Computing Price Trajectories in Combinatorial Auctions with Proxy Bidding

Peter R. Wurman, Jie Zhong and Gangshu Cai

Computer Science Department
North Carolina State University
Raleigh, NC 27695-7535
{*wurman, jzhong, gcai*}*@ncsu.edu*

Abstract

Proxy bidding has proven useful in a variety of real auction formats—most notably eBay—and has been proposed for the nascent field of combinatorial auctions. Previous work on proxy bidding in combinatorial auctions requires the auctioneer to run the auction with myopic bidders to determine the outcome. In this paper we present a radically different approach that computes the bidders' allocation of their attention across the bundles only at "inflection points." Inflections are caused by the introduction of a new bundle into an agent's demand set, a change in the set of currently competitive allocations, or the withdrawal of an agent from the set of active bidders. This approach has several advantages over alternatives, including that it computes exact solutions and is invariant to the magnitude of the bids.

Key words: Combinatorial Auctions, Proxy Bidding, Auction Algorithms

1 Introduction

Iterative combinatorial auctions are auctions that combine the expressive power of combinational bids with the progressive march towards a solution common in iterative, single-item mechanisms like the English auction (see [9] for an introduction). Iterative combinatorial auctions are attractive for several reasons, but particularly because they require bidders to determine exact values on items, or combinations of items, only if those items are relevant to the final allocation. In cases where value determination is costly, bidders need not compute the value of bundles that they do not expect to win in the auction.

Recently, mechanism designers have expressed interest in enabling *proxy bidding* in iterative combinatorial auctions. Proxy bidding is the process in which the bidder expresses a bid—typically greater than strictly necessary to be accepted by the auction—to an agent that then bids incrementally on behalf of the bidder until it either wins or exhausts the authority granted it. In a combinatorial auction, the bidder sends a message to the proxy that expresses value for some or all of the bundles. The proxy bidder, guided by the message, bids incrementally on behalf of the user by following some prescribed bidding policy.

Proxy bidding has the advantage of speeding up the auction by allowing bidders to place larger bids which will be executed only to the extent necessary to outbid competitors [2,10,11]. Also, by restricting the strategic flexibility of the bidders, mechanism designers may be better able to design successful auctions and predict their outcomes. Indeed, Ausubel and Milgrom [2] show that a semi-sincere equilibrium always exists under proxy bidding in their Ascending Package Auction (APA).

In addition, enabling an auction with proxy bidding may reduce the need for the bidders to accurately estimate the valuations of the other participants in the auction. For example, an equilibrium strategy in a first-price sealed-bid auction requires estimating the value of the second highest bidder. However, when a first-price, sealed-bid auction is enhanced with proxy bidding it reduces to a Vickrey auction and each bidder's equilibrium strategy is to submit her true value [12]. Iterative combinatorial auctions are more complex than the English auction format used on sites like eBay, and proxy bidding does not necessarily produce incentive compatibility. We do not address strategic issues in this paper. Instead, we focus on the computational issues faces by the auctioneer. Nevertheless, there may be many problems for which proxy bidding simplifies the bidders' strategy selection problems.

Once the auction has collected the proxy bids, it must compute the final prices and allocation entailed by the bids and the proxy bidding policy. We refer to this task as solving the *Proxy Auction Problem* (PAP). A natural approach, which we refer to as *solving by simulation*, is to instantiate the proxy bidders as algorithms that, using the bidders' messages, incrementally bid until the winners and prices are determined. Recently [14], we provided an alternative algorithm that directly computes the outcome of the auction by computing the bidding patterns engendered by the proxy statements, and the subsequent price trajectories. The previous version of the algorithm provides the basic framework of the approach but was limited by a simplifying assumption on the events that cause changes in bidding behavior. In this paper, we relax that assumption and present a technique that solves the general problem.

In Section 2 we define a simple combinatorial auction with proxy bidding. Section 3 describes our novel algorithm for solving PAP, with particular attention to the task of computing the bidders' behavior. Section 4 walks through a portion of the calculations necessary to solve the running example problem. The final section contains observations on aspects of the algorithm and a discussion of future work.

2 A Simple Combinatorial Proxy Auction

We consider an iterative combinatorial auction that accepts bids on bundles of items, and generates non-linear bundle-prices, that is, the price of a bundle may not be the sum of the prices of individual items. Several proposed iterative combinatorial auctions fit into this category, including various versions of *i*Bundle [8,10], A1BA [15], and, with some interpretation, APA [2]. The set of rules we employ here define a simplified iterative combinatorial auction that combines aspects of the existing proposals. The details of the auction format described in this paper were chosen because of their natural interpretation within the framework of the proxy solution algorithm.

The problem faced by the auctioneer is to allocate n heterogeneous items to m buyers. Let $\mathcal{J}$ be the set of items, and $\mathcal{I}$ be the set of buyers, and index the sets with k and i, respectively. The set of all bundles, $\mathcal{B}$, contains 2^n different combinations of items, including the empty set, $\emptyset$. Let $\boldsymbol{b} \in \{0,1\}^n$ where $\boldsymbol{b}^k = 1$ implies that item k is an element of the bundle $\boldsymbol{b}$.

The auction proceeds in rounds, and in each round the buyers may place offers on a subset of bundles. We denote buyer i's bid at round t as a collection of mutually exclusive offers on bundles of the form $r_i^t(\boldsymbol{b})$, where $r_i^t(\boldsymbol{b}) \in \mathcal{R}_+$, [1] and we make the standard assumption that $r_i^t(\emptyset) = 0$. The auction remembers each bidder's last offer on each item. Thus, if i does not update her bid in round $t+1$, $r_i^{t+1}(\boldsymbol{b}) = r_i^t(\boldsymbol{b})$.

It is convenient for our reduction to the algorithm for PAP to assume that the bidding rules require that an agent either pass or improve its bid on exactly one bundle by increasing it to the current price plus a small increment, δ. This restriction could also be implemented as part of the proxy agent's bidding strategy.

After the bids are recieved, the auction computes the winning combination of bids, a task called solving the *Winner Determination Problem* (WDP). Algorithms for solving the WDP have been widely studied in recent years [1,3,13]. At the end of each round the new bundle-prices are announced and the auction tells the bidders which bundles, if any, they are winning.

The announced prices are simply the highest bid received on a particular item and are *anonymous* in that all bidders are given the same information. However, the prices are not necessarily *separating* [16] because the optimal allocation may include bidders whose last offer on their winning bundle is less than the current price; hence we need to directly inform bidders of their winning status. Let $\pi_{\boldsymbol{b}}$ denote the

[1] This standard XOR format is fully expressive, although not the most concise method of expressing bids [7].

price associated with bundle $\boldsymbol{b}$, and

$$\pi_{\boldsymbol{b}} = \max_i r_i^t(\boldsymbol{b}).$$

In the proxy-enabled auction, the bidder submits to the proxy agent a (not necessarily truthful) value statement, v_i, that defines a willingness to pay for some subset of bundles. The proxy agent submits the incremental bids, $r_i^t(\boldsymbol{b})$, on behalf of the user by following a *straightforward bidding policy* [6]. If it is told by the auction that it is winning, the agent does not increase its bid. Otherwise, the agent bids on the bundle, $\boldsymbol{b}'$, that maximizes its real surplus at the given prices. That is,

$$\boldsymbol{b}' = \arg\max_{\boldsymbol{b}} \left\{ v_i(\boldsymbol{b}) - (\pi_{\boldsymbol{b}} + \delta) \right\}, \tag{1}$$

where δ is the minimum bid increment and $\boldsymbol{\pi}$ is the set of announced bundle-prices. If more than one bundle satisfies (1), the agent selects one randomly to bid on. We refer to the set of bundles that satisfy (1) as the agent's *demand set* [2] and denote it D_i^t. [3]

Note that agents will often reach a point at which no bundle provides positive surplus at the current prices. When this occurs, the agent will stop bidding in the auction, though its most recent bids remain. The auction terminates when no new bids are received in a round.

We designate the auction defined by the combination of proxy bidding and these simple rules as the *Simple Combinatorial Proxy Auction* (SCPA). Given a set of value statements, the auctioneer must now solve the *Proxy Auction Problem* (PAP), that is, it must compute the prices and allocation that result when all of the agents follow straightforward bidding. A natural approach is to iteratively compute incremental bids until the termination criteria is reached. However, this *simulated bidding* approach has several undesirable properties. First, the outcome is dependent upon implementation details such as the tie-breaking rule, the order of bidding, and the bid increment. In fact, the randomizations described in the SCPA description were chosen to avoid biases and produce a consistent outcome over a sufficient number of iterations. Second, the accuracy is a function of the bid increment. We can improve the accuracy by decreasing the bid increment, but doing so will greatly increase the number of iterations and the amount of time the process takes. This is particularly undesirable because each iteration requires the auction solve an NP-complete WDP and compute new prices. Although some researchers [5] have begun looking at techniques for reusing previous solutions in iterative combinatorial auctions, the techniques are at an early stage of development.

[2] The demand set is also referred to as the *best response set* in the literature.

[3] For expository clarity, we leave the t superscript off the symbols when it is not ambiguous.

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

Table 1
An example with four buyers bidding on the combinations of three objects.

We advocate a novel approach to solving PAPs rooted in the observation that the straightforward bidding policy generates predictable patterns of bidding behavior that results in complex, but computable, trajectories for bundle-prices. The approach outlined in the next section allows us to compute the results by taking large steps in a manner analogous to the way the outcome of proxy bidding in eBay can be computed by setting the price to δ above the second highest bid. Moreover, we can compute an exact outcome, one that is independent of the bid increment or tie-breaking rules, and whose run time is independent of the magnitude of the bids.

3 Solving PAP

3.1 Concepts

Table 1 shows a scenario we use throughout the paper. Figure 1 shows the progression of the price of each bundle through the course of the proxied auction with the agents given buyer values as shown in Table 1. Both the simulation (when δ is sufficiently small) and our new algorithm generate the price curves in the figure. The figure has obvious structure: prices increase at steady rates with occasional events that cause them to change trajectories.

The central concept of our approach is to compute each bidder's allocation of *attention* among the bundles. Attention represents the proportion of its bidding opportunities that the agent will use to increase its offer on a bundle. Each agent has one unit of attention per unit of time to allocate among the bundles and the passing action. Attention can also be thought of as dollars per unit time; for the purposes of this exposition we will use a normalized value of one dollar per unit time. We denote agent i's attention to bundle $\boldsymbol{b}$ as $\theta_{i,\boldsymbol{b}}$. If the agent is told it is winning by the auctioneer, it will pass. Denote the proportion of time agent i spends passing as $\theta_{i,\text{pass}}$.

The trajectory of a bundle is simply the sum of the attention being paid to it. Let

$\theta^t_{\boldsymbol{b}}$, the trajectory of bundle $\boldsymbol{b}$ at instant t, be computed as

$$\theta^t_{\boldsymbol{b}} = \sum_i \theta^t_{i,\boldsymbol{b}}.$$

In Figure 1, at time zero, the price of every bundle is zero. Consider the time interval, T_1, between zero and $t = 2/3$. Each agent's demand set is the singleton ABC and the four agents spend their time outbidding one another. However, although the price of ABC increases rapidly, it does not increase at slope equal to the number of bidders. The bidding pattern that is established during the initial period has one (randomly selected) bidder passing because it is announced as the winner of ABC, and the other three bidders increasing their bids. At any given instant, each agent has a 0.25 probability of being announced as the winner, and a 0.75 probability of increasing its bid because it is not winning. Four agents allocating 0.75 units of attention creates the slope of 3.0 seen in interval T_1.

In order to correctly determine the price trajectories, the algorithm must also keep track of the *competitive allocations* (CAs). We use set notation to indicate the structure of an allocation, where the lexigraphical position of a bundle in the set corresponds to the agent to which it is assigned. During interval T_1, there are four CAs: $\{$ABC, –, –, –$\}$, $\{$–, ABC, –, –,$\}$, $\{$–, –,ABC, –,$\}$, and $\{$–, –, –, ABC$\}$. Importantly, the set of competitive allocations is identical to the set of solutions to the WDP.

More formally, let $\boldsymbol{f}$ denote a feasible allocation of the objects, that is, $\boldsymbol{f} : \mathcal{J} \rightarrow \mathcal{I}$. Let F be the set of all feasible allocations. Denote agent i's allocation in $\boldsymbol{f}$ as $\boldsymbol{f_i}$. The value of allocation $\boldsymbol{f}$ at time t is

$$V(\boldsymbol{f}) = \sum_i r^t_i(\boldsymbol{f_i}). \tag{2}$$

Let F^* denote the set of allocations that maximize (2).

The CA notion is critical because agents will pass if and only if they are a member of a CA, and they will pass as often as a CA to which they belong is selected by the auctioneer. The values of the CAs that are significant in the running example are depicted in Figure 2. The CAs on the envelope are competitive. Notice that the four allocations in which ABC is allocated begin as CAs, but at $t = 2/3$ the allocation $\{$ABC, –, –, –$\}$ flatlines; Agent 1 does not bid on ABC again. Similarly, $\{$–, ABC, –, –,$\}$ and $\{$–, –, –, ABC$\}$ plateau when the price of ABC reaches 11 and 14, respectively. Interestingly, the allocation that eventually wins the auction, $\{$A, BC, –, –$\}$, does not become competitive until around time 17. Thereafter, it remains on the envelope until time 40 at which point the bidders supporting $\{$–, –, ABC, –$\}$ and $\{$–, –, C, AB$\}$ give up.

The proxy solution algorithm is an iterative process in which each step involves (1) computing the allocation of attention, and (2) computing the duration for which the

the former holds. In the following sections, we deal with these two issues in more detail, starting with the latter.

3.2 Computing Intervals

We use the same methodology to compute the duration of an interval described elsewhere [14], but here extend it to the general case. There are two potential events that define the end of an interval:

(1) the prices of bundles reach a point where one or more bidders become attracted to one or more bundles that were not previously in their demand sets, or
(2) an allocation that was not formerly competitive reaches a value that makes it competitive.

Unlike our previous work, we do not restrict these events to consist of a single allocation or a single agent and bundle. The complexity that arises when we lift this restriction is dealt with primarily in the next section.

To establish the duration of the interval starting at time t, we need the *first* change in demand sets among all agents; the trajectories will change at the first inflection point, rendering the other computed collisions obsolete. Consider a bundle, $\boldsymbol{c}$, that is not in i's demand set, and a bundle, $\boldsymbol{b}$, that is. The amount of time it will take for $\boldsymbol{c}$ to become as attractive to i as $\boldsymbol{b}$ is

$$\Delta t_i^{\boldsymbol{b},\boldsymbol{c}} = \frac{v_i(\boldsymbol{b}) - \pi_{\boldsymbol{b}}^t - v_i(\boldsymbol{c}) + \pi_{\boldsymbol{c}}^t}{\theta_{\boldsymbol{b}}^t - \theta_{\boldsymbol{c}}^t}.$$

Note that cases where $\Delta t_i^{\boldsymbol{b},\boldsymbol{c}}$ is negative or undefined indicate that the two trajectories do not collide in the future because they are parallel or diverging. Such pairs are excluded.

The duration of time until the next demand set change among all agents is

$$\Delta t^{\mathrm{DS}} = \min_i \{ \min_{\boldsymbol{b} \in D_i^t, \boldsymbol{c} \notin D_i^t} \Delta t_i^{\boldsymbol{b},\boldsymbol{c}} \}.$$

Similarly, we need to compute the first non-competitive allocation that will become competitive under the current allocation of attention. First, recognize that the trajectory of an allocation is the sum of the trajectories of the components that are actively being bid upon by members of the allocation. Let $\gamma_{\boldsymbol{f}}^t$ be the trajectory of $\boldsymbol{f}$, computed as

$$\gamma_{\boldsymbol{f}}^t = \sum_{i | \boldsymbol{f_i} \in D_i^t} \theta_{\boldsymbol{f_i}}^t. \qquad (3)$$

Competitive allocation, $\boldsymbol{f}$, will collide with non-competitive allocation, $\hat{\boldsymbol{f}}$, when the values of the two become equal. If the current time is t, the collision will occur in $\Delta t^{\boldsymbol{f},\hat{\boldsymbol{f}}}$ time increments, where

$$\Delta t^{\boldsymbol{f},\hat{\boldsymbol{f}}} = \frac{V^t(\hat{\boldsymbol{f}}) - V^t(\boldsymbol{f})}{\gamma^t_{\boldsymbol{f}} - \gamma^t_{\hat{\boldsymbol{f}}}}.$$

A non-positive value for $\Delta t^{\boldsymbol{f},\hat{\boldsymbol{f}}}$ means that, with the current allocation of attention, $\hat{\boldsymbol{f}}$ will not become competitive with $\boldsymbol{f}$. Finally, among the positively valued $\Delta t^{\boldsymbol{f},\hat{\boldsymbol{f}}}$'s, we find the first:

$$\Delta t^{\mathrm{CA}} = \min_{\boldsymbol{f} \in F^*, \hat{\boldsymbol{f}} \notin F^*} \Delta t^{\boldsymbol{f},\hat{\boldsymbol{f}}}.$$

The next inflection point is determined by taking the minimum of Δt^{DS} and Δt^{CA}. In addition, we keep track of which allocations and combinations of agents and bundles triggered the inflection point.

3.3 Computing the Allocation of Attention

The main contribution of this paper is in defining how to compute the allocation of attention in the general case where an arbitrary number of bundles are added to demand sets and/or one or more allocations become competitive. The assumptions made in the original formulation of the algorithm [14] allowed us to simplify the maintenance of the demand set information at the inflection points. In the general case, demand sets interact in complex ways.

This section details a mixed integer-linear program (MILP) that simultaneously computes which allocations remain competitive, which bundles should be kept in the demand set for each agent, and how much attention each agent allocates to each item in its demand set. The domain of the MILP is defined by the *potential* demand sets, $\hat{D}_i$, and the *potential* competitive allocations, $\hat{F}^*$. Although all of the identified bundles and allocations are instantaneously active at the inflection at time t, we are concerned with determining which will remain active going forward. Mathematically, MILP: $\{\hat{D}_i, \hat{F}^*\} \rightarrow \{D_i, F^*, \theta_{i,\boldsymbol{b}}\}$.

The potential demand set for each agent is the union of the bundles that were previously in the agent's demand set and those bundles being introduced by the agent at this inflection point; in other words, the set of bundles that satisfy (1) at time t. The integer variable $y_{i,\boldsymbol{b}}$ takes the value one if bundle $\boldsymbol{b}$ will remain in agent i's demand set during the following interval, and zero otherwise.

The potential CA set, $\hat{F}^*$, is the union of the previous CAs and those allocations being introduced at this inflection point; no allocation can become competitive without being one of the causes of the inflection. The integer variable $x_{\boldsymbol{f}}$ takes the value one if allocation $\boldsymbol{f}$ will remain competitive during the interval, and zero otherwise.

It is clear that $\forall i \in \mathcal{I}$,

$$y_{i,\boldsymbol{b}} \geq \theta_{i,\boldsymbol{b}}, \quad \forall \boldsymbol{b} \in \hat{D}_i. \tag{4}$$

The straightforward bidding policy that each agent implements implies that when bundles $\boldsymbol{b}$ and $\boldsymbol{c}$ are both in D_i over an interval, $\theta_{\boldsymbol{b}} = \theta_{\boldsymbol{c}}$. It follows that when $y_{i,\boldsymbol{b}} = 1$,

$$\begin{aligned} &\text{if } \ y_{i,\boldsymbol{c}} = 1, \text{then } \ \theta_{\boldsymbol{b}} = \theta_{\boldsymbol{c}}, \\ &\text{if } \ y_{i,\boldsymbol{c}} = 0, \text{then } \ \theta_{\boldsymbol{b}} < \theta_{\boldsymbol{c}}. \end{aligned}$$

Expressed as integer constraints, $\forall i \in \mathcal{I}$ and $\boldsymbol{b}, \boldsymbol{c} \in \hat{D}_i$,

$$\theta_{\boldsymbol{c}} - \theta_{\boldsymbol{b}} + N y_{i,\boldsymbol{b}} \leq N, \tag{5}$$

$$1 - y_{i,\boldsymbol{c}} \leq N(\theta_{\boldsymbol{c}} - \theta_{\boldsymbol{b}}) + N^2(1 - y_{i,\boldsymbol{b}}). \tag{6}$$

where N is a sufficiently large constant.

Active agents have one unit of attention to allocate, while agents that no longer achieve positive surplus on any bundle (and therefore have empty demand sets) will allocate no attention. Let K_i be a constant used during problem construction where $K_i = 0$ if $\hat{D}_i$ is empty, and $K_i = 1$ otherwise. We conserve attention with the constraint

$$\sum_{\boldsymbol{b} \in \mathcal{B}} \theta_{i,\boldsymbol{b}} + \theta_{i,\text{pass}} = K_i, \quad \forall i \in \mathcal{I}. \tag{7}$$

We now turn our attention to the constraints that capture the influence of the competitive allocations. Let $\beta_{\boldsymbol{f}} \in [0, 1]$ be the frequency with which allocation $\boldsymbol{f}$ is announced as the winner, enabling the members of that allocation to pass. At every iteration, one of the CAs must be announced as the winning bundle. Thus,

$$\sum_{\boldsymbol{f} \in \hat{F}^*} \beta_{\boldsymbol{f}} = 1. \tag{8}$$

If allocation $\boldsymbol{f}$ is not competitive then $\beta_{\boldsymbol{f}} = 0$. In other words,

$$\beta_{\boldsymbol{f}} \leq x_{\boldsymbol{f}}. \tag{9}$$

By definition, competitive allocations increase their value at the same rate, while if a potential CA turns out to be not competitive, its slope must be less than those allocations that are competitive. Consider the case where $\boldsymbol{f}$ is competitive, that is, $x_{\boldsymbol{f}} = 1$.

$$\begin{aligned}
&\text{If } \ x_{\hat{\boldsymbol{f}}} = 1, \text{then } \ \gamma_{\boldsymbol{f}} = \gamma_{\hat{\boldsymbol{f}}}, \\
&\text{if } \ x_{\hat{\boldsymbol{f}}} = 0, \text{then } \ \gamma_{\boldsymbol{f}} \geq \gamma_{\hat{\boldsymbol{f}}}.
\end{aligned}$$

An integer-linear form of this logical constraint is

$$\gamma_{\hat{\boldsymbol{f}}} - \gamma_{\boldsymbol{f}} + N x_{\boldsymbol{f}} \leq N. \tag{10}$$

However, the slope of the allocation $\boldsymbol{f}$ depends upon which elements of the potential demand sets are retained in the actual demand sets of the agents. In particular, in equation (3), the slope is the sum of the aggregate attention paid to bundles only when the agent to which the bundle is allocated has the bundle in its active demand set. Formally,

$$\gamma_{\boldsymbol{f}} = \sum_i \theta_{\boldsymbol{f}_i} y_{i,\boldsymbol{f}_i},$$

which is nonlinear. To convert it to a linear constraint, we introduce the variable $\alpha_{i,\boldsymbol{b}}$ where $\alpha_{i,\boldsymbol{b}} = \theta_{i,\boldsymbol{b}}$ if $\boldsymbol{b}$ is in i's active demand set, and $\alpha_{i,\boldsymbol{b}} = 0$ otherwise. This relationship is captured in the constraints

$$\alpha_{i,\boldsymbol{b}} \leq N y_{i,\boldsymbol{b}}, \tag{11}$$
$$\alpha_{i,\boldsymbol{b}} - \theta_{\boldsymbol{b}} + N y_{i,\boldsymbol{b}} \leq N, \tag{12}$$
$$\theta_{\boldsymbol{b}} - \alpha_{i,\boldsymbol{b}} + N y_{i,\boldsymbol{b}} \leq N. \tag{13}$$

The introduction of the variable $\alpha_{i,\boldsymbol{b}}$ allows us to express the slope of an allocation as

$$\gamma_{\boldsymbol{f}} = \sum_{i \in \boldsymbol{f}} \alpha_{i,\boldsymbol{f}_i}.$$

Every competitive allocation will have some probability of being selected as the winning allocation. It is not true that all competitive allocations are selected equally often during an interval; they are only equally likely to be chosen when they are tied. Consider the bidding pattern in a two-agent, two-item scenario where Agent 1 is bidding only on AB, and Agent 2 is alternating between bidding on A and bidding on AB. In those rounds in which Agent 2 bids on A, {–, AB} is the only CA. In the alternate rounds, Agent 2 may increase its bid on AB, thereby tying Agent 1 and having a 0.5 probability of being declared the current winner. Thus, Agent 2 will bid on A half the time, on AB a quarter of the time, and pass the remaining quarter of the time.

This analysis suggest a connection between agent i's behavior and the frequency with which i is not a member of the winning allocation. Agent i bids whenever one of the other competitive allocations pass. We employ another constant that is used during problem construction: let $G_{\boldsymbol{f},i} = 1$ if i is allocated a bundle in $\boldsymbol{f}$, and $G_{\boldsymbol{f},i} = 0$ otherwise.

$$1 - \theta_{i,\text{pass}} = \sum_{\boldsymbol{f} \in \hat{F}^*} (1 - K_i G_{\boldsymbol{f},i}) \beta_{\boldsymbol{f}}. \tag{14}$$

The final piece of the MILP is the objective function. We take as our objective the maximization of the number of competitive allocations.

$$\text{maximize} \sum_{\boldsymbol{f} \in \hat{F}^*} x_{\boldsymbol{f}}.$$

Solving the MILP outlined above will compute the bundles that are in each agent's best response set going forward (those that have $y_{i,\boldsymbol{b}} = 1$), which allocations will be competitive (those for which $x_{\boldsymbol{f}} = 1$), and how much attention each agent pays to each bundle in its demand set. From the attention we can compute the slope of the price of each bundle and the slope of the allocations, both of which are necessary to determine the duration of the interval. The complete MILP constructed from equations (4–14) is shown in Figure 3.

4 Worked Example

The results of applying the algorithm to the example in Table 1 are shown in Figures 1 and 2. In this section, we highlight some of the steps in the process. Table 2 shows the computations involved at six of the ten steps involved in solving the auction. Each step shows the prices at the designated time, the potentially competitive allocations (with those that remain competitive in the interval designated with an asterisk), each agent's potential demand set, and each agent's allocation of attention. The bundles that are determined to be in each agent's active demand set have a value in the corresponding attention cell, even when the value is zero, while bundles that were in $\hat{D}_i^t$ but not D_i^t have empty cells. The attention columns are summed to give the trajectories of the prices going forward.

At $t = 0$, the prices are zero and ABC is the sole element in each agent's demand set. Since there are four CAs, and no other bundles distracting the bidders, each CA wins one fourth of the time. At $t = 2/3$, the bundles AB and AC enter Agent 1's demand set, and BC enters Agent 2's. Both agents focus their attention on the new elements.

Step 1, $t = 0$		Prices:	0	0	0	0	0	0	0	
$\boldsymbol{f} \mid V(\boldsymbol{f}) = 0$	Agent	$\hat{D}_i$	A	B	AB	C	AC	BC	ABC	pass
*{ABC, –, –, –}	1	ABC							0.75	0.25
*{–, ABC, –, –}	2	ABC							0.75	0.25
*{–, –, ABC, –}	3	ABC							0.75	0.25
*{–, –, –, ABC}	4	ABC							0.75	0.25
		$\theta_{\boldsymbol{b}}$	0	0	0	0	0	0	3	

Step 2, $t = 2/3$		Prices:	0	0	0	0	0	0	2	
$\boldsymbol{f} \mid V(\boldsymbol{f}) = 2$	Agent	$\hat{D}_i$	A	B	AB	C	AC	BC	ABC	pass
{ABC, –, –, –}	1	AB, AC, ABC			0.5		0.5			
*{–, ABC, –, –}	2	BC, ABC						1	0	
*{–, –, ABC, –}	3	ABC							0.5	0.5
*{–, –, –, ABC}	4	ABC							0.5	0.5
		$\theta_{\boldsymbol{b}}$	0	0	0.5	0	0.5	1	1	

⋮ ⋮ ⋮

Step 7, $t = 17\frac{1}{3}$		Prices:	2	3	10	1	10	12	14	
$\boldsymbol{f} \mid V(\boldsymbol{f}) = 14$	Agent	$\hat{D}_i$	A	B	AB	C	AC	BC	ABC	pass
*{A, BC, –, –}	1	A, AB, AC	5/14		0		1/14			4/7
{–, ABC, –, –}	2	B, BC, ABC		3/14				3/14		4/7
*{–, –, ABC, –}	3	ABC							4/7	3/7
{–, –, –, ABC}	4	AB, C, AC, ABC			5/14	5/14	4/14			
		$\theta_{\boldsymbol{b}}$	5/14	3/14	5/14	5/14	5/14	3/14	4/7	

Step 8, $t = 31\frac{1}{3}$		Prices:	7	6	15	6	15	15	22	
$\boldsymbol{f} \mid V(\boldsymbol{f}) = 22$	Agent	$\hat{D}_i$	A	B	AB	C	AC	BC	ABC	pass
*{A, BC, –, –}	1	A, AB, AC	1/7							6/7
*{A, –, –, BC}	2	B, BC		1/7				1/7		5/7
*{–, –, ABC, –}	3	C, ABC				3/7			3/7	1/7
	4	B, AB, C, AC, BC		1/7	2/7		2/7	1/7		1/7
		$\theta_{\boldsymbol{b}}$	1/7	2/7	2/7	3/7	2/7	2/7	3/7	

Step 9, $t = 34\frac{5}{6}$		Prices:	7.5	7	16	7.5	16	16	23.5	
$\boldsymbol{f} \mid V(\boldsymbol{f}) = 23.5$	Agent	$\hat{D}_i$	A	B	AB	C	AC	BC	ABC	pass
*{A, BC, –, –}	1	A	1/4							3/4
{A, –, –, BC}	2	B, BC		1/8				1/8		3/4
*{–, –, ABC, –}	3	C, ABC				3/8			3/8	1/4
*{–, –, C, AB}	4	$\emptyset$								
		$\theta_{\boldsymbol{b}}$	1/4	1/8	0	3/8	0	1/8	3/8	

Step 10, $t = 36\frac{5}{6}$		Prices:	8	7.25	16	8.25	16	16.25	24.25	
$\boldsymbol{f} \mid V(\boldsymbol{f}) = 24.25$	Agent	$\hat{D}_i$	A	B	AB	C	AC	BC	ABC	pass
*{A, BC, –, –}	1	A, AB, AC	0		0		0			1
*{A, –, BC, –}	2	B, BC		1/4				1/4		1/2
*{–, –, ABC, –}	3	C, BC, ABC				1/4		0	1/4	1/2
*{–, –, C, AB}	4	$\emptyset$								
*{AB, –, C,–}		$\theta_{\boldsymbol{b}}$	0	1/4	0	1/4	0	1/4	1/4	

Step 11, $t = 39\frac{5}{6}$		Prices:	8	8	16	9	16	17	25	

Agent 3 stops bidding and either {A, BC, –, –} or {A, B, C, –} wins.

Table 2
Some of the steps in the computation of the auction result for the example in Table 1. Asterisks indicate which of the potential CAs are determined to be competitive.

We then skip to step 7 where we see a complex allocation of attention across the bundles, yet one that continues to satisfy the constraints of the problem and exactly matches the simulation pattern. It is also interesting to note that the inflection at step 7 is caused by the allocation {A, BC, –, –} entering the CA set. The effect of this is clearly visible in Figure 2. At step 8, we see an interesting effect when BC enters Agent 4's demand set. The price of BC at the time is 15, so when Agent 4 joins the bidding, the allocation {A, –, –, BC} becomes competitive, as seen by the dramatic rise in its value. As a side effect, Agent 1 finds that it is winning often, and AB and AC are increasing too fast (a rate of 2/7) and are dropped from Agent 1's demand set. At step 9, we have our first agent drop out because no bundles provide positive surplus, and simultaneously we see the allocation {–, –, C, AB} become competitive. It is also interesting that the new competitive allocation includes Agent 4 even though it has stopped bidding. At step 11, Agent 3 drops out when it can no longer achieve positive surplus. At the same time, allocation {A, B, C, –} becomes competitive. The auctioneer then selects from among the only two remaining allocations: {A, B, C, –} or {A, BC, –, –}. Note that both Agent 1 and Agent 2—the only remaining active bidders—receive a bundle in the two candidate final allocations, and thus do not need to bid further. Further, the bidding policies ensure that the two active agents are indifferent between the two candidate allocations because they generate the same surplus at the final prices, respectively.

5 Discussion and Future Work

Aside from research on algorithms to solve singular instances of the WDP [1,3,13], there is little research that studies the computation over the life of an iterative combinatorial auction. One exception is the recent work by Kastner, et al. [5], in which the authors study the costs of maintaining previous solutions to the WDP as bidders incrementally increase their bids. While relevant to scenarios in which bidders have unconstrained bidding strategies, when proxy bidding is allowed, maintaining previous solutions to the WDP is a natural byproduct of the algorithm presented in this paper.

The solution-by-simulation approach has been investigated by Hoffman, et al. [4]. They present initial computational results for several methods of adjusting the size of the bid increment in order to reduce the number of iterations required to find the solution. The algorithm presented by Hoffman, et al. is heuristic in nature, and has routines for rolling back bids if it determines that the bid increment was too large (essentially moving the bidders past an inflection point). A direct comparison between the computational costs of Hoffman's approach and ours is a subject for future research. We expect our method to involve fewer, but more costly, computations.

We intend to continue with this line of research and to expand the framework to encompass the other auctions in the literature. For example, the SCPA presented in this paper differs from the Ausubel and Milgrom's APA mechanism [2] principally in the amount of attention agents are permitted to allocate. In APA, the bidders raise their offer on *all* elements of their best-response set, whereas in our mechanism the bidder randomly selects one element of its demand set on which to bid. We feel there is a strong correspondence between SCPA and A1BA [15], and posit that A1BA's distinctive method of price determination can be applied *ex-post*. We would like to try to represent at least the anonymous price variations of *i*Bundle [8] in the framework, and it may be possible to capture the discriminatory mechanisms.

We also plan to study the computational complexity of the process. We expect the MILP to be NP-complete, and at least as hard as the WDP. While we clearly benefit by solving for a limited number of inflection points, the value of the overall approach will depend upon how much harder it is to compute the allocation of attention than to solve the WDP. At the same time, we have made no effort yet to reduce the overhead of the algorithm, and our experience so far suggests that there are many ways in which to reduce the computational costs. For example, we described the interval computations as a comparison between all combinations of elements in a set with those not in the set. With some clever bookkeeping, a much smaller number of comparisons need to be made. There are also obvious ways in which a computer program could avoid creating some instances of the constraints in the MILP. Because these optimizations obfuscate the central ideas in the approach, we omitted them in this presentation.

Acknowledgments

This research was funded by NSF CAREER award 0092591 and the E-Commerce program NC State. The views and conclusions contained herein are those of the authors, as are any errors. We thank Ashish Sureka, who helped with the earlier version of this algorithm, David Parkes, for his encouragement and suggestions, and the attendees of the 2003 FCC Combinatorial Bidding Conference for their comments.

References

[1] A. Andersson, M. Tenhunen, and F. Ygge. Integer programming for combinatorial auction winner determination. In *Fourth International Conference on Multiagent Systems*, pages 39–46, 2000.

[2] L. M. Ausubel and P. R. Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1):1–42, 2002.

[3] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, 1999.

[4] K. Hoffman, D. Menon, and T. Wilson. Observations and near-direct implementations of the ascending proxy auction. December 2003.

[5] R. Kastner, C. Hsieh, M. Potkonjak, and M. Sarrafzadeh. On the sensitivity of incremental algorithms for combinatorial auctions. In *IEEE International Workshop on Advanced Issues of E-Commerce & Web-Based Information Systems (WECWIS)*, 2002.

[6] P. Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108(2):245–272, 2000.

[7] N. Nisan. Bidding and allocation in combinatorial auctions. In *Second ACM Conference on Electronic Commerce*, pages 1–12, 2000.

[8] D. C. Parkes. *i*Bundle: An efficient ascending price bundle auction. *First ACM Conference on Electronic Commerce*, pages 148–157, November 1999.

[9] D. C. Parkes. Iterative combinatorial auctions. In P. Crampton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. to appear, 2003.

[10] D. C. Parkes and L. H. Ungar. Preventing strategic manipulation in iterative auctions: Proxy agents and price-adjustment. In *Seventeenth National Conference on Artificial Intelligence*, pages 82–89, 2000.

[11] J. Rodríguez, P. Noriega, C. Sierra, and J. Padget. FM96.5 a java-based electronic auction house. In *Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '97)*, 1997.

[12] A. E. Roth and A. Ockenfels. Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the Internet. *American Economic Review*, 92(4):1093–1103, June 2002.

[13] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Cabob: A fast optimal algorithm for combinatorial auctions. In *International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001.

[14] P. R. Wurman, G. Cai, J. Zhong, and A. Sureka. An algorithm for computing the outcome of combinatorial auctions with proxy bidding. In *Fifth International Conference on Electronic Commerce*, pages 1–8, Pittsburgh, PA, 2003.

[15] P. R. Wurman and M. P. Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Second ACM Conference on Electronic Commerce*, pages 21–29, 2000.

[16] P. R. Wurman, M. P. Wellman, and W. E. Walsh. A parametrization of the auction design space. *Games and Economic Behavior*, 35(1-2):304–338, April-May 2001.

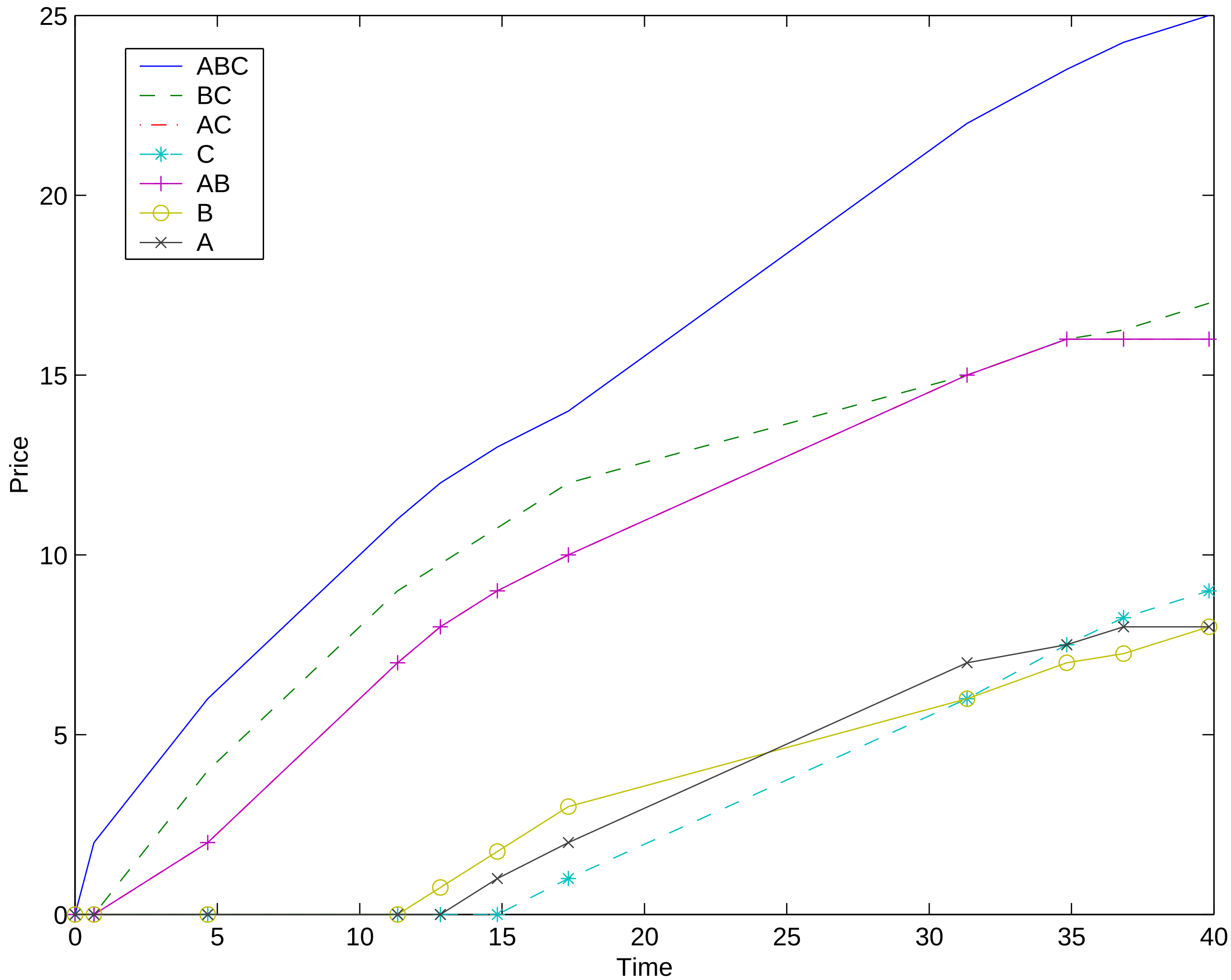

Fig. 1. The prices over time when proxy agents bid on the example in Table 1.

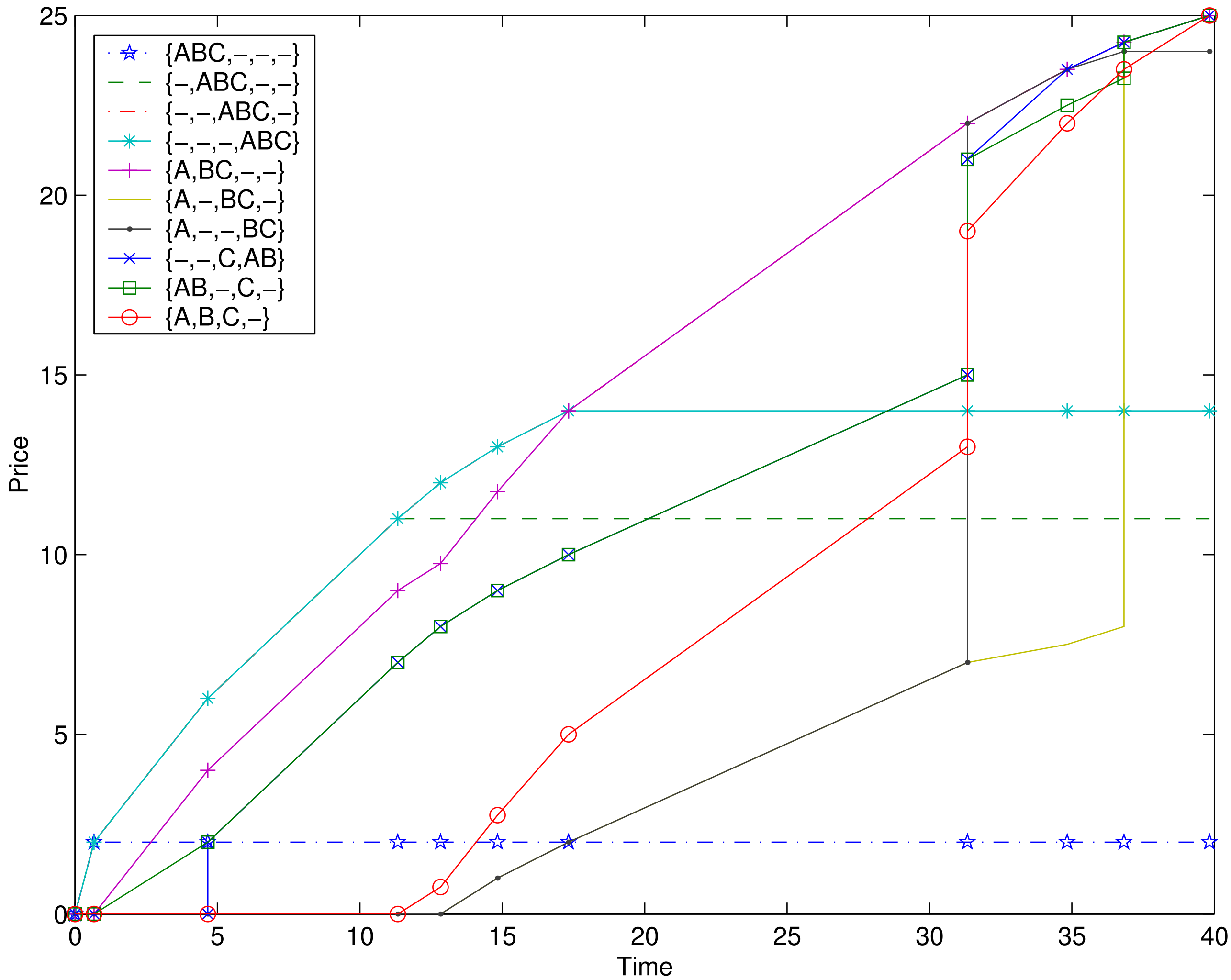

Fig. 2. The values of the competitive allocations for the example in Table 1.

$$
\begin{cases}
\max \sum_{\boldsymbol{f} \in \hat{F}^*} x_{\boldsymbol{f}} \\
\text{s.t.} \quad y_{i,\boldsymbol{b}} \geq \theta_{i,\boldsymbol{b}}, \quad \forall i \in \mathcal{I},\ \boldsymbol{b} \in \hat{D}_i \\
\quad \sum_{j \in \mathcal{I}} \theta_{j,\boldsymbol{b}} - \sum_{j \in \mathcal{I}} \theta_{j,\boldsymbol{c}} + N y_{i,\boldsymbol{b}} \leq N, \quad \forall i \in \mathcal{I}, \text{and } \boldsymbol{b}, \boldsymbol{c} \in \hat{D}_i \\
\quad y_{i,\boldsymbol{b}} + N(\sum_{j \in \mathcal{I}} \theta_{j,\boldsymbol{b}} - \sum_{j \in \mathcal{I}} \theta_{j,\boldsymbol{c}}) - N^2 y_{i,\boldsymbol{c}} \geq 1 - N^2, \quad \forall i \in \mathcal{I}, \text{and } \boldsymbol{b}, \boldsymbol{c} \in \hat{D}_i \\
\quad \alpha_{i,\boldsymbol{b}} \leq N y_{i,\boldsymbol{b}}, \quad \forall i \in \mathcal{I},\ \boldsymbol{b} \in \hat{D}_i \\
\quad \alpha_{i,\boldsymbol{b}} - \sum_{j \in \mathcal{I}} \theta_{j,\boldsymbol{b}} + N y_{i,\boldsymbol{b}} \leq N, \quad \forall i \in \mathcal{I},\ \boldsymbol{b} \in \hat{D}_i \\
\quad \sum_{j \in \mathcal{I}} \theta_{j,\boldsymbol{b}} - \alpha_{i,\boldsymbol{b}} + N y_{i,\boldsymbol{b}} \leq N, \quad \forall i \in \mathcal{I},\ \boldsymbol{b} \in \hat{D}_i \\
\quad \sum_{i \in \hat{\boldsymbol{f}}} \alpha_{i,\hat{\boldsymbol{f}}_i} - \sum_{i \in \boldsymbol{f}} \alpha_{i,\boldsymbol{f}_i} + N x_{\boldsymbol{f}} \leq N, \quad \forall \boldsymbol{f}, \hat{\boldsymbol{f}} \in \hat{F}^* \\
\quad \beta_f \leq x_{\boldsymbol{f}}, \quad \forall \boldsymbol{f} \in \hat{F}^* \\
\quad \theta_{i,\text{pass}} + \sum_{\boldsymbol{f} \in \hat{F}^*} (1 - K_i G_{\boldsymbol{f},i}) \beta_{\boldsymbol{f}} = 1, \quad \forall i \in \mathcal{I} \\
\quad \theta_{i,\text{pass}} + \sum_{\boldsymbol{b} \in B} \theta_{i,\boldsymbol{b}} = K_i, \quad \forall i \in \mathcal{I} \\
\quad \sum_{\boldsymbol{f} \in \hat{F}^*} \beta_{\boldsymbol{f}} = 1 \\
\quad \theta_{i,\text{pass}} \geq 0, \forall i \in \mathcal{I} \\
\quad \beta_{\boldsymbol{f}} \geq 0, \quad x_{\boldsymbol{f}} \in \{0,1\}, \quad \forall \boldsymbol{f} \in \hat{F}^* \\
\quad \theta_{i,\boldsymbol{b}} \geq 0, \quad \alpha_{i,\boldsymbol{b}} \geq 0, \quad y_{i,\boldsymbol{b}} \in \{0,1\}, \quad \forall i \in \mathcal{I},\ \boldsymbol{b} \in \hat{D}_i
\end{cases}
$$

Fig. 3. The complete mixed-integer program for computing the allocation of attention in the proxy auction.