running Linux operating system.

As the results indicate, even for auctions of size 500,000 we are able to achieve real-time performance; the full incremental update takes at most around 12.69 milliseconds for any live bid (dead bids can be discarded without performing any updates). Once sub-auction revenues are updated, any winning and deadness level can be calculated at the demand of bidders. For any given span, calculating either the winning or deadness level takes less than a microsecond.

Note that the ability to provide real-time bidder support for 500,000-item SIMU-OR auctions is a huge difference compared to MISU-OR auctions (Adomavicius and Gupta 2005), where it was possible to handle auctions of a few dozen heterogeneous items in real-time. This performance difference is attributable to the fact that SIMU-OR auctions have much fewer sub-auctions that need to be processed after each incoming bid and, generally, to the fact that SIMU-OR auction incremental update problem, while being NP-hard, allows a faster (i.e., pseudo-polynomial time) algorithm, as mentioned earlier.

SIMU-XOR: Homogenous-Item Combinatorial Auction with XOR Bids

Next, we discuss SIMU auctions with XOR bidding, or SIMU-XOR auctions. For SIMU-XOR auctions, multiple bids made by the same bidder are not allowed to win the auction simultaneously. In other words, at any state of a SIMU-XOR auction, any two winning bids must come from two different bidders. The constraint of XOR bidding brings significant theoretical and computational complexity, over and above the SIMU-OR auctions. In this section, we present the theoretical results about the dynamics of SIMU-XOR auctions.

Preliminaries of SIMU-XOR Auctions

Basic Definitions and Notation

Let N be the *auction size* (i.e., the number of identical items to be auctioned). Let P be the *bidder set* (i.e., the set of all possible bidders participating in the auction). We represent a bid in SIMU-XOR auctions somewhat differently than that

Table 2. Average Auction Incremental Update Time for One Live Bid (in Milliseconds)

Auction size (N)	Update All Sub-Auction Revenues	Update All LastWinBid	Full Incremental Updates
10,000	0.01	0.05	0.06
20,000	0.03	0.08	0.11
50,000	0.05	0.22	0.27
100,000	0.11	0.42	0.53
200,000	0.20	1.09	1.29
500,000	1.25	11.44	12.69

in SIMU-OR auction, to accommodate for the fact that, in XOR auctions, the same bidder often submits multiple XOR bids *simultaneously*. Specifically, a *general* bid b in SIMU-XOR auctions is represented by the tuple $b = (\vec{v}, p, t)$, where

$\vec{v} = (v_1, v_2, \dots, v_N)$ is a vector of nonnegative values denoting the *bid values* that are placed on every possible number of items (i.e., v_i represents the *value* placed on i items, $1 \leq i \leq N$; $v_i = 0$ means that the bidder chooses not to submit a bid on a bundle of size i); p denotes the *person* who places the bid ($p \in P$); and t denotes the *time* when the bid is placed. We use the notations $p(b)$ and $t(b)$ to refer to the person and the time of the bid, respectively, and $v_i(b)$ to represent bid value on span i . We also define an *atomic bid*, represented by the tuple $\sigma = (s, v, p, t)$, as a component of bid b placed by person p at time t that has value v on s items. For an atomic bid, we use the notations $s(\sigma)$, $v(\sigma)$, $p(\sigma)$, $t(\sigma)$ to refer to the span (i.e., number of units), value (i.e., bid amount), person (i.e., bidder id), and time of that atomic bid, respectively. For convenience, we consider the following two notations of a bid b to be equivalent: (1) $b = (\vec{v}, p, t)$ and (2) $b = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$,

where $s(\sigma_i) = i$, $v(\sigma_i) = v_i(b)$, $p(\sigma_1) = p(\sigma_2) = \dots = p(\sigma_N) = p(b)$, and $t(\sigma_1) = t(\sigma_2) = \dots = t(\sigma_N) = t(b)$.⁶ Again, we introduce the atomic bid representation because, according to the definition of SIMU-XOR auction, while each bidder can place a general bid with multiple atomic bids simultaneously, no more than one of those atomic bids is allowed to win the auction.

Similar to the SIMU-OR auction, we assume that there exists a strict chronological order to general bids (i.e., for any two different general bids b_i and b_j we always have $t(b_i) \neq t(b_j)$). Therefore, all bids in an auction can be ordered according to this order (i.e., b_1, b_2, \dots). As a result, the SIMU-XOR auction is discretized into auction states B_k ($k = 1, 2, \dots$) based on each incoming bid (i.e., $B_k = \{b_1, b_2, \dots, b_k\}$). Equivalently, we can denote the auction states using the

notation of atomic bids as

$$B_k = \left\{ \sigma_{p(b_1)1}, \dots, \sigma_{p(b_1)N}, \sigma_{p(b_2)1}, \dots, \sigma_{p(b_2)N}, \dots, \sigma_{p(b_k)1}, \dots, \sigma_{p(b_k)N} \right\}$$

where $\sigma_{p(b_i)j}$ is the atomic bid placed by bidder $p(b_i)$ with span j . For the purpose of notational completeness, we define $B_0 = \emptyset$. Note that while two bids at different states must have different time, they could be submitted by the same bidder. In other words, in SIMU-XOR auctions, a bidder is allowed to submit multiple general XOR bids at different time.

The definition of feasible allocation in SIMU-XOR auction is different from that in SIMU-OR auction, due to the constraint of XOR bidding. Specifically, given an arbitrary set of bids B in a SIMU-XOR auction of size N , a *feasible allocation* in B is defined as a set of *atomic bids* C (where $C \subseteq B$) if the total number of items across all atomic bids in C does not exceed N , and *no two atomic bids in C are placed by the same person*, or, more formally, $\sum_{\sigma \in C} s(\sigma) \leq N$ and $\forall \sigma_1, \sigma_2 \in C, p(\sigma_1) \neq p(\sigma_2)$. Let C_k denote the set of all feasible allocations possible at auction state k , that is, $C_k = \{C \subseteq B_k | \sum_{\sigma \in C} s(\sigma) \leq N \text{ and } \forall \sigma_1, \sigma_2 \in C, p(\sigma_1) \neq p(\sigma_2)\}$. Straightforwardly, we can extend the notions of span, value, person, and time from atomic bids to feasible allocations. If C is a feasible allocation, then $s(C) = \sum_{\sigma \in C} s(\sigma)$, $v(C) = \sum_{\sigma \in C} v(\sigma)$, $p(C) = \cup_{\sigma \in C} p(\sigma)$, and $t(C) = \max_{\sigma \in C} t(\sigma)$. For completeness, we define

the span, value, person, and time of an empty allocation \emptyset as: $s(\emptyset) = 0$, $v(\emptyset) = 0$, $p(\emptyset) = \emptyset$, and $t(\emptyset) = 0$.

Allocative Fairness Principle for Determining Winners

The tie-breaking mechanism for an SIMU-OR auction also needs modification to accommodate for SIMU-XOR auctions, due to the XOR bidding constraint. Because multiple atomic bids of the same general bid can be placed simultaneously by

⁶For SIMU-XOR auctions, unless specifically noted, we refer to a “bid” as a general bid (i.e., the collection of atomic bids made by a person at a particular time moment), not the individual atomic bids.

a particular bidder, they may end up in different feasible allocations, causing these allocations to have the same timestamp. More formally, for any two feasible allocations C' and C'' , although $(C' \setminus C'') \cap (C'' \setminus C') = \emptyset$, there is no guarantee that $t(C' \setminus C'') \neq t(C'' \setminus C')$. Denote the atomic bid in $(C' \setminus C'')$ with latest time (i.e., arriving last) as σ_1 , and the atomic bid in $(C'' \setminus C')$ with latest time as σ_2 ; it is possible that σ_1 and σ_2 belong to the same general bid and, therefore, $t(\sigma_1) = t(C' \setminus C'') = t(C'' \setminus C') = t(\sigma_2)$. To ensure *fairness*, the bidder who bids last should be allocated as few items as possible. In other words, when two feasible allocations (1) have the same value and (2) are tied by time after removing overlapping atomic bids, our tie-breaking mechanism chooses the feasible allocation whose last arriving atomic bid has a *smaller* span. This is a strict tie-breaking mechanism, because if σ_1 and σ_2 belong to the same general bid, we know $s(\sigma_1) \neq s(\sigma_2)$. Using the *strict total order* notation $<$ on feasible allocations, we will state that feasible allocation C'' is better than C' (denoted as $C' < C''$), if any of the following three conditions is true: (1) $v(C') < v(C'')$; or (2) $v(C') = v(C'')$ and $t(C' \setminus C'') > t(C'' \setminus C')$; or (3) $v(C') = v(C'')$, $t(C' \setminus C'') = t(C'' \setminus C')$, and $s(\sigma_1) > s(\sigma_2)$, where $\sigma_1 \in C' \setminus C''$, $\sigma_2 \in C'' \setminus C'$, and $t(\sigma_1) = t(C' \setminus C'') = t(C'' \setminus C') = t(\sigma_2)$. In other words, if there are several feasible allocations with equal value, the earliest allocation is chosen over the later ones, in order to prevent later atomic bids from winning simply by matching the auction revenue. If there are several feasible allocations with equal value and same time, the one whose last arriving atomic bid has the smallest span is chosen, in order to fulfill earlier atomic bids as much as possible. This tie-breaking mechanism properly captures the proposed fairness principle in addition to revenue maximization. Below is an illustrative example.

Illustration 6. Consider the following three bids of a SIMU-XOR auction with three units (i.e., $N = 3$) and three bidders $P = \{A, B, C\}$, which were submitted in the following sequence:

$b_1 = \sigma_{A1} \text{ XOR } \sigma_{A2} = (1, \$4, A, 1) \text{ XOR } (2, \$6, A, 1)$. That is, bidder A bids \$4 on one unit and \$6 on two units, as two atomic XOR bids, denoted as σ_{A1} and σ_{A2} ;
 $b_2 = \sigma_{B1} \text{ XOR } \sigma_{B2} = (1, \$5, B, 2) \text{ XOR } (2, \$9, B, 2)$;
 $b_3 = \sigma_{C1} \text{ XOR } \sigma_{C2} = (1, \$7, C, 3) \text{ XOR } (2, \$8, C, 3)$.

Consider the following two feasible allocations $C_1 = \{\sigma_{A1}, \sigma_{B1}, \sigma_{C1}\}$ and $C_2 = \{\sigma_{B2}, \sigma_{C1}\}$. That is, C_1 consists of one-unit bids from bidders A, B, and C, while C_2 consists of the two-unit bid from B and one-unit bid from C. We can see that the revenues of these allocations are identical, $v(C_1) = v(C_2) = \$16$. We can also see that, after removing overlapping atomic bid $\{\sigma_{C1}\}$ (i.e., $\{\sigma_{C1}\} = C_1 \cap C_2$), which does not influence the tie-breaking between the two, the two allocations are tied on time $t(C_1 \setminus C_2) = t(\{\sigma_{A1}, \sigma_{B1}\}) = 2 = t(\{\sigma_{B2}\}) = t(C_2 \setminus C_1)$. Accordingly, σ_{B1} and σ_{B2} are the last arriving atomic

bids for $C_1 \setminus \{\sigma_{C1}\}$ and $C_2 \setminus \{\sigma_{C1}\}$. Because $s(\sigma_{B1}) = 1 < 2 = s(\sigma_{B2})$, we have $C_2 < C_1$. Thus, according to the tie-breaking rule, C_1 has precedence over C_2 , and bidder B should be allocated only one unit (not two); this reflects the fairness principle, because allocating more units to B would mean deallocating units from bidder A without improving auction revenue (even though A bid before B).

Key Elements of Real-Time Bidder Support

Finally, using the aforementioned notations and definitions, the winner determination problem is simply $WIN_k = \max_k C_k$. The auction revenue is $REV_k = v(WIN_k)$. To provide real-time bidder support, we extend the same theoretical definitions of deadness level and winning level metrics to a new atomic bid, σ . In particular, for span x and bidder p , we theoretically define the winning level, $WL_k(x, p)$, to be an amount that satisfies: $\sigma \in WIN_{k+1} \Leftrightarrow v(\sigma) > WL_k(x, p)$. We theoretically define the deadness level, $DL_k(x, p)$, to be an amount that satisfies: $\sigma \in LIVE_{k+1} \Leftrightarrow v(\sigma) > DL_k(x, p)$. Note from the notations that, unlike in SIMU-OR auctions, winning and deadness levels in SIMU-XOR auctions are “personalized” (i.e., depend on specific bidder p , as will be discussed later in the paper).

The Sub-Auction Concept for SIMU-XOR Auctions

The concept of sub-auction in SIMU-XOR auctions needs to be specifically modified from that in SIMU-OR, to account for the constraint of XOR bidding. More precisely, given auction state k , item count x ($1 \leq x \leq N$), and a subset of bidders P ($\emptyset \subseteq P \subseteq \mathcal{P}$), we define $B_k[x, P]$ to include only those atomic bids from B_k that are placed on x or fewer items, and that are placed by bidders from set P (i.e., $B_k[x, P] = \{\sigma \in B_k | s(\sigma) \leq x, p(\sigma) \in P\}$). We say that $B_k[x, P]$ defines a *sub-auction* of the overall auction of size N .

Given auction state k , lot size x , and a subset of all bidders P , we further can define $C_k[x, P]$ as the set of all feasible allocations that “cover” up to x items and involve bidders in the bidder set P (i.e., $C_k[x, P] = \{C \subseteq B_k[x, P] | s(C) \leq x, p(C) \subseteq P \text{ and } \forall \sigma_1, \sigma_2 \in C, p(\sigma_1) \neq p(\sigma_2)\}$). Similar to SIMU-OR auctions, the winner determination problem for each sub-auction can be formulated as $WIN_k[x, P] = \max_k C_k[x, P]$, where $WIN_k[x, P]$ represents the winning allocation for sub-auction $[x, P]$ at auction state k . Again, for the purpose of notational completeness we define $WIN_k[x, P] = \emptyset$ and $WIN_k[x, \emptyset] = \emptyset$ for any bidder set P and span x . Also, let $REV_k(x, P)$ denote the revenue of sub-auction $[x, P]$ at auction

state k or, more formally, $REV_k(x, P) = v(WIN_k[x, P])$. Based on this definition of sub-auction revenue, the following properties of $REV_k(x, P)$ are straightforward: (1) $REV_k[x, P] \leq REV_{k+1}(x, P)$; (2) $(x \leq y) \Rightarrow REV_k(x, P) \leq REV_k(y, P)$; and (3) $(P \subseteq Q) \Rightarrow REV_k(x, P) \leq REV_k(x, Q)$. In other words, the sub-auction revenue is monotonically nondecreasing with auction state k , sub-auction size x , and bidder set P .

Similar to SIMU-OR auctions, we again provide the following theorems regarding three key aspects: (1) the change in winning allocation as a new bid (potentially consisting of N nonzero atomic bids) arrives; (2) the calculation of winning and deadness levels for each span and bidder; (3) the maximum number of live bids at any given auction state.

Suppose that we are at auction state k , and a new general bid b_{k+1} is placed by bidder $p(b_{k+1})$. In particular, $b_{k+1} = (\vec{v}, p(b_{k+1}), k+1)$, where $\vec{v} = (v_1, v_2, \dots, v_N)$ and $\forall i \in \{1, 2, \dots, N\}, v_i \geq 0$. In other words, the new bid consists of at most N nonzero atomic bids, denoted as $\{\sigma_1, \sigma_2, \dots, \sigma_N\}$ where $\sigma_1 = (i, v, p(b_{k+1}), k+1)$. The following theorem explains the dynamics of the winning allocation for each sub-auction $[x, P]$.

Theorem 6. For a SIMU-XOR auction, given auction state k and new bid $b_{k+1} = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$, $\forall x \in \{1, 2, \dots, N\}$, $\forall P \subseteq P$:

1. If $p(b_{k+1}) \notin P$, then $WIN_{k+1}[x, P] = WIN_k[x, P]$.
2. If $p(b_{k+1}) \in P$, then $WIN_{k+1}[x, P] = \max_{\sigma \in \{b_{k+1}\} \cup \{P\}} \{WIN_k[x, P], WIN_k[x-1, P \setminus \{p(b_{k+1})\}] \cup \{\sigma_1\}, WIN_k[x-2, P \setminus \{p(b_{k+1})\}] \cup \{\sigma_2\}, \dots, WIN_k[1, P \setminus \{p(b_{k+1})\}] \cup \{\sigma_{x-1}\}, \{\sigma_x\}\}$.

In summary, Theorem 6 says that, for any sub-auction that to which b_{k+1} does not belong, the winning allocation does not change. Instead, for any sub-auction to which b_{k+1} does belong, the winning allocation can change only if some atomic bid σ_i of b_{k+1} and the winning bids in the $[x - s(\sigma_i), P \setminus \{p(b_{k+1})\}]$ sub-auction (the bidder $p(b_{k+1})$ of b_{k+1} must be excluded in order to preserve the XOR requirement of SIMU-XOR auctions) combine to be better than the previous winning allocation of sub-auction $[x, P]$.

Calculating Winning and Deadness Levels of Bids in SIMU-XOR Auctions

As was done with SIMU-OR auctions, we would like to know when exactly the submitted atomic bids are winning, live, or dead in SIMU-XOR auctions as well, in order to be able to provide comprehensive feedback to the bidders. This is accomplished, again, by exploring the idea of sub-auctions

further and by analyzing how incoming bids affect the auction dynamics. Winning level calculation is fairly straightforward and logically similar to that of SIMU-OR auctions. Deadness level calculation, however, turns out to be very different from that of SIMU-OR auctions, due to the XOR constraint. Below, we first present the theoretical results regarding the necessary and sufficient conditions for a newly submitted atomic bid to be winning.

Theorem 7. For a SIMU-XOR auction, given auction state k and new atomic bid σ : $\sigma \in WIN_{k+1} \Leftrightarrow v(\sigma) > REV_k(N, P) - REV_k(N - s(\sigma), P \setminus \{p(\sigma)\})$.

Corollary 7a. For a SIMU-XOR auction k , the winning level at span x for bidder p^* is calculated as

$$WL_k(x, p^*) = REV_k(N, P) - REV_k(N - x, P \setminus \{p^*\})$$

The above theoretical results indicate that, for a newly submitted atomic bid to be winning, it has to outbid the difference between the revenue of the entire auction and the revenue of bids placed on the “complementary” item lot by other bidders (in order not to violate the XOR constraint). Note that the winning levels for SIMU-XOR auctions can differ for different bidders, unlike the winning levels for SIMU-OR auctions. Finally, the necessary and sufficient conditions for winning a particular sub-auction $[i, P]$ can be analogously derived, as shown in Corollary 7b.

Corollary 7b. For a SIMU-XOR auction, given auction state k and new atomic bid σ , and for all bidder set P such that $p(\sigma) \in P \subseteq P$:

$$\sigma \in WIN_{k+1}[i, P] \Leftrightarrow v(\sigma) > REV_k(i, P) - REV_k(i - s(\sigma), P \setminus \{p(\sigma)\}).$$

Next, we present the necessary and sufficient conditions for an atomic bid to be live.

Theorem 8. For a SIMU-XOR auction, given auction state k and any atomic bid $\sigma \in B_k$:

1. If $s(\sigma) \leq N - |P|$, then $\sigma \in LIVE_k \Leftrightarrow \sigma = WIN_k[x(\sigma), p(\sigma)]$;
2. If $s(\sigma) > N - |P|$, then $\sigma \in LIVE_k \Leftrightarrow \exists Q \subseteq P$ with $p(\sigma) \in Q$ and $|Q| = |P| - (N - s(\sigma))$ such that $\sigma = WIN_k[s(\sigma), Q]$.

Theorem 8 states that, for an existing atomic bid of span x (i.e., $s(\sigma) = x$), (1) if $x \leq N - |P|$, then it is live as long as it is not outbid by existing bids placed by the same bidder with equal or smaller span, and (2) if $x > N - |P|$, then it is live if and only if it wins at least one sub-auction with span x and bidder set Q , where $|Q| = |P| - (N - x)$, or equivalently, with $N - x$ other bidders removed. The intuition of “removing” N

– x other bidders is that, if one considers a future allocation where each of these $N - x$ removed bidders places a high-valued bid on 1 item, they will “block” their own existing bids (due to XOR constraint), and the focal atomic bid can consequently win the auction together with this allocation. The following corollary can be derived based on the results from Theorem 8, regarding the calculation of deadness level for any given span and bidder.

Corollary 8a. For a SIMU-XOR auction, given auction state k , the deadness level at span x for bidder p^* is calculated as follows:

1. If $x \leq N - |P|$, then $DL_k(x, p^*) = REV_k(x, p^*)$;
2. If $x > N - |P|$, then $DL_k(x, p^*) = \min_{\forall Q \subseteq P, p^* \in Q, |Q|=|P|-(N-x)} REV_k(x, Q)$

Importantly, the results from Corollary 8a show that, in SIMU-XOR auctions, the deadness level is personalized (i.e., it can be different for different bidders), which is distinct from the deadness levels in SIMU-OR auctions. The following example illustrates how the bid deadness level can be calculated in SIMU-XOR auctions.

Illustration 7. Consider the three-unit auction from Illustration 6, and the same three submitted bids:

$$\begin{aligned} b_1 &= \sigma_{A1} \text{ XOR } \sigma_{A2} = (1, \$4, A, 1) \text{ XOR } (2, \$6, A, 1); \\ b_2 &= \sigma_{B1} \text{ XOR } \sigma_{B2} = (1, \$5, B, 2) \text{ XOR } (2, \$9, B, 2); \\ b_3 &= \sigma_{C1} \text{ XOR } \sigma_{C2} = (1, \$7, C, 3) \text{ XOR } (2, \$8, C, 3). \end{aligned}$$

Suppose we want to calculate the deadness level of two units for bidder A (i.e., $DL_3(2, A)$). According to Corollary 8a, $DL_3(2, A) = \min_{\forall Q \subseteq \{A, B, C\}, A \in Q, |Q|=3-(3-2)} REV_k(x, Q)$.

There are two bidder sets that satisfy $A \in Q$ and $|Q| = 3 - (3 - 2)$: $\{A, B\}$ and $\{A, C\}$. Therefore,

$$\begin{aligned} \min_{\forall Q \subseteq \{A, B, C\}, A \in Q, |Q|=2} REV_k(x, Q) &= \min(REV_3(2, \{A, B\}), \\ &REV_3(2, \{A, C\})) = \min(v(\{\sigma_{A1}, \sigma_{B1}\}), v(\{\sigma_{A1}, \sigma_{C1}\})) = \\ &= \min(\$9, \$11) = \$9. \end{aligned}$$

Therefore, $DL_3(2, A) = \$9$ (i.e., bidding above \$9 on two units will result in a live bid for bidder A).

Note that bidding *exactly* \$9 on two units is not sufficient to be live, because both σ_{B2} and $\{\sigma_{A1}, \sigma_{C1}\}$ are two-unit bids that already have values equal to or higher than \$9. However, bidding strictly above \$9 on two units (e.g., \$10) is sufficient to be live, as it can combine with σ_{C1} to win the auction.

Recall that in both MISU-OR and SIMU-OR auctions, at any state of the auction, the set of live bids is precisely the same as the set of all winning bids across all the sub-auctions. However, this is not true in SIMU-XOR auctions. Based on

our particular construction of sub-auctions, an atomic bid winning a sub-auction is not necessarily live. To see this, consider an atomic bid σ that is currently dead; it could nonetheless be winning its “own” sub-auction $[s(\sigma), p(\sigma)]$. Instead, the set of live atomic bids needs to be constructed based on results in Theorem 8. This finding highlights an important distinction between OR and XOR bidding.

Finally, we can derive several properties about winning and deadness level, listed in Theorem 9. Note that statements 1–5 are similar to the ones in SIMU-OR auctions (see Theorem 4), reflecting some common themes underlying the two auction types (e.g., statements 4–5 suggest that one needs to bid higher in order to stay in the auction as it progresses and as the number of items to bid on increases). However, statements 6–7 differ from the ones in SIMU-OR auctions. They indicate that if an atomic bid is currently winning (live), its value must be *exactly* equal to the current winning (deadness) level of the span and bidder of this atomic bid (i.e., sub-auction $[x, p]$).

Theorem 9. In a SIMU-ORX auction, for any auction state k , the following statements are true:

1. $DL_k(x, p) \leq REV_k(x, P)$
2. $DL_k(x, p) \leq WL_k(x, p)$
3. $\forall p \in P, DL_k(N, p) = WL_k(N, p) = REV_k(N, P)$
4. $DL_k(x, p) \leq DL_{k+1}(x, p)$
5. $x \leq y \Leftrightarrow WL_k(x, p) \leq WL_k(y, p)$ and $DL_k(x, p) \leq DL_k(y, p)$
6. $\sigma \in WIN_k \Rightarrow WL_k(s(\sigma), p(\sigma)) = v(\sigma)$
7. $\sigma \in LIVE_k \Rightarrow DL_k(s(\sigma), p(\sigma)) = v(\sigma)$

Tight Upper Bound on the Number of Live Bids in SIMU-XOR Auctions

Using the theoretical results from the previous section we will derive a tight upper bound for the number of live atomic bids, as stated in the following theorem and Corollary.

Theorem 10. For a SIMU-XOR auction, given auction state k , there can be no more than $\min(N - x + 1, |P|)$ live atomic bids with span x .

Corollary 10. For a SIMU-XOR auction, given auction state k , the maximum possible number of live atomic bids is

$$\sum_{x=1}^N \min(x, |P|). \text{ That is, } |LIVE_k| \leq \sum_{x=1}^N \min(x, |P|).$$

Notably, if $|P| \geq N$, the maximum number live atomic bids will be $\sum_{x=1}^N \min(x, |P|) = \sum_{x=1}^N x = N(N + 1) / 2$. In other words, regardless of the number of bidders in a SIMU-XOR

auction, the total number of live atomic bids at any auction state is always bounded by $N(N + 1)/2$. The above result marks another significant difference between SIMU-OR auctions and SIMU-XOR auctions. In SIMU-OR auctions, there cannot be more than N live bids at any auction state. In SIMU-XOR auctions, the maximum number of live atomic bids at any auction state is quadratic with respect to auction size. Of course, the number of live atomic bids is even smaller if $|P| < N$. Thus, unlike MISU-OR auctions, which can have exponentially many live bids, both SIMU-OR and SIMU-XOR auctions have much fewer live bids. To demonstrate that the upper bound derived in Corollary 10 is tight, below we describe an example scenario where the number of live atomic bids is exactly $\sum_{x=1}^N \min(x, |P|)$.

Illustration 8. Consider the following auction of size N and bidder set P . For any span $x \in \{1, 2, \dots, N\}$ and bidder $p \in P$, denote the corresponding atomic bid as σ_{px} . For convenience, let $P = \{1, 2, \dots, |P|\}$, where each element is both an identifier for a unique bidder and the time when the bidder places the bid. In other words, bidder 1 places his/her bid at time 1, bidder 2 places his/her bid at time 2, and so on. The value of each atomic bid is constructed as follows: $v(\sigma_{p1}) = p$ and $\forall x \geq 2, v(\sigma_{px}) = 2|P|(x - 1) + (p - 1)$.

The above construction has the following property: $\forall x_1, x_2 \in \{1, 2, \dots, N\}$ and $\forall p_1, p_2 \in \{1, 2, \dots, |P|\}, p_1 \neq p_2$, we have $v(\sigma_{p_1 x_1}) + v(\sigma_{p_2 x_2}) < v(\sigma_{1(x_1 + x_2)})$. In other words, any combination of two atomic bids from spans x_1 and x_2 is dominated by even the smallest atomic bid from span $x_1 + x_2$. As a result, any sub-auction at span x can only be won by atomic bids of span x . By doing so, we can have exactly $\min(N - x + 1, |P|)$ live bids for span x , and exactly $\sum_{x=1}^N \min(x, |P|)$ live bids in the entire auction.

Here is a small numeric example that satisfies this construction. Consider a three-item auction of three bidders (bidder 1, 2, and 3), with three bids made at states 1, 2, and 3: $b_1 = (1, \$1) \text{ XOR } (2, \$6) \text{ XOR } (3, \$12)$, $b_2 = (1, \$2) \text{ XOR } (2, \$7) \text{ XOR } (3, \$13)$, and $b_3 = (1, \$3) \text{ XOR } (2, \$8) \text{ XOR } (3, \$14)$. At state three, there are six live atomic bids: all three atomic bids at span 1, two atomic bids at span 2 with largest values, and one atomic bid at span 3 with largest value.

Implementing Bidder Support in SIMU-XOR Auctions

Straightforward Implementation

The theoretical results in the previous section provide key insights toward implementation of bidder support. Assuming

that we assign an integer identifier to each bidder participating in an auction from the set $\{1, 2, \dots, |P|\}$, every possible set of bidders can be represented as a bitmap of length $|P|$. Throughout the auction, we keep two arrays REV and LastWinBid to store important bid information, defined as follows:

- REV is a two-dimensional array where $\text{REV}[x, P]$ stores the current revenue of sub-auction $[x, P]$, where $x \in \{1, 2, \dots, N\}$ and P is the bidder set represented as a bitmap (i.e., each unique subset of bidders from \emptyset to $|P|$ is represented by a unique integer from 0 to $2^{|P|} - 1$).
- LastWinBid is a two-dimensional array where $\text{LastWinBid}[x, P]$ stores the temporally latest winning bid of sub-auction $[x, P]$. In particular, $\text{LastWinBid}[x, P].\text{span}$, $\text{LastWinBid}[x, P].\text{value}$, and $\text{LastWinBid}[x, P].\text{bidder}$ stores the span, value, and bidder index information, respectively. Also, for every P and x , we define both $\text{LastWinBid}[0, P]$ and $\text{LastWinBid}[x, \emptyset]$ as having span 0, value 0, and bidder index 0.

The space complexity of both arrays is $O(N \cdot 2^{|P|})$, because there are in total $2^{|P|}$ subsets of bidders for each particular span. Algorithm 2.1 updates the revenues and last winning bids for all sub-auctions after each new general bid.

ALGORITHM 2.1: Given a new general bid, update the revenues and winners of all sub-auctions.

```

Inputs:  $v[1..N]$  (array of values for each span of a newly submitted
general bid)
 $p$  (bidder of a newly submitted bid)
For  $x = 1$  To  $N$  Do
  For every bidder set  $S$  where  $p \in S$  Do
    For  $i = 1$  To  $x$  Do
      // if this atomic bid wins sub-auction  $[x, S]$ 
      If  $v[i] > \text{REV}[x, S] - \text{REV}[x - i, S \setminus p]$  Then
        // update the sub-auction revenue
         $\text{REV}[x, S] = v[i] + \text{REV}[x - i, S \setminus p]$ 
        // update the LastWinBid array
         $\text{LastWinBid}[x, S].\text{value} = v[i]$ 
         $\text{LastWinBid}[x, S].\text{span} = i$ 
         $\text{LastWinBid}[x, S].\text{bidder} = p$ 

```

The computational complexity of this algorithm is $O(N^2 2^{|P|-1})$, because there are in total $N \cdot 2^{|P|-1}$ sub-auctions to consider, each taking at most $O(N)$ to update. Note that this is the worst-case complexity, where the newly submitted general bid has N nonzero atomic bids. It is important to note that the above implementation automatically satisfies the requirement of our tie-breaking mechanism and allocative fairness. First,

similarly to Algorithm 1.1, this implementation ensures revenue maximization and, in case there are several revenue-maximizing feasible allocations, chooses the earliest one. Second, for any span x , by looping from 1 to x (i.e., from small to large), we ensure that LastWinBid only stores the winning atomic bid with the smallest possible span.

After the sub-auction revenues are updated, winning levels and deadness levels can be calculated by querying the REV array. In particular, winning level of span x and bidder p is simply $REV[N, P] - REV[N - x, P \setminus \{p\}]$, and deadness level of the same span and bidder is $REV[x, p]$ if $x \leq N - |P|$ or $\min_{Q \subseteq P, p^* \in Q, |Q|=|P|-(N-x)} REV[x, Q]$, otherwise. The com-

plexity of querying the winning level is $O(1)$. The complexity of querying deadness level is $O(1)$ if $x \leq N - |P|$ or

$O\left(C_{|P|-1}^{N-x}\right)$ otherwise,⁷ as there are $O\left(C_{|P|-1}^{N-x}\right)$ possible $REV_k(x, Q)$ to consider, where $p^* \in Q$ and $|Q| = |P| - (N - x)$. For a given span x , querying deadness level is costlier when the number of bidders $|P|$ increases; it is costliest when $N - x = (|P| - 1)/2$. Finally, it takes $O(|P|N)$ time to calculate winning levels across all spans and all bidders, and exponential time between $O(2^N)$ and $O(2^{|P|-1})$ to calculate deadness levels across all spans and all bidders.⁸

The up-to-date information stored in LastWinBid can be used to recover the winning allocation, in a similar way as with SIMU-OR auctions. Algorithm 2.2, with inputs CurrSpan = N and BidderSet = P , can recover the winning allocation of the auction within $O(\min\{N, |P|\})$ time in the worst-case. Importantly, based on Theorem 8, the union of winning atomic bids across sub-auctions with x items and $\max\{|P| - (N - x), 1\}$ bidders is the set of all *live* atomic bids at span x . Therefore, we can use the same algorithm to recover the set of all live bids. In particular, at any given auction state, we can run Algorithm 2.2 once for each relevant sub-auction. All atomic bids that are returned in this process will comprise the set of live bids.

⁷ C_A^B denotes a binomial coefficient or, in other words, “A choose B”.

⁸The complexity of calculating deadness levels for all spans and bidders is $O\left(\sum_{x=1}^N C_{|P|-1}^{N-x}\right)$, which equals $O(2^{|P|-1})$ if $|P| \leq N$. If $|P| > N$, then $\sum_{x=1}^N C_{|P|-1}^{N-x} \leq \sum_{x=1}^N C_{|P|-1}^{N-x} < \sum_{x=1}^{|P|-1} C_{|P|-1}^x$. Therefore, the complexity is larger than $O(2^N)$ but smaller than $O(2^{|P|-1})$.

ALGORITHM 2.2: Find Winning Allocation of a Sub-Auction

Inputs: CurrSpan, BidderSet // span and bidder set of the sub-auction

While (CurrSpan > 0) and (BidderSet $\neq \emptyset$) **Do**

$v = \text{LastWinBid}[\text{CurrSpan}, \text{BidderSet}].\text{value}$

$s = \text{LastWinBid}[\text{CurrSpan}, \text{BidderSet}].\text{span}$

$p = \text{LastWinBid}[\text{CurrSpan}, \text{BidderSet}].\text{bidder}$

Print $\langle v, s, p \rangle$

CurrSpan = CurrSpan - s

BidderSet = BidderSet $\setminus \{p\}$

Table 3 shows the amount of time it takes to update all sub-auction revenues, bid evaluation metrics, and the LastWinBid array for one incoming live bid. For each configuration of auction size and total bidder number, we simulate 1,000 *general* bids (i.e., 1,000 auction states in total), each containing N nonzero *live atomic bids*. The bidder of each general bid is randomly drawn uniformly from $\{1, 2, \dots, |P|\}$, and the value of each atomic bid in a general bid is assigned as the current *winning* level of its span and bidder plus a random integer drawn uniformly from $\{1, 2, 3\}$. In other words, each atomic bid always has higher value than the current winning level at its span and bidder; this was purposefully chosen in order to evaluate the proposed approach on the “worst case” scenario that maximizes the amount of computation needed for incremental update. The full incremental update is run *after each general bid*. The reported time is the average running time over 1,000 general bids. The experiments were done on a 2.6GHz Intel Core i5 computer with 6G RAM running Linux operating system.

We can see from Table 3 that, while adding more bidders increases the running time of revenue updates exponentially, adding more items only increases running time of revenue updates quadratically, which is consistent with the theoretical computational complexity of Algorithm 2.1 mentioned earlier. When there are 10 bidders in total, we can handle SIMU-XOR auctions of up to 800 items in real-time. Once sub-auction revenues are updated, calculating the winning level for a particular bidder at a given span takes less than a microsecond, again consistent with the fact that winning levels can be obtained by constant time lookup. The time to calculate deadness level for a bidder at a given span ranges from less than a microsecond to 7.83 milliseconds, depending on the total number of bidders and the span value. Overall, our proposed implementation is highly practical, because the number of unique bidders is typically fairly small in many real-world auctions, whereas the number of items can be quite large.

The computational performance of the above straightforward implementations can be improved by conducting at least two preprocessing procedures before the incremental update.

Table 3. Average Auction Incremental Update Time for One Live Bid (in Milliseconds)

Auction Size (N)	Bidder Number (P)	Update All Sub-Auction Revenues	Update All LastWinBid	Full Incremental Updates
100	10	5.47	0.56	6.03
500		221.29	34.87	256.16
800		823.05	115.82	938.87
10	10	0.05	0.02	0.07
	15	1.35	0.49	1.84
	20	72.69	18.70	91.39
	22	361.94	122.95	484.89

First, over the course of the auction, one bidder may submit multiple bids at different times, but not all of them need to be stored. Because of the XOR nature of the auction, multiple bids by the same bidder can be consolidated according to two rules: (1) at a given span, only the atomic bid with the highest value needs to be stored, and (2) if there are multiple atomic bids at a given span with the same highest value, only the earliest one needs to be stored. Second, due to the fact that there can be no more than $(N - x + 1)$ live atomic bids at span x at any auction state, at most $N(N + 1)/2$ atomic bids can “matter” in the calculation of any bidder support metrics (again assuming $N \leq |P|$; otherwise even fewer atomic live bids). Therefore, at any given auction state, we only need to consider the set of bidders who are involved in placing the live atomic bids. Denote these bidders as P_{live} , clearly $|P_{live}| \leq N(N + 1)/2$. This reduces the total number of sub-auctions from $N \cdot 2^{|P|}$ down to $N \cdot 2^{|P_{live}|}$, which can be particularly helpful when $|P|$ is very large (i.e., when $|P| \gg |P_{live}|$).

Efficient Implementation for Revenue and Winning Level Calculation in SIMU-XOR

In combinatorial auction literature, many solutions to the winner determination problem only consider finding *one* allocation (e.g., a set of winning bids) that achieves maximum auction revenue (e.g., Sandholm 1999, 2002; Sandholm et al. 2002; Tennenholtz 2000; Xia et al. 2005). In other words, when multiple allocations all have the maximum revenue, these solutions typically do not try to identify the winner according to certain systematic tie-breaking mechanisms. Our proposed method in the previous section is capable of finding the winner effectively while ensuring the allocation fairness. However, it is useful to note that the winner determination and certain bidder support problems can be solved more efficiently, if one does not consider the allocation fairness issue.

To find revenues and compute some of the bid evaluation metrics more efficiently, we need a faster algorithm to com-

pute $REV_k(x, P)$ for any given span x and bidder set P . While the incremental updating procedure presented in the previous section keeps track of revenues for all sub-auctions, not all of these sub-auction revenues are needed for bid evaluation metric calculation at any given time. Therefore, we can save time by only calculating the sub-auction revenues that are necessary, at the time when they are needed. For notational simplicity, we refer to bidders as $\{1, 2, \dots, |P|\}$. We compute $REV_k(x, P)$ incrementally by sequentially considering bids of each bidder in bidder set P . Let $REV_k^p(x, P)$ be the intermediate value of $REV_k(x, P)$ after the first p bidders in the bidder set P are considered. Therefore, $REV_k^{|P|}(x, P) = REV_k(x, P)$. Meanwhile, $REV_k(x, p)$ is the revenue of sub-auction $[x, p]$ (i.e., essentially the value of the highest atomic bid made by the p^{th} bidder up to span s). Using the dynamic programming approach, the recurrence equation of this problem is as follows:

$$REV_k^p(x, P) = \max \{REV_k^{p-1}(x, P), REV_k^{p-1}(x-1, P) + REV_k(1, p), REV_k^{p-1}(x-2, P) + REV_k(2, p), \dots, REV_k^{p-1}(1, P) + REV_k(x-1, p), REV_k(x, p)\},$$

with the base case $\forall x, P, REV_k^1(x, P) = REV_k(x, 1)$. Note that the XOR constraint is satisfied because at most one atomic bid from the p^{th} bidder can be selected in the above maximization (subject to tie-breaking). The implementation is outlined as Algorithm 3 below. We use array $REV[x]$ to store the revenue of span x , and revise it as each bidder in the bidder set P is considered. We use array $v[x, p]$ to store the revenue of sub-auction $[x, p]$ (i.e., $REV_k(x, p)$).

Algorithm 3: Calculate auction revenues for all spans given a particular set of bidders.

```

Input: P (a set of bidders indexed by 1, 2, ..., |P|);
v[x, p] (revenue of sub-auction [x, p])
// initialize REV array based on the bids made by the first bidder
For x = 0 To N Do REV[x] = v[x, 1]
For p = 2 To |P| Do
    NewREV = REV
    For x = 1 To N Do
        For s = 1 To x Do
            If v[s, p] + REV[x-s] > NewREV[x] Then
                NewREV[x] = v[s, p] + REV[x-s]
    REV = NewREV

```

Unlike Algorithm 2.1, while being able to calculate precise auction and sub-auction winning revenues, the above algorithm is not designed to find winners that satisfy allocative fairness. However, it has significant computational efficiency over Algorithm 2.1. The computational complexity of Algorithm 3 is $O(N^2|P|)$. When the input bidder set is the set of all bidders, the above algorithm can find the maximum auction revenue, stored in $REV[N]$. To find the winners (not considering allocative fairness), one can implement the same LastWinBid data structure used in Algorithm 2.1 (i.e., store the last winning atomic bid after considering the p^{th} bidder for span x in LastWinBid[p, x], and use Algorithm 2.2 to recover the winning atomic bids). To calculate the winning levels for bidder p^* at span x , we need to run this algorithm (1) once for the entire set of bidders to obtain $REV_k(N, P)$, and (2) once for all bidders excluding p^* to obtain $REV_k(N - x, P \setminus \{p^*\})$. In other words, to calculate winning levels for all bidders at all possible spans, the algorithm needs to run for no more than $|P| + 1$, or $O(P)$, times. This algorithm can be used to calculate auction revenue or revenues needed for winning level calculations after the arrival of each new bid. It can reduce both storage (i.e., no need to store revenues for all $N \cdot 2^{|P|}$ possible sub-auctions) and computation (i.e., no need to do incremental updates for exponential amount of sub-auction revenues), compared to Algorithm 2.1. Furthermore, once a particular $REV_k^P(x, P)$ is calculated, we can store it in memory. In subsequent computations within auction state k , the same intermediate result can be directly retrieved (does not need to be recomputed).

Table 4 compares the average running time of calculating all winning levels after the arrival of one live bid for different sizes of SIMU-XOR auctions, using Algorithm 3 and Algorithm 2.1. Clearly, Algorithm 3 can compute all winning levels (i.e., revenue updates and winning level updates) significantly faster than Algorithm 2.1. With 10 bidders, we can handle up to 2,500 items in real time (i.e., within 1

second). More remarkably, with 10 items, we can handle up to 2,500 bidders in real time—a significant improvement over the capability of Algorithm 2.1. This improvement is possible because we do not need to store or update an exponential number of sub-auctions. However, it is important to reiterate that the computational advantage of this approach is limited to the tasks of revenue and winning level calculation as well as winner identification (up to ties) calculation; this approach does not extend to the more comprehensive real-time bidder support that includes deadness level calculation or the identification of auction winners according to the allocative fairness principles.

Summary and Future Work

In this paper, we discuss continuous combinatorial auctions where multiple identical units of one homogeneous item are being auctioned, and bidders place bids on particular units of such item. We consider two variations of SIMU auction, namely the SIMU-OR and SIMU-XOR auctions. While multiple bids of the same bidder are allowed to win a SIMU-OR auction simultaneously, at most one of them is allowed to win a SIMU-XOR auction. We develop comprehensive theoretical characterizations of the two types of SIMU auctions, as well as computational frameworks that use exhaustive state space but can respond to bidder queries in real time. Our theoretical results highlight the similarities and differences between SIMU-OR and SIMU-XOR auctions, in terms of properties of basic metrics that are necessary to provide real-time feedback to bidders, including winning levels and deadness levels. For example, we show that, in contrast to heterogeneous-item combinatorial auctions, where the number of possible live bids in an auction can be exponential in the auction size, the number of possible live bids in a SIMU-OR (SIMU-XOR) auction is linear (quadratic) in the auction size. In addition, our proposed approach has the desired property of ensuring *strict allocative fairness* (i.e., the ability to identify the winners—a single feasible allocation that should win the auction according to systematic fairness principles—which has been underexplored in prior work).

The understanding of SIMU auction dynamics then allows us to design viable computational infrastructure that can compute bidder support information in real-time. We present results from scalable incremental update algorithms for both types of SIMU auctions. For SIMU-OR auctions, we show that real-time bidder support is achievable for auctions up to 500,000 units. For SIMU-XOR auctions, due to their highly combinatorial nature, our computational design for real-time bidder support requires exponential storage with respect to the number of unique bidders in the auction (which is typically

Table 4. Average Winning Levels Calculation Time for One Live Bid (in Milliseconds)

Auction Size (N)	Bidder Number (P)	Calculate All Winning Levels Using Algorithm 2.1	Calculate All Winning Levels Using Algorithm 3
1000	10	2014.19	128.03
1500		4152.93	288.49
2000		8186.92	528.73
2500		13133.39	831.33
10	1000	Cannot fit in memory	142.70
	1500	Cannot fit in memory	317.22
	2000	Cannot fit in memory	573.85
	2500	Cannot fit in memory	895.16
100	100	Cannot fit in memory	142.86
150	150	Cannot fit in memory	680.60

fairly small in many real-world auctions), but only linear storage with respect to the number of units in the auction (which can be quite large). Nonetheless, our computational approach can provide real-time bidder support for SIMU-XOR auctions of reasonable, practically relevant sizes. Moreover, we also propose a more efficient algorithm for more restricted bidder support settings, which can determine winners (up to ties) and calculate winning levels for either thousands of items or thousands of bidders for SIMU-XOR auctions.

Providing real-time bidder support in continuous SIMU auctions not only facilitates bidders' decision making, but also benefits the auctioneers in at least two major ways. First, having a real-time bidder support system greatly increases the *usability* and *feasibility* of the auctions, which is imperative for the *acceptance* of such mechanism. Adomavicius, Curley, et al. (2013b) found that providing information feedback effectively increases the perceived usefulness, perceived ease of use, and the intention to use in continuous combinatorial auctions. Clearly, acceptance of the mechanism is a prerequisite before any meaningful economic benefit can be realized and extracted. Second, prior research has shown that providing feedback information can significantly increase the allocative efficiency (i.e., the *total value* generated in the trading process) of the auctions, and the auctioneers can accordingly extract more revenue (discussed in Adomavicius, Curley et al. 2013a for MISU auctions).⁹

⁹In general, the net value generated in an auction is distributed between bidders (as bidder surplus) and auctioneers (as revenue), and the specific configurations of information feedback can affect that distribution. Auctioneers can customize the comprehensive bidder support system (e.g., provide only a subset of available feedback information) to accommodate their specific goals and contexts.

Throughout the paper, we have highlighted several similarities and differences between SIMU auctions and MISU auctions. In summary, a key difference is that the total number of possible bundles to bid on is exponential in MISU auctions but linear in SIMU auctions. This results in different space complexity in implementations (i.e., exponential space is required to keep track of all sub-auctions in a MISU auction, but only linear space is needed in a SIMU auction). Interestingly, despite such difference, the incremental update of sub-auction revenue and the calculation of winning levels are conceptually very similar for the two auction types. Another similarity is the utility of the sub-auction concept in the theoretical analysis and practical implementation of both auction types. As with MISU-OR auctions discussed in Adomavicius and Gupta (2005) and MISU-XOR auctions discussed in Petrakis et al. (2012), the sub-auction concept continues to prove crucial in understanding the dynamics of SIMU auctions.

Our work has significant value for the practical applications of SIMU auctions. For existing applications, such as industrial procurement, commodity auctions, and network capacity auctions, implementing the real-time bidder support capability proposed in this paper provides more transparency to the participants about the continuous combinatorial auction dynamics and, thus, can facilitate better decision-making of bidders, and thereby improve both the economic outcomes and the user acceptance of such auctions (Adomavicius, Curley et al. 2013a, 2013b). Especially in SIMU auctions that put higher participation burden on its participants (e.g., auctions with large sizes, with dynamic (real-time) auction progressions, and with high diversity in bundle preferences among bidders), the benefits of real-time bidder support would be most evident. Equipped with real-time bidder support capability, one can seek to deploy the continuous combinatorial auction mechanism not only in many business-

oriented domains, but also in consumer-oriented domains, where the implementation of this mechanism is nonexistent.

Our research also opens up a number of important future research directions, one of which is to consider the impact of bidder support on the game-theoretic properties of various specific instances of SIMU auctions. A typical problem concerns the *equilibrium bidding strategies* in such auctions. In general, little is known about the equilibrium bidding strategy in combinatorial auctions (Pekec and Rothkopf 2003) except for very special cases (e.g., de Vries et al. 2007), and there is evidence that equilibrium bidding may not lead to allocative efficiency or desirable bidder surplus in these auctions (Banks et al. 1989; Katok and Roth 2004). Meanwhile, Bichler et al. (2017) has shown that providing information feedback (e.g., deadness levels) can make *straightforward bidding* (i.e., always bid on the packages with highest profit potential) an equilibrium strategy, under certain assumptions of bidder valuations. Therefore, it is valuable to understand the equilibrium bidding strategies in various specific implementations of SIMU auctions (if one exists), as well as the effect of providing bidder support on such strategies. Another direction is to design *incentive compatible* mechanisms for SIMU auctions. For instance, Edelman et al. (2007) found that the generalized second price process is incentive incompatible for multi-unit auctions. Future research should investigate alternative mechanisms that promote incentive compatible bidding behaviors.

Overall, providing real-time bidder support addresses the fundamental *acceptance and transparency* issue of continuous combinatorial auctions. Our work, by tackling this issue, informs future research to study the above game-theoretic problems, via theoretical analyses and/or empirical examinations, for various kinds of specialized SIMU auctions (e.g., clock auctions, proxy auctions) under particular settings of bidder/auctioneer behaviors (e.g., bidding strategies and valuation distributions).

In addition to studying the game-theoretic and mechanism design aspects of SIMU auctions, there are several other interesting directions for future work. Probably the most general type of combinatorial auctions is multi-item multi-unit (MIMU) auctions (see Gonen and Lehmann 2000; Leyton-Brown et al. 2000) where multiple heterogeneous items, each with multiple identical units, are being sold. Future work can investigate the possibilities to examine the theoretical and algorithmic developments for providing real-time bidder support in MIMU auctions. Because the sub-auction construct plays an important role in understanding both MISU and SIMU auctions, one future research direction is to address bidder support issues in MIMU auctions via the use of sub-auctions.

References

- Adomavicius, G., Curley, S. P., Gupta, A., and Sanyal, P. 2012. "Effect of Information Feedback on Bidder Behavior in Continuous Combinatorial Auctions," *Management Science* (58:4), pp. 811-830.
- Adomavicius, G., Curley, S. P., Gupta, A., and Sanyal, P. 2013a. "Impact of Information Feedback in Continuous Combinatorial Auctions: An Experimental Study of Economic Performance," *MIS Quarterly* (37:1), pp. 55-76.
- Adomavicius, G., Curley, S. P., Gupta, A., and Sanyal, P. 2013b. "User Acceptance of Complex Electronic Market Mechanisms: Role of Information Feedback," *Journal of Operations Management* (31:6), pp. 489-503.
- Adomavicius, G., and Gupta, A. 2005. "Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions," *Information Systems Research* (16:2), pp. 169-185.
- Adomavicius, G., Gupta, A., and Sanyal, P. 2012. "Effect of Information Feedback on the Outcomes and Dynamics of Multisourcing Multiattribute Procurement Auctions," *Journal of Management Information Systems* (28:4), pp. 199-230.
- Adomavicius, G., Sanyal, P., Gupta, A., and Curley, S. 2007. "Design and Effects of Information Feedback in Continuous Combinatorial Auctions," in *Proceedings of the 28th International Conference on Information Systems*, Montreal, pp. 1-17.
- Ausubel, L. M., Cramton, P. C., and Milgrom, P. 2005. "The Clock-Proxy Auction: A Practical Combinatorial Auction Design," in *Handbook of Spectrum Auction Design*, M. Bichler and J. K. Goeree (eds.), Cambridge, MA: MIT Press, pp. 120-140.
- Ausubel, L. M., and Milgrom, P. R. 2002. "Ascending Auctions with Package Bidding," *Advances in Theoretical Economics* (1:1), Article 1.
- Ausubel, L. M., and Cramton, P. C. 1995. "Demand Revelation and Inefficiency in Multi-Unit Auctions," Mimeo, University of Maryland.
- Banks, J. S., Ledyard, J. O., and Porter, D. P. 1989. "Allocating Uncertain and Unresponsive Resources: An Experimental Approach," *The Rand Journal of Economics* (20:1), pp. 1-25.
- Bichler, M., Davenport, A., Hohner, G., and Kalagnanam, J. 2006. "Industrial Procurement Auctions," in *Combinatorial Auctions*, P. Cramton, Y. Shoham, and R. Steinberg (eds.), Cambridge, MA: MIT Press, pp. 593-612.
- Bichler, M., and Goeree, J. K. 2017. *Handbook of Spectrum Auction Design*, Cambridge, UK: Cambridge University Press.
- Bichler, M., Gupta, A., and Ketter, W. 2010. "Research Commentary—Designing Smart Markets," *Information Systems Research* (21:4), pp. 688-699.
- Bichler, M., Hao, Z., and Adomavicius, G. 2017. "Coalition-Based Pricing in Ascending Combinatorial Auctions," *Information Systems Research* (28:1), pp. 159-179.
- Bichler, M., Shabalin, P., and Pikhovskiy, A. 2009. "A Computational Analysis of Linear Price Iterative Combinatorial Auction Formats," *Information Systems Research* (20:1), pp. 33-59.
- Brewer, P. J., and Plott, C. R. 1996. "A Binary Conflict Ascending Price Bicap Mechanism for the Decentralized Allocation of the

- Right to Use Railroad Tracks,” *International Journal of Industrial Organization* (14:6), pp. 857-886.
- Caplice, C. 2007. “Electronic Markets for Truckload Transportation,” *Production and Operations Management* (16:4), pp. 423-436.
- Cramton, P. 2013. “Spectrum Auction Design,” *Review of Industrial Organization* (42:2), pp. 161-190.
- Cramton, P., Shoham, Y., and Steinberg, R. (eds.). 2006. *Combinatorial Auctions*, Cambridge, MA: MIT Press.
- Cramton, P., Shoham, Y., and Steinberg, R. 2006. “Introduction to Combinatorial Auctions,” *Combinatorial Auction*, Cambridge, MA: MIT Press, pp. 1-13.
- de Vries S, Schummer J, and Vohra R. 2007. “On Ascending Vickrey Auctions for Heterogeneous Objects,” *Journal of Economic Theory* (132:1), pp. 95-118.
- Edelman, B., Ostrovsky, M., and Schwarz, M. 2007. “Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords,” *The American Economic Review* (97:1), pp. 242-259.
- Eso, M., Ghosh, S., Kalagnanam, J., and Ladanyi, L. 2005. “Bid Evaluation in Procurement Auctions with Piecewise Linear Supply Curves,” *Journal of Heuristics* (11:2), pp. 147-173.
- Goeree, J. K., and Holt, C. A. 2010. “Hierarchical Package Bidding: A Paper and Pencil Combinatorial Auction,” *Games and Economic Behavior* (70:1), pp. 146-169.
- Gonen, R., and Lehmann, D. 2000. “Optimal Solutions for Multi-Unit Combinatorial Auctions: Branch and Bound Heuristics,” in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, New York: ACM, MN, pp. 13-20.
- Granados, N., and Gupta, A. 2013. “Transparency Strategy: Competing with Information in a Digital World,” *MIS Quarterly* (37:2), pp. 637-641.
- Granados, N. F., Gupta, A., Kauffman, R. J. 2010. “Information Transparency in Business-to-Consumer Markets: Concepts, Framework, and Research Agenda,” *Information Systems Research* (21:2), pp. 207-226.
- Guler, K., Bichler, M., and Petrakis, I. 2016. “Ascending Combinatorial Auctions with Risk Averse Bidders,” *Group Decision and Negotiation* (25:3), pp. 609-639.
- Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. “Design Science in Information Systems Research,” *MIS Quarterly* (28:1), pp. 75-105.
- Hu, T. C. 1969. *Programming and Network Flows*, Reading, MA: Addison-Wesley.
- Katok, E., and Roth, A. E. 2004. “Auctions of Homogeneous Goods with Increasing Returns: Experimental Comparison of Alternative “Dutch” Auctions,” *Management Science* (50:8), pp. 1044-1063.
- Kellerer, H., Pferschy, U., and Pisinger, D. 2004. *Introduction to NP-Completeness of Knapsack Problems*, Berlin: Springer.
- Kelly, F., and Steinberg, R. 2000. “A Combinatorial Auction with Multiple Winners for Universal Service,” *Management Science* (46:4), pp. 586-596.
- Kothari, A., Parkes, D. C., and Suri, S. 2005. “Approximately-Strategy Proof and Tractable Multiunit Auctions,” *Decision Support Systems* (39:1), pp. 105-121.
- Leyton-Brown, K., Pearson, M., and Shoham, Y. 2000. “Towards a Universal Test Suite for Combinatorial Auction Algorithms,” in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, New York: ACM, pp. 66-76.
- Leyton-Brown, K., Shoham, Y., and Tennenholtz, M. 2000. “An Algorithm for Multi-Unit Combinatorial Auctions,” in *Proceedings of the 17th National Conference on Artificial Intelligence*, Atlanta, pp. 56-61.
- Maskin, E., Riley, J., and Hahn, F. 1989. “Optimal Multi-Unit Auctions,” in *The Economics of Missing Markets, Information, and Games*, F. Hahn (ed.), Oxford, UK: Oxford University Press, pp. 312-335.
- Milgrom, P. 2000. “An Economist’s Vision of the B-to-B Marketplace,” Executive White Paper, Perfect.com, Palo Alto, CA.
- Murillo, J., Muñoz, V., López, B., and Busquets, D. 2008. “A Fair Mechanism for Recurrent Multi-Unit Auctions,” in *Multiagent System Technologies: Lecture Notes in Computer Science* (Vol. 55244), R. Bergman, G. Lindemann, S. Kirn, and M. Pěchouček (eds.), Berlin: Springer, pp. 147-158.
- Narahari, Y., and Dayama, P. 2005. “Combinatorial Auctions for Electronic Business,” *Sadhana* (30:2-3), pp. 179-211.
- Nisan, N. 2000. “Bidding and Allocation in Combinatorial Auctions,” in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, New York: ACM, pp. 1-12.
- Parkes, D. C. 1999. “iBundle: An Efficient Ascending Price Bundle Auction,” in *Proceedings of the 1st ACM Conference on Electronic Commerce*, New York: ACM, pp. 148-157.
- Pekec, A., and Rothkopf, M. H. 2003. “Combinatorial Auction Design,” *Management Science* (49:11), pp. 1485-1503.
- Petrakis, I., Ziegler, G., and Bichler, M. 2012. “Ascending Combinatorial Auctions with Allocation Constraints: On Game Theoretical and Computational Properties of Generic Pricing Rules,” *Information Systems Research* (24:3), pp. 768-786.
- Porter, D., Rassenti, S., Roopnarine, A., and Smith, V. 2003. “Combinatorial Auction Design,” *Proceedings of the National Academy of Sciences* (100:19), 11153-11157.
- Rassenti, S. J., Smith, V. L., and Bulfin, R. L. 1982. “A Combinatorial Auction Mechanism for Airport Time Slot Allocation,” *The Bell Journal of Economics* (13:2), pp. 402-417.
- Rothkopf, M. H., Pekec, A., and Harstad, R. M. 1998. “Computationally Manageable Combinatorial Auctions,” *Management Science* (44:8), pp. 1131-1147.
- Sandholm, T. 1999. “An Algorithm for Optimal Winner Determination in Combinatorial Auctions,” in *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm, pp. 542-547.
- Sandholm, T. 2002. “Algorithm for Optimal Winner Determination in Combinatorial Auctions,” *Artificial Intelligence* (135:1), pp. 1-54.
- Sandholm, T., Suri, S., Gilpin, A., and Levine, D. 2002. “Winner Determination in Combinatorial Auction Generalizations,” in *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems* (Part 1), New York: ACM, pp. 69-76.
- Scheffel, T., Piskovsky, A., Bichler, M., and Guler, K. 2011. “An Experimental Comparison of Linear and Nonlinear Price Combinatorial Auctions,” *Information Systems Research* (22:2), pp. 346-368.
- Sen, A., and Bagchi, A. 2012. “Providing Information Feedback to Bidders in Online Multi-Unit Combinatorial Auctions,” *Pro-*

- ceedings of the 18th Americas Conference on Information Systems, Seattle, WA.
- Simon, H. A. 1982. *Models of Bounded Rationality: Empirically Grounded Economic Reason*, (Vol. 3), Cambridge, MA: MIT Press.
- Tennenholtz, M. 2000. "Some Tractable Combinatorial Auctions," in *Proceedings of the 17th National Conference on Artificial Intelligence*, Austin, TX, pp. 98-103.
- TradeSlot. 2017. "Timber Supply Auctions—VicForests" (<http://www.tradeslot.com/vicforests-b2b-timber-supply-auctions/>).
- Vazirani, V. 2003. *Approximation Algorithms*, New York: Springer.
- Xia, M., Stallaert, J., and Whinston, A. B. 2005. "Solving the Combinatorial Double Auction Problem," *European Journal of Operational Research* (164:1), pp. 239-251.

About the Authors

Gediminas Adomavicius is a professor of Information and Decision Sciences at the University of Minnesota, where he also holds Carolyn I. Anderson Chair in Business Education Excellence. He received his Ph.D. in Computer Science from New York University. His research interests include recommender systems, machine learning, and electronic market mechanisms. His work has been published in leading Information Systems and Computer Science journals. He received the NSF CAREER Award for his research on recommender systems. He has served on the editorial boards of several leading academic journals, including as senior editor for

Information Systems Research and *MIS Quarterly*, and is a Distinguished Fellow of the INFORMS Information Systems Society.

Alok Gupta is the Associate Dean for Faculty and Research at the Carlson School of Management, University of Minnesota. He is Curtis L. Carlson Schoolwide Chair in Information Management, and the former chair of the Information and Decision Sciences Department. He was awarded the prestigious NSF CAREER Award for his research on dynamic pricing mechanisms on the Internet in 2001, and was named an INFORMS Information Systems Society Distinguished Fellow in 2014. He currently serves as the editor-in-chief of *Information Systems Research*. He served as a senior editor for *Information Systems Research* and has been serving as an associate editor for *Management Science*. His research has appeared in several information systems, economics, and computer science journals including *Management Science*, *Information Systems Research*, and *MIS Quarterly*.

Mochen Yang is an assistant professor in the Department of Operation and Decision Technologies at Indiana University's Kelley School of Business. He received a Ph.D. from Carlson School of Management at University of Minnesota, and a bachelor's degree from the School of Economics and Management at Tsinghua University. His research focuses on designing computational artifacts to facilitate decision making in complex market mechanisms. His research has been published in *Information Systems Research*, and presented at the Conference on Information Systems and Technology, the Workshop on Information Technology and Systems, and the Winter Conference on Business Analytics.

Designing Real-Time Feedback for Bidders In Homogeneous-Item Continuous Combinatorial Auctions

Gediminas Adomavicius and Alok Gupta

Carlson School of Management, University of Minnesota, Minneapolis, MN 55455 U.S.A.
{gedas@umn.edu} {gupta037@umn.edu}

Mochen Yang

Kelley School of Business, Indiana University, Bloomington, IN 47405 U.S.A.
{yangmo@iu.edu}

Appendix A

Proofs of all Theorems, Corollaries, and Lemmas for SIMU-OR Auctions

Theorem 1. For a SIMU-OR auction, given auction state k and new bid b_{k+1} :

1. $(\forall x < s(b_{k+1}))(WIN_{k+1}[x] = WIN_k[x]);$
2. $(\forall x \geq s(b_{k+1}))(WIN_{k+1}[x] = \max_{<}[WIN_k[x], \{b_{k+1}\} \cup WIN_k[x - s(b_{k+1})]])$

Proof. The first statement follows immediately from the definition of $WIN_{k+1}[x]$, because $b_{k+1} \notin WIN_{k+1}[x] \Rightarrow WIN_{k+1}[x] = WIN_k[x]$. The second statement immediately follows from two facts: (a) the only way to have $WIN_{k+1}[x] \neq WIN_k[x]$ is when $b_{k+1} \in WIN_{k+1}[x]$; and (b) since bid b_{k+1} “covers” only $s(b_{k+1})$ items, the best possible allocation among $C_{k+1}[x]$ involving b_{k+1} should also involve the best possible prior bids (i.e., from B_k) covering the remaining $x - s(b_{k+1})$ items. Hence, the only alternative to $WIN_k[x]$ would be $\{b_{k+1}\} \cup WIN_k[x - s(b_{k+1})]$. ■

Theorem 2. For a SIMU-OR auction, given auction state k and new bid b_{k+1} ,

$$b_{k+1} \in WIN_{k+1} \Leftrightarrow v(b_{k+1}) > REV_k(N) - REV_k(N - s(b_{k+1}))$$

Proof. Immediate from Theorem 1, since $b_{k+1} \in WIN_{k+1}$ if and only if $WIN_k < \{b_{k+1}\} \cup WIN_k[N - s(b_{k+1})]$. Based on the definition of strict total order $<$ and taking into account that WIN_k chronologically precedes allocation $\{b_{k+1}\} \cup WIN_k[N - s(b_{k+1})]$, we have that $v(b_{k+1}) > REV_k(N) - REV_k(N - s(b_{k+1}))$. ■

Corollary 2a. For a SIMU-OR auction, given auction state k , the winning level at span x is calculated as $WL_k(x) = REV_k(N) - REV_k(N - x)$

Proof. Immediate from Theorem 2 and the definition of winning level. ■

Corollary 2b. For a SIMU-OR auction, given auction state k and new bid b_{k+1} :

$$b_{k+1} \in WIN_{k+1}[i] \Leftrightarrow v(b_{k+1}) > REV_k(i) - REV_k(i - s(b_{k+1}))$$

Proof. Immediate from Theorem 2, by considering sub-auction i . ■

Theorem 3. For a SIMU-OR auction, given auction state k and any bid $b \in B_k$:

$$b \in LIVE_k \Leftrightarrow (\exists x \geq s(b))(b \in WIN_k[x])$$

Proof. First, we prove $(\exists x \geq s(b))(b \in WIN_k[x]) \Rightarrow b \in LIVE_k$. Consider two following cases for x :

[Case 1: $x = N$] $b \in WIN_k[N] \Rightarrow b \in WIN_k \Rightarrow b \in LIVE_k$.

[Case 2: $x < N$] Consider new bid b_{k+1} , such that $s(b_{k+1}) = N - x$ and $v(b_{k+1}) > REV_k - v(WIN_k[x])$. Furthermore, let's consider allocation $\alpha = \{b_{k+1}\} \cup WIN_k[x]$. By definition, $s(\alpha) = s(b_{k+1}) + s(WIN_k[x]) \leq N - x + x = N$ and $v(\alpha) = v(b_{k+1}) + v(WIN_k[x]) > REV_k$. I.e., $s(\alpha) \leq N$ and $v(\alpha) > REV_k$. Hence, $\alpha = WIN_{k+1}$. Because $b \in WIN_k[x]$ and $WIN_k[x] \subseteq \alpha$, we have that $b \in WIN_{k+1}$. Consequently, $b \in LIVE_k$.

Next, we prove $b \in LIVE_k \Rightarrow (\exists x \geq s(b))(b \in WIN_k[x])$. From $b \in LIVE_k$ we have that there exists an auction state l ($l \geq k$) such that $b \in WIN_l$. Denote WIN_l as: $WIN_l = \{b\} \cup \beta_{1,k} \cup \beta_{k+1,l}$, where $\beta_{1,k} = \{b' \in B_k \wedge b' \neq b\}$ and $\beta_{k+1,l} = \{b' \in WIN_l | b' \notin B_k\}$. Thus, bid sets $\{b\}$, $\beta_{1,k}$, and $\beta_{k+1,l}$ are pairwise disjoint. Suppose otherwise, i.e., $(\forall x \geq s(b))(b \notin WIN_k[x])$. Consider bid set $\alpha = \{b\} \cup \beta_{1,k}$. Denote $x' = s(\alpha) = s(b) + s(\beta_{1,k})$. Since $x' \geq s(b)$, we have that $b \notin WIN_k[x']$. Furthermore, by definition, $\alpha \subseteq B_k[x']$ and $b \in \alpha$. Therefore, $\alpha < WIN_k[x']$. Consequently, $WIN_l = \alpha \cup \beta_{k+1,l} < WIN_k[x'] \cup \beta_{k+1,l}$, i.e., we have found a set of bids $WIN_k[x'] \cup \beta_{k+1,l} \subseteq B_l$ that is better than WIN_l , which is a contradiction to the definition of WIN_l . ■

Corollary 3a. For a SIMU-OR auction, given auction state k , the deadness level at span x is calculated as $DL_k(x) = \min_{i \in \{x, \dots, N\}} [REV_k(i) - REV_k(i - x)]$

Proof. Immediate from Theorem 3 and Corollary 2b. I.e., let $s(b_{k+1}) = x$, then $b_{k+1} \in LIVE_{k+1} \Leftrightarrow (\exists i \geq x)(b_{k+1} \in WIN_{k+1}[i]) \Leftrightarrow (\exists i \geq x)(v(b_{k+1}) > REV_k(i) - REV_k(i - x)) \Leftrightarrow v(b_{k+1}) > \min_{i \in \{x, \dots, N\}} [REV_k(i) - REV_k(i - x)]$. The results follow based on the definition of deadness level. ■

Theorem 4. For a SIMU-OR auction, given any auction state k and sub-auctions x and y , the following statements are true:

1. $DL_k(x) \leq REV_k(x)$
2. $DL_k(x) \leq WL_k(x)$
3. $DL_k(N) = WL_k(N) = REV_k(N)$
4. $DL_k(x) \leq DL_{k+1}(x)$
5. $x \leq y \Rightarrow DL_k(x) \leq DL_k(y)$ and $WL_k(x) \leq WL_k(y)$
6. $b \in WIN_k \Rightarrow WL_k(s(b)) \leq v(b)$
7. $b \in LIVE_k \Rightarrow DL_k(s(b)) \leq v(b)$

Proof. Statements 1-7 follow immediately from the definitions of $DL_k(x)$ and $WL_k(x)$ as well as the definition and properties of $REV_k(x)$. ■

Theorem 5. In a SIMU-OR auction of size N , for any auction state k we have: $|LIVE_k| \leq N$.

Proof. Based on Corollary 3b, we have that $LIVE_k = \bigcup_{1 \leq i \leq N} WIN_k[i]$. Therefore, instead of analyzing set $LIVE_k$, we will focus on the set $\bigcup_{1 \leq i \leq N} WIN_k[i]$. Specifically, we will show that $|\bigcup_{1 \leq i \leq x} WIN_k[i]| \leq x$ for all $x = 1, \dots, N$. In other words, given any $x = 1, \dots, N$, we will show that, if we consider *all* sub-auctions from 1 to x , no more than x different bids can appear in the winning allocations of these sub-auctions.

We will prove that $|\bigcup_{1 \leq i \leq x} WIN_k[i]| \leq x$ by induction on x . Obviously, the base case $x = 1$ holds, since $WIN_k[1]$ is either a singleton set (i.e., it represents the largest-valued 1-item bid that was submitted to the auction) or an empty set (if no 1-item bids were submitted so far). In other words, $|WIN_k[1]| \leq 1$. Now let's assume that this statement holds for x , i.e., $|\bigcup_{1 \leq i \leq x} WIN_k[i]| \leq x$, and prove that it then must hold for $x + 1$ as well, i.e., $|\bigcup_{1 \leq i \leq x+1} WIN_k[i]| \leq x + 1$.

Let's assume otherwise, i.e., $|\bigcup_{1 \leq i \leq x+1} WIN_k[i]| \geq x + 2$, which means that there must exist at least two bids $b' \in WIN_k[x + 1]$ and $b'' \in WIN_k[x + 1]$, such that $b' \notin \bigcup_{1 \leq i \leq x} WIN_k[i]$ and $b'' \notin \bigcup_{1 \leq i \leq x} WIN_k[i]$. Let's denote the spans of b' and b'' as s' and s'' (obviously, $s', s'' \geq 1$), respectively, and rewrite $WIN_k[x + 1]$ as $WIN_k[x + 1] = \{b'\} \cup \{b''\} \cup WIN_k[x + 1 -$

$s' - s''$], where $WIN_k[x + 1 - s' - s'']$ denotes the “rest” of $WIN_k[x + 1]$, besides b' and b'' . We can write this way for 3 reasons: (a) we know that b' and b'' belong to $WIN_k[x + 1]$; (b) $WIN_k[x + 1]$ should involve the *best* possible allocation that can cover the remaining $x + 1 - s' - s''$ items of sub-auction $x + 1$, not covered by b' and b'' , hence $WIN_k[x + 1 - s' - s'']$; (c) since $b', b'' \notin \bigcup_{1 \leq i \leq x} WIN_k[i]$, we have that $b', b'' \notin WIN_k[x + 1 - s' - s'']$, i.e., we are not “double counting” b' and b'' .

Now consider the following allocation $C' = \{b'\} \cup WIN_k[x + 1 - s' - s'']$. Obviously, $s(C') \leq x + 1 - s''$. Also, since $b' \in C'$ and $b' \notin WIN_k[x + 1 - s'']$, we have that $C' \neq WIN_k[x + 1 - s'']$. Or, more precisely, $C' < WIN_k[x + 1 - s'']$. Finally, we go back to the expression for $WIN_k[x + 1]$ and plug in the results, i.e., $WIN_k[x + 1] = \{b'\} \cup \{b''\} \cup WIN_k[x + 1 - s' - s''] = C' \cup \{b''\} < WIN_k[x + 1 - s''] \cup \{b''\}$. Note that there is no danger of “double counting” in the latest expression either, since $b'' \notin WIN_k[x + 1 - s'']$. Therefore, we derive that $WIN_k[x + 1] < WIN_k[x + 1 - s''] \cup \{b''\}$, i.e., there exists an allocation with the span less than or equal $x + 1$ that is better than the best allocation with that span, $WIN_k[x + 1]$. Contradiction. Therefore, our assumption that $|\bigcup_{1 \leq i \leq x+1} WIN_k[i]| \geq x + 2$, was incorrect. This leads us to the result that $|\bigcup_{1 \leq i \leq x+1} WIN_k[i]| \leq x + 1$, which completes the proof by induction.

Therefore, we have $|\bigcup_{1 \leq i \leq x} WIN_k[i]| \leq x$ for all $x = 1, \dots, N$. The proof of the theorem is concluded by choosing $x = N$, i.e., $|LIVE_k| = |\bigcup_{1 \leq i \leq N} WIN_k[i]| \leq N$. ■

Appendix B

Proofs of all Theorems, Corollaries, and Lemmas for SIMU-XOR Auctions

Theorem 6. For a SIMU-XOR auction, given auction state k and new bid $b_{k+1} = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$, $\forall x \in \{1, 2, \dots, N\}$, $\forall P \subseteq \mathbb{P}$:

1. If $p(b_{k+1}) \notin P$, then $WIN_{k+1}[x, P] = WIN_k[x, P]$.
2. If $p(b_{k+1}) \in P$, then

$$WIN_{k+1}[x, P] = \max_{<} \{WIN_k[x, P], WIN_k[x - 1, P \setminus \{p(b_{k+1})\}] \cup \{\sigma_1\}, WIN_k[x - 2, P \setminus \{p(b_{k+1})\}] \cup \{\sigma_2\}, \dots, WIN_k[1, P \setminus \{p(b_{k+1})\}] \cup \{\sigma_{x-1}\}, \{\sigma_x\}\}.$$

Proof. Let $p^* = p(b_{k+1})$. The first statement follows immediately from the definition of $WIN_{k+1}[x, P]$. I.e., $p^* \notin P \Rightarrow \forall \sigma \in b_{k+1}, \sigma \notin WIN_{k+1}[x, P] \Rightarrow WIN_{k+1}[x, P] = WIN_k[x, P]$. The second statement immediately follows from two facts: (a) the only way to have $WIN_{k+1}[x, P] \neq WIN_k[x, P]$ is when $\exists \sigma \in b_{k+1}, \sigma \in WIN_{k+1}[x, P]$; and (b) since any atomic bid σ “covers” only $s(\sigma)$ items, the best allocation among $C_{k+1}[x, P]$ involving σ should also involve the best possible prior bids (i.e., from B_k) covering the remaining $x - s(\sigma)$ items, made by the set of bidders excluding $p(\sigma) = p^*$. Hence, the only alternative to $WIN_k[x, P]$ is $WIN_k[x - s(\sigma), P \setminus \{p^*\}] \cup \{\sigma\}$. Following the same logic, when we consider a new *general* bid $b = (\sigma_1, \sigma_2, \dots, \sigma_N)$, the possible alternatives to $WIN_k[x, P]$ include $WIN_k[x - 1, P \setminus \{p^*\}] \cup \{\sigma_1\}$, $WIN_k[x - 2, P \setminus \{p^*\}] \cup \{\sigma_2\}$, ..., $WIN_k[1, P \setminus \{p^*\}] \cup \{\sigma_{x-1}\}$, and $\{\sigma_x\}$, and the exact winning allocation is determined by total order $<$ that satisfies allocative fairness. ■

Theorem 7. For a SIMU-XOR auction, given auction state k and new atomic bid σ . Then $\sigma \in WIN_{k+1} \Leftrightarrow v(\sigma) > REV_k(N, \mathbb{P}) - REV_k(N - s(\sigma), \mathbb{P} \setminus \{p(\sigma)\})$

Proof. Let $s(\sigma) = x$ and $p(\sigma) = p^*$. Immediate from Theorem 6, because $\sigma \in WIN_{k+1}$ if and only if $WIN_k < \{\sigma\} \cup WIN_k[N - x, \mathbb{P} \setminus \{p^*\}]$. Based on the definition of strict total order $<$ and the fact that WIN_k chronologically precedes allocation $\{\sigma\} \cup WIN_k[N - x, \mathbb{P} \setminus \{p^*\}]$, we have $v(\sigma) > REV_k(N, \mathbb{P}) - REV_k(N - x, \mathbb{P} \setminus \{p^*\})$. ■

Corollary 7a. For a SIMU-XOR auction, given auction state k , the winning level at span x for bidder p^* is calculated as $WL_k(x, p^*) = REV_k(N, \mathbb{P}) - REV_k(N - x, \mathbb{P} \setminus \{p^*\})$

Proof. Immediate from Theorem 7 and the definition of winning level. ■

Corollary 7b. For a SIMU-XOR auction, given auction state k and new atomic bid σ . For all bidder set P such that $p(\sigma) \in P \subseteq \mathbb{P}$

$$\sigma \in WIN_{k+1}[i, P] \Leftrightarrow v(\sigma) > REV_k(i, P) - REV_k(i - s(\sigma), P \setminus \{p(\sigma)\})$$

Proof. Immediate from Theorem 7, by considering sub-auction $[i, P]$. ■

Theorem 8. For a SIMU-XOR auction, given auction state k and any atomic bid $\sigma \in B_k$.

1. If $s(\sigma) \leq N - |\mathbb{P}|$, then: $\sigma \in LIVE_k \Leftrightarrow \sigma = WIN_k[s(\sigma), p(\sigma)]$;
2. If $s(\sigma) > N - |\mathbb{P}|$, then: $\sigma \in LIVE_k \Leftrightarrow \exists Q \subseteq \mathbb{P}$ with $p(\sigma) \in Q$ and $|Q| = |\mathbb{P}| - (N - s(\sigma))$ such that $\sigma = WIN_k[s(\sigma), Q]$

Proof. [Case 1] Note that, if $s(\sigma) \leq N - |\mathbb{P}|$, then the atomic bid is automatically live, as long as it is not outbid by other bids submitted by the same bidder with same or smaller span (i.e., $\sigma = WIN_k[s(\sigma), p(\sigma)]$). This is because, for an arbitrary atomic bid σ , we can easily construct a “complementary” future allocation β which satisfies $s(\beta) = N - s(\sigma)$, $p(\beta) = \mathbb{P} \setminus \{p(\sigma)\}$, and $v(\beta) > REV_k(N, \mathbb{P}) - REV_k(s(\sigma), p(\sigma))$. It follows that $\{\sigma\} \cup \beta = WIN_{t(\beta)}$ and $\sigma \in LIVE_k$.

[Case 2] We first prove $\exists Q \subseteq \mathbb{P}$ with $p(\sigma) \in Q$ and $|Q| = |\mathbb{P}| - (N - s(\sigma))$ such that $\sigma = WIN_k[s(\sigma), Q] \Rightarrow \sigma \in LIVE_k$. If $s(\sigma) = N$, then $N - s(\sigma) = 0$ and $|Q| = |\mathbb{P}|$, which implies that $WIN_k[s(\sigma), Q] = WIN_k$. Therefore, $\sigma = WIN_k[s(\sigma), Q] \Rightarrow \sigma = WIN_k \Rightarrow \sigma \in LIVE_k$. If instead $s(\sigma) < N$, consider a future allocation β where $s(\beta) = N - s(\sigma)$, $|p(\beta)| = N - s(\sigma)$, and $p(\beta) \cap Q = \emptyset$. In other words, β covers $N - s(\sigma)$ items and contains bidders that are not in Q . Suppose $v(\beta) > REV_k(N, \mathbb{P}) - REV_k(x, Q)$, then $WIN_k[s(\sigma), Q] \cup \beta = WIN_{t(\beta)}$. Therefore, $\sigma \in LIVE_k$.

We then prove $\sigma \in LIVE_k \Rightarrow \exists Q \subseteq \mathbb{P}$ with $p(\sigma) \in Q$ and $|Q| = |\mathbb{P}| - (N - s(\sigma))$ such that $\sigma = WIN_k[s(\sigma), Q]$. Instead of directly proving this, we prove the contrapositive statement, i.e., $\forall Q \subseteq \mathbb{P}$ with $p(\sigma) \in Q$ and $|Q| = |\mathbb{P}| - (N - s(\sigma))$, $\sigma \neq WIN_k[s(\sigma), Q] \Rightarrow \sigma \in DEAD_k$. Consider an allocation α of the whole auction that contains σ , i.e., $s(\alpha) \leq N$ and $\sigma \in \alpha$. Let $\beta = \alpha \setminus \{\sigma\}$, thus, we know that $s(\beta) \leq N - s(\sigma)$. Because each bidder in an allocation has to bid on at least 1 item, we have $|p(\beta)| \leq N - s(\sigma)$, or $|\mathbb{P}| - |p(\beta)| \geq |\mathbb{P}| - (N - s(\sigma))$. Consider $WIN_k[s(\sigma), \mathbb{P} \setminus p(\beta)]$. Because $\beta = \alpha \setminus \{\sigma\}$, we know $p(\sigma) \in \mathbb{P} \setminus p(\beta)$. Therefore, there exists $Q \subseteq \mathbb{P} \setminus p(\beta)$ that satisfies $|Q| = |\mathbb{P}| - (N - s(\sigma))$ and $p(\sigma) \in Q$. We either have $WIN_k[s(\sigma), Q] < WIN_k[s(\sigma), \mathbb{P} \setminus p(\beta)]$, or $WIN_k[s(\sigma), Q] = WIN_k[s(\sigma), \mathbb{P} \setminus p(\beta)]$. Furthermore, for any bidder set Q such that $|Q| = |\mathbb{P}| - (N - x)$ and $p(\sigma) \in Q$, $\{\sigma\}$ itself is a feasible allocation for sub-auction $[s(\sigma), Q]$. Thus, the fact that $\sigma \neq WIN_k[s(\sigma), Q]$ implies $\{\sigma\} < WIN_k[s(\sigma), Q]$. Therefore, we always have $\{\sigma\} < WIN_k[s(\sigma), \mathbb{P} \setminus p(\beta)]$. As a result, $\alpha = \beta \cup \{\sigma\} < \beta \cup WIN_k[s(\sigma), \mathbb{P} \setminus p(\beta)]$, and therefore $\sigma \in DEAD_k$. ■

Corollary 8a. For a SIMU-XOR auction, given auction state k , the deadness level at span x for bidder p^* is calculated as follows:

1. If $x \leq N - |\mathbb{P}|$, then: $DL_k(x, p^*) = REV_k(x, p^*)$;
2. If $x > N - |\mathbb{P}|$, then: $DL_k(x, p^*) = \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q| = |\mathbb{P}| - (N - x)} REV_k(x, Q)$

Proof. Consider a new atomic bid, σ , to be submitted at state $k + 1$. Let $s(\sigma) = x$ and $p(\sigma) = p^*$. If $x \leq N - |\mathbb{P}|$, then based on Theorem 8, $\sigma \in LIVE_{k+1} \Leftrightarrow \sigma = WIN_{k+1}[x, p^*] \Leftrightarrow WIN_k[x, p^*] < \{\sigma\} \Leftrightarrow v(\sigma) > REV_k(x, p^*) \Leftrightarrow DL_k(x, p^*) = REV_k(x, p^*)$.

If instead $x > N - |\mathbb{P}|$, then based on Theorem 7, $\sigma \in LIVE_{k+1} \Rightarrow \exists P \subseteq \mathbb{P}$ with $p^* \in P$ and $|P| = |\mathbb{P}| - (N - x)$ such that $\sigma = WIN_{k+1}[x, P]$. Because $WIN_k[x, P]$ precedes $WIN_{k+1}[x, P]$ in time, it follows that $v(\sigma) = REV_{k+1}(x, P) > REV_k(x, P) \geq \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q| = |\mathbb{P}| - (N - x)} REV_k(x, Q)$. Therefore, we know that if σ is live, then $v(\sigma) > \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q| = |\mathbb{P}| - (N - x)} REV_k(x, Q)$. Conversely, if $v(\sigma) > \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q| = |\mathbb{P}| - (N - x)} REV_k(x, Q)$, suppose

$\min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q|=|\mathbb{P}|-(N-x)} REV_k(x, Q) = REV_k(x, P)$ where $p^* \in P$ and $|P| = |\mathbb{P}| - (N - x)$, then $WIN_k[x, P] < \{\sigma\}$. One can construct a “complementary” future bid combination β such that $s(\beta) = N - x$, $p(\beta) = \mathbb{P} \setminus P$. Then, $\beta \cup WIN_k[x, P] < \beta \cup \{\sigma\}$, and therefore σ is live. Together, it follows that $\sigma \in LIVE_{k+1} \Leftrightarrow v(\sigma) > \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q|=|\mathbb{P}|-(N-x)} REV_k(x, Q) \Leftrightarrow DL_k(x, p^*) = \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q|=|\mathbb{P}|-(N-x)} REV_k(x, Q)$. ■

Theorem 9. In a SIMU-XOR auction, for any auction state k , the following statements are true:

1. $DL_k(x, p) \leq REV_k(x, \mathbb{P})$
2. $DL_k(x, p) \leq WL_k(x, p)$
3. $\forall p \in \mathbb{P}, DL_k(N, p) = WL_k(N, p) = REV_k(N, \mathbb{P})$
4. $DL_k(x, p) \leq DL_{k+1}(x, p)$
5. $x \leq y \Rightarrow WL_k(x, p) \leq WL_k(y, p)$ and $DL_k(x, p) \leq DL_k(y, p)$
6. $\sigma \in WIN_k \Rightarrow WL_k(s(\sigma), p(\sigma)) = v(\sigma)$
7. $\sigma \in LIVE_k \Rightarrow DL_k(s(\sigma), p(\sigma)) = v(\sigma)$

Proof. Statements 1-5 follow immediately from the definitions of $DL_k(x, p)$, $WL_k(x, p)$, and $REV_k(x, p)$.

For statement 6, denote $WIN_k = \{\sigma\} \cup \beta$ where $s(\beta) \leq N - s(\sigma)$ and $p(\sigma) \notin p(\beta)$. By definition, $v(\sigma) + v(\beta) = REV_k(N, \mathbb{P}) = WL_k(s(\sigma), p(\sigma)) + REV_k(N - s(\sigma), \mathbb{P} \setminus \{p(\sigma)\})$. If $v(\beta) < REV_k(N - s(\sigma), \mathbb{P} \setminus \{p(\sigma)\})$, there must be another allocation β' with $s(\beta') \leq N - s(\sigma)$ and $p(\sigma) \notin p(\beta')$ that has $v(\beta') > v(\beta)$. As a result, $WIN_k < \{\sigma\} \cup \beta'$, contradiction. Therefore, $v(\beta) = REV_k(N - s(\sigma), \mathbb{P} \setminus \{p(\sigma)\})$ and $v(\sigma) = WL_k(s(\sigma), p(\sigma))$.

For statement 7, based on Theorem 8, if $s(\sigma) \leq N - |\mathbb{P}|$, then $\sigma \in LIVE_k \Rightarrow \sigma = WIN_k[s(\sigma), p(\sigma)] \Rightarrow v(\sigma) = REV_k(s(\sigma), p(\sigma))$. Meanwhile, Corollary 8a states that in this case, $DL_k(s(\sigma), p(\sigma)) = REV_k(s(\sigma), p(\sigma))$. Therefore, $DL_k(s(\sigma), p(\sigma)) = v(\sigma)$. Instead, if $s(\sigma) > N - |\mathbb{P}|$, based on Theorem 8, $\sigma \in LIVE_k \Rightarrow \sigma = WIN_k[s(\sigma), P]$ where $p(\sigma) \in P \subseteq \mathbb{P}$ and $|P| = |\mathbb{P}| - (N - s(\sigma))$, which implies $v(\sigma) = REV_k(s(\sigma), P)$. Meanwhile, Corollary 8a states that in this case, $DL_k(s(\sigma), p(\sigma)) = \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q|=|\mathbb{P}|-(N-s(\sigma))} REV_k(s(\sigma), Q)$. Since $p(\sigma) \in Q$, we know that $REV_k(s(\sigma), Q) \geq v(\sigma)$. Overall, $v(\sigma) = REV_k(s(\sigma), P) = \min_{\forall Q \subseteq \mathbb{P}, p^* \in Q, |Q|=|\mathbb{P}|-(N-s(\sigma))} REV_k(s(\sigma), Q) = DL_k(s(\sigma), p(\sigma))$. ■

Theorem 10. For a SIMU-XOR auction, given auction state k , there can be no more than $\min(N - x + 1, |\mathbb{P}|)$ live atomic bids with span x .

Proof. At auction state k , let the atomic bids of span x of each bidder be $\sigma_1, \sigma_2, \dots, \sigma_{|\mathbb{P}|}$. Without loss of generality, assume that $\sigma_{|\mathbb{P}|} < \sigma_{|\mathbb{P}|-1} < \dots < \sigma_1$. Suppose $|\mathbb{P}| \geq N - x + 1$, consider the atomic bid σ_{N-x+1} , i.e., the $(N - x + 1)^{\text{th}}$ highest atomic bid at that span x . There are precisely $|\mathbb{P}| - (N - x) - 1$ atomic bids that are inferior to σ_{N-x+1} , namely $\sigma_{N-x+2}, \dots, \sigma_{|\mathbb{P}|}$. Therefore, for any bidder set Q such that $|Q| = |\mathbb{P}| - (N - x)$, it must contain *at least* one bidder who placed the atomic bid σ_t where $1 \leq t \leq N - x + 1$. Based on the assumed order, $\forall s$ such that $N - x + 2 \leq s \leq |\mathbb{P}|$, we have $\sigma_s < \sigma_t$. Because $\{\sigma_t\}$ itself is a feasible allocation for sub-auction $[x, Q]$, we know that either $\sigma_t = WIN_k[x, Q]$ or $\sigma_t < WIN_k[x, Q]$ is true. However, in either case, we always have $\sigma_s < WIN_k[x, Q]$. Because the choice of bidder set Q is arbitrary, it follows that $\sigma_s \in DEAD_k$. I.e., only atomic bids among $\sigma_1, \sigma_2, \dots, \sigma_{N-x+1}$ are possible to be live at auction state k . Suppose instead $|\mathbb{P}| < N - x + 1$, following the same logic above, we can see that all $|\mathbb{P}|$ atomic bids at span x can potentially be live. Overall, there cannot be more than $\min(N - x + 1, |\mathbb{P}|)$ live atomic bids at span x . ■

Corollary 10. For a SIMU-XOR auction, given auction state k , the maximum possible number of live atomic bids is $\sum_{x=1}^N \min(x, |\mathbb{P}|)$. I.e., $|LIVE_k| \leq \sum_{x=1}^N \min(x, |\mathbb{P}|)$

Proof. Based on Theorem 10, by summing the live bids across all spans we have that $|LIVE_k| \leq \sum_{x=1}^N \min(N - x + 1, |\mathbb{P}|) = \sum_{x=1}^N \min(x, |\mathbb{P}|)$. ■

Appendix C

Additional Theoretical Results

Here we list two additional theoretical results for SIMU auctions, which are not discussed in the main paper but may provide further insights to understand SIMU auctions.

First, in SIMU-OR auctions, even by very quick, naïve calculations, one can show that can never have more than $O(N \cdot \log N)$ live bids at a time, as explained below.

In SIMU-OR auctions, for any auction state k , $|LIVE_k| = O(N \cdot \log N)$.

Proof. The proof of the lemma follows immediately from the fact that in an auction of size N , there can be no more than N live 1-item bids (i.e., bids with span 1), no more than $N/2$ live 2-item bids, or, more generally, no more than N/x live x -item bids, where $x = 1, 2, \dots, N$. And from elementary mathematical analysis we have that $N + \frac{N}{2} + \dots + \frac{N}{N} = N \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{N}\right) = O(N \cdot \log N)$. ■

Note that the above result could be written as a non-asymptotic upper bound as well, i.e., without using $O(\cdot)$ notation. In such case, the naïve upper bound could be stated as: $|LIVE_k| \leq \sum_{x=1}^N \lfloor N/x \rfloor$. The analogous result about the number of winning bids is obvious: $|WIN_k| \leq N$, i.e., there can be at most N winning bids in an auction of size N , because each bid has to bid on at least one item.

Second, in a SIMU-XOR auction, if a bidder submits multiple atomic bids simultaneously, each of which is above the current winning level at its span, then only one of those atomic bids can be winning (due to the XOR constraint), and the winner will be the atomic bid with largest margin over its current winning level. If several atomic bids all have the largest margin, the one with smallest span will win (due to fairness-based tie-breaking). This is summarized as follows.

For a SIMU-XOR auction, given auction state k and a new general bid $b_{k+1} = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$, where $\forall x \in \{1, 2, \dots, N\}$, $p(\sigma_x) = p^*$, $s(\sigma_x) = x$, and $v(\sigma_x) > WL_k(x, p^*)$. Let s^* be the **smallest** span s that satisfies $s \in \operatorname{argmax}_{x \in \{1, 2, \dots, N\}} (v(\sigma_x) - WL_k(x, p^*))$. Then $\sigma_{s^*} \in WIN_{k+1}$.

Proof. For atomic bid $\sigma_x \in b_{k+1}$, if it were to win the auction, the resulting winning allocation would be $\{\sigma_x\} \cup WIN_k(N - x, \mathbb{P} \setminus \{p^*\})$, which has the revenue of $v(\sigma_x) + REV_k(N - x, \mathbb{P} \setminus \{p^*\})$. Because $v(\sigma_x) + REV_k(N - x, \mathbb{P} \setminus \{p^*\}) = WL_k(x, p^*) + REV_k(N - x, \mathbb{P} \setminus \{p^*\}) + v(\sigma_x) - WL_k(x, p^*) = REV_k(N, \mathbb{P}) + v(\sigma_x) - WL_k(x, p^*)$, the maximum revenue is achieved by span s that satisfies $s \in \operatorname{argmax}_{x \in \{1, 2, \dots, N\}} (v(\sigma_x) - WL_k(x, p^*))$. The conclusion follows based on our tie-breaking mechanism. Furthermore, *only* σ_{s^*} in b_{k+1} will win in state $k + 1$ because of XOR bidding. ■

Appendix D

Benefits of a Real-Time Bidder Support System

In our paper, the difference between a real-time bidder support system (one that provides up-to-date information feedback after each bid) versus a non-real-time system (one that provides information feedback only at certain pre-specified intervals, i.e., which inevitably results in providing some outdated information) can have at least two implications for performance.

First, the two systems differ in their *usability/feasibility*. Because we consider continuous combinatorial auctions, where no formal “rounds of bidding” are imposed and the participations are completely asynchronous, it is critical to have a *real-time* bidder support system, simply because the auctions would be infeasible to conduct otherwise. Imagine an auction where bidders are free to join and leave as they wish, but have to wait (e.g., minutes or even hours) to obtain information feedback and construct their bids, such a mechanism will have staggeringly *low usability* and may not be adopted at all.

Second, the two systems can also result in different auction convergence outcomes and bidder experiences. While bidders can receive up-to-date information from a real-time system, they may receive obsolete information from a non-real-time system that provides feedback with some (potentially significant) delay, during which multiple bids could have been submitted. We illustrate this point using a numeric example and a set of stylized simulation examples. In our illustration, we assume bidders are relatively conservative and use deadness levels as bidding guidelines, and we show how a non-real-time system can result in slower auction convergence.

Numeric Example. Consider a SIMU-OR auction of 4 units with the following 4 OR bids already submitted:
 $b_1 = (1, \$1)$, $b_2 = (1, \$1)$, $b_3 = (1, \$1)$, $b_4 = (2, \$4)$

At auction state 4 (i.e., after these 4 bids), the sub-auction revenues and deadness levels are as follows:

$REV_4(1) = \$1$, $REV_4(2) = \$4$, $REV_4(3) = \$5$, $REV_4(4) = \$6$
 $DL_4(1) = \$1$, $DL_4(2) = \$2$, $DL_4(3) = \$5$, $DL_4(4) = \$6$

Now suppose a bidder wants to bid on 2 units and makes a request to see the deadness level for 2 units. Under a real-time bidder support system, the bidder sees the timely and correct deadness level of \$2 for 2 units, and can make informed bidding decisions accordingly. However, under a non-real-time bidder support system, it takes a longer time to update feedback metrics. For instance, suppose 2 bids are submitted during the time it takes to update feedback metrics, i.e., at auction state 4, the bidder still only has access to the deadness level calculated based on bids at auction state 2 (i.e., based on b_1 and b_2 , deadness level for 2 units was \$0), which is already obsolete. Consequently, the bidder may place \$1 on 2 units, which appears to be live based on (obsolete) information calculated at state 2, but in fact is already dead based on (timely and correct) information at state 4. Therefore, having a non-real-time bidder support system that provides obsolete feedback information can lead bidders to make incorrect decisions (e.g., submitting bids that are already dead), which may slow down the convergence of the auction.

Simulation Experiments. Consider a SIMU-OR auction of 100 units and a set of bidders. We simulate both a real-time bidder support system and a non-real-time system as follows:

1. Real-time system: at auction state k , a bidder makes a bid by following two steps: (1) randomly pick span $s \in \{1, \dots, 100\}$; (2) assign bid value $v = DL_k(s) + 1$.
2. Non-real-time system: assuming the system takes a longer time to update feedback metrics, during which t new bids are submitted. At auction state k , a bidder makes a bid by following two steps: (1) randomly pick span $s \in \{1, \dots, 100\}$; (2) assign bid value $v = DL_{k-r-t}(s) + 1$, where $r = (k \bmod t)$ and $k - r - t$ represents the most recent state when deadness levels are updated. Also, for auction states $k < t$, bidders only have access to deadness levels at state 0.

In other words, under a real-time system, new bids have values higher than current deadness levels, and are live by construction. Under a non-real-time system, new bids have values higher than most recently updated deadness levels, which could be outdated. For simulation simplicity and illustrative purposes, we set a fixed auction revenue

level of \$1000 and count the number of bids it takes to reach that level, as a measure of the speed of auction progression. We conduct 500 simulation runs and report two quantities: (1) the average number of bids submitted before auction revenue hits \$1000 and (2) the average percentage of bids that are already dead upon submission. In other words, the first quantity represents the speed of auction progression, and the second quantity reflects the amount of “wasted effort” in the auction. We report the results in the following table.

Update Speed	Average Number of Bids before Revenue Hits \$1000	Average Percentage of Dead Bids upon Submission
Real-time	3586.70	0%
$t = 1$	3976.21	13%
$t = 5$	6324.71	52%
$t = 10$	8559.97	68%
$t = 15$	10347.07	74%

Based on the results, as t grows larger (i.e., as the system becomes less real-time), we can see that it takes more bids to reach \$1000 auction revenue, and a higher percentage of those bids are already dead upon submission, because of the obsolete deadness level information.

In summary, compared to a real-time bidder support system, a non-real-time system may result in significantly slower auction progression and substantial “wasted effort”, and hence slower auction convergence as well as potentially lower bidder satisfaction.

Appendix E

Comparison Between OR Versus XOR Bidding Language

Bidder's Perspective

From the bidder's perspective, the OR and XOR bidding languages differ on at least two aspects: *expressiveness* and *simplicity* (Nisan 2000).

Expressiveness: the XOR bidding language is strictly more expressive than the OR bidding language, i.e., any valuations expressed in OR bids can be expressed in XOR bids, and XOR language can also directly express substitutability among bids. For example, consider a SIMU auction of 10 units, we provide two canonical cases to illustrate this aspect.

Example E1: Suppose a bidder is willing to pay \$6 for 4 units or \$7 for 5 units, but has \$0 valuation for fewer than 4 units and \$7 valuation for more than 5 units. Under XOR language, the bidder can directly express the preference on 4 or 5 units by making 2 XOR bids, respectively \$6 on 4 units and \$7 on 5 units. Under OR language, the bidder *cannot* express this preference perfectly. By placing \$6 on 4 units and \$7 on 5 units, the bidder may end up having to pay \$13 for 9 units, despite her \$7 valuation for 9 units. Alternatively, by placing \$6 on 4 units and \$1 on 1 unit, the bidder may end up having to pay \$1 for 1 unit, despite her \$0 valuation for 1 unit. Either way, the bidder is exposed to the risk of disutility.

Example E2: Suppose a bidder has non-zero valuations for any number of units, and her valuations are increasing with diminishing margins with respect the number of units. I.e., let v_i represent her valuation for i units, then $\forall i$, $v_i > 0$ and $v_{i+1} - v_i < v_i - v_{i-1}$. Under XOR language, the bidder can directly express this preference by making 10 XOR bids, respectively v_i on i units. Under OR language, the bidder can express this preference, but must express it differently. In particular, she needs to make 10 OR bids, each on only 1 unit, with values $v_1, (v_2 - v_1), \dots, (v_{10} - v_9)$. Suppose the bidder (mistakenly) places v_1 on 1 unit and v_2 on 2 units, she may end up having to pay $v_1 + v_2$ for 3 units, which is higher than her valuation for 3 units (v_3), because of the concavity of valuations.

Simplicity: the XOR language is less simple than OR language (i.e., it may take fewer OR bid elements to express certain preference than XOR). To see this, consider a simple SIMU auction of 3 units. If a bidder is willing to pay \$5 for 1 unit, \$10 for 2 units, and \$15 for 3 units, he/she only needs to make two OR bids (i.e., a \$5 bid on 1 unit and a \$10 bid on 2 units), but would have to make three XOR bids (i.e., a \$5 bid on 1 unit, a \$10 bid on 2 units, and a \$15 bid on 3 units). Given these two differences, the bidding language used in an auction should be sufficiently expressive for its specific application needs and also simple for bidders to use.

Auctioneer's Perspective

Assuming the auctioneer's goal is to maximize auction revenue, which of the two bidding languages leads to higher revenue depends heavily on many different factors, including, at the very least: (1) bidders' valuations; (2) bidders' bidding behaviors (i.e., how they choose to express their preferences under different bidding languages); (3) auction progression (i.e., how bidders revise their bids during the course of an auction); and (4) other auction-specific activity rules. Below we construct 3 numeric examples to show, under some *specific assumptions about bidder valuations and bidding behaviors*, the OR language may result in higher or equal revenue than the XOR language. In these examples, we explicitly fix bidders' valuations, and assume they make bids to express their valuations in a one-shot manner (i.e., without iteratively revising their bids later). These stylized examples are intended to illustrate only a small portion of the potential intricacies of this problem.

Example E3: OR leads to higher revenue than XOR.

Consider a SIMU auction of 4 units and 2 bidders, A and B, with the following valuations:

- Bidder A is willing to pay \$5 for 2 units, \$8 for 3 units;
- Bidder B is willing to pay \$6 for 3 units or \$9 for 4 units.

Under OR bidding language, suppose bidders choose to make the following bids:

- Bidder A makes two OR bids: (\$3 on 1 unit) OR (\$5 on 2 units);
- Bidder B makes two OR bids: (\$3 on 1 unit) OR (\$6 on 3 units).

As the result, the auction revenue is \$11, by allocating 3 units to bidder A for price \$3+\$5 and allocating 1 unit to bidder B for price \$3.

On the other hand, under XOR language, suppose bidders choose to make the following bids:

- Bidder A makes two XOR bids: (\$5 on 2 units) XOR (\$8 on 3 units);
- Bidder B makes two XOR bids: (\$6 on 3 units) XOR (\$9 on 4 units).

As the result, the auction revenue is \$9, by allocating all 4 units to bidder B. In other words, the OR bidding language results in \$2 higher revenue than XOR language.

Example E4: OR leads to equal revenue as XOR

Consider the same auction in the previous example. Under OR bidding language, suppose bidders choose to make the following bids:

- Bidder A makes two OR bids: (\$5 on 2 units) OR (\$8 on 3 units);
- Bidder B makes two OR bids: (\$6 on 3 units) OR (\$9 on 4 units).

Since the combination of any two bids contains more than 4 units, which cannot be fulfilled, the auction revenue is \$9, by allocating all 4 units to bidder B. This is the same auction revenue under the XOR language.

Example E5: OR leads to equal revenue as XOR

Consider a SIMU auction of 3 units and 2 bidders, A and B. Each bidder is willing to pay \$3 for 1 unit, \$5 for 2 units, or \$6 for 3 units (i.e., both bidders have decreasing marginal valuations).

Under OR bidding language, suppose bidders choose to make the following bids:

- Bidder A makes three OR bids: (\$3 on 1 unit) OR (\$2 on 1 unit) OR (\$1 on 1 unit);
- Bidder B makes three OR bids: (\$3 on 1 unit) OR (\$2 on 1 unit) OR (\$1 on 1 unit).

As the result, the auction revenue is \$8, by allocating 2 units to bidder A for price \$3+\$2 (assuming bidder A bids first) and allocating 1 unit to bidder B for price \$3.

On the other hand, under XOR language, suppose bidders choose to make the following bids:

- Bidder A makes three XOR bids: (\$3 on 1 unit) XOR (\$5 on 2 units) XOR (\$6 on 3 units);
- Bidder B makes three XOR bids: (\$3 on 1 unit) XOR (\$5 on 2 units) XOR (\$6 on 3 units).

As the result, the auction revenue is also \$8, by allocating all 2 units to bidder A for price \$5 (assuming bidder A bids first) and allocating 1 unit to bidder B for price \$3.

Example E3 represents the scenario where bidders cannot express their valuations perfectly under OR language. Specifically, by making 1-unit bids, they are exposed to the risk of winning 1 unit, even if they may not want it. However, they can directly and perfectly express their valuations under XOR language. As such, the auction revenue under OR is higher than under XOR. Example E4 and E5 represent two special cases where bidders can perfectly express their valuations under both OR and XOR languages, and therefore result in the same auction revenue.

In addition to the above numerical examples, we also provide theoretical analyses of a stylized case, where OR language may lead to higher or equal auction revenue as XOR language.

Consider a SIMU auction of N units and 2 bidders, A and B, with the following valuations:

- Bidder A is willing to pay $\$a_x$ for x units or $\$a_y$ for y units;
- Bidder B is willing to pay $\$b_x$ for x units or $\$b_y$ for y units.

Furthermore, assume that $x < y < x + y \leq N < 2y$ and $2y - x \leq N$. Also assume that $a_x < a_y$, $b_x < b_y$, i.e., the bid value is higher for more units.

Under XOR language, bidders can perfectly express their preferences by making the following bids:

- Bidder A makes two XOR bids: ($\$a_x$ on x units) XOR ($\$a_y$ on y units);
- Bidder B makes two XOR bids: ($\$b_x$ for x units) XOR ($\$b_y$ for y units).

As a result, the auction revenue is $\max\{a_x + b_y, a_y + b_x\}$.

On the other hand, under OR bidding language, due to our assumptions on x and y , bidders cannot perfectly express their preferences without being exposed to the risk of acquiring allocations they don't value. Here, we consider four different situations:

Situation 1:

- Bidder A makes two OR bids: ($\$a_x$ on x units) OR ($\$a_y$ on y units).
- Bidder B makes two OR bids: ($\$b_x$ for x units) OR ($\$b_y$ for y units).

The resulting auction revenue is $\max\{a_x + a_y, a_x + b_y, a_y + b_x, b_x + b_y\}$.

Situation 2:

- Bidder A makes two OR bids: ($\$a_x$ on x units) OR ($\$(a_y - a_x)$ on $y - x$ units).
- Bidder B makes two OR bids: ($\$b_x$ for x units) OR ($\$(b_y - b_x)$ for $y - x$ units).

The resulting auction revenue is $\max\{a_x + b_x + (b_y - b_x), b_x + a_x + (a_y - a_x)\} = \max\{a_x + b_y, a_y + b_x\}$.

Situation 3:

- Bidder A makes two OR bids: ($\$a_x$ on x units) OR ($\$(a_y - a_x)$ on $y - x$ units).
- Bidder B makes two OR bids: ($\$b_x$ for x units) OR ($\$b_y$ for y units).

The resulting auction revenue is $\max\{a_x + b_y, b_x + a_x + (a_y - a_x), (a_y - a_x) + b_y\} = \max\{a_x + b_y, a_y + b_x, (a_y - a_x) + b_y\}$ (revenue of $(a_y - a_x) + b_y$ is possible because we have assumed $2y - x \leq N$).

Situation 4:

- Bidder A makes two OR bids: ($\$a_x$ on x units) OR ($\$a_y$ on y units).
- Bidder B makes two OR bids: ($\$b_x$ for x units) OR ($\$(b_y - b_x)$ for $y - x$ units).

The resulting auction revenue is $\max\{a_y + b_x, a_x + b_x + (b_y - b_x), (b_y - b_x) + a_y\} = \max\{a_y + b_x, a_x + b_y, (b_y - b_x) + a_y\}$ (revenue of $(b_y - b_x) + a_y$ is possible because we have assumed $2y - x \leq N$).

Overall, in situation 2, OR results in the same revenue as XOR; in situations 1, 3, and 4, OR results in no less revenue than XOR (i.e., OR may result in higher or equal revenue as XOR, depending on the specific values of a_x, a_y, b_x, b_y).

Generalizing from these cases, we argue that, assuming bidders make truthful bids based on their valuations in a one-shot manner, (1) OR language results in *no less* auction revenue than XOR *if bidders cannot perfectly express their preferences under OR*; and (2) OR language results in *equal* auction revenue than XOR *if bidders can perfectly express their preferences under OR*.

Importantly, in addition to bidding language choice, auction revenue depends on bidders' specific valuations and how they choose to express their valuations, including bidders' strategies for updating their bids continuously over time (i.e., in non-one-shot contexts). We would also like to emphasize that the choice of OR versus XOR has other implications in addition to revenue. For instance, in the above Example E3, bidder B is allocated 1 unit, which may be deemed undesirable by the bidder (the bidder only has non-zero valuations for 3 or 4 units). This can potentially lead to lower bidder satisfaction and discourage participation. Therefore, the auctioneer needs to consider multiple aspects (not just revenue) when choosing bidding language.

Copyright of MIS Quarterly is the property of MIS Quarterly and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.