

Visualizing combinatorial auctions

Joe Ping-Lin Hsiao · Christopher G. Healey

Published online: 19 April 2011
© Springer-Verlag 2011

Abstract We propose a novel scheme to visualize combinatorial auctions; auctions that involve the simultaneous sale of multiple items. Buyers bid on complementary sets of items, or bundles, where the utility of securing all the items in the bundle is more than the sum of the utility of the individual items. Our visualizations use concentric rings divided into arcs to visualize the bundles in an auction. The arcs' positions and overlaps allow viewers to identify and follow bidding strategies. Properties of color, texture, and motion are used to represent different attributes of the auction, including active bundles, prices bid for each bundle, winning bids, and bidders' interests. Keyframe animations are used to show changes in an auction over time. We demonstrate our visualization technique on a standard testbed dataset generated by researchers to evaluate combinatorial auction bid strategies, and on recent Federal Communications Commission (FCC) auctions designed to allocate wireless spectrum licenses to cell phone service providers.

Keywords Combinatorial auction · Ecommerce · Perception · Visualization

1 Introduction

Combinatorial auctions are auctions where multiple items are sold simultaneously. Buyers place bids on bundles of items rather than bidding separately on individual items.

This ability to bid on bundles makes combinatorial optimization and mathematical programming relevant for solving the distribution of items that maximizes all buyers' utilities. Thus, combinatorial auctions are studied in economics, operations research, and computer science [10, 12, 17].

Combinatorial auctions allow buyers to fully express their preferences, particularly when items are complementary. A complement is a set of items $S = \{i_1, \dots, i_n\}$ that has a greater utility than the sum of the utility for each individual item, $u(S) > \sum_{j=1}^n u(i_j)$. For example, a pair of shoes is more valuable than a left shoe or a right shoe alone. The value of an individual item $i_j \in S$ may differ between bidders, particularly if a bidder already possesses other items in S .

Combinatorial auctions have been employed in a variety of industries: transportation routing, airport arrival and departure scheduling, and allocating radio spectrums for communications services [3, 11]. In each case, the motivation for using a combinatorial auction is the complementary nature of the desired resources. For example, a trucker's cost for handling shipping to a new destination depends on his original routes. If the new destination is near his existing path, the cost for shipping goods there will be relatively lower than shipping to a location not near the path.

To date, combinatorial auction results are normally presented as data tables showing active bundles, bid prices, bidders' demand sets, competitive allocations of items to winners, and so on. Table 1 shows two rounds of an auction, and the final bundle prices, involving four bidders and three items. The table is effective for showing quantitative results of the auction. However, it can be difficult and time consuming to gain an overall understanding of the relationships between different bidders, their interests, and their bid strategies, particularly as these properties change during the lifetime of the auction. Our goal for this project is to provide a

J.P.-L. Hsiao · C.G. Healey (✉)
Department of Computer Science, North Carolina State
University, Raleigh, NC, USA
e-mail: healey@csc.ncsu.edu

visualization that complements the data table. Specifically, we want to:

1. Produce a concise spatial layout to increase the amount of information contained in our visualizations.
2. Visualize bundle prices, competitive allocations, and bidders' interests using colors and textures selected using guidelines from visual perception.
3. Visualize changes in an auction over time using motion, again selected based on guidelines from human perception.

Researchers generally analyze results in two ways: (1) as a post-processing operation after an auction completes; or (2) as an auction runs. Our technique is designed to support both needs, allowing viewers to track an auction's state in real-time as it unfolds, or to visualize auction logs to see how an auction progressed. Our expectation is that the visualizations will be used to quickly explore an auction to identify interesting bundle prices, bid strategies, allocations of items, changes in a bidder's interests, and so on. These events can then be examined in more detail by studying related values in the data table.

2 Iterative proxy combinatorial auctions

Our first testbed dataset involves a type of combinatorial auction known as an iterative proxy combinatorial (IPC) auction [16]. The auction iterates in rounds, with proxy agents acting in place of the actual buyers. Each buyer submits to his agent the amounts that he is willing to pay for each bundle before the auction begins. Agents bid on behalf of their buyers based on these amounts. An agent's profit for each bundle is the difference between the price the buyer pays and the price the agent pays to win the bundle. Since agents will never bid above the buyer's price, profit for any bundle won is always greater than or equal to zero. All agents act to try to increase their profit.

At the start of each round, agents submit offers for bundles where they think they can maximize their profit. The auctioneer receives the bids and solves the *winner determination problem* (WDP) to assign items to agents. The auctioneer announces the winners and winning prices to end the round. This process continues until the auction ends. The iterative nature of the auction allows agents to offer provisional prices, then adjust their allocations based on other agents' bids.

Each agent maintains a demand set of bundles that will maximize the agent's profit, based on the bundles' current prices. As the auction progresses and bundle prices change, an agent's demand set will also change. At the beginning of a round, an agent may increase the bid on a preferred bundle or pass. If an agent wins a bundle, or if the agent cannot

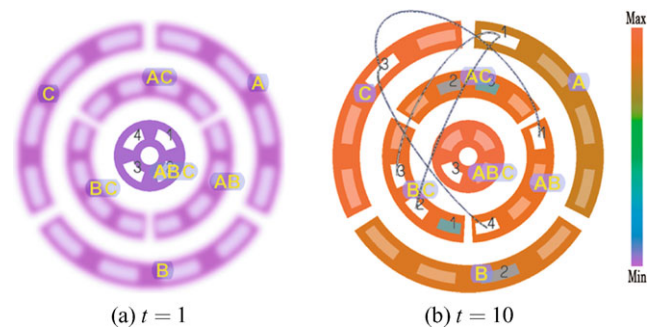


Fig. 1 The same two steps from Table 1, displayed using our visualization strategy

generate profit from the bundle, he will pass for the next round.

To simplify the auction, certain rules are applied to our testbed. An agent may have multiple bundles in his demand set, but he can increase his bid on only one bundle per round. The bid increment adds a fixed amount to the preferred bundle's current price. Given these constraints, during each round an agent can: (1) continue bidding on the most recently bid-on bundle; (2) start bidding on a different bundle in the demand set; (3) add a new bundle to the demand set and bid on it; or (4) pass for the round.

After receiving a round's bids, the auctioneer assigns bundles to winning agents to produce a *competitive allocation* that: (1) maximizes the sum of the bid prices—the revenue—that the auctioneer receives; and (2) assigns each item to at most one agent. If multiple allocations with the same revenue exist, the auctioneer chooses one at random.

Set notation is used to represent competitive allocations. Positions in the set correspond to an ordered list of agents. Bundles appearing at position i imply that agent i has won the bundle. For example, assume four agents are bidding on three items $D = \{A, B, C\}$. In the first round, each agent bids on the bundle containing all three items (Table 1 and Fig. 1a). The possible competitive allocations are therefore $\{ABC, -, -, -\}$, $\{-, ABC-, -\}$, $\{-, -, ABC, -\}$, or $\{-, -, -, ABC\}$. The auctioneer will choose the allocation corresponding to the agent who offers the highest bid price.

The auction itself is represented as a subset of rounds where one of the following *inflection points* occurs: (1) a new bundle enters an agent's demand set; (2) competitive allocations change; or (3) an agent withdraws as an active bidder.

Figure 1 shows the two timesteps in Table 1, displayed using our visualization technique. Bundles of decreasing size are represented as concentric rings, with all the bundles of a given size occupying a common ring. Sharpness represents bidders' interests, color represents the bid price for each bundle, saturation represents winning bids,

Table 1 Two steps and the final prices for four agents {1, 2, 3, 4} and three items {A, B, C} showing current bundle prices, agents' active bundle sets, demand sets D_i , and probabilities of bidding on a bundle

Step 2, $t = 1$	Prices:	0	0	0	0	0	0	0	
	D_i	A	B	AB	C	AC	BC	ABC	pass
{ABC, -, -, -} ₁	ABC							$\frac{3}{4}$	$\frac{1}{2}$
{-, ABC, -, -} ₂	ABC							$\frac{3}{4}$	$\frac{1}{2}$
{-, -, ABC, -} ₃	ABC							$\frac{3}{4}$	$\frac{1}{2}$
{-, -, -, ABC} ₄	ABC							$\frac{3}{4}$	$\frac{1}{2}$
...									
Step 10, $t = 36\frac{5}{6}$	Prices:	8	7.25	16	8.25	16	16.25	24.25	
	D_i	A	B	AB	C	AC	BC	ABC	pass
{A, BC, -, -} ₁	A, AB, AC	0		0		0			1
{A, -, BC, -} ₂	B, BC		$\frac{1}{4}$				$\frac{1}{4}$		$\frac{1}{2}$
{-, -, ABC, -} ₃	C, BC, ABC				$\frac{1}{4}$		0	$\frac{1}{4}$	$\frac{1}{2}$
{-, -, C, AB} ₄	\emptyset								
{AB, -, C, -} ₄									
Step 11, $t = 39\frac{5}{6}$	Prices:	8	8	16	9	16	17	25	
	A	B	AB	C	AC	BC	ABC	pass	

and textured links connect items in a common bundle set.

3 Combinatorial datasets

The set of all possible bundles in a combinatorial auction forms a combinatorial dataset. For D of size n , a combinatorial dataset C is the set of all D_i , $0 \leq i \leq n$, where D_i is the set of all combinations of elements from D of size i . If the original D has n elements, the cardinality of the combinatorial dataset will be $\sum_{k=0}^n \binom{n}{k}$. By the binomial theorem, $\sum_{k=0}^n \binom{n}{k} r^k = (1+r)^n$, so $r = 1$ gives $\sum_{k=0}^n \binom{n}{k} = 2^n$. For example, for $D = \{A, B, C, D\}$ of size $n = 4$, the following D_i with a total of $2^4 = 16$ combinations exist:

$$\begin{aligned}
 D_0 &= \{\emptyset\}, |D_0| = \binom{4}{0} = 1 \\
 D_1 &= \{A, B, C, D\}, |D_1| = \binom{4}{1} = 4 \\
 D_2 &= \{AB, AC, AD, BC, BD, CD\}, |D_2| = \binom{4}{2} = 6 \quad (1) \\
 D_3 &= \{ABC, ABD, ACD, BCD\}, |D_3| = \binom{4}{3} = 4 \\
 D_4 &= \{ABCD\}, |D_4| = \binom{4}{4} = 1
 \end{aligned}$$

The following characteristics present challenges for visualizing a combinatorial dataset.

1. The cardinality of C grows exponentially as n increases, $|C| = \sum_{k=0}^n \binom{n}{k} = 2^n$.

2. The growth of the cardinality produces complex superset-subset relationships between consecutive D_i .
3. The number of subsets $|D_i| = \binom{n}{i}$ of size i is not constant nor even monotonically increasing or decreasing. For $0 \leq i \leq \lfloor \frac{n}{2} \rfloor$, $|D_i|$ is increasing, and for $\lceil \frac{n}{2} \rceil < i \leq n$, $|D_i|$ is decreasing. This makes traditional hierarchical visualization techniques difficult to apply.

3.1 Previous work

Two types of diagrams are commonly used to represent combinatorial datasets: *Hasse diagrams* [5] and *Venn diagrams* [13]. Figure 2a illustrates a Hasse diagram of $D = \{A, B, C, D, E\}$. Subsets of the same size are drawn on the same row, with D_n on top and D_0 on the bottom. Every D_i , $0 \leq i < n$, emits edges to its supersets D_{i+1} to show a “belongs to” relationship between consecutive levels. One disadvantage of Hasse diagrams is that as n increases the number of edges becomes large, making it difficult to understand superset-subset relationships.

Figure 2b shows the Venn diagram for the same D . No edges are drawn; instead, the Venn diagram uses regions to represent subsets and overlaps of regions to represent relationships between subsets. This is a more efficient use of space than a Hasse diagram. Different colors are used to help viewers maintain a visual separation of nodes in different levels. In this example, yellow, red, and purple represent subsets D_1 , D_2 , and D_3 of sizes one, two, and three, respectively. Unfortunately, region sizes in a Venn diagram

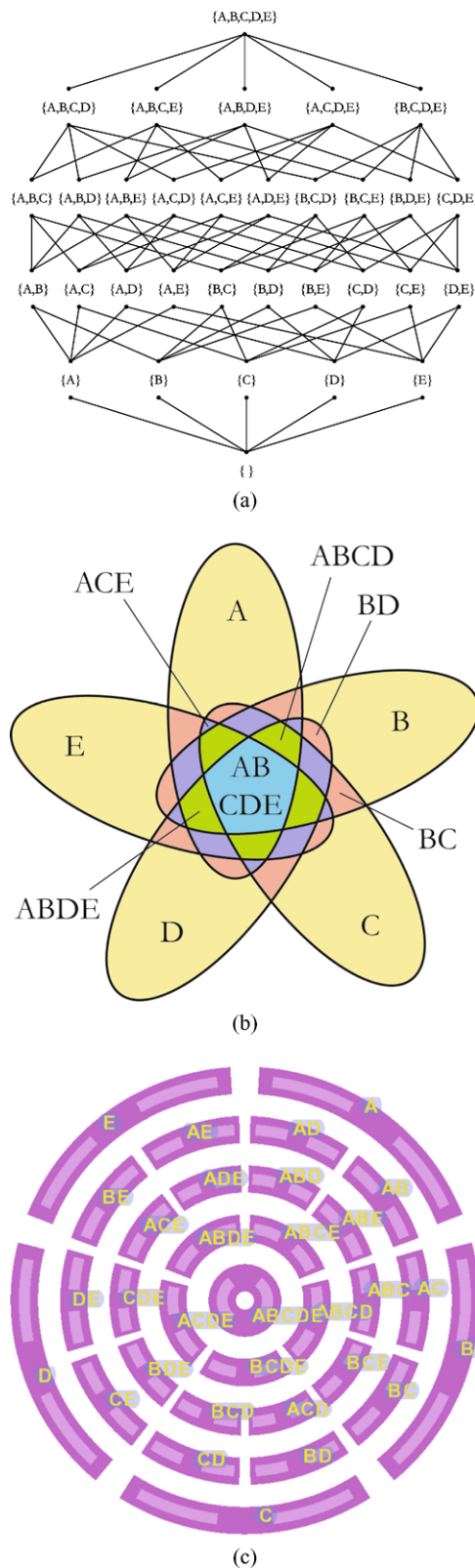


Fig. 2 Combinatorial dataset visualization of the family of subsets of $\{A, B, C, D, E\}$: (a) Hasse diagram; (b) Venn diagram; (c) our concentric ring layout

are nonuniform. The regions in the core and near the edges are larger than the interior regions. Again, as n increases regions will shrink in size, possibly to a point that makes them impractical for certain applications.

4 Concentric ring visualization

We chose to maintain the concentric layout of the Venn diagram to preserve the useful spatial relationships it provides. The foundation of our visualization technique is a collection of equal width concentric rings separated by a fixed distance and divided into multiple arcs.

Figure 2c shows an example layout for the same $D = \{A, B, C, D, E\}$ containing $n = 5$ elements. The five rings in the visualization correspond to combinatorial datasets D_1 – D_5 —all subsets of size one on the outermost ring—through D_5 —all subsets of size five on the innermost ring. Each ring is subdivided into arcs, one for each of the $|D_i| = \binom{n}{i}$ possible combinations of elements. As noted in Eq. 1, for $n = 5$, the number of arcs grows over the range $1 \leq i \leq 3$, and shrinks over the range $3 < i \leq 5$.

Subsets in each D_i are distributed onto individual arcs. The yellow tags on each arc in Fig. 2c name their corresponding subsets (i.e., bundles). The center ring is always a whole circle that maps to the subset D_n containing all of D 's elements.

4.1 Rings and arcs

The visualization's rings and the arcs within each ring serve to group D 's subsets, that is, the bundles of items in an auction in a logical manner. The outermost ring contains subsets of size one, the first interior ring contains subsets of size two, and so on until the largest subset is reached. Therefore, the number of rings equals the number of elements n in D .

4.2 Completely concise forms

We would like to arrange the rings so that arcs with common elements overlap one another. This would allow a viewer, for example, to locate bundle A on the outermost arc, then follow a straight path toward the center of the rings, viewing all the bundles that contain A as one of their items. This would allow viewers to quickly determine how different items are being allocated: either independently, or as part of some other bundle.

Each of the n arcs in the outermost ring defines a pie-shaped region, or sector, that connects the arc's endpoints to the center of the circle (Fig. 3). To obtain the overlap we want, we must arrange the arcs so that the sector representing element i_j intersects every inner arc whose superset contains i_j . We call a layout that achieves this for every $i_j \in D$ a completely concise form.

Fig. 3 Ring and arc layouts:
 (a) a completely concise form for subsets of $D = \{A, B, C\}$;
 (b) an incomplete form for subsets of $D = \{A, B, C, D\}$

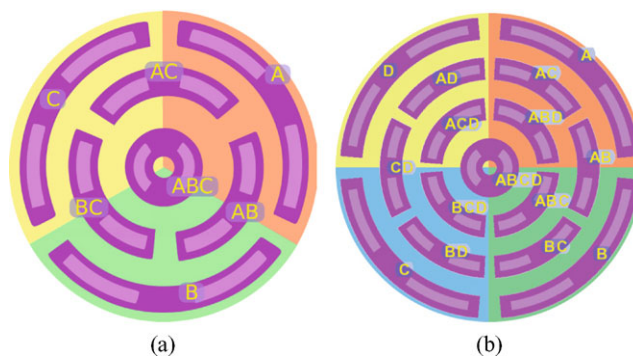


Figure 3a shows a completely concise form for $D = \{A, B, C\}$. The red, green, and yellow areas indicate the sectors for $\{A\}$, $\{B\}$, and $\{C\}$, respectively. $\{A, B\}$ maps to an arc that overlaps with the sectors from $\{A\}$ and $\{B\}$. Similarly, $\{B, C\}$ maps to an arc that overlaps the sectors from $\{B\}$ and $\{C\}$, and $\{A, C\}$ maps to an arc that overlaps with the sectors from $\{A\}$ and $\{C\}$. $\{A, B, C\}$ maps to the core circle which overlaps all three sectors.

Unfortunately, a completely concise form does not exist for datasets with $n > 3$ elements. To see this, consider the outer ring’s layout for a dataset $D = \{i_1, \dots, i_n\}$, $n > 3$. Each arc has exactly two neighbors. i_1 , for example, has neighbors i_n and i_2 . By definition, the first interior ring has more arcs than the outer ring, so its arcs are shorter than arcs on the outer ring. This implies that any arc in an interior ring can overlap at most two neighboring sectors on the outer ring. Any subset involving i_1 and a non-neighboring element (e.g., $\{i_1, i_3\}$) therefore cannot be mapped to an arc on the interior ring that overlaps both $\{i_1\}$ ’s and $\{i_3\}$ ’s sectors. Thus no completely concise form exists.

Figure 3b shows the layout for $D = \{A, B, C, D\}$. In the first inner ring, subsets $\{A, B\}$ and $\{C, D\}$ are mapped to arcs in completely concise locations, but $\{A, C\}$, $\{A, D\}$ and $\{B, C\}$, and $\{B, D\}$ are not. Since completely concise forms do not exist for $n > 3$, our technique approximates a concise layout as closely as possible by introducing arc penalty scores.

4.3 Penalties

We generate the required arcs in each ring using a penalty score that measures how close to completely concise a candidate mapping is. Consider a subset of items $\{i_j, \dots, i_k\}$, $j < k$ assigned to an arc on an interior ring. To calculate the subset’s penalty in its current position, we first extract its elements i_j, i_{j+1} , and so on. For each i_j , we count how many arcs it would need to hop to arrive at an arc that intersects i_j ’s sector. We chose the direction to hop—clockwise or counterclockwise—that produces the smallest penalty. A subset’s penalty is the sum of all its elements’ penalties.

The sum of all subset penalties represents the ring’s overall penalty. Thus, a ring with a lower penalty is closer to a completely concise form.

For example, consider subset $\{A, C\}$ in Fig. 3b. The arc holding $\{A, C\}$ intersects A ’s sector, so $A \in \{A, C\}$ receives a penalty score of zero. This arc does not intersect element C ’s sector, however. Element C would need to hop three arcs clockwise or two arcs counterclockwise to intersect C ’s sector. The penalty for $C \in \{A, C\}$ is therefore the smaller hop value, two. The overall penalty for the arc holding $\{A, C\}$ is $0 + 2 = 2$. This same procedure can be applied to all the arcs in the ring. Summing the arc penalties generates an overall penalty for the entire ring.

4.4 Reducing penalties

Although we have shown it is not possible to achieve an interior ring penalty of zero for datasets of size $n > 3$, we would like to minimize the penalty for each ring. Intuitively, there are two ways to do this.

First, we can try to increase the number of sectors each interior arc intersects. For example, in Fig. 3b, the arc holding $\{A, D\}$ intersects D ’s sector, but not A ’s. If we rotated the ring clockwise by a just a few degrees, however, the arc would intersect both A ’s and D ’s sectors, reducing its penalty from one to zero. This rotation would also reduce arc $\{B, C\}$ ’s penalty in the same way. Moreover, none of the other arcs will have changed the sectors they overlap, so none of their penalties will change. In particular, no arc penalty increases. Therefore, increasing the number of sectors the arcs intersect reduces the ring’s overall penalty by two.

Second, we can try to modify the mapping of subsets to arcs in a way that reduces the distance to the subsets’ sectors and, therefore, reduces the ring’s penalty. The first inner ring in Fig. 3b applies this logic. Each subset is assigned to an arc that is “close” to its elements’ sectors. There is no pairwise swapping of two arcs that will decrease the sum of the penalties for the arcs.

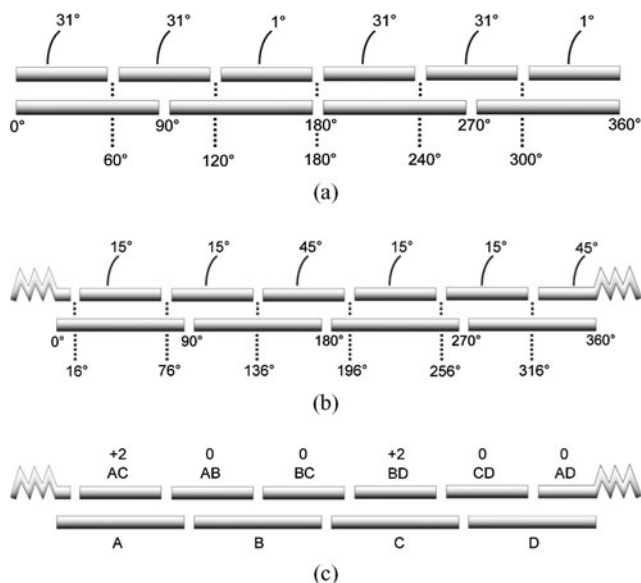


Fig. 4 Maximizing overlap and assigning subsets to arcs: (a) unwound versions of the outermost and first interior rings in Fig. 3b; (b) a 16° clockwise rotation to maximize arc to sector overlap; (c) assignment of subsets of size two to arcs that minimizes arc and overall ring penalties

4.4.1 Maximizing overlap

We apply an iterative approach to rotate each interior ring to a position that maximizes the overlaps between its arcs and the sectors of the outermost ring. Consider the outermost and first interior rings of Fig. 3b. We begin by unwinding the rings to form straight line segments. Aligning the segments horizontally provides an easy way to count the number of intersections. Figure 4a shows the interior ring on the upper line segment, and the outermost ring on the lower line segment. The angular distance between neighboring segments defines how the arcs are separated. For the interior ring, by symmetry a 60° rotation in either direction places the arcs back in their original configuration. This means we only need to consider rotating the inner ring in the range 0° . . . 60° when searching for a configuration that maximizes the number of overlaps.

We do not need to test every possible rotation within this range. Instead, we only consider rotations by angles that produce one of the following critical events:

1. An inner arc moves out of a sector.
2. An inner arc moves into a new sector.

In Fig. 4a, the angles above the inner arcs represent the critical angles for a right shift (a clockwise rotation). A 1° right shift will move the third and sixth arcs into a new sector, increasing their overlap from one to two. A 31° shift will move arcs one and four into a new sector, increasing their overlap from one to two, but it will also move arcs two and five out of a sector, decreasing their overlap from two to one.

The first two critical events occur at 1° and 31°, so shifting by 1° or more, but less than 31°, will invoke the first set of critical events—increasing the overlap of the third and sixth arcs from one to two. Although we could choose any shift angle $1 \leq \theta_1 < 31$, we want to make the overlaps between arcs and segments as obvious as possible. We choose the median between the two critical events, $\theta_1 = 1 + \frac{31-1}{2} = 16^\circ$.

Figure 4b shows the new arc positions after a 16° shift. The number of intersections has increased from $1 + 2 + 1 + 1 + 2 + 1 = 8$ (Fig. 4a) to $1 + 2 + 2 + 1 + 2 + 2 = 10$. After the rotation, the critical angles may change. In our example, they now occur at 15° and 45°. We apply another $\theta_2 = 15 + \frac{45-15}{2} = 30^\circ$ rotation to invoke the next set of critical events. We continue this process until the total shift $\Theta = \sum_{i=1}^n \theta_i$ exceeds the amount needed to reach a symmetric realignment, that is, $\Theta \geq 60$ in our example. We then select the shift θ_i that produced the largest overlap, and rotate the ring by that amount.

4.4.2 Assigning subsets to arcs

Once arc positions are chosen to maximize overlap, we must decide which subset to map to each arc. For w subsets mapped to w arcs in a ring there are $w!$ possible mappings. It is not feasible to compute the penalties for all the possibilities, especially for larger w . We instead assign subsets in two stages.

We begin by mapping subsets to arcs that overlap two sectors. Our goal is to try to produce penalties of zero. If no such mappings exist, subsets are assigned to arcs to make the penalties as small as possible. For example, for $D = \{A, B, C, D, E\}$ with $n = 5$ elements, on the ring holding subsets of size three, candidates $\{A, B, C\}$ and $\{A, B, E\}$ are most suitable for an arc intersecting sectors A and B , since the penalty for either assignment is one: hop one sector clockwise to match $C \in \{A, B, C\}$ to its sector, or hop one sector counterclockwise to match $E \in \{A, B, E\}$ to its sector.

After two-sector arcs are filled, we map the remaining subsets to arcs that intersect a single sector. We choose a candidate at random from the remaining subsets, then assign it to the arc that minimizes the penalty value. Following this stage, every subset is assigned to an arc, and every arc holds exactly one subset.

As an example, consider the outermost and first inner rings in the concentric visualization in Fig. 3b. The configuration in Fig. 4b maximizes the number of overlaps. We first assign $\{A, B\}$, $\{B, C\}$, $\{C, D\}$, and $\{A, D\}$ to the arcs intersecting A and B , B and C , C and D , and D and A , respectively. The remaining subsets $\{A, C\}$ and $\{B, D\}$ are assigned to arcs that minimize the penalty value to two. The ring penalty for this arrangement is therefore $2 + 0 + 0 + 2 + 0 + 0 = 4$ (Fig. 4c).

Table 2 Four bidders and three items $\{A, B, C\}$ from our IPC auction, with integers showing the maximum each bidder is willing to pay for different bundles of items

	A	B	AB	C	AC	BC	ABC
Bidder 1	10	3	18	2	18	10	20
Bidder 2	4	9	15	3	12	18	20
Bidder 3	1	3	11	9	16	17	25
Bidder 4	7	7	16	7	16	16	20

5 Visualization design

The first practical testbed for our visualization system is a proxy agent combinatorial auction generated in Dr. Peter Wurman’s E-commerce Laboratory [16]. They provided a dataset where four buyers bid on combinations of three objects, $D = \{A, B, C\}$. Each buyer has exactly one agent bidding in the auction. Before the auction begins, buyers provide the agents with the highest value they are willing to pay for each possible bundle. Table 2 shows the values used in this testbed.

Table 1 shows two inflection points from the original tabular representation for the combinatorial auction results. Recall that inflection points are times when a critical event occurs: a new bundle enters an agent’s demand set, competitive allocations change, or an agent withdraws as an active bidder. The table includes bundle prices at the designated time, all potentially competitive allocations, an asterisk, each agent’s potential demand set, and each agent’s allocation of attention.

These same attribute values form the source of our input data. Similar to Table 1, we focused on four attributes: the times of inflection points, bundle prices, competitive allocations, and each agent’s allocation of attention. We use results from visual perception to combine color, texture, and motion to visualize each attribute, and to animate changes within the auction over time.

5.1 Bundle prices

Two visual features, color, and blur, are used to visualize bundle prices. The minimum and maximum prices for each bundle are known before the auction begins, allowing us to normalize a bundle’s price onto the range $[0, 1]$. Recall that bundles prices can only increase as the auction runs.

We use knowledge of color perception to build a perceptually balanced color scale [7, 14] (Fig. 1). A balanced color scale ensures that, for two colors located anywhere along the scale, if the colors are a constant distance apart, they will produce a constant perceived color difference. Our color scale is built in the CIE LUV color model [2], looping once around the luminance pole as it passes through dark blues and greens at one end to bright oranges and reds at the

other. Embedding a monotonically increasing, perceptually balanced luminance within the scale allows it to properly highlight high spatial frequency differences in the data.

Blur is applied to bundles that do not receive new bids, to orient a viewer’s focus of attention to bundles receiving new bids. Blur was originally suggested by Kosara et al., who created a *semantic depth of field* by blurring less relevant data to make more important information stand out [9]. This is similar to using depth-of-field in photography to focus on a target subject and imply other details through an out-of-focus background.

5.2 Allocation of attention

Rectangular subregions are embedded within each arc in the visualization to represent proxy agents. We assign color and transparency to each proxy’s embedded region to represent how much “attention” he has allocated to the corresponding bundle:

- *white* to show that the agent is a candidate to win this bundle,
- *grey* to show that the agent is interested in this bundle, and he is increasing his bid,
- *dark green* to show that the agent is interested in this bundle, but he is not increasing his bid, and
- *transparent* to show that the agent is not interest and will not bid on this bundle.

We initially tried to visualize each agent’s allocation of attention as a numeric value. To measure this, an agent’s attention for a bundle is divided by the total attention from all agents for the bundle. The result represents the strength of an agent’s attention relative to the other agents who are also interested in securing the bundle. We mapped allocation strength to the same color scale used for bundle prices, then assigned the color to the agent’s subregion. The colors changed over time to reflect an agent’s increase or decrease in attention for a particular bundle.

Unfortunately, this produced a visually complex scene with many colored regions of various sizes competing for a viewer’s attention. In hindsight, this is not surprising, since psychophysical experiments conducted in our laboratory showed that a viewer is limited to distinguishing between seven or fewer isoluminant colors [6]. Because of this, we switched to a simpler representation where allocation of attention is categorized into one of four states: winning, interested and bidding, interested but not bidding, and not interested. For subregions that are either winning or interested and bidding, we display a proxy agent ID to clarify who the subregion represents.

5.3 Competitive allocations

Our visualization identifies all competitive allocations that could be chosen by the auctioneer at a given inflection point.

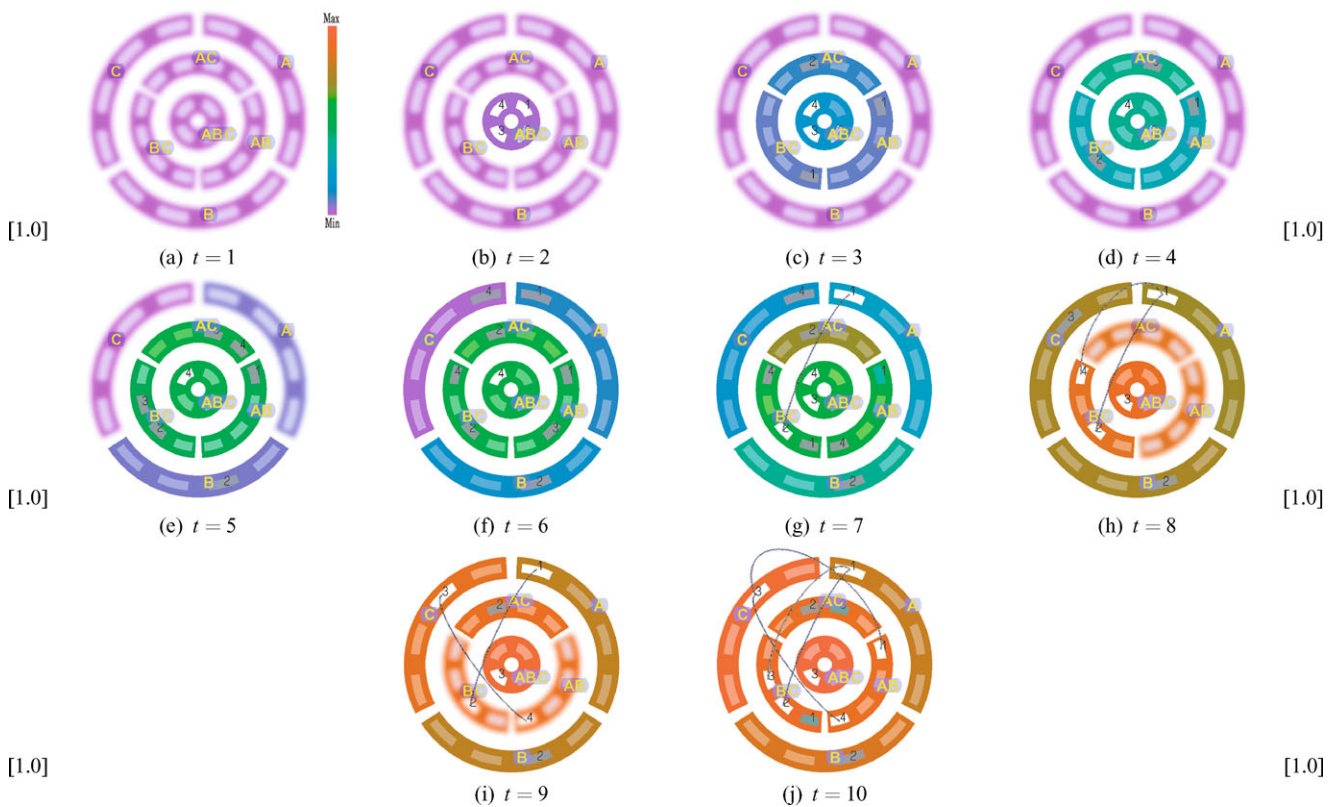


Fig. 5 Snapshots of the ten inflection points in our testbed auction: bundle price → arc color, active bundles → sharpness, attention → subregion color, competitive allocation → textured lines

As noted above, winning agents are visualized as white subregions within an arc (bundle). Collections of these winning bundles form competitive allocations.

To avoid interference, Ware recommends assigning new data to the least used graphical dimension in a visualization [15]. Color represents bundle prices and allocation of attention, so we apply textures to distinguish the assignment of winning bundles to competitive allocations. The bundles that make up a particular allocation are connected with textured curved lines. Different competitive allocations use different texture patterns that are perceptually distinguishable [7].

New inflection points can have new competitive allocations, which means the specific winning bundles will change. Our visualization shows the competitive allocations at the current inflection point. As new inflection points are visualized, the competitive allocations are updated simultaneously by translating the endpoints of the curved lines to reflect new combinations of winning bundles.

5.4 Inflection points

Our visualizations are not static. We animate them to show the progression of time during an auction. Color, blur, transparency, and textured lines vary between timesteps to high-

light changes in the auction's state. This is especially useful for showing changes at inflection points. Properties of motion like flicker rate, direction, and velocity have been shown to be perceptually salient, and have been recommended as ways to draw a viewer's focus of attention to important areas of change in an image [1, 8].

Before the animation begins, we retrieve information about the first two inflection points. These are assigned to a current state and a next state, respectively. The difference in a bundle's price between inflection points is used to smoothly vary its arc's color and blur. The colors of allocation of attention subregions and endpoints of competitive allocation lines are reset as each new inflection point is reached.

A local timer tracks the auction's progression. When the timer exceeds the timestamp of the next state, attribute values are updated, and the current and next states are updated based on the next inflection point. This process continues until the auction ends.

Figure 5 visualizes all ten timesteps in our testbed auction. From our visualizations, we can observe a general bidding strategy. In Fig. 5a, there are no bids on any bundles other than the entire collection of items $\{A, B, C\}$ at the core, innermost ring. Beginning in Fig. 5b, some agents attempt to bid on other bundles of size two. In Fig. 5e, the first

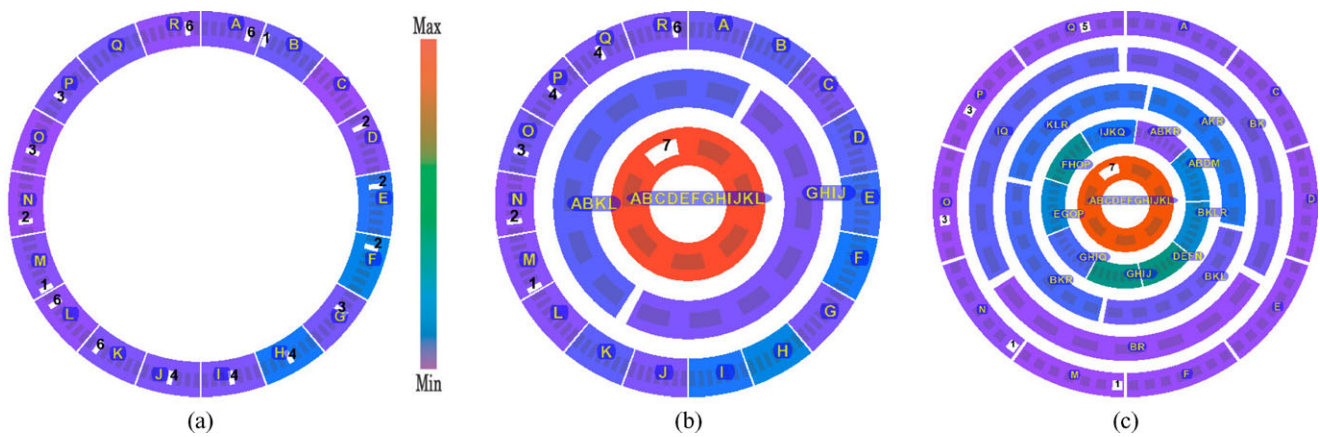


Fig. 6 FCC auction visualizations: (a) SMR auction; (b) TPB auction; (c) MPB auction

bid on a bundle with a single item B appears. A simple interpretation of these images shows that initially all of the agents try to win the bundle containing all available items. As this bundle’s price increases, agents distribute their bids to other bundles with fewer items to try to avoid the highly competitive single bundle. This process continues until the last inflection point in the auction (Fig. 5j), where four competitive allocations $\{A, BC, -, -\}$, $\{A, -, BC, -\}$, $\{AB, -, C, -\}$, and $\{-, -, C, AB\}$ exist. Following the final bid step, allocations $\{A, BC, -, -\}$ and $\{A, B, C, -\}$ remain competitive (Table 1). The auctioneer will randomly choose one of these distributions as the winning allocation.

Anecdotal feedback from our domain experts has been positive. They feel the visualizations do a much better job than the data tables at showing the overall state and progression of an auction, particularly given the ability to “scrub” back and forth in time to see how properties of the auction are changing. Although the data tables are still needed to determine exact attribute values, the visualizations are used first to obtain an overview of the auction, and to identify inflection points where a more detailed analysis could be useful.

5.5 FCC auctions

To further explore the usability and scalability of our technique, we visualized three different auction strategies being studied by the Federal Communications Commission (FCC). The FCC is investigating the use of combinatorial auctions to allocate wireless spectrum licenses to interested parties like cell phone and Internet service providers. The FCC has already completed one auction of this type, but it was extremely limited in its scope. There were only two bidders bidding on six items. Moreover, only one of the bidders actually participated in the auction, winning five of the six items in the second round. The auction terminated at this point, since there were no other bidders to offer higher prices on any of the bundles.

As an alternative, we instead visualized data from one of the FCC’s auction design experiments [4]. Here, 18 spectrum licenses $\{A, B, C, \dots, Q, R\}$ were auctioned to seven individuals who acted as bidders. Each individual is an expert in the wireless spectrum domain, and all the individuals had previous experience with the auction system. The main goal of the experiment was to compare the performance properties of three types of auctions: a simultaneous multiround (SMR) auction, a tiered package bidding (TPB) auction, and a modified flexible package bidding (MPB) auction. In an SMR auction all licenses are put up for bid simultaneously, and bidders can submit bids on individual licenses only, that is, there are no bundles with more than one item in an SMR auction. In a TPB auction, bidders can submit bids either on individual licenses, or on a set of predefined, non-overlapping bundles specified by the auctioneer. Finally, an MPB auction runs similar to the first testbed: bidders may bid on bundles of licenses of their choosing.

Figures 6a–c visualize the final round of the SMR, TPB, and MPB auctions, respectively. We visualize the final round only because the datasets provided by the FCC do not contain intermediate round information like bid prices or provisional winners. To handle the potentially exponential number of bundles in the TPB and MPB auctions, we only visualize bundles that received one or more bids during the auction (note that we can see *which* bundles were bid on during intermediate rounds). This reduces significantly the size and number of rings. The absence of an arc can also provide interesting insights into an auction. For example, in Fig. 6c, the outermost ring contains ten arcs, showing that only 10 of the 18 individual licenses received bids.

To address the goal of comparing the different auction types, we need to visualize how licenses are sold across different auctions. An overall maximum and minimum bid price for each bundle is calculated across the three auctions. Bundle arcs are then colored based on this

global value range. This allows a viewer to compare a bundle's price in each of the three auctions by comparing the color of the bundle's arc in the different visualizations. For example, in the TPB and MPB auctions the central bundle $\{ABCDEFGHIJKL\}$ is red, suggesting it sold for a higher price than the sum of the individual items $\{A, B, D, E, F, G, H, I, J, K, L\}$ in the SMR auction, where each arc is purple or blue. Examination of the actual bundle prices confirms this finding.

Other useful details are also present in the visualizations. We found that all the licenses were sold in the TPB auction, while in the MPB auction R was unsold, and in the SMR auction C and Q were unsold. More bidders won wireless spectrum licenses in the TPB auction versus the MPB auction. Finally, only two types of bundles were sold in the TPB and MPB auctions: bundles within individual items, and the bundle containing $\{ABCDEFGHIJKL\}$. This occurred even though MPB allows bidders to construct their own bundles for bids, versus the predefined bundles in the TPB auction.

Another finding of the FCC, that the TPB auction produced higher overall auction revenues, can be inferred by observing that bundle prices were similar in both the TPB and MPB auctions, but that more licenses were sold in the TPB auction (18) versus the MPB auction (17). Again, this inference needs to be verified by computing the actual revenues generated in each auction. This demonstrates our expectation, stated in the Introduction, that the visualizations can be used to quickly explore an auction to identify interesting activity, but that these "interesting" events would then be examined in more detail by studying related values in the data tables.

6 Conclusions

This article describes a new method for visualizing combinatorial auctions. Combinatorial auctions use combinatorial datasets as their underlying data structure. Hasse diagrams and Venn diagrams are the conventional methods used to represent these exponentially growing datasets. Neither of them is designed for combinatorial auctions, however, and both have additional limitations. In a Hasse diagram, edges representing the subsets' relationships usually cross each other, producing a complex graph with many crowded lines. In a Venn diagram, screen space is assigned in an unbalanced manner. Furthermore, these nonuniform regions can shrink to sizes that are impractical as the number of elements n grows.

We present a concentric ring visualization to address these issues. Our layout is designed to provide a stable, intuitive structure. We use multiple rings to represent different bundle sizes, and multiple arcs within a ring to represent

all possible bundles of a given size. A well-organized visual representation will have bundles with common items intersecting the items' outermost arc sectors. Although this arrangement is impossible for D of size $n > 3$, our layout minimizes the number and the size of the mismatches we generate. Rather than trying all possible combinations of subsets to arcs to locate an optimal assignment, we explore a small portion of the search space to identify high-quality representations.

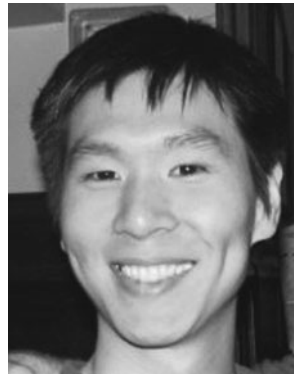
Color, blur, transparency, texture, and motion are used to represent data attributes within a combinatorial auction. The data is divided across inflection points that identify times when important auction properties change. We visualize four data attributes: bundle prices, each agent's allocation of attention, competitive allocations, and inflection point locations. Inflection points trigger our visualization to retrieve information about future auction states, and to update the display of the current state shown to the viewer. Bundle prices and price changes are represented by perceptually balanced colors and blur, respectively. Agents' allocations of attention are reduced to one of four static states displayed as subregions with different colors and transparencies. Finally, potential winners in a competitive allocation are connected using textured curve lines. Sets of winning bundles for different competitive allocations are distinguished by curves with different textures.

Our technique is currently limited to the four data attributes we are visualizing. Including additional information may produce a more comprehensive analysis tool, but only if the new data can be included in ways that does not reduce the effectiveness or the perceptual salience of the current representation.

References

1. Bartram, L., Ware, C., Calvert, T.: Filtering and integrating visual information with motion. *Inf. Vis.* 1(1), 66–79 (2002)
2. CIE: CIE Publication No. 15, Supplement Number 2 (E-1.3.1, 1971): Official Recommendations on Uniform Color Spaces, Color-Difference Equations, and Metric Color Terms. Commission Internationale de L'Éclairage (1978)
3. Cramton, P.: Spectrum auctions. In: Cave, M., Majumdar, S., Vogelsang, I. (eds.) *Handbook of Telecommunications Economics*. Elsevier, Amsterdam (2002)
4. Goeree, J.K., Holt, C.A., Ledyard, J.O.: An experimental comparison of flexible and tiered package bidding. http://wireless.fcc.gov/auctions/data/papersAndStudies/fcc_report_052507_final.pdf (2007)
5. Golubic, M.C., Ben-Arroyo Hartman, I. (eds.): *Graph Theory, Combinatorics, and Algorithms: Interdisciplinary Applications*. Springer, New York (2005)
6. Healey, C.G.: Choosing effective colours for data visualization. In: *Proceedings of the 7th IEEE Visualization Conference (Vis '96)*, pp. 263–270. San Francisco, California (1996)

7. Healey, C.G., Enns, J.T.: Large datasets at a glance: combining textures and colors in scientific visualization. *IEEE Trans. Vis. Comput. Graph.* **5**(2), 145–167 (1999)
8. Huber, D.E., Healey, C.G.: Visualizing data with motion. In: *Proceedings of the 16th IEEE Visualization Conference (Vis 2005)*, pp. 527–534, Minneapolis, Minnesota (2005)
9. Kosara, R., Miksch, S., Hauser, H.: Focus + context taken literally. *IEEE Comput. Graph. Appl.* **22**(1), 22–29 (2002)
10. Ono, C., Nishiyama, S., Horiuchi, H.: Designing a double-sided combinatorial auction system. In: *Proceedings of the International Conference on Information and Knowledge Engineering 2003*, pp. 445–451, Las Vegas, Nevada (2003)
11. Rassenti, S.J., Smith, V.L., Bulfin, R.L.: A combinatorial auction mechanism for airport time slot allocation. *Bell J. Econ.* **13**, 402–417 (1982)
12. Song, J., Regan, A.: Approximation algorithms for the bid construction problem in combinatorial auctions for the procurement of freight transportation contracts. *Transp. Res., Part B, Methodol.* **39**(10), 914–933 (2005)
13. Venn, J.: On the diagrammatic and mechanical representation of propositions and reasonings. *London Edinburgh Dublin Philos. Mag. J. Sci.* **9**(9), 1–18 (1880)
14. Ware, C.: Color sequences for univariate maps: theory, experiments, and principles. *IEEE Comput. Graph. Appl.* **8**(5), 41–49 (1988)
15. Ware, C.: *Information Visualization: Perception for Design*, 2nd edn. Morgan Kaufmann, San Francisco (2004)
16. Wurman, P.R., Zhong, J., Cai, G.: Computing price trajectories in combinatorial auctions with proxy bidding. *Electron. Commer. Res. Appl.* **3**(4), 329–340 (2004)
17. Zurel, E., Nisan, N.: An efficient approximate allocation algorithm for combinatorial auctions. In: *Proceedings ACM Electronic Commerce Conference*, pp. 125–136, Tampa, Florida (2001)



Joe Ping-Lin Hsiao received a B.S. from the Yuan-Ze University in Chung-Li, Taiwan, and an M.S. and Ph.D. from North Carolina State University in Raleigh, North Carolina. His research interests include visualization, text analytics, and computer graphics.



Christopher G. Healey is an Associate Professor in the Department of Computer Science at North Carolina State University. He received a B.Math. from the University of Waterloo in Waterloo, Canada, and an M.Sc. and Ph.D. from the University of British Columbia in Vancouver, Canada. He is an Associate Editor for *ACM Transactions on Applied Perception*. His research interests include visualization, graphics, visual perception, and areas of applied mathematics, databases, artificial intelligence, and aesthetics related to visual analysis and data management.