## ABSTRACT

ZHONG, JIE. A Price Trajectory Algorithm for Solving Iterative Auction Problems. (Under the direction of Associate Professor Peter R. Wurman.)

Many types of auctions are discussed in the literature such as single item auctions, sequential auctions, and combinatorial auctions. Proxy bidding has proven useful in solving iterative auction problems in many real-world auction formats. In this dissertation, I propose a new type of iterative auction called the Simple Combinatorial Proxy Auction.

A popular method for solving the iterative proxy auction problems is simulating the incremental bidding decisions of the agents. However, this approach has some disadvantages. In this dissertation, I present a new approach called the Price Trajectory Algorithm to solve iterative auction problems. This approach computes the agents' allocation of their attention across the bundles only at "inflection points" – the points at which agents change their behavior. The proposed algorithm tracks the behavior of agents and the competitive allocations of items in order to establish a connection between them. With the allocation of agents' attention, one can compute the slopes of price curves to get the bundle prices and speed up the computation by jumping from one inflection point to the next.

The price trajectory algorithm can be applied to the Ascending Package Auction, the Ascending $k$-Bundle Auction, and the Simple Combinatorial Proxy Auction. The price trajectory algorithm has several advantages over other alternatives: (1) The price trajectory algorithm computes exact solutions. (2) The solutions are independent of the bid increment or tie-breaking rules. (3) The solutions are invariant to the magnitude of the bids.

To ensure security, I present a cryptographic protocol for the price trajectory algorithm. The cryptographic protocol guarantees that only the auctioneer obtains the correct and necessary information from the agents.

## Biography

ZHONG, JIE, was born in 1976 in Sichuan, China. He attended the Department of Mathematical Science at Nankai University, Tianjin, China, in 1994, and received his Bachelor's Degree in Computational Mathematics in 1998. From 1998 to 2001, Jie studied at the Department of Mathematical Science at Tsinghua University, Beijing, China, and earned his Master's Degree in Operations Research. In Fall 2001, Jie moved to the United States of America, and enrolled in the Ph.D. program of Operations Research and Computer Science at North Carolina State University, Raleigh, NC. In October, 2004, Jie began working at SAS Institute, Cary, NC, as a senior developer of price optimization.

## Acknowledgements

I want to express my deep appreciation and gratitude to my advisor Dr. Peter R. Wurman, for his invaluable advice and guidance that extended far beyond technical assistance. I am fortunate and happy to have had the opportunity to work in Dr. Wurman's Intelligent Commerce Research Group. Also, I thank Dr. Shu-Cherng Fang, Dr. Yahya Fathi, and Dr. Carla D. Savage for their willingness to be on my advisory committee, and for providing useful suggestions, enlightening discussions, and contributing to my success. In particular, I am very grateful to Dr. Fang, who helped me with my research, and also offered many helpful suggestions for my personal and professional life. I am very thankful to Dr. Xiuli Chao for his help and encouragement on both my research and personal life. Without his help and confidence in me, I would not have achieved my goals.

I would like to thank the Graduate Program in Operations Research and Computer Science of North Carolina State University for supporting me during these years. I was fortunate enough to have made many nice and helpful friends at North Carolina State University and SAS Institute, namely, Carole Beam, Gangshu Cai, Hao Cheng, Yue Dai, Xiaoli Ling, Jim Sheedy, Raj Solanki, Ashishi Sureka, Yan Xu, Yong Wang, Qing Zhang, Wei Zhang, and Xiang Zhou. I thank all of them for their friendship.

Finally, I thank my parents for their love, support, and confidence in me, and I am forever indebted to my wife, Yaxing Liu, who unconditionally loves and supports me. She unceasingly encouraged and motivated me to complete my Ph.D. studies. She is a constant companion who always stays with me to overcome whatever difficulties that I may face in all my endeavors and in my daily life.

# Contents

# List of Tables

# List of Figures

# List of Symbols and Abbreviations

| | |
|---|---|
| $I$ : | Set of all agents (bidders). |
| $B$ : | Set of all bundles. |
| $v_{ib}$ : | Agent $i$'s valuation on bundle $b$. |
| $r_{ib}$ : | Agent $i$'s bid price on bundle $b$. |
| $\theta_{ib}$ : | Agent $i$'s bid attention on bundle $b$. |
| $\theta_b$ : | The slope of bundle $b$. |
| $s_i$ : | Agent $i$'s surplus. |
| $\pi_b$ : | Price of bundle $b$. |
| $D_i$ : | Agent $i$'s demand set. |
| $\hat{D}_i$ : | Agent $i$'s potential demand set. |
| $F^*$ : | The set of competitive allocations. |
| $\hat{F}^*$ : | The set of potential competitive allocations. |
| $f^*$ : | A competitive allocation. |
| $V(f^*)$ : | Value of competitive allocation $f^*$. |
| $I_{f^*}$ : | Set of winning agents in a competitive allocation $f^*$. |
| $B_{f^*}$ : | Set of allocated bundles in a competitive allocation $f^*$. |
| $f_i^*$ : | The bundle that is assigned to agent $i$ in competitive allocation $f^*$. |
| $i_b$ : | The agent who receives bundle $b$ in competitive allocation $f^*$. |
| $\beta_f$ : | The frequency of allocation $f$ being announced as competitive allocation. |

| | |
|---|---|
| WDP : | The winner determination problem. |
| VCG : | The Vickrey-Clarke-Groves auction. |
| APA : | The ascending package auction. |
| A$k$BA : | The ascending $k$-bundle auction. |
| SCPA : | The simple combinatorial proxy auction. |
| PTA : | The price trajectory algorithm. |
| AAM : | The attention allocation method. |
| IPM : | The inflection point method. |

# Chapter 1

# Introduction

An auction is the process of buying and selling things by offering them up for bids, taking bids, and then selling the items to the bidder who submitted the highest bid price. While the date of the first auction is not known, it is clear that auctions have been around for a long time. Now, more and more companies use auctions as an important channel for marketing their products. Millions of people shop at Internet auction sites such as eBay, Priceline.com, and Yahoo Auctions. Specifically, eBay reported record consolidated Q3-04 net revenues of 805.9 million dollars, up 52% year over year [16].

Conventional auctions usually let the bidders bid for just one item with one price. The English Auction [21], used on sites like eBay, is probably the most common type. Bidders start at a reserve price provided by the auctioneer, and increase their bid to either the highest price that they are willing to pay for an item or the winning price. Bidding activity stops when the auctioneer declares the auction complete. The item is sold to the highest bidder at his bid price. This method of determining a winner and a payment is called the pay-your-bid auction. The Sealed-bid First-Price Auction, also called the Simultaneous

Auction [22], requires that all bidders simultaneously submit bids so that no bidder knows the bid of any other participant. The bidder with the highest bid wins and pays the bid price. The Vickrey Auction [39], a second-price mechanism, is also designed for selling a single item. The highest bidder obtains the item at the second highest price. The advantage of the Vickrey Auction is that bidders are motivated to bid what they think the item is worth without worrying what others will bid. Thus, bidders in a Vickrey Auction strive to bid an item's value honestly. In the traditional Dutch Auction [21], the auctioneer begins with a high asking price that is lowered until a participant is willing to accept the auctioneer's price, or a predetermined minimum price is reached. The winning participant pays the last announced price.

Combinatorial auctions allow the bidders to better express their bids over heterogeneous items because they may not have linear valuations on combinations of items [3, 10, 27]. For example, if two bicycle wheels and one bicycle frame are sold separately in two auctions, a bidder may value two wheels or a single frame at 0 dollars, but may value the combination of two wheels and one frame at 200 dollars. If forced to purchase each component in separate auctions, the bidder has a dilemma: bidding enough to win the components that are sold first may results in a financial loss if he fails to win the components that are sold later. This dilemma can be overcome by selling all goods simultaneously and allowing buyers to submit bids on bundles (combinations of items). Such combinatorial bids are offers to pay a certain amount only if all units are awarded, but pay nothing otherwise.

Combinatorial auctions could be used in many situations. For example, radio spectrum licenses can cover an entire nation or be split among smaller areas. The Federal Communications Commission (FCC) will use combinatorial auctions to sell spectrum licenses in the Upper 700 MHz band. In FCC Auction No. 31, the FCC will permit bids for any of the 4095 possible packages of the twelve licenses on offer. Combinatorial auctions could also be useful for asset sales auctions, which often accept bids on the combinations of pieces such as the house and barn, the arable land, other land, and water rights. A variety of industries have employed combinatorial auctions. For instance, they have been used for truckload transportation, bus routes, and procurement in electronic commerce between business and suppliers, and have been proposed for airport arrival and departure slots at airport gates across self-interested airlines [28].

Combinatorial auctions have two inherent difficulties:

1. Determining who the winning bidders are and which bundles they won, and

2. Determining the winning bidders' payments.

For single item auctions, these two problems do not cause computational difficulty. As discussed above, the item is always assigned to the bidder who offers the highest bid price. The payment varies in different auction implementations, but the computation is pretty straightforward.

In contrast, the computation is usually complex for efficiently allocating multiple items to bidders that maximizes the value of assigned combinations of items. This is known as the Winner Determination Problem (WDP) and was shown to be NP-hard by Rothkopf [34], although tractable special cases exist and some specialized algorithms have been developed [12, 36, 38]. Auctioneers need only pick the highest bidder as winner in single item auctions, but they must solve a mixed integer linear program [1, 23] to determine the efficient allocation in combinatorial auctions, due to the exponential number of combinations of items.

The most famous mechanism in combinatorial auction is the Vickrey-Clarke-Groves

(VCG) mechanism, named for the pioneering work of Vickrey [39], Clarke [8] and Groves [13]. As a second-price mechanism, it encourages bidders to bid bundles' value honestly. In the VCG mechanism, bidders submit their valuations on all combinations of items to the auctioneer, and the auctioneer computes the socially efficient allocation. Each winner receives a discount from his actual bid that is computed by removing his offers from the auction. In other words, each winner's payment is equal to the difference in everyone else's value in the efficient allocation with all bidders and that of the efficient allocation when he is absent.

Although the VCG mechanism has several desirable properties and is widely discussed in the combinatorial auction literature, it is not common in practice. The pay-your-bid (or first-price) auction is the most obvious alternative mechanism. However, it also has disadvantages. In a sealed-bid auction, the first-price payment rule encourages the bidders to submit bids that are just barely enough to achieve the efficient allocation. But if a bidder is trying to predict the minimum amount needed to win his bundle, he may under bid because he knows little about the other bids. Thus, using this payment rule, the auction may fail to achieve an efficient outcome. These problems have led researchers to look to iterative combinatorial auctions.

Iterative versions of auctions are attractive for several reasons, in particular because they allow bidders to base bidding decisions on approximate value information and postpone the computation of exact values until it becomes clear which items are relevant to the final allocation. In each iteration, an allocation based on the current bids is computed and then announced so bidders can assimilate those computations into their bids in the next iteration. The iterative auction ends when there is no further competition among bidders and the competitive allocation in the last iteration is the final allocation. In other words, iterative auctions allow bidders to revise combinational bids as the final allocation and payments evolve.

In iterative combinatorial auctions, it is possible to determine an efficient allocation without bidders reporting, or even determining, exact values for all combinations of items; it is also possible that the efficient allocation will never be reached. In contrast, any efficient sealed-bid auction requires bidders to determine and report their value for all possible combinations of items. Basically, the main advantage of iterative auctions is that they only require bidders to determine exact values on items, or combinations of items, if those items are relevant to the final allocation. Moreover, they protect the confidentiality of the bidders' private value because bidders only need to submit partial and essential information about their valuation of all combinations of items.

Several researchers have worked on iterative combinatorial auctions with ascending bids in which the bids start from 0 and increase in response to feedback from the auctioneer. Ausubel and Milgrom [2] developed the Ascending Package Auction (APA) and have shown that a semi-sincere equilibrium always exists and the final allocation is in the core. The Ascending $k$-Bundle Auction (A$k$BA) is another family of iterative auctions presented by Wurman and Wellman [41]. The price technique of A$k$BA leads to a range of equilibrium price lattices. Parkes et al., [24, 26] described $i$Bundle, yet another ascending-price iterative combinatorial auction. There are three basic variations of $i$Bundle: $i$Bundle(2) assigns anonymous prices to bundles for all bidders and every bidder sees the same price; $i$Bundle(3) generates discriminatory prices for bundles and different agents may see different prices for the same bundle. The third variation $i$Bundle(d) dynamically switches from $i$Bundle(2) to $i$Bundle(3) to support efficient allocations. $i$Bundle(d) is guaranteed to compute the optimal bundle allocations to bidders who follow a best-response bidding strategy.

Recently, mechanism designers have shown an interest in proxy bidding [40, 43] for iterative combinatorial auctions. In proxy bidding, the bidder expresses a value statement to a software agent that then follows a prescribed bidding strategy to bid incrementally on behalf of the bidder until it either wins a bundle or exhausts the authority granted to it.

Proxy bidding has the advantage of accelerating the auction by allowing bidders to place larger bids that are executed only to the extent necessary to outbid competitors [2, 26, 31]. By restricting the strategic flexibility of the bidders, mechanism designers may be better able to design successful auctions and predict their outcomes. In addition, allowing proxy bidding may reduce the need for 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 effectively becomes a Vickrey Auction and each bidder's equilibrium strategy is to submit his true value [32].

A natural method for using proxy bidding to solve the problems of iterative combinatorial auctions is to simulate a bidding process that progresses step by step. Although simple, this method has disadvantages. First, the outcome is dependent upon implementation details, such as the tie breaking rule and the bid increment. Second, the accuracy of the outcome is a function of the bid increment. We can improve the method by decreasing the bid increment, but doing so increases the number of iterations and the amount of time that the process takes because each iteration requires the auctioneer to solve an NP-complete WDP. Finally, the running time of the outcome is sensitive to the magnitude of values, to the ordering of agents, and to the tie-breaking rules.

In this dissertation, a new auction called the Simple Combinatorial Proxy Auction (SCPA) is presented where "simple" reflects the fact that defining bundle prices is straightforward. SCPA actually gives the same final allocations as A1BA does. Therefore, one can easily get the final allocation of A1BA by running SCPA. The major contribution of the dissertation is that an algorithm is introduced to solve iterative combinatorial auctions with proxy bidding that can be applied to SCPA, A$k$BA, APA, and possibly others. This approach has several advantages over alternatives; in particular, it computes exact solutions that are independent of the bid increment or tie-breaking rules and are invariant to the magnitude of the bids.

In Chapter 2 of this dissertation, current literature is reviewed and the main features of combinatorial auctions are introduced. Common notations and definitions are presented, and generalized Vickrey Auctions are examined in detail. Iterative auctions are also examined, such as APA, A$k$BA, and $i$Bundle. The chapter ends with a discussion of the winner determination problem.

In Chapter 3, the simple combinatorial proxy auction is presented. The auction rules are defined and the bidding policies are discussed. The most natural algorithm, simulation, is then used to solve the SCPA problem. An example is introduced to demonstrate the agents's behavior in bundling price patterns in SCPA simulation rounds. The chapter ends with a proof that SCPA generates the same allocation as A$k$BA and a demonstration of SCPA's strengths through several examples.

In Chapter 4, an algorithm called the Price Trajectory Algorithm (PTA) is proposed to solve iterative auction problems. This approach includes two parts. The first, called the Attention Allocation Method, computes the bidders' allocation of their attention across the bundles, which leads to the slopes of all bundle prices. The second part, called the Inflection Point Method, determines the time duration needed for the bundle prices to increase with the slopes of the first method. Using SCPA as an example, the chapter presents a mixed integer linear program to compute the allocation of attention. From the auction rules and bidding policies, the next inflection point is calculated. An example, with details, is used to illustrate how the PTA works. Finally, multi-stage proxy auctions are introduced.

In the fifth chapter, the correctness of PTA is discussed and a theorem states that the simulation result is a feasible solution to the attention allocation model. From the complexity point of view, PTA is an NP-hard algorithm. But in some special situations, the mixed integer linear program can be simplified to a linear program. Next, some computational results are given. Finally, the comparison between PTA and other approaches is presented.

In Chapter 6, PTA is applied to APA problems. The mathematical model is constructed from the auction rules and bidding policies. Compared to SCPA's implementation, it has fewer binary variables in constraints. An example is used to illustrate to processes. Finally, computational results are given to show PTA's advantages in APA.

A cryptographic protocol is presented to preserve information of PTA in Chapter 7. The millionaire problem is introduced in the first section. The second section describes the protocol. Using this protocol, an agent could protect its private data and reveal only the information necessary for the auctioneer to successfully run the auction. Fraud detection is discussed in the last section.

In the last chapter, Chapter 8, a review of the new iterative proxy auction and the PTA algorithm is summarized. Some future work related to the dissertation is presented.

# Chapter 2

# Literature Review

## 2.1 Preliminaries

Let $G$ denote the set of $n$ items, and $I = \{1, \cdots, m\}$ be the set of all $m$ bidders (proxy agents). Without ambiguity, let $\bar{I} = \{0, 1, \cdots, m\}$, where the auctioneer is presented as "bidder" 0. A combination of items is called a *bundle*. Let $B$ be the set of $2^n - 1$ combinations of $n$ different items, excluding the empty set. Denote bidder $i$'s *valuation* of bundle $b$ as $v_{ib}$. Since the bidders are free to place bids on combination of items, rather than just on individual items, this kind of auction is called a combinatorial auction.

In a combinatorial auction, the auctioneer assigns the bundles to bidders. The assignment of bundles to bidders is called an *allocation*. Denote the set of all possible allocations $F$. In an allocation, a bidder may either obtain one bundle or get nothing. Denote $I_f$ as the set of the winning bidders who obtain a bundle, and $B_f$ as the set of the allocated bundles that are assigned to bidders in an allocation $f$. In this dissertation, an allocation, $f$, is represented by an ordered set where the lexicographical position corresponds to the bidder and the value corresponds to the bundle. $i \in f$ means that bidder $i$ gets a bundle, $f_i$, in allocation $f$, and $i \notin f$ means bidder $i$ receives nothing in allocation $f$. For example, there are four bidders, denoted {1, 2, 3, 4}, and three items denoted {A, B, C} in a combinatorial auction. One possible allocation, $f$, is that Bidder 1 obtains bundle B, Bidder 3 receives bundle AC, and Bidders 2 and 4 get nothing. Therefore, $f$={B, –, AC, –}, $1 \in f, f_1 = B, 3 \in f, f_3 = AC$ and $2 \notin f, 4 \notin f$.

Denote $r_{ib}$ as the *bid* of bidder $i$ on bundle $b$. The *value* of an allocation $f$ is computed as $V(f) = \sum_{i \in f} r_{if_i}$. In order to get the *competitive allocation*, $f^*$, the auctioneer needs to solve the winner determination problem (details in Section 2.4). The competitive allocation maximizes the allocation value based on the bidders' bids while the *efficient allocation* is computed from the bidders' valuations.

Denote $\pi_b$ as the *price*[1] associated with bundle $b$. Based on the bundle price, bidder $i$'s *surplus* $s_{ib}$ on bundle $b$ is calculated as $s_{ib} = v_{ib} - \pi_b$ if $v_{ib} > \pi_b$, $s_{ib} = 0$ otherwise.

We define bidder $i$'s best response set, called his *demand set* $D_i$, as

$$D_i = \left\{ b \in B : s_{ib} > 0,\ s_{ib} \geq s_{ib'}\ \forall b' \in B \right\}. \tag{2.1}$$

It means that the bundles with maximal positive surplus are in the bidder's demand set. This dissertation assumes that bidders use a best response strategy in auctions, which is a straightforward bidding policy.

## 2.2 Generalized Vickrey Auctions

The Vickrey Auction [39] is a type of sealed-bid auction where the highest bidder wins, but pays the price of the second highest bidder. In an independent-values setting, such

[1]Different auctions may compute $\pi_b$ in different ways, or not at all. For example, there is no bundle price defined in Ascending Package Auction. More details are in Section 2.3.2.

sealed-bid, second-price processes have several desirable properties. First, the equilibrium strategy is that the bidder bids his true value. Second, at equilibrium the auction always leads to economic efficiency, in which the bidder with the highest value always wins.

Vickrey's original paper considered only auctions where a single, indivisible item is being sold. When multiple identical items are being sold in a single auction, the most obvious generalization is to have all bidders pay the amount of the highest, non-winning bid. This is known as a uniform-price auction [9]. However, the uniform-price auction does not result in bidders bidding their true valuations as they do in a second-price auction except for those who have demand for a single unit.

The Vickrey Auction is well studied in economic literature [4, 5], but is not particularly common in practice [35]. The most obvious reason is that it does not maximize the auctioneer's revenue. The fear that the auctioneer might cheat and the lack of privacy of the valuation are also explanations. If there are multiple items that any bidder wishes to bid for, the advantages may be overcome. There are some additional reasons, such as bidder asymmetry, non-independent values, and bidder risk aversion.

The Generalized Vickrey Auction (GVA) is one instance of the Vickrey-Clarke-Groves (VCG) mechanism [8, 13, 39]. The GVA can also handle combinatorial auctions and has good theoretical results such as incentive compatibility and Pareto efficiency [2, 33]. The VCG payment of bidder $i$ is computed as

$$p_i = v_{if_i^*} - (V(f^*) - V(f^{-i*})),$$

where $f^*$ is the efficient allocation and $f^{-i*}$ is the efficient allocation based on the valuation of all bidders except for bidder $i$. Ausubel and Milgrom [2] proved that the VCG allocation $f^*$ is buyer Pareto-dominant if the VCG payment is in the core. Also, the VCG payment

Table 2.1: Valuations of four buyers on the combinations of three 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 |

is in the core if and only if the core contains the buyer Pareto-dominant allocation.

The following example illustrates the VCG payment computation. Suppose four bidders have valuations for three items A, B, and C, as shown in Table 2.1. Assume all bidders submit their bids with the true values shown in the above table. The auctioneer solves the winner determination problem and gets the efficient allocation: $f^*$={A, BC, –, –}[2]. The value of $f$ is $V(f^*) = \$28$. Bidder 1's payment is computed as follows. Assume Bidder 1 did not submit his valuation. Then the efficient allocation, $f^{-1*}$, becomes ={–, B, C, A} or {–, BC, –, A} or {–, B, AC, –} or {–, B, –, AC} or {–, –, C, AB} or {–, –, ABC, –}. The value of $f^{-1*}$, $V(f^{-1*})$ is \$25. So, Bidder 1's payment is $p_1 = v_{1A} - (V(f^*) - V(f^{-1*})) = 10 - (28 - 25) = \$7$. Similarly, assume Bidder 2 did not submit his valuation. Then the efficient allocation, $f^{-2*}$, becomes {A, –, BC, –} or {AB, –, C, –}. The value of $f^{-2*}$ is $V(f^{-2*}) = \$27$. So, Bidder 2's payment is $p_2 = v_{2BC} - (V(f^*) - V(f^{-2*})) = 18 - (28 - 27) = \$17$. Because Bidders 3 and 4 receive nothing, their payment is \$0.

## 2.3 Iterative Ascending Auctions

The VCG auction is not desirable in practice for several reasons. First, it does not maximize the auctioneer's profit. Second, the auctioneer may use shill[3] bids to increase her

[2] $f^*$={A, B, C, –} is another efficient allocation.

[3] A shill is an associate of the auctioneer who pretends no association to the auctioneer and assumes the air of an enthusiastic customer.

profit. Third, the auctioneer's profit is not monotonic with regard to the bidders' valuation. Thus, the pay-your-bid (or first-price) auction is an obvious alternative mechanism, though it, too, has disadvantages. In a sealed-bid auction, the first-price payment rule encourages the bidders to submit bids that just barely achieve the efficient allocation. But if a player is trying to predict the minimum amount needed to win his bundle, he may not bid enough to win due to his incomplete information about the bids of others. Thus, the procedure is not guaranteed to find the efficient allocation. Another disadvantage is that the sealed-bid auction requires complete and exact computations.

Iterative auctions [29] with proxy bidding can avoid the disadvantages of bidding more than the minimum amount necessary to win a particular bundle if the proxy bids are based on a relatively small increment.

The following steps represent the basic framework associated with solving the iterative auction problems by simulating the bidding that would occur if the bidders follow simple strategies.

**Step 0.** The auctioneer initializes the bidding process.

**Step 1.** The auctioneer solves the competitive allocation and bundle prices (if the prices exist) and announces them to all agents.

**Step 2.** Each agent follows simple bidding policies to bid or pass.

**Step 3.** The auctioneer goes to Step 1 if there are still some agents bidding, or stops otherwise.

The above procedure is a general framework for solving iterative combinatorial auctions by simulated bidding. Different auctions have different implementation details.

For example, an agent will bid on all bundles in its demand set in an Ascending Package Auction (APA), but will only bid on one bundle in its demand set in Ascending $k$-Bundle Auctions (A$k$BA). There are no bundle prices in APA, but the auctioneer will compute and announce the prices in each iterative round in A$k$BA.

### 2.3.1 Ascending *k*-Bundle Auctions

The Ascending $k$-Bundle Auction (A$k$BA), presented by Wurman and Wellman [42], is a family of progressive auctions that use equilibrium bundle prices. It requires that a bidder either passes or improves his bid on exactly one bundle by increasing the current price with a small increment, $\delta$. After the bids are received, the auctioneer solves the winner determination problem and announces a potential allocation. The auctioneer will solve two linear programs to get the bundle prices. The announced prices are anonymous, i.e., all bidders are given the same information.

In the proxy version of the A$k$BA, we assume that proxy agents submit incremental bids on behalf of the bidder by following a straightforward bidding policy [22]. Agent $i$ bids on an element of its demand set,

$$D_i = \left\{ b \in B : s_{ib} > 0,\ s_{ib} \geq s_{ib'}\ \forall b' \in B \right\}.$$

That is, if an agent is told by the auctioneer that it is winning, the agent does not increase its bid. Otherwise, the agent will bid on the bundle that maximizes its surplus at the given prices.

The procedure of solving A$k$BA by simulation with straightforward bidding is

**Step 0.** Initialization. $n = 0, r_{ib}^{n} = 0$ for $i \in I, b \in B$, increment $\delta > 0$.

**Step 1.** The auctioneer solves the winner determination problem, i.e., computes the competitive allocation $f^*$ based on $r_{ib}^n$ and announces $f^*$ to all agents.

**Step 2.** The auctioneer computes the bundle prices $\pi^n$.

**Step 3.** Each agent $i = 1, 2, \cdots, m$ computes its demand set $D_i^n$.

If $i \notin f_i^*$ and $D_i^n \neq \emptyset$, then agent $i$ will increase its bid on one bundle, $b_0$, in its demand set. That is,

$$r_{ib}^{n+1} \leftarrow r_{ib}^n, \quad \forall b \in B \text{ and } b \neq b_0,$$

and

$$r_{ib_0}^{n+1} \leftarrow \pi_{b_0}^n + \delta.$$

Otherwise, it will pass,

$$r_{ib}^{n+1} \leftarrow r_{ib}^n, \quad \forall b \in B.$$

**Step 4.** The auctioneer checks for termination.

If some agents increased bids in Step 3, $n \leftarrow n + 1$ and go to Step 1.

Otherwise, the auction stops. The current competitive allocation $f^*$ is the final allocation and the winning agents' payments are the prices of the bundles that they win.

In order to get equilibrium prices, two linear programs used by Leonard [20] are introduced in Step 2. Let $f^*$ be a competitive allocation, $s_i$ be agent $i$'s surplus and $\pi_b$ be the price of allocated bundle $b$ in the competitive allocation $f^*$.

$$\text{LP}_{\text{lower}} \begin{cases} \min \quad h(\pi) = \sum_{b \in B_{f^*}} \pi_b & (2.2) \\ \text{s.t.} \quad s_i + \pi_b \geq r_{ib}, \quad \forall i \in I, b \in B_{f^*} & (2.3) \\ s_i, \pi_b \geq 0, \qquad \forall i \in I, b \in B_{f^*} & (2.4) \\ \sum_{i \in I} s_i + \sum_{b \in B_{f^*}} \pi_b = V(f^*) & (2.5) \end{cases}$$

$$\text{LP}_{\text{upper}} \begin{cases} \min \quad g(s) = \sum_{i \in I} s_i & (2.6) \\ \text{s.t.} \quad s_i + \pi_b \geq r_{ib}, \quad \forall i \in I, b \in B_{f^*} & (2.7) \\ s_i, \pi_b \geq 0, \qquad \forall i \in I, b \in B_{f^*} & (2.8) \\ \sum_{i \in I} s_i + \sum_{b \in B_{f^*}} \pi_b = V(f^*) & (2.9) \end{cases}$$

$\text{LP}_{\text{lower}}$ maximizes each agent's surplus with respect to the bids and $\text{LP}_{\text{upper}}$ minimizes each agent's surplus with respect to the bids, within the range of equilibrium prices that support the optimal solution to the allocation $f$ [41]. Suppose $(\bar{s}_i^*, \underline{\pi}_g^*)$ and $(\underline{s}_i^*, \bar{\pi}_g^*)$ to be the optimal solutions of $\text{LP}_{\text{lower}}$ and $\text{LP}_{\text{upper}}$, respectively. Then we have the prices as follows

$$\pi_b = k\bar{\pi}_b^* + (1-k)\underline{\pi}_b^*, \quad \forall b \in B_{f^*} \text{ and } k \in [0,1]. \tag{2.10}$$

$$\pi_b = k \max_i \{r_{ib} - \underline{s}_i^*\} + (1-k) \max_i \{r_{ib} - \bar{s}_i^*\}, \quad \forall b \notin B_{f^*} \text{ and } k \in [0,1]. \tag{2.11}$$

By referring to auctions that set prices by the above expression as *k-bundle auctions*, Wurman and Wellman presented the Ascending $k$-Bundle Auction (A$k$BA). They proved that A$k$BA uses competitive equilibrium bundle prices that support the efficient allocation under the free disposal condition. In their paper, they examined a particular instance of the A$k$BA family, called A1BA, and presented some empirical data on its performance.

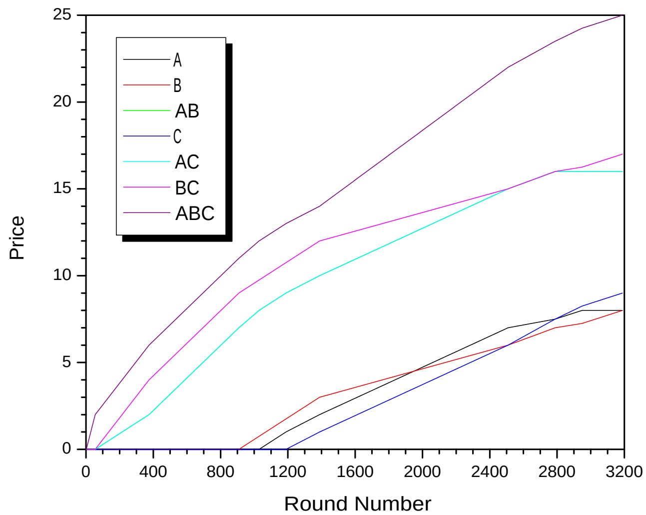


Figure 2.1: A$k$BA bundle prices of solving the example in Table 2.1 by simulation.

Figure 2.1 shows the bundle prices of A1BA increase when the bidding rounds move given the buyer valuations shown in Table 2.1. The x-axis indicates the round number and the y-axis indicates the bundle price. The result is from the simulation method with an increment $\delta = 0.01$. At the beginning of the processes, all agents are bidding bundle ABC because this bundle gives them maximal surplus. After the price of ABC reaches 2, bundles AB and AC also give Agent 1 the maximal surplus and bundle BC gives Agent 2 the maximal surplus. Thus, they start to bid on AB, AC, and BC. This event leads an inflection on the price curve of bundle ABC. From the figure, we can observe that the bundle prices increase at a steady rate for a large number of rounds. The final allocation is $\{$A, BC, –, –$\}$ and Agent 1's payment is 8 and Agent 2's payment is 16.99.

### 2.3.2 Ascending Package Auctions

Ausubel and Milgrom [2] presented a type of iterative combinatorial auction called the Ascending Package Auction (APA) in which bidders may determine their own packages on which to bid. There are no bundle prices in APA. With the proxy bidding, the outcome is socially efficient and in the core for the bidders' valuations. They proved that bidders submitting true values leads to a Nash Equilibrium and the outcome coincides with the Vickrey Auction outcome when payments are linear and items are substitutes. Compared to the Vickrey Auction, APA has some significant advantages. It generates higher equilibrium prices, is less vulnerable to shill bidding and collusion by coalitions of losing bidders, and can handle budget constraints more robustly.

In the APA bidding processes, an agent either passes or bids on some bundles. The auction does not allow the winning agents in one round to bid in the next. In other words, the winning agents must hold their bids in the next round. To simplify the agents' bidding strategy, we assume each proxy agent bids straightforwardly at each round. That is, the agent's demand set is defined as

$$D_i^n = \left\{ b \in B : v_{ib} > r_{ib} \text{ and } v_{ib} - r_{ib} \geq v_{ib'} - r_{ib'} \ \forall b' \in B \right\}. \tag{2.12}$$

An agent would increase its bid on all bundles in its demand set by $\delta$ if it was not winning in the previous round, but hold its bids otherwise. In each round, the auctioneer collects the bids from all agents and solves the WDP to get the competitive allocation. Then she announces the competitive allocation to all the agents.

Solving APA by simulation with best response bidding strategy involves the following steps:

**Step 0.** $n = 0, r_{ib}^n = 0, i \in I, b \in B$ and increment $\delta > 0$.

**Step 1.** The auctioneer solves the winner determination problem, i.e., computes the competitive allocation $f^*$ based on $r_{ib}^n$ and announces $f_i^*$ to agent $i$.

**Step 2.** If agent $i$ does not win a bundle and its demand set $D_i^n$ is not empty, it increases its bid on all bundles in $D_i^n$. That is,

$$r_{ib}^{n+1} \leftarrow r_{ib}^n + \delta, \quad \forall b \in D_i^n.$$

Otherwise, it passes,

$$r_{ib}^{n+1} \leftarrow r_{ib}^n, \quad \forall b \in D_i^n.$$

**Step 3.** The auctioneer updates the round counter, $n \leftarrow n + 1$, and goes to Step 1 if there are still some agents bidding in Step 2. Otherwise, she stops the auction processes and announces that the current allocation $f^*$ is the final allocation and the winning agents' payments are their last bid price, $r_{if_i^*}^n$.

Figure 2.2 shows the four agents' progressions of the bids in APA given the buyer values shown in Table 2.1. The x-axis indicates the round number and the y-axis indicates the bid price. I ran the simulation with increment $\delta = 0.01$. At the very beginning of the auction, the bid on each bundle is zero. Each agent puts all its interest in bundle ABC because ABC gives the maximal surplus from the valuation in Table 2.1. When Agent 1's bid price of ABC reaches 2, it begins to bid on bundles AB and AC because these two bundles also give the maximal surplus. Similarly, when Agent 2's bid price of ABC reaches 2, it would take bundle BC into its demand set and starts to bid on BC. From the figure, we can see that the bid prices increase with a steady rate in a large amount of rounds. The final allocation is {A, BC, –, –} and Agent 1's payment is 8 and Agent 2's payment is 17. Note that these payments are different than the VCG payments.

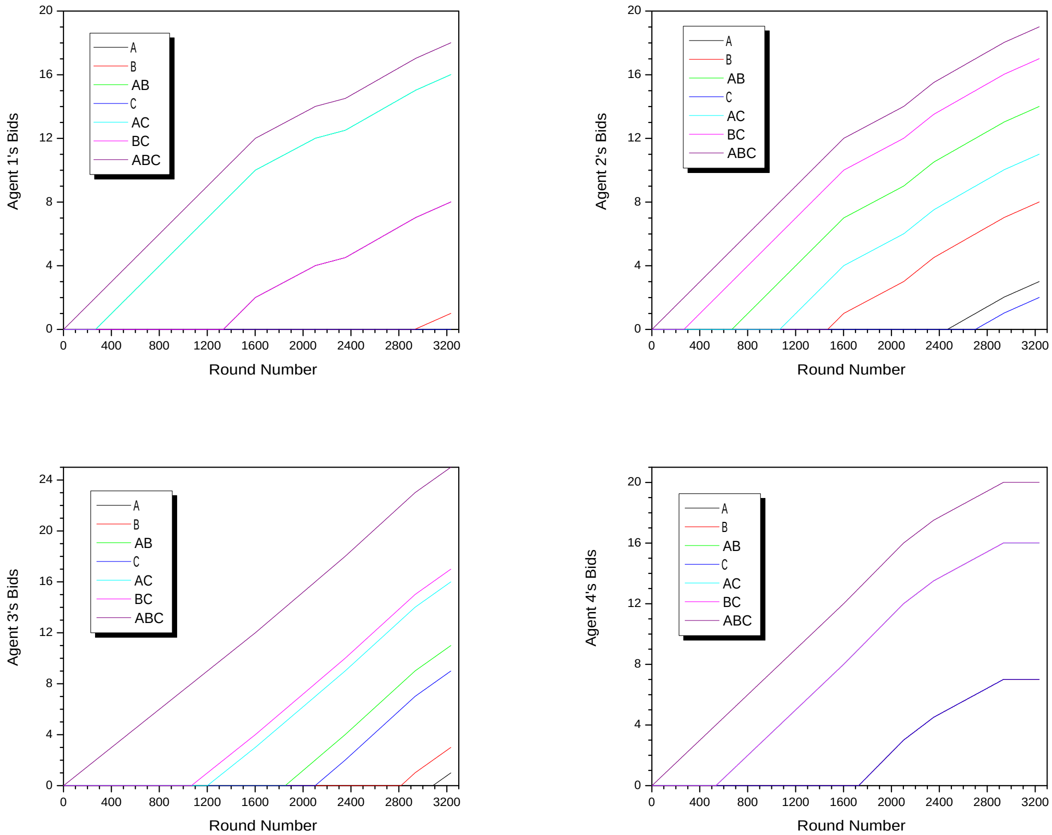


Figure 2.2: APA bundle bids of solving the example in Table 2.1 by simulation.

### 2.3.3 *i*Bundle Auctions

Parkes [24] introduced *i*Bundle, an ascending-price combinatorial auction that provides an XOR (exclusive-or) bidding language.There are three basic variations of *i*Bundle auctions: *i*Bundle(2) assigns anonymous prices to bundles for all agents and every agent sees the same prices; *i*Bundle(3) generates discriminatory prices for bundles and different agents may see different prices for the same bundle. The third variation *i*Bundle(d) dynamically switches from *i*Bundle(2) to *i*Bundle(3) to support efficient allocations.

In $i$Bundle auctions, the auctioneer solves the WDP to get the competitive allocation and announces the prices and the competitive allocation to all agents. Agents submit their new bids in the next iterative round if they are not winning in the current round. The auction terminates when every agent either (1) receives a bundle in the competitive allocation, or (2) repeats the same bids in successive rounds.

To simplify the bidding strategies, assume all agents use straightforward bidding strategy and submit the bids on all bundles in their demand set. The $i$Bundle(2) is used as an example to show the bidding processes. Since $i$Bundle(2) variation uses anonymous bundle prices, agent $i$'s demand set is defined as (2.1), i.e.,

$$D_i = \left\{ b \in B : s_{ib} > 0,\ s_{ib} \geq s_{ib'}\ \forall b' \in B \right\}.$$

An agent would increase its bid on all bundles in its demand set by $\delta$ if it was not winning in the previous round, and keep its bids otherwise.

Solving the $i$Bundle(2) by simulating the bidding processes includes the following steps.

**Step 0.** $n = 0, r_{ib}^n = 0, i \in I, b \in B$ and increment $\delta > 0$.

**Step 1.** The auctioneer solves the winner determination problem, i.e., computes the competitive allocation $f^*$ based on $r_{ib}^n$ and announces $f_i^*$ to agent $i$.

**Step 2.** The auctioneer computes the price $\pi^n$ and announces the price to all agents.

**Step 3.** All agents compute their demand set $D_i^n$. If agent $i$ does not win a bundle and its demand set $D_i^n$ is not empty, it increases its bid on all bundles in $D_i^n$. That is,

$$r_{ib}^{n+1} \leftarrow \pi_b^n + \delta, \quad \forall b \in D_i^n,$$

and

$$r_{ib}^{n+1} \leftarrow r_{ib}^{n}, \quad \forall b \notin D_i^n.$$

Otherwise, it passes,

$$r_{ib}^{n+1} \leftarrow r_{ib}^{n}, \quad \forall b \in B.$$

**Step 4.** The auctioneer updates the round counter, $n \leftarrow n+1$, and goes to Step 1 if there are still some agents bidding in Step 2. Otherwise, she stops the auction processes and announces that the current allocation $f^*$ is the final allocation and the winning agent's payment is $r_{if_i^*}^{n}$.

Figure 2.3 shows the bundle prices during $i$Bundle(2) with the buyer values shown in Table 2.1. The x-axis indicates the round number and the y-axis indicates the bundle price. I ran the simulation method with the increment $\delta$ taking the value of 0.01. The final allocation is {A, BC, –, –} and Agent 1's payment is 7.99 and Agent 2's payment is 16.99. This example ends with an efficient allocation. As we discussed before, this is not always true for $i$Bundle(2). But $i$Bundle(d) was proven to achieve the efficient allocation when agents follow straightforward bidding strategies [25].

## 2.4 Winner Determination Problem

In combinatorial auctions, given a set of bids, the auctioneer's goal is to maximize her *profit*, which is the summation of all winning agents' bid on allocated bundles. In other words, the auctioneer is looking for an allocation whose value is the largest. This fundamental problem in combinatorial auctions is called the Winner Determination Problem (WDP). Let $b \in \{0,1\}^n$ where $b^j = 1$ implies that item $j$ is an element of the bundle $b$.

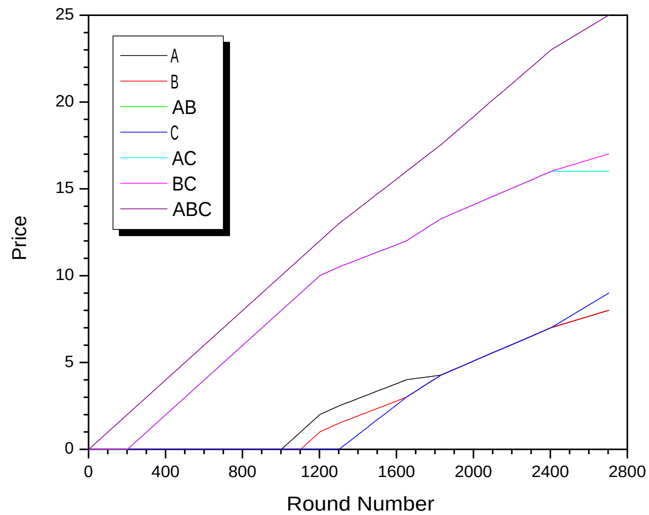


Figure 2.3: $i$Bundle(2) bundle prices of solving the example in Table 2.1 by simulation.

For two bundles, $b$ and $c$, we use the superset notation $b \subset c$ to indicate that, for all $j$, $b^j \leq c^j$. Further, we invoke free disposal, which allows us to assume valuations increase monotonically as items are added to a bundle. Let $x_{ib} \in \{0, 1\}$ take the unit value only when $b$ is assigned to agent $i$. Then the WDP is defined as follows

$$\text{WDP} \left\{ \begin{array}{ll} \max & \sum_{i \in I} \sum_{b \in B} r_{ib} x_{ib} \\ \text{s.t.} & \sum_{b \in B} x_{ib} \leq 1, \quad i = 1, \cdots, m \\ & \sum_{i \in I} \sum_{b \in B} b^j x_{ib} \leq 1, \quad j = 1, \cdots, n \\ & x_{ib} \in \{0, 1\} \end{array} \right.$$

For example, the following $x$ is a feasible solution to the problem in Table 2.1,

$$x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The matrix defines the allocation in which Agent 1 gets bundle B, Agent 3 receives bundle AC, and Agents 2 and 4 get nothing. In this dissertation, the shorter form $f$={B, –, AC, –} (see Section 2.1) is used to represent the allocation.

A competitive allocation is an allocation that maximizes the above expression. $F^*$, the set of competitive allocations is

$$F^* = \left\{ f^* \in F : f^* = \arg\max_{f \in F} \{V(f)\} \right\}, \tag{2.13}$$

where $F$ is the feasible set of the WDP.

Tree search algorithms work by enumerating all possible ways of making the decisions. Thus, once the search finishes, the optimal set $F^*$ will have been found and proven optimal. However, in practice, the space is too large to search exhaustively. The science of search is in techniques that selectively search the space while still provably finding an optimal solution. The general approaches to standard mixed integer linear programming problems, such as cutting planes, branching and bound, and heuristic algorithms, can be used to solve the WDP. Since the 1990s, the WDP has attracted some researchers. Nisan [23] expressed the WDP as a standard mixed integer linear programming problem. Rothkopf *et al.* [34] proved that the WDP is NP-hard. Sandholm [36] and Fujishima, *et al.* [12] proposed a special-purpose search algorithm for WDP based on the branch-on-items that used depth-first-search strategy, where the search always proceeds from a node to an unvisited child, if one exists. If not, the search backtracks. Instead of branching on items, a newer faster algorithm using the branch-on-bids is presented by Sandholm [37]. Kelly [19] showed that the WDP is a generalized knapsack problem. Kellerer *et al.* [18] used a dynamic programming approach to solve the WDP.

## 2.5 Summary

The current literature is reviewed and the main features of combinatorial auctions are introduced in this chapter. The first section presented some common notations and definitions that will be used in the following chapters. The VCG auction was introduced in the second section and an example was used to illustrate how to compute the payment of VCG. Section 2.3 addressed the iterative ascending auctions including the A$k$BA, APA, and $i$Bundle Auctions. The auction rules and policies were presented in this section. An example was used to demonstrate how to get the results of these iterative combinatorial auctions. The winner determination problem was discussed in Section 2.4.

# Chapter 3

# Simple Combinatorial Proxy Auction

This chapter introduces a new iterative combinatorial auction with proxy bidding called the Simple Combinatorial Proxy Auction (SCPA). This auction is simple in that the price definition is naive and the payment is pay-your-bid. SCPA may give different results from other combinatorial auction mechanisms, including the VCG, iBundle, and APA, and does not have the same theoretical results.

In Section 3.1, the auction rules and bidding policies are presented. The simulation method is implemented to solve the SCPA problem in Section 3.2. An example is used to show how the SCPA mechanism works in Section 3.3. Section 3.4 addresses the proof that SCPA's final allocation is equal to the final allocation of A1BA, and the last section demonstrates the advantages of SCPA with several examples.

## 3.1 Auction Rules and Bidding Policies

The Simple Combinatorial Proxy Auction (SCPA) has the following rules:

- Several unique items are sold;
- The auction is iterative and accepts bid on any bundle;
- The bundle prices start at \$0 and are anonymous to all agents
- The bundle price is the maximal bid value among all bids on the bundle;
- If there is more than one competitive allocation, the auctioneer randomly picks one and announces it to all agents;
- The auctioneer only accepts bids from non-winning agents of the announced allocation;
- The auction requires that an agent cannot withdraw its bids.

Let $r_{ib}$ be agent $i$'s bid on bundle $b$. The current price of bundle $b$ is simply the highest bid among all bidders on the bundle. That is

$$\pi_b = \max_{i \in I} \{r_{ib}\}. \tag{3.1}$$

The price is anonymous in that all bidders are given the same price information and is nonlinear in that the price of a bundle may not be the summation of the prices of individual items.

To simplify the agents' strategy space, we assume that all agents submit incremental bids on behalf of the bidder by following a straightforward bidding strategy. That is, agent $i$'s demand set is computed as

$$D_i = \left\{ b \in B : s_{ib} > 0,\ s_{ib} \geq s_{ib'}\ \forall b' \in B \right\},$$

where $s_{ib}$ is agent $i$'s surplus of on bundle $b$, $s_{ib} = v_{ib} - \pi_b$. The agent is active if its demand set is not empty. Suppose there is no preference on the bundles in an agent's demand set and the agent's bid increment $\delta$ is a constant. In this case, the agent's bidding strategies are:

- If agent $i$ is told by the auctioneer that it is winning, it does not increase its bid in the next round;
- Otherwise, agent $i$ submits one and only one bid on a bundle in its demand set $D_i$ with the current price plus the increment $\delta$.

After the bids are received, the auctioneer computes the bundle prices and solves the WDP to get a competitive allocation. The she announces the competitive and the bundle prices. The announced prices are not necessarily separating [42] because the optimal allocation may include agents whose last offer on their winning bundle is less than the current price. To compensate for this, SCPA directly informs agents of their winning status. Note that an inactive agent may be a winner in some special cases because the agents cannot withdraw their bids once the bids are submitted to the auctioneer.

## 3.2 Simulation Method

A superscript on the notations is used to indicate the number of a round in the simulation method. In this dissertation, the round number is omitted if there is no ambiguity. The simulation method for solving SCPA with straightforward bidding involves the following steps:

**Step 0.** Initialization: Increment $\delta > 0$, round counter $n = 0$, bids $r_{ib}^n = 0$ for $i \in I, b \in B$.

**Step 1.** The auctioneer solves the WDP based on $r_{ib}^n$ to get the competitive allocation $f^*$, and then $f^*$ to all agents.

**Step 2.** The auctioneer computes the bundle prices $\pi_b^n = \max_{i \in I}\{r_{ib}^n\}$ and announces them to all agents.

**Step 3.** Each agent $i$ computes the demand set, $D_i^n$, by following the best response strategy.

**Step 4.** If agent $i$ does not win a bundle in $f^*$, i.e., $i \notin f^*$ and its demand set is not empty, $D_i^n \neq \emptyset$, it will increase its bid on one bundle, $b_0$, in its demand set. That is ,

$$r_{ib}^{n+1} = r_{ib}^n, \quad \forall b \in B \text{ and } b \neq b_0,$$

and

$$r_{ib_0}^{n+1} = \pi_{b_0}^n + \delta.$$

Otherwise, it will pass (hold its bid in next round),

$$r_{ib}^{n+1} = r_{ib}^n, \quad \forall b \in B.$$

**Step 5.** Check for the termination.

If there are still some agents bidding in Step 4, the auctioneer updates the round counter $n \leftarrow n + 1$ and goes to Step 1.

Otherwise, the auctioneer stops the bidding processes and announces that the current competitive allocation $f^*$ is the final allocation and the winning agents' payments are their last bids on the bundle that they win.

Table 3.1: Example with four buyers bidding on the combinations of three items.

| | A | B | AB | C | AC | BC | ABC |
|---|---|---|---|---|---|---|---|
| Buyer 1 | 13 | 20 | 27 | 20 | 28 | 22 | 50 |
| Buyer 2 | 19 | 1 | 20 | 17 | 34 | 18 | 38 |
| Buyer 3 | 28 | 17 | 38 | 3 | 31 | 19 | 44 |
| Buyer 4 | 5 | 12 | 15 | 26 | 29 | 32 | 43 |

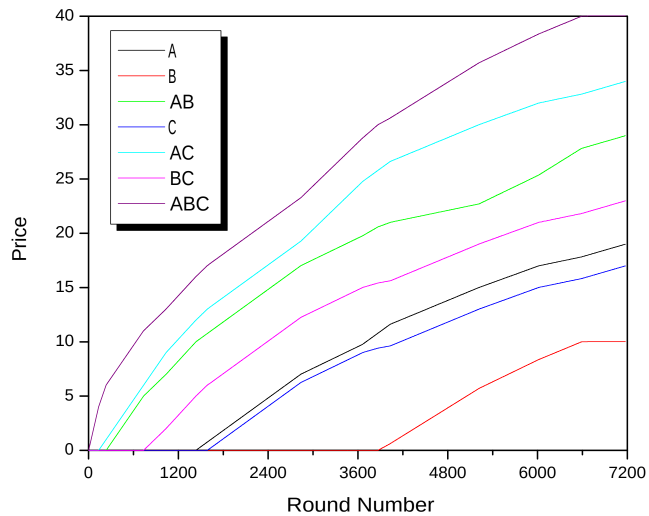


Figure 3.1: SCPA bundle prices of solving the example in Table 3.1 by simulation.

## 3.3 An Example

Figure 3.1 shows the progression of the prices of bundles in SCPA with the agents given buyer values shown in Table 3.1. The increment in the simulation rounds takes the value of 0.01. At the beginning of the auction, each agent puts all its interest on bundle ABC because ABC gives the maximal surplus. When the price of ABC reaches 4, Agent 2 changes its behavior. It will begin to bid bundle AC because bundle AC gives the same maximal surplus as bundle ABC does. It is interesting that Agent 2 does not bid on bundle ABC any more in the following rounds because Agent 1, Agent 2, and Agent 3's continuing bidding on ABC makes the price of ABC increasing faster than the new active bundle and Agent 2 has no chance to bid on ABC. After the price of ABC reaches the value of 6, bundle AB becomes active to Agent 3 because AB shows the maximal surplus as bundle ABC. Thus, the price of bundle AB increases, as shown in the figure.

It is obvious from the figure that, for a large number of round periods, the prices progress in a steady fashion, and that these periods are punctuated by events that change the rates at which the prices proceed. For example, from round 3900 to 5200, the slopes of all bundle prices remain constant. In this range, the agents' demand sets do not change and the competitive allocation set is stable. More details about this stability are addressed in Chapter 4.

The final bundle prices found using the simulation method depend upon the increment $\delta$. Define the average bundle price as

$$\bar{\pi} = \frac{\sum_{b \in B} \pi_b}{|B|},$$

and the average bundle error as

$$\bar{E}(\delta) = \frac{|\pi_b - \pi_b^{\mathrm{SCPA}}(\delta)|}{|B|},$$

where $\pi_b$ and $\pi_b^{\mathrm{SCPA}}(\delta)$ are the actual price and SCPA simulation price of bundle $b$, respectively. Suppose the relative bundle error as a function of relative bundle increment $\bar{\delta} = \frac{\delta}{\bar{\pi}} \times 100\%$ is defined by

$$E(\bar{\delta}) = \frac{\bar{E}(\delta)}{\bar{\pi}} \times 100\%.$$

I tested the example shown in Table 3.1 using different increments, and the relative bundle errors are shown in Figure 3.2 where the x-axis indicates the value of the relative bundle increment and the y-axis is the relative bundle error. Results for the increments, 4, 2, 1,

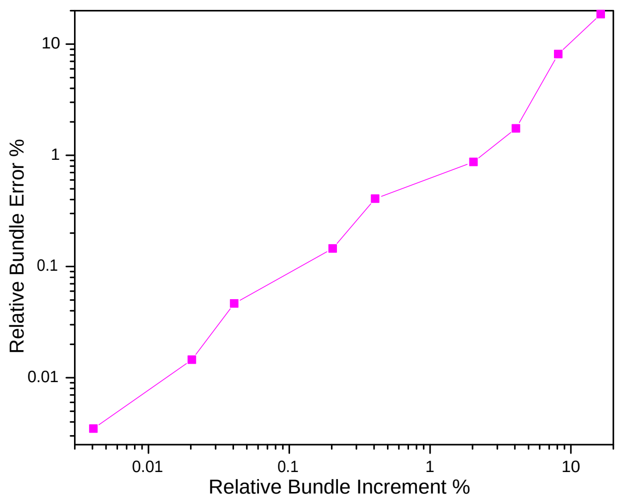


Figure 3.2: Relationship between the accuracy and the simulation increment.

0.5, 0.1, 0.05, 0.01, 0.005, and 0.001 are shown. From the figure, it is clear that a smaller increment leads to a more accurate price. Section 5.3 provides some results showing that the smaller increment costs more computing time.

## 3.4 Equivalence to A*k*BA

This section shows that SCPA is equal to A1BA in that the final allocation of SCPA is the same as that of A1BA if their initial values, the increments $\delta$, and the agents' bidding strategy are the same. This result is important because it means that, rather than solving the linear program $\mathrm{LP}_{\mathrm{upper}}$ to compute the bundle prices in every round of A1BA, one can just take the SCPA prices, which are easy to get, as the intermediate A1BA bundle prices, and in the last round of A1BA, he/she solves an $\mathrm{LP}_{\mathrm{upper}}$ to get the final A1BA payments.

### 3.4.1 The Equivalence of A1BA and SCPA

The equivalence of A1BA and SCPA can be proved by induction by showing that the bids of all agents on all bundles will be the same at round $n+1$ if they are the same at round $n$.

Suppose the bids in both A1BA and SCPA are $r_{ib}^n, i \in I, b \in B$ at round $n$. By solving the same WDP based on the bids $r_{ib}^n$, we get the same competitive allocation $f^*$. Thus, the non-winning agents in both auctions are the same. Because the winning agents will keep their bids in round $n+1$ and only the non-winning agents will increase their bids, it is sufficient to prove the non-winning agents' bids are the same in round $n+1$. As we know, the non-winning agent will increase, by an increment, its bid on the current price of a bundle in its demand set. That is, the non-winning agents' bid of A1BA in round $n+1$ is $r_{ib}^{n+1} = \pi_b^n + \delta$, and their bid of SCPA in round $n+1$ is $r_{ib}^{n+1} = \max\{r_{ib}^n\} + \delta$. So, our task is to show that in A1BA, $\pi_b^n = \max\{r_{ib}^n\}$ for any non-winning agent, $i$, and any bundle, $b \in D_i^n$. Before the proof of this conclusion is given, Let us take a look of some lemmas.

In A1BA, the agents' demand sets are based on the anonymous bundle prices. The agents are bidding on the bundles in their demand set straightforwardly. That is, the bids $r^{n+1}$ are based on demand $D^n$. So, it is easy to get the following lemma.

**Lemma 3.4.1** *In* A1BA*, if a bundle, $b$, is in agent $i$'s demand set $D_i$, agent $i$'s bid on $b$ is the highest value among the values of all agents. That is,*

$$r_{ib} = \max_{j \in I}\{r_{jb}\},$$

*where $\delta$ is the bidding increment.*

**Lemma 3.4.2** *Suppose $f^*$ is a competitive allocation and $(s^*, \pi^*)$ is the optimal solution of* $\mathrm{LP}_{\mathrm{upper}}$ *in* A1BA*. Then any non-winning agent's surplus is zero.*

**Proof.** Suppose agent $i_0$ gets nothing in the competitive allocation $f^*$ and $s^*_{i_0} > 0$. From (2.9),

$$V(f^*) = \sum_{i \in I} s^*_i + \sum_{b \in B_{f^*}} \pi^*_b.$$

Rewrite $V(f^*)$,

$$V(f^*) = \sum_{i \in I_{f^*}} s^*_i + \sum_{i \notin I_{f^*}} s^*_i + \sum_{b \in B_{f^*}} \pi^*_b.$$

Since $s^*_i \geq 0$, we have

$$V(f^*) \geq \sum_{i \in I_{f^*}} s^*_i + s^*_{i_0} + \sum_{b \in B_{f^*}} \pi^*_b.$$

Because $s^*_{i_0} > 0$, we get

$$V(f^*) > \sum_{i \in I_{f^*}} s^*_i + \sum_{b \in B_{f^*}} \pi^*_b = \sum_{i \in I_{f^*}} (s^*_i + \pi^*_{f^*_i}). \tag{3.2}$$

By rearranging (2.7), it is easy to get

$$s^*_i + \pi^*_{f^*_i} \geq r_{i f^*_i}. \tag{3.3}$$

Take (3.3) into (3.2), we have

$$V(f^*) > \sum_{i \in I_{f^*}} r_{i f^*_i}. \tag{3.4}$$

Since $V(f^*)$ is the value of the competitive allocation,

$$V(f^*) = \sum_{i \in I_{f^*}} r_{i f^*_i}. \tag{3.5}$$

Thus, Equations (3.4) and (3.5) contradict. Therefore, a non-winning agent's surplus must be zero. ■

From this lemma, we can get the following corollary.

**Corollary 3.4.3** *Suppose $f^*$ is a competitive allocation and $(s^*, \pi^*)$ is the optimal solution of* $\text{LP}_{\text{upper}}$ *in* A1BA. *For any bundle $b$, the non-winning agent's bid on $b$ is always less than or equal to $\pi_b^*$, that is,*

$$r_{ib} \leq \pi_b^*, \quad \forall i \notin I_{f^*}.$$

**Proof.** Suppose agent $i_0$ is a non-winning agent. Then, from Lemma 3.4.2, we have $s_{i_0}^* = 0$. If $b \notin B_{f^*}$ is an unallocated bundle, from (2.11) with $k = 1$,

$$\pi_b^* = \max_{i \in I}\{r_{ib} - s_i^*\} \geq r_{i_0 b - s_{i_0}^*} = r_{i_0 b}.$$

Otherwise, $b \in B_{f^*}$ is an allocated bundle. From (2.7) and Lemma 3.4.2, we get

$$s_{i_0 b}^* + \pi_b^* = \pi_b^* \geq r_{i_0 b}.$$

So, Corollary 3.4.3 is true. ■

**Lemma 3.4.4** *Suppose $f^*$ is a competitive allocation and $(s^*, \pi^*)$ is the optimal solution of* $\text{LP}_{\text{upper}}$ *in* A1BA. *For any winning agent, $i$, the following equation is true,*

$$s_i^* + \pi_{f_i^*}^* = r_{if_i^*}.$$

**Proof.** From (2.7), we have

$$s_i^* + \pi_{f_i^*}^* \geq r_{if_i^*}, \quad i \in I_{f^*}, \tag{3.6}$$

and from (2.9), we get

$$V(f^*) = \sum_{i \in I_{f^*}} s_i^* + \sum_{b \in B_{f^*}} \pi_b^*.$$

From Lemma 3.4.2,

$$V(f^*) = \sum_{i \in I_{f^*}} (s_i^* + \pi_{f_i^*}^*). \tag{3.7}$$

Notice that $V(f^*)$ is the value of the competitive allocation $f^*$, i.e.,

$$V(f^*) = \sum_{i \in I_{f^*}} r_{if_i^*}. \tag{3.8}$$

Thus, from (3.7) and (3.8), we have

$$\sum_{i \in I_{f^*}} (s_i^* + \pi_{f_i^*}^*) = \sum_{i \in I_{f^*}} r_{if_i^*}. \tag{3.9}$$

Thus, from (3.6) and (3.9), the lemma holds, i.e.,

$$s_i^* + \pi_{f_i^*}^* = r_{if_i^*}.$$

Therefore, the winning agent $i$'s bid price on bundle $f_i^*$ is equal to the summation of its surplus and the bundle price. ■

**Lemma 3.4.5** *Suppose $f^*$ is a competitive allocation and $(s^*, \pi^*)$ is the optimal solution of* $\mathrm{LP_{upper}}$ *in* A1BA*. Then, for any bundle $b \in B$,*

$$\pi_b^* \leq \max_{i \in I}\{r_{ib}\}.$$

**Proof.** If $b \notin B_{f^*}$ is an unallocated bundle, from (2.8)and (2.11) with $k = 1$ ,

$$\pi_b^* = \max_{i \in I}\{r_{ib} - s_i^*\} \leq \max_{i \in I}\{r_{ib}\}. \tag{3.10}$$

Suppose $b \in B_{f^*}$ is an allocated bundle. Denote $i_b$ as the agent who receives bundle $b$ in the competitive allocation $f^*$. From Lemma 3.4.4,

$$s_{i_b}^* + \pi_b^* = r_{i_b b}.$$

From (2.8), we have

$$\pi_b^* \leq r_{i_b b} \leq \max_{i \in I}\{r_{ib}\}. \tag{3.11}$$

Therefore, (3.10) and (3.11) prove the lemma. ■

Suppose $(s^*, \pi^*)$ is the optimal solution of the corresponding $\mathrm{LP}_{\mathrm{upper}}$ in A1BA. From Lemma 3.4.2, we know that non-winning agents have 0 surplus. To the winning agents, we have the following theorem.

**Theorem 3.4.6** *Suppose $f^*$ is a competitive allocation. Let $(s^*, \pi^*)$ denote the optimal solution of* $\mathrm{LP}_{\mathrm{upper}}$ *in* A1BA*. Then there must exist one winning agent whose surplus is zero.*

**Proof.** Suppose all winning agents have zero surplus. That is, $s_i^* > 0$ for all $i \in I_{f^*}$. Let

$$\varepsilon = \min_{i \in I_{f^*}} \{s_i^*\} > 0.$$

Define $(\hat{s}, \hat{\pi})$ as follows,

$$\hat{s}_i = \begin{cases} s_i^* - \varepsilon, & i \in I_{f^*} \\ 0, & i \notin I_{f^*} \end{cases}$$

$$\hat{\pi}_b = \pi_b^* + \varepsilon, \quad b \in B_{f^*}.$$

It is easy to confirm that $(\hat{s}, \hat{\pi})$ is a feasible solution to $\mathrm{LP}_{\mathrm{upper}}$. The objective value of $(\hat{s}, \hat{\pi})$ is $g(\hat{s}) = \sum_{i \in I_{f^*}} \hat{s}_i$. Take the values of $\hat{s}_i$, $i \in I_{f^*}$ into $g(\hat{s})$, we have

$$g(\hat{s}) = \sum_{i \in I_{f^*}} (s_i^* - \varepsilon) = g(s^*) - |f^*|\varepsilon,$$

where $|f^*|$ is the number of winning agents in $f^*$. Because $\varepsilon > 0$, we obtain that $g(\hat{s}) > g(s^*)$. It contradicts the optimality of $s^*$. Therefore, the theorem holds. ■

Suppose $(s^*, \pi^*)$ is the optimal solution of the corresponding $\mathrm{LP}_{\mathrm{upper}}$ in A1BA. The value of objective function $g(s)$ at $s^*$ is either zero or positive. To the first case, we have the following theorem.

**Theorem 3.4.7** *Suppose $f^*$ is a competitive allocation. Let $(s^*, \pi^*)$ denote the optimal solution of* $\text{LP}_{\text{upper}}$ *in* A1BA*. If the optimal objective value $g(s^*) = 0$, the bundle prices of* A1BA *are the same as* SCPA*, i.e.,*

$$\pi_b^* = \max_{i \in I}\{r_{ib}\}, \quad b \in B.$$

**Proof.** From (2.6), the optimal objective

$$g(s^*) = \sum_{i \in I} s_i^* = 0.$$

Thus, from (2.8), i.e., $s_i^* \geq 0\ \forall i \in I$, we get

$$s_i^* = 0\ \forall i \in I. \tag{3.12}$$

Because $V(f^*) = \sum_{i \in I} s^* + \sum_{b \in B_{f^*}} \pi_b^*$, we obtain

$$V(f^*) = \sum_{b \in B_{f^*}} \pi_b^*. \tag{3.13}$$

For any unallocated bundle $b$, from (2.11) with $k = 1$,

$$\pi_b^* = \max_{i \in I}\{r_{ib} - s_i^*\} = \max_{i \in I}\{r_{ib}\} \quad \forall b \notin B_{f^*}. \tag{3.14}$$

Suppose $b$ is an allocated bundle, i.e., $b \in B_{f^*}$. From the feasibility of the optimal solution $(s^*, \pi^*)$, the following expression holds for all winning agents,

$$s_i^* + \pi_b^* \geq r_{ib}, \quad \forall i \in I. \tag{3.15}$$

Take the value of $s^* = 0$ from (3.12) into the above expression (3.15), and it is clear that the bundle price is greater than or equal to all winning agents' bid price,

$$\pi_b^* \geq r_{ib}, \quad \forall i \in I,$$

which implies that price of bundle $b$ is greater than or equal to the maximal bid on bundle $b$ over all agents, i.e.,

$$\pi_b^* \geq \max_{i \in I}\{r_{ib}\}. \tag{3.16}$$

Expression (3.13) and (3.16) convert to

$$V(f^*) \geq \max_{i \in I}\{r_{ib}\}. \tag{3.17}$$

Obviously, for $i \in I_{f^*}$, agent $i$'s bid price is less than or equal to the max bid of bundle $f_i^*$,

$$r_{if_i^*} \leq \max_{j \in I}\{r_{jf_i^*}\}.$$

So, the value of the competitive allocation $f^*$ is less than or equal to the summation of the maximal bids of allocated bundles,

$$V(f^*) = \sum_{i \in I_{f^*}} r_{if_i^*} \leq \sum_{b \in B_{f^*}} \max_{i \in I}\{r_{ib}\}. \tag{3.18}$$

Thus, from (3.17) and (3.18), the following inequalities hold,

$$\sum_{b \in B_{f^*}} \max_{i \in I}\{r_{ib}\} \geq \sum_{b \in B_{f^*}} \pi_b^* \geq \sum_{b \in B_{f^*}} \max_{i \in I}\{r_{ib}\},$$

which implies that price of bundle $b$ is the maximal value of bids on bundle $b$ over all agents, i.e.,

$$\pi_b^* = \max_{i \in I}\{r_{ib}\}, \quad \forall b \in B_{f^*}. \tag{3.19}$$

Therefore, (3.14) and (3.19) prove the theorem. ■

Theorem 3.4.7 states that under the condition that the optimal objective value of $\mathrm{LP}_{\mathrm{upper}}$ is zero, the bundle prices of A1BA are the same values as in SCPA at round $n$. Note that the non-winning agents are also the same at round $n$. Thus, the bids of all agents on all bundles in A1BA and SCPA are the same at round $n+1$.

However, the optimal objective value of $\text{LP}_{\text{upper}}$ may not be zero in all rounds. The following theorem completes the proof by showing that even if $g(s^*) > 0$ at round $n$, the non-winning agents' bids in both auctions are still the same at round $n+1$. Consider the case that the optimal objective value $g(s^*)$ of $\text{LP}_{\text{upper}}$ is not zero. From Lemma 3.4.4, $\pi_{if_i^*} = r_{if_i^*} - s_i^*$, thus, the price of allocated bundle may be less than the highest value of all agents' bids. But, as the following theorem shows, this never happens to the bundles that are in the demand set of non-winning agents.

**Theorem 3.4.8** *Suppose $f^*$ is the competitive allocation at round $n$ and agent $j$ does not win a bundle in $f^*$. Denote $(s^*, \pi^*)$ to be the optimal solution of* $\text{LP}_{\text{upper}}$ *in* A1BA*. Then the price of any bundle in its demand set $D_j^n$ is*

$$\pi_b^* = \max_{i \in I}\{r_{jb}^n\}, \quad b \in D_j^n.$$

**Proof.** Suppose bundle $b$ is in agent $j$'s demand set $D_j^n$ at round $n$. We consider two cases: bundle $b$ is either unallocated in $f^*$ or allocated in $f^*$.

Case (i): $b$ is an unallocated bundle. From (2.11) with $k = 1$, the price of $b$ is

$$\pi_b^* = \max_{i \in I}\{r_{ib}^n - s_i^*\} \geq r_{jb}^n - s_j^*.$$

From Lemma 3.4.2, we know that $s_j^* = 0$. Thus,

$$\pi_b^* = \max_{i \in I}\{r_{ib}^n - s_i^*\} \geq r_{jb}^n. \tag{3.20}$$

Because $b$ in agent $j$'s demand set, $D_j^n$, from Lemma 3.4.1, we have

$$r_{jb}^n = \max_{i \in I}\{r_{ib}^n\}.$$

Consider $s_i^* \geq 0$ for all $i \in I$, we obtain

$$r_{jb}^n = \max_{i \in I}\{r_{ib}^n\} \geq \max_{i \in I}\{r_{ib}^n - s_i^*\}. \tag{3.21}$$

From (3.20) and (3.21), we get

$$\pi_b^* = \max_{i \in I}\{r_{ib}^n\}.$$

Case (ii): $b$ is an allocated bundle. From the feasibility of $(s^*, \pi^*)$,

$$s_i^* + \pi_b^* \geq r_{ib}, \quad \forall i \in I.$$

For agent $j$, the above expression gives

$$s_j^* + \pi_b^* \geq r_{jb}.$$

From Lemma 3.4.1 and 3.4.2, we have

$$\pi_b^* \geq r_{jb} = \max_{i \in I}\{r_{ib}\}. \tag{3.22}$$

From Lemma 3.4.5,

$$\pi_b^* \leq \max_{i \in I}\{r_{ib}\}. \tag{3.23}$$

Combing (3.22) and (3.23) gives

$$\pi_b^* = \max_{i \in I}\{r_{ib}\}.$$

Therefore, (i), together with (ii), proves the theorem. ■

The main benefit of understand the relationship between A1BA and SCPA is that it costs less to compute the bundle prices of SCPA by solving (3.1) than those of A1BA by solving $\mathrm{LP}_{\mathrm{upper}}$ and equation (2.10) and (2.11). Moreover, any algorithm that has successfully been applied to SCPA can be directly used to solve A1BA with the final adjustments made at the end. In other words, A1BA and SCPA can share solution methods.

### 3.4.2 Example and Discussion

It is clear that the bundle prices are monotonically increasing in SCPA, while A1BA may produce oscillations in bundle prices. In the previous section, we proved that

Table 3.2: Example with three buyers bidding on the combinations of two items.

| | A | B | AB |
|---|---|---|---|
| Agent 1 | 6 | 6 | 0 |
| Agent 2 | 3 | 0 | 0 |
| Agent 3 | 0 | 0 | 8 |

A1BA is equal to SCPA in the sense that identical bids are submitted in all simulation rounds. As a result, there is no need to solve the $\mathrm{LP}_{\mathrm{upper}}$ in every round of an A1BA simulation if we do not care about the bundle prices in the processes. At the last round, the $\mathrm{LP}_{\mathrm{upper}}$ is solved to get the final bundle prices of A1BA and then to get the payments of A1BA.

Let us take a look at an example in Table 3.2. By using simulation with a bid increment of 0.1, both SCPA and A1BA need about 500 rounds to reach the final results shown in Figure 3.3. In the simulation rounds, the competitive allocations are {B, A, –} and {–, –, AB}. From round 1 to round 300 in A1BA, the solution of the $\mathrm{LP}_{\mathrm{upper}}$ is $s_i^* = 0$ for all agents no matter which competitive allocation is announced. Theorem 3.4.7 shows that the A1BA bundle prices would be $\pi_b = \max\{r_{1b}, r_{2b}, r_{3b}\}$ for $b =$A, B, AB, which are exactly the same as the SCPA bundle prices. The A1BA price oscillations begin at round 301 in Figure 3.3.

To see what causes the oscillations, let us look at round 350, at which $r_{1\mathrm{A}} = 3.5, r_{1\mathrm{B}} = 3.5, r_{3\mathrm{AB}} = 6.5$, Agent 2 is inactive and it keeps its bid $r_{2\mathrm{A}} = 3$.

- If the competitive allocation {–, –, AB} is announced by the A1BA auction, the $\mathrm{LP}_{\mathrm{upper}}$ gives us the solution $s_1^* = s_2^* = s_3^* = 0, \pi_{\mathrm{A}}^* = 3.5, \pi_{\mathrm{B}}^* = 3.5, \pi_{\mathrm{AB}}^* = 6.5$. In the next round, Agent 1 will bid on either bundle A or bundle B with price $r_{1\mathrm{A}} = 3.5 + 0.1 = 3.6$, and the winning agent, Agent 3, will keep its bid as 6.5.

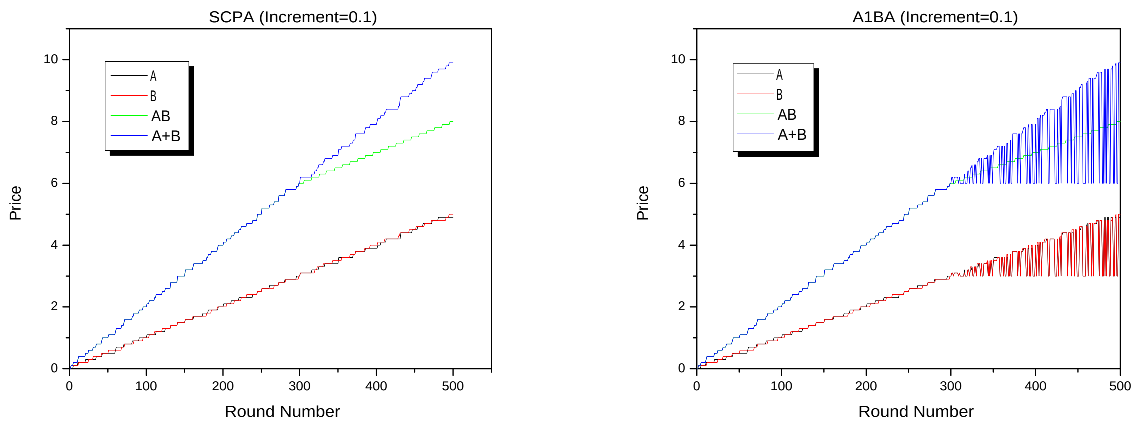


Figure 3.3: Bundle prices of SCPA and A1BA of solving the example in Table 3.2 by simulation.

- If the competitive allocation {B, A, −} is announced, the solution of the $\text{LP}_{\text{upper}}$ is $s_1^* = 0.5, s_2^* = s_3^* = 0, \pi_{\text{A}}^* = 3, \pi_{\text{B}}^* = 3, \pi_{\text{AB}}^* = 6.5$. In the next round, Agents 1 and 2 are the winning agents, they won't bid (Agent 2 is inactive). Agent 3 is the non-winning agent and will bid. Notice that only bundle AB is in Agent 3's demand set. So, it will place the new bid on AB with $r_{3\text{AB}} = 6.5 + 0.1 = 6.6$.

Thus, as the auction oscillates between the two solutions, the price of bundle B must be decreased to reflect the fact that Agent 2 is winning, and the price of bundle A to balance the surplus for Agent 1. Then, both prices bounce back up again in the rounds when Agent 3 is winning. Consider the SCPA's price definition, the SCPA price is always at the top of the A1BA price oscillations.

## 3.5 Uniqueness of SCPA

In this section, some examples are used to show that SCPA is different from the existing auctions. Let us take a look at the example in Table 3.3:

Table 3.3: Example that shows the SCPA is different from the existing auctions.

| | A | B | AB |
|---|---|---|---|
| Bidder 1 | 8 | 7 | 9 |
| Bidder 2 | 1 | 3 | 9 |
| Bidder 3 | 2 | 1 | 10 |

Table 3.4: Results of different auctions running the example in Table 3.3.

| Auctions | Allocation | Payments |
|---|---|---|
| GVA | {A, B, –} | (7, 2, 0) |
| APA | {A, B, –} | (7.5, 2.5, 0) |
| *i*Bundle | {A, B, –} | (7.5, 2.5, 0) |
| SCPA | {–, –, AB} | (0, 0, 9) |

The results of SCPA, *i*Bundle, VCG and APA are listed in Table 3.4 with the increment taking the value of 0.001 in iterative auctions. It is clear that, at least for some problems, the final allocation in SCPA is different from that of GVA and APA. That means that SCPA is not equal to VCG and APA. The example in Table 3.2 has already shown SCPA is not equal to A1BA. So, SCPA is a new type of iterative auction.

The results in Table 3.4 shows that SCPA is not equal to VCG because their final allocations are different. Actually, even their final allocation is the same, their payments may be different. By running SCPA with the example in Table 2.1, the final allocation if {A, BC,–, –} and Bidder 1's payment is 8 and Bidder 2's payment is 17. From the discussion in Chapter 2, the VCG gives the same allocation but Bidder 1's payment is 7 and Bidder 2's payment is 17. Therefore, like the other iterative auctions, the SCPA payment may be different from VGA, even if the allocation is the same.

# Chapter 4

# Price Trajectory Algorithm

In this chapter, an algorithm named the Price Trajectory Algorithm (PTA) is presented to solve iterative combinatorial auction problems. The Simple Combinatorial Proxy Auction (SCPA) is used as an example to illustrate the core concepts and constraints of the algorithm. Based on the straightforward bidding strategy, the discrete bidding rounds are changed to continuous time by letting the increment go to zero. The agents' bidding attention is analyzed and a mathematical model is proposed to compute the allocation of agent's attention. A numerical example is given to show the details in the SCPA version of the PTA ($\mathrm{PTA}_{\mathrm{SCPA}}$) iterations. Finally, the multi-stage auctions are discussed and the $\mathrm{PTA}_{\mathrm{SCPA}}$ is used in the multi-stage auctions.

## 4.1 Framework of Price Trajectory Algorithm

Consider the bundle prices in Figure 3.1 again. For large periods of time, the prices progress in a steady fashion, and these periods are punctuated by events that cause a change in the trajectories. The auction starts off with all four agents bidding on the bundle ABC because ABC gives them the maximal surplus. The four agents put their unit attention either on ABC or pass. This pattern remains until an event happens: when the price on bundle ABC reaches 4, Agent 3 becomes indifferent to bundles AB and ABC. In other words, the event is that Agent 3's demand set changes. Now, Agent 3 alternates bidding on the bundle AB in its demand set and the price of bundle AB increases, and other agents still bid on bundle ABC. Similar to the first period, the pattern remains until another event happens.

Rather than model bidding behavior explicitly, we construct a model of the price or bid trajectories and keep track of each agent's contribution toward the trajectories. Changes in the trajectory correspond to points in time at which price changes cause an agent to change its behavior. The PTA is based on the following behavior of agents.

1. An agent will pass if it is currently winning or the current bundle prices are all greater than or equal to its valuations.

2. An agent will begin bidding on a bundle that is not in its previous demand set when the surplus of this bundle becomes the same as the surplus of the bundles in its demand set.

3. An agent will stop bidding on a bundle if the bundle's price is increasing faster than other bundles that the agent has in its demand set.

Roughly speaking, the PTA is an iterative process in which each iteration involves (1) computing the allocation of agents' attention, and (2) computing the duration for which the former holds. We will discuss these two issues in the following sections.

### 4.1.1 Demand Set

An agent becomes inactive when its potential surplus becomes 0. The inactive agents do not bid because their demand sets are empty. Each active agent $i$ has a unit of *attention* that it can allocate between the bundles in its demand set $D_i$. Attention can be thought of as the proportion of the agent's time that will be spent bidding on a particular bundle. Let $\theta_{ib}$ be the amount of attention that agent $i$ gives to bundle $b$ over a unit time interval, and $\theta_{i\text{pass}}$ be the proportion of time the agent will not bid because it is already winning. An agent does not bid on a bundle $b$ that is not in its demand set, $\theta_{ib} = 0$. The slope of bundle $b$ is easily computed by $\theta_b = \sum_i \theta_{ib}$.

At inflection points, we know the *potential* demand sets, $\hat{D}_i$ for all $i \in I$, which are is the sets of bundles that satisfy (2.1) at time $t$. The potential set 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. Although all of the identified bundles are instantaneously active at the inflection point of time $t$, they may not all remain active going forward. We care about determining which bundles will remain active in the ensuing time period. During a period between two inflection points, the actual demand set consists of all the bundles on which the agent really puts attention.

### 4.1.2 Competitive Allocations

An agent's bidding behavior is also influenced by whether or not an agent is part of a winning coalition. If the agent is not winning, and has any surplus left to commit, it will bid again. While the agent is winning, it will not bid because it is already in a state in which the bundle that it is winning presents the greatest surplus.

Recall that allocation $f$ is a feasible solution to the WDP, that is, $f : J \rightarrow I$. In addition to the trajectories of the bundle prices, the PTA maintains information about the trajectories and values of the allocations. The concept of competitive allocations is critical because agents will pass if and only if they are members of a competitive allocation, and they will pass as often as a competitive allocation to which they belong is selected by the auctioneer.

Although competitive allocation $f$ is an optimal solution to the WDP, we do not solve WDP directly to get the competitive allocations. At the beginning of an interval, the *potential* competitive allocation set $\hat{F}^*$ consists of the competitive allocations in the previous interval and the allocations that become competitive at this particular time. Similar to the way we handle potential demand sets, from among the competitive allocations that are instantaneously competitive at the inflection at time $t$, we must determine which allocations will remain competitive going forward. During a period between two inflection points, the real competitive allocation set $F^*$ contains all the allocations that give the maximal value.

Figure 4.1 shows the values of the competitive allocations during the course of solving the example of Table 2.1. The four allocations in which bundle ABC is allocated begin as competitive allocations, but at $t = \frac{2}{3}$ 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, one of the allocations 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 Agent 3 and Agent 4 give up.

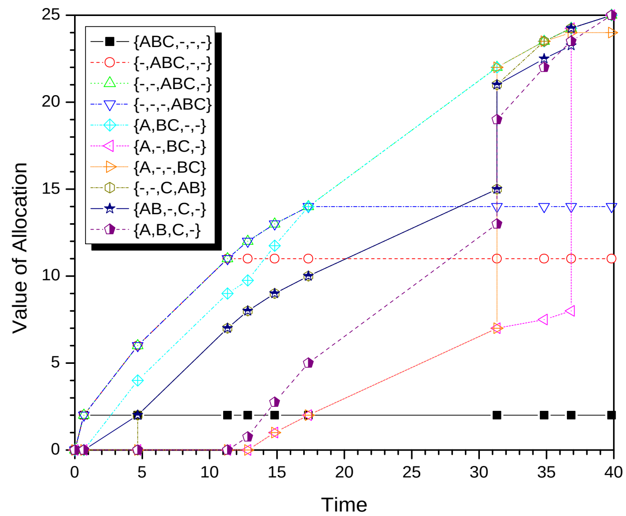


Figure 4.1: SCPA allocation values of solving the example in Table 2.1 by the PTA.

### 4.1.3 Attention Allocation Method

An agent's allocation of attention determines whether the bundle in a potential demand set remains in the actual demand set, and whether the potential competitive allocation is still competitive. There is a close tie between the demand sets and the competitive allocations, and it is not generally possible to separate them. Based on this, a mixed-integer linear program algorithm named the Attention Allocation Method for SCPA ($\text{AAM}_{\text{SCPA}}$) is developed to identify the actual demand sets, competitive allocations, and the allocation of attention. Mathematically,

$$\text{AAM}_{\text{SCPA}} : \{\hat{D}_i, \hat{F}^*\} \rightarrow \{D_i, F^*, \theta_{ib}\}.$$

### 4.1.4 Inflection Point Method

The PTA is an iterative process. In each iteration, AAM computes the agent attention allocation based on the agents' potential demand sets and the potential competitive allocations. Then, the slopes of bundle prices are computed from the agent attentions. After that, the Inflection Point Method (IPM) is used to get the next inflection point. The IPM computes the time interval by choosing the inflection points at which the agents' demand sets or the set of competitive allocations change. Three potential events define the end of an interval:

1. the prices/bids 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. the agents' surplus reaches zero, or
3. an allocation that was not formerly competitive reaches a value that makes it competitive.

The second kind of event can be thought of as a special case of the fist event if we introduce a null bundle. The valuation of the null bundle is zero. When the agent is attracted to the null bundle, it means that its surplus across all other bundles has become zero.

## 4.2 Mixed Integer Linear Programming

In this section, $\text{AAM}_{\text{SCPA}}$ is constructed from the bidding strategy to solve the agents' attention allocation problem.

Between time $t_1$ and $t_2$, we can compute the trajectory $\theta_b^{t_1 t_2}$ of a bundle as the summation of the attention being paid to it, which is denoted as $\theta_b$ if there is no ambiguity,

$$\theta_b^{t_1 t_2} = \sum_i \theta_{ib}.$$

Alternately, suppose $\pi_b^{t_1}$ and $\pi_b^{t_2}$ are the prices of bundle $b$ at times $t_1$ and $t_2$, respectively. Then the slope of bundle $b$ can be computed as

$$\theta_b^{t_1 t_2} = \frac{\pi_b^{t_2} - \pi_b^{t_1}}{t_2 - t_1}. \tag{4.1}$$

### 4.2.1 Constraints

**Proposition 4.2.1** *If both bundles $b$ and $c$ are in the demand set of agent $i$ at time $t_1$, and $b$ still remains in agent $i$'s demand set at time $t_2$ ($t_2 > t_1$) but $c$ does not, the slope of bundle $b$ is less than the slope of $c$,*

$$\theta_b < \theta_c.$$

**Proof.** At time $t_1$, since both $b$ and $c$ are in the demand set $D_i^{t_1}$, the surpluses of bundle $b$ and $c$ are the same,

$$v_i(b) - \pi_b^{t_1} = v_{ic} - \pi_c^{t_1}.$$

At time $t_2$, bundle $b$ is in the demand set $D_i^{t_2}$ and $c$ is not. From the definition of demand set, the surplus of bundle $b$ is larger than the slope of $c$,

$$v_{ib} - \pi_b^{t_2} > v_{ic} - \pi_c^{t_2}.$$

So, we get the following inequality by merging the two expressions above,

$$\pi_b^{t_2} - \pi_b^{t_1} < \pi_c^{t_2} - \pi_c^{t_1}.$$

Divide both sides by $t_2 - t_1$,

$$\frac{\pi_b^{t_2} - \pi_b^{t_1}}{t_2 - t_1} < \frac{\pi_c^{t_2} - \pi_c^{t_1}}{t_2 - t_1}.$$

Then substitute for equation (4.1), we get

$$\theta_b < \theta_c,$$

which proves the proposition. ■

**Proposition 4.2.2** *If both bundles $b$ and $c$ are continuously in the demand set $D_i$ of agent $i$ between time $t_1$ and $t_2$, the slope of bundle $b$ is equal to the slope of $c$,*

$$\theta_b = \theta_c.$$

**Proof.** Similar as the proof of Proposition 4.2.1. ■

Active agents have one unit of attention to allocate. We can easily account for agents that no longer achieve positive surplus on any bundle (and therefore have empty demand sets) by giving them zero attention to allocate. 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. Because the auction rules require agents to bid on just one bundle in a round, agent $i$'s attention has the following constraints,

$$\sum_{b \in \hat{D}_i} \theta_{ib} + \theta_{i\text{pass}} = K_i, \tag{4.2}$$

$$0 \leq \theta_{ib} \leq 1$$

An agent may stop bidding on a bundle if the bundle's price is increasing more quickly than other bundles in its demand set. Thus, we differentiate between the *potential demand set* at time $t_1$ and the actual demand set maintained by the agent in the interval $[t_1, t_2]$. In other words, although all of the identified bundles are instantaneously active at the inflection at time $t_1$, we are concerned with determining which will remain active going forward. For example, consider three agents named Agent 1, Agent 2, and Agent 3 and the two items, A and B. At a particular time $t_1$, Agent 1's potential demand set $D_1$ is {A, AB}, and Agents 2 and 3's potential demand sets are the same, $D_2 = D_3$={AB}. We know that bundle AB's price increases faster than bundle A's price because both Agents 2 and 3 keep bidding on bundle AB and only Agent 1 bids on bundle A. Therefore, bundle AB is not in Agent 1's demand set after time $t_1$ and Agent 1 does not bid further on AB.

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, it is the set of bundles that satisfy (2.1) at time $t$. The integer variable $y_{ib}$ takes the value of 1 if bundle $b$ will remain in agent $i$'s demand set during the following interval, and 0 otherwise.

It is clear that $\forall i \in I$,

$$y_{ib} \quad \geq \quad \theta_{ib}, \quad \forall b \in \hat{D}_i. \tag{4.3}$$

Also, if $b, c \in \hat{D}_i$, from Proposition 4.2.1 and Proposition 4.2.2,

$$\text{if} \quad y_{ib} = 1 \quad \text{and} \quad y_{ic} = 1, \text{then} \quad \theta_b = \theta_c,$$

$$\text{if} \quad y_{ib} = 1 \quad \text{and} \quad y_{ic} = 0, \text{then} \quad \theta_b < \theta_c.$$

We can rewrite the above rules as an integer linear constraint, $\forall i \in I$ and $b, c \in \hat{D}_i$,

$$N(\theta_c - \theta_b) \geq 1 - y_{ic} - N^2(1 - y_{ib}), \tag{4.4}$$

where $N$ is a sufficiently large positive constant.

Now we turn our attention to the constraints that capture the influence of the competitive allocations. Let $\gamma_f^{t_1 t_2}$, which are as $\gamma_f$ if there is no ambiguity, be the slope of allocation $f$, then

$$\gamma_f^{t_1 t_2} = \frac{V^{t_2}(f) - V^{t_1}(f)}{t_2 - t_1}.$$

**Proposition 4.2.3** *If both allocation $f$ and $\hat{f}$ are in the competitive allocation set $F^*$ at time $t_1$, and $f$ is still in $F^*$ at time $t_2$ ($t_2 > t_1$) but $\hat{f}$ is not, the slope of $f$ is larger than the slope of $\hat{f}$,*

$$\gamma_f > \gamma_{\hat{f}}.$$

**Proof.** At time $t_1$, since both $f$ and $\hat{f}$ are in the competitive allocation set $F^*$, their values are the same at time $t_1$,

$$V^{t_1}(f) = V^{t_1}(\hat{f}).$$

At time $t_2$, $f \in F^*$ and $\hat{f} \notin F^*$. Thus the value of $f$ is larger than that of $\hat{f}$,

$$V^{t_2}(f) > V^{t_2}(\hat{f}).$$

By subtracting $V^{t_1}(f)$ and $V^{t_1}(\hat{f})$ on both sides of the above expression, we get

$$V^{t_2}(f) - V^{t_1}(f) > V^{t_2}(\hat{f}) - V^{t_1}(\hat{f}).$$

Thus,

$$\frac{V^{t_2}(f) - V^{t_1}(f)}{t_2 - t_1} > \frac{V^{t_2}(\hat{f}) - V^{t_1}(\hat{f})}{t_2 - t_1}.$$

We finish our proof by substituting $\gamma_f$ and $\gamma_{\hat{f}}$ into above expression,

$$\gamma_f^t > \gamma_{\hat{f}}^t.$$

Therefore, the slope of $f$ is larger than the slope of $\hat{f}$. ■

**Proposition 4.2.4** *If both allocation $f$ and $\hat{f}$ are in the competitive allocation set $F^*$ between time $t_1$ and $t_2$, their slopes are equal,*

$$\gamma_f = \gamma_{\hat{f}}.$$

**Proof.** Similar to the proof of Proposition 4.2.3. ■

Let $\beta_f \in [0,1]$ be the frequency with which allocation $f$ is announced as the winner, enabling the members of that allocation to pass. At every iteration of the simulation method, one and only one of the competitive allocations must be announced as the winning bundle. Thus,

$$\sum_{f \in F^*} \beta_f = 1. \tag{4.5}$$

The potential competitive allocation set, $\hat{F}^*$, is the union of the previous competitive allocations and those allocations being introduced at this inflection point. The integer variable $x_f$ takes the value of 1 if allocation $f$ remains competitive during the interval, and 0 otherwise.

If allocation $f$ is not competitive, $\beta_f = 0$. In other words, the following constraint holds,

$$\beta_f \leq x_f. \tag{4.6}$$

From Proposition 4.2.3 and Proposition 4.2.4,

$$\begin{aligned} &\text{if} \quad x_f = 1 \quad \text{and} \quad x_{\hat{f}} = 1, \text{then} \quad \gamma_f = \gamma_{\hat{f}}, \\ &\text{if} \quad x_f = 1 \quad \text{and} \quad x_{\hat{f}} = 0, \text{then} \quad \gamma_f \geq \gamma_{\hat{f}}. \end{aligned}$$

An integer linear form of these logical constraints is

$$\gamma_{\hat{f}} - \gamma_f + N x_f \leq N, \quad \forall f, \hat{f} \in \hat{F}^*. \tag{4.7}$$

The slope of the allocation $f$ depends upon which elements of the potential demand sets are retained in the actual demand sets of the agents. Formally,

$$\gamma_f = \sum_i \theta_{f_i} y_{i f_i},$$

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

$$\alpha_{ib} \leq N y_{ib}, \tag{4.8}$$

$$\alpha_{ib} \leq \theta_b, \tag{4.9}$$

$$\theta_b - \alpha_{ib} + N y_{ib} \leq N. \tag{4.10}$$

The introduction of the variable $\alpha_{ib}$ allows us to express the slope of an allocation as

$$\gamma_f = \sum_{i \in f} \alpha_{i f_i}.$$

Every competitive allocation will have some frequency 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. Suppose the competitive allocations are {AB, –} and {–, AB}. In those rounds in which Agent 2 bids on A, {AB, –} is the only competitive allocation. In the alternate rounds, Agents 1 and 2 may increase their bids on AB, and {AB, –} and {–, AB} are equally announced. So, Agent 1 has a 0.75 frequency of being declared the current winner. Therefore, Agent 1 will bid on AB a quarter of the time and pass three quarters of the time, and 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 suggests a connection between agent $i$'s behavior and the frequency with which agent $i$ is not a member of the winning allocation. Agent $i$ passes whenever one of the competitive allocations, which contain agent $i$, is announced. Therefore, we have the following proposition:

**Proposition 4.2.5** *Let indicator function $1_{[i \in f]}$ take the value 1 if $i$ is allocated a bundle in $f$, and 0 otherwise. The agent's pass ratio is equal to the summation of the frequency of all competitive allocations that includes this agent,*

$$\theta_{i\text{pass}} = \sum_{f \in \hat{F}^*} K_i \cdot 1_{[i \in f]} \cdot \beta_f.$$

### 4.2.2 The Mathematical Model

The trajectory algorithm outlined above determines three variables: (1) which bundles from agent's demand set go forward (those that have $y_{ib} = 1$), (2) which allocations are competitive (those for which $x_f = 1$), and (3) 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 the slopes are necessary to determine the duration of the interval. The complete $\text{AAM}_{\text{SCPA}}$ constructed from equations (4.3)–(4.11) is shown in Figure 4.2 where $N$ is a large positive number.

## 4.3 Duration of The Time Interval

The last component of the algorithm is to compute the duration of a time interval. There are three kinds of events that could make an agent change its behavior: a new bundle cames into an agent's demand set; an agent's demand set becomes empty; and a non-competitive allocation becomes competitive. We calculate the first event.

We need to compute the first change in demand sets among all agents. Consider a bundle, $c$, which is not in agent $i$'s current demand set, and a bundle, $b$, which is. From the definition of the demand set, the surplus of bundle $b$ is greater than the surplus of bundle

$$
\mathrm{AAM}_{\mathrm{SCPA}}
\begin{cases}
y_{ib} \geq \theta_{ib}, \quad \forall i \in I,\ b \in \hat{D}_i & (4.11) \\
N(\sum_{j \in I} \theta_{jb} - \sum_{j \in I} \theta_{jc}) + N^2(1 - y_{ic}) \geq 1 - y_{ib}, \quad \forall i \in I,\ b, c \in \hat{D}_i & (4.12) \\
\alpha_{ib} \leq N y_{ib}, \quad \forall i \in I,\ b \in \hat{D}_i & (4.13) \\
\alpha_{ib} - \sum_{j \in I} \theta_{jb} \leq 0, \quad \forall i \in I,\ b \in \hat{D}_i & (4.14) \\
\sum_{j \in I} \theta_{jb} - \alpha_{ib} \leq N(1 - y_{ib}), \quad \forall i \in I,\ b \in \hat{D}_i & (4.15) \\
\sum_{i \in \hat{f}} \alpha_{i\hat{f}_i} - \sum_{i \in f} \alpha_{if_i} \leq N(1 - x_f), \quad \forall f, \hat{f} \in \hat{F}^* & (4.16) \\
\beta_f \leq x_f, \quad \forall f \in \hat{F}^* & (4.17) \\
\sum_{f \in \hat{F}^*} \beta_f = 1, & (4.18) \\
K_i \sum_{f \in \hat{F}^*} 1_{[i \in f]} \cdot \beta_f + \sum_{b \in \hat{D}_i} \theta_{ib} = K_i, \quad \forall i \in I & (4.19) \\
\beta_f \geq 0, \quad x_f \in \{0, 1\}, \quad \forall f \in \hat{F}^* & \\
\alpha_{ib} \geq 0, \quad 0 \leq \theta_{ib} \leq 1, \quad y_{ib} \in \{0, 1\}, \quad \forall i \in I,\ b \in \hat{D}_i &
\end{cases}
$$

Figure 4.2: Mathematical model of the AAM of the PTA for SCPA.

$c$. However bundle $c$ may come into the agent's demand set if the slope of $b$ is greater than the slope of $c$. In other words, when the surplus of $b$ decreases faster than that of $c$, their surplus could eventually be the same. If $\theta_b > \theta_c$, the amount of time it will take for $c$ to become attractive to agent $i$ is

$$\Delta t_i^{b,c} = \frac{v_{ib} - \pi_b - v_{ic} + \pi_c}{\theta_b - \theta_c}.$$

Now consider the case where the surplus of an agent becomes zero and the agent drops out of the bidding. Let bundle $b$ be a member of agent $i$'s demand set with current bid $r_{ib}$. The amount of time that it will take for agent $i$'s surplus to become zero is

$$\Delta t_i^b = \frac{v_{ib} - \pi_b}{\theta_b}$$

Therefore, the duration during which agent $i$'s demand set does not change is, for any $b$ in $D_i$,

$$\Delta t_i^{DS} = \min\left\{\Delta t_i^b, \min_{c \notin D_i}\{\Delta t_i^{b,c}\}\right\}.$$

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

$$\Delta t^{DS} = \min_i \left\{\Delta t_i^{DS}\right\}.$$

The other cause of an inflection is when a non-competitive allocation becomes competitive. We need to compute the first non-competitive allocation that will become competitive under the current allocation of attention. Competitive allocation, $f$, will collide with non-competitive allocation, $\hat{f}$, when the values of the two become equal. For such an event to happen, the slope of $\hat{f}$ must be larger than the slope of $f$. So, if $\gamma_{\hat{f}} > \gamma_f$, the collision will occur in $\Delta t^{f,\hat{f}}$ time increments, where

$$\Delta t^{f,\hat{f}} = \frac{V(f) - V(\hat{f})}{\gamma_{\hat{f}} - \gamma_f}.$$

For $f$ in $F^*$, among the $\Delta t^{f,\hat{f}}$s, we find the first non-competitive allocation that will become competitive

$$\Delta t^{CA} = \min_{\hat{f} \notin F^*}\{\Delta t^{f,\hat{f}} | \Delta t^{f,\hat{f}} > 0\}.$$

Therefore, the next inflection point is determined by taking the minimum of $\Delta t^{DS}$ and $\Delta t^{CA}$,

$$\Delta t = \min\{\Delta t^{DS}, \Delta t^{CA}\}.$$

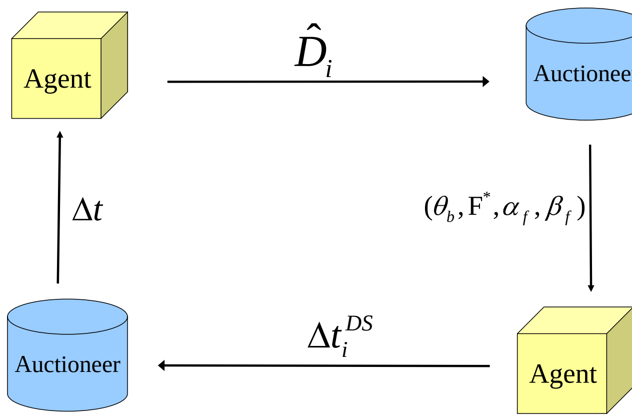


Figure 4.3: Framework of the PTA.

## 4.4 The Price Trajectory Algorithm

The PTA is an iterative process with interactions between the agents and the auctioneer. Figure 4.3 shows the iterative process of the $\text{PTA}_{\text{SCPA}}$.

The algorithm starts at an initial inflection point with announced bundle prices, $\pi$. Assume all agents follow the straightforward bidding strategy. Based on the bundle prices at the inflection point, the agents figure out their potential demand set, $\hat{D}_i$, and submit it to the auctioneer.

The auctioneer receives the potential demand sets, $\hat{D}_i$, from all agents. Meanwhile, she computes the potential competitive allocation set, $\hat{F}^*$. Next, she constructs the $\text{AAM}_{\text{SCPA}}$ model based on the potential demand sets and the potential competitive allocation set to solve the agents' actual demand set, $D_i$, the agents' allocation of the attention, $\theta_{ib}$, among bundles in demand set, the competitive allocation set, $F^*$, the slope of competitive allocation, $\alpha_f$, and the frequency, $\beta_f$, of the competitive allocations being selected. Then, she gets the bundle slope, $\theta_b = \sum_i \theta_{ib}$. At last, she announces the following information to all agents: $\theta_b, F^*, \alpha_f$, and $\beta_f$.

Each agent computes its demand set $D_i$ based on the bundle slopes, $D_i = \{b \in \hat{D}_i : \theta_b \leq \theta_c, \forall c \in \hat{D}_i\}$ and its pass value $\theta_{i\text{pass}} = \sum\limits_{f \in F^*} 1_{[i \in f]} \cdot \beta_f$. After that, the agent calculates the time duration, $\Delta t_i^{DS}$, during which its demand set does not change, and submits the time duration to the auctioneer.

Based on the time duration during which the agent's demand set does not change and the time duration during which the competitive allocation set does not change, the auctioneer computes the next inflection point $\Delta t$ and then gets the bundle prices at the next inflection point. Finally, she announces the new bundle prices to all agents. The processes move to the next iteration.

The termination condition of the $\text{PTA}_{\text{SCPA}}$ is that all non-winning agents submit an empty potential demand set to the auctioneer.

## 4.5 A Worked Example

The bundle prices of applying the price trajectory algorithm to the example in Table 3.1 are shown in Figures 4.4 where the x-axis indicates the time of the auction processes and the y-axis is the bundle price. We can see that the bundle price trajectories in this figure perfectly match the trajectories in Figure 3.1. The $\text{PTA}_{\text{SCPA}}$ only needs fourteen steps to find the winners and the payments in this example, while the simulation needs thousands of rounds to reach the final allocation and payments. In this example, the final allocation is {B, –, A, C} and Agent 1's payment is 10, Agent 3's payment is 19, and Agent 4's payment is 17.

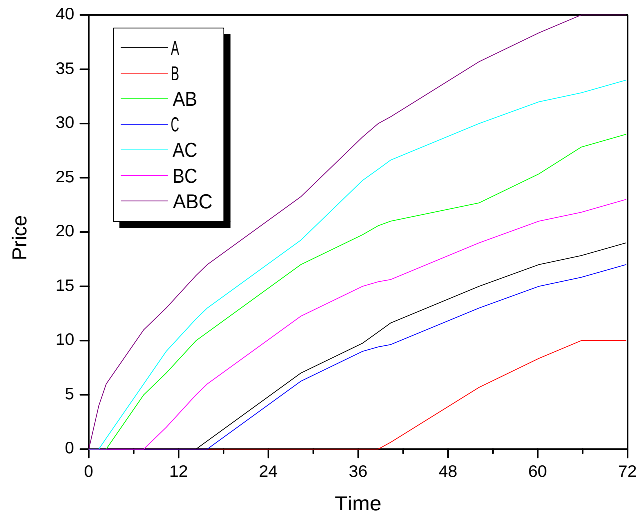


Figure 4.4: SCPA bundle prices of solving the example in Table 3.1 by the PTA.

The values of competitive allocations while applying the price trajectory algorithm to the example in Table 3.1 are shown in Figure 4.5 where the x-axis indicates the time of the auction processes and the y-axis is the value of the allocation. This figure contains all allocations that have chance to be competitive. The top curve represents the value of the competitive allocations. From the figure, we can see that the final allocation, {B, –, A, C}, is below the top curve until the last step. At the beginning of the last step, its value jumps to meet the competitive allocation curve and it becomes competitive. Let us take a look of an interesting allocation, {–, ABC, –, –}. It is competitive at in first step, but it becomes non-competitive in the second, third, and fourth steps because Agent 2 does not bid on bundle ABC in these steps. It is interesting that Agent 2 bids on ABC in Step 5 again, thus {–, ABC, –, –} jumps into the competitive allocation set and remains competitive until Step 8. In Step 9, it is non-competitive because Agent 2 does not bid on bundle ABC. And it comes back to the potential competitive allocation set at the beginning of Step 11 again

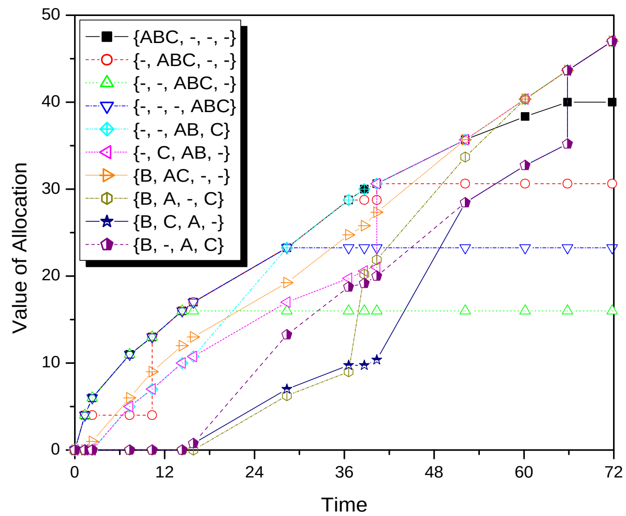


Figure 4.5: SCPA allocation values of solving the example in Table 3.1 by the PTA.

and then leaves the competitive allocation set forever.

Table 4.1 and 4.2 show the computations involved at eight of the fourteen steps required to solve the auction. Each step shows the following:

- the prices at the designated time,
- the potential competitive allocations (with those that remain competitive in the interval designated with an asterisk and the value of the frequency of being selected),
- each agent's potential demand set,
- and each agent's allocation of attention.

The bundles that are determined to be in each agent's actual 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 in $D_i^t$ have empty cells. The attention columns are summed to give the trajectories of the prices going forward.

Table 4.1: Some steps of solving the example in Table 3.1 by applying SCPA version of the PTA.

Step 1, $t = 0$

| | | Prices: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 0$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{ABC, –, –, –} 1/4 | 1 | ABC | | | | | | | 0.75 | 0.25 |
| *{–, ABC, –, –} 1/4 | 2 | ABC | | | | | | | 0.75 | 0.25 |
| *{–, –, ABC, –} 1/4 | 3 | ABC | | | | | | | 0.75 | 0.25 |
| *{–, –, –, ABC} 1/4 | 4 | ABC | | | | | | | 0.75 | 0.25 |
| | | $\theta_b$ | 0 | 0 | 0 | 0 | 0 | 0 | 3 | |

Step 2, $t = 4/3$

| | | Prices: | 0 | 0 | 0 | 0 | 0 | 0 | 4 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 4$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{ABC, –, –, –} 1/3 | 1 | ABC | | | | | | | 2/3 | 1/3 |
| {–, ABC, –, –} | 2 | AC, ABC | | | | | 1 | | | |
| *{–, –, ABC, –} 1/3 | 3 | ABC | | | | | | | 2/3 | 1/3 |
| *{–, –, –, ABC} 1/3 | 4 | ABC | | | | | | | 2/3 | 1/3 |
| | | $\theta_b$ | 0 | 0 | 0 | 0 | 1 | 0 | 2 | |

⋮ ⋮ ⋮

Step 5, $t = 10\frac{1}{3}$

| | | Prices: | 0 | 0 | 7 | 0 | 9 | 2 | 13 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 13$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{ABC, –, –, –} 5/8 | 1 | ABC | | | | | | | 3/8 | 5/8 |
| *{–, ABC, –, –} 1/8 | 2 | AC, ABC | | | | | 3/4 | | 1/8 | 1/8 |
| *{–, –, ABC, –} 1/8 | 3 | AB, ABC | | | 3/4 | | | | 1/8 | 1/8 |
| *{–, –, –, ABC} 1/8 | 4 | BC, ABC | | | | | | 3/4 | 1/8 | 1/8 |
| | | $\theta_b$ | | | 3/4 | | 3/4 | 3/4 | 3/4 | |

⋮ ⋮ ⋮

Step 9, $t = 36\frac{7}{12}$

| | | Prices: | 9.75 | 0 | 19.7.5 | 9 | 24.75 | 15 | 28.75 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 28.75$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{–, –, AB, C} 3/5 | 1 | ABC | | | | | | | 0.6 | 0.4 |
| *{ABC, –, –, –} 2/5 | 2 | A, AC, ABC | 0.5 | | | | 0.5 | | | |
| {–, ABC, –, –} | 3 | A, AB | | | 0.4 | | | | | 0.6 |
| | 4 | C, BC | | | | 0.2 | | 0.2 | | 0.6 |
| | | $\theta_b$ | 0.5 | | 0.4 | 0.2 | 0.5 | 0.2 | 0.6 | |

Table 4.2: Continuation of Table 4.1.

Step 10, $t = 38\frac{2}{3}$

| | | Prices: | $10\frac{19}{24}$ | 0 | $20\frac{7}{12}$ | $9\frac{5}{12}$ | $25\frac{19}{24}$ | $15\frac{5}{12}$ | 30 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 30$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| | 1 | B, ABC | | 3/8 | | | | | 3/8 | 1/4 |
| *{–, –, AB, C} 3/4 | 2 | A, AC | 1/2 | | | | 1/2 | | | |
| *{ABC, –, –, –} 1/4 | 3 | AB | | | 1/4 | | | | | 3/4 |
| | 4 | C, BC | | | | 1/8 | | 1/8 | | 3/4 |
| | | $\theta_b$ | 1/2 | 3/8 | 1/4 | 1/8 | 1/2 | 1/8 | 3/8 | |

Step 11, $t = 40\frac{1}{3}$

| | | Prices: | $11\frac{5}{8}$ | $\frac{5}{8}$ | 21 | $9\frac{5}{8}$ | $26\frac{5}{8}$ | $15\frac{5}{8}$ | $30\frac{5}{8}$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 30\frac{5}{8}$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{–, C, AB, –} 3/7 | 1 | B, ABC | | 3/7 | | | | | 3/7 | 1/7 |
| *{–, –, AB, C} 3/7 | 2 | A, C, AC, ABC | 2/7 | | | 0 | 2/7 | | | 3/7 |
| *{ABC, –, –, –} 1/7 | 3 | AB | | | 1/7 | | | | | 6/7 |
| {–, ABC, –, –} | 4 | C, BC | | | | 2/7 | | 2/7 | | 3/7 |
| | | $\theta_b$ | 2/7 | 3/7 | 1/7 | 2/7 | 2/7 | 2/7 | 3/7 | |

⋮ ⋮ ⋮

Step 13, $t = 60\frac{7}{48}$

| | | Prices: | 17 | $8\frac{17}{48}$ | $25\frac{17}{48}$ | 15 | 32 | 21 | $38\frac{17}{48}$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 40\frac{17}{48}$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{B, A, –, C} 3/7 | 1 | B, ABC | | 2/7 | | | | | 2/7 | 3/7 |
| *{B, AC, –, –} 0 | 2 | A, C, AC | 1/7 | | | 1/7 | 1/7 | | | 4/7 |
| *{–, C, AB, –} 1/7 | 3 | AB | | | 3/7 | | | | | 4/7 |
| *{–, –, AB, C} 3/7 | 4 | C, BC | | | | 0 | | 1/7 | | 6/7 |
| | | $\theta_b$ | 1/7 | 2/7 | 3/7 | 1/7 | 1/7 | 1/7 | 2/7 | |

Step 14, $t = 65\frac{29}{32}$

| | | Prices: | $17\frac{79}{96}$ | 10 | $27\frac{79}{96}$ | $15\frac{79}{96}$ | $32\frac{79}{96}$ | $21\frac{79}{96}$ | 40 | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f \mid V(f) = 43\frac{31}{48}$ | Agent | $\hat{D}_i$ | A | B | AB | C | AC | BC | ABC | pass |
| *{B, A, –, C} 1/5 | 1 | B, ABC | | 0 | | | | | 0 | 1 |
| *{B, C, A, –} 1/5 | 2 | A, C, AC | 1/5 | | | 1/5 | 1/5 | | | 2/5 |
| *{B, –, A, C} 3/5 | 3 | A, AB | 0 | | 1/5 | | | | | 4/5 |
| *{–, C, AB, –} 0 | 4 | C, BC | | | | 0 | | 1/5 | | 4/5 |
| *{–, –, AB, C} 0 | | $\theta_b$ | 1/5 | 0 | 1/5 | 1/5 | 1/5 | 1/5 | 0 | |

Step 15, $t = 71\frac{19}{24}$

| | Prices: | 19 | 10 | 29 | 17 | 34 | 23 | 40 |
|---|---|---|---|---|---|---|---|---|

Agent 2 stops bidding and {B, –, A, C} wins.

At $t = 0$, the prices are zero and bundle ABC is the sole element in each agent's demand set. Since there are four competitive allocations, and no other bundles distracting the agents, each competitive allocation wins one fourth of the time. At $t = 4/3$, the bundle AC enters Agent 2's demand set. In Step 2, Agent 2 focuses its attention on the new elements and bundle ABC leaves Agent 2's demand set because the other three agents bid on ABC and make ABC's slope larger than the slope of AC. In other words, Agent 2 does not bid on ABC because the surplus of ABC is less than that of AC.

An interesting event happens at Step 5. Bundle ABC comes back to Agent 2's demand set and from this step to Step 8, Agent 2 bids on ABC. Bundle ABC leaves the demand set of Agent 2 at Step 9. It returns to Agent 2's potential demand set at the beginning of Step 11. But in this step, it does not remain in the demand set and leaves the demand set forever.

We take a look of Step 11 where we see a complex allocation of attention across the bundles, yet one that continues to satisfy the constraints of the AAM and exactly matches the simulation pattern. This step starts when bundle C and ABC enter into Agent 2's potential demand set. Within the step, Agent 2 does not bid on bundle ABC but bids on bundle C that has a price value of $9\frac{5}{8}$ at the starting time. It makes the value of the allocation {–, C, AB, –} jump to hit the value of the competitive allocation. It is also interesting to note that Step 11 ($t = 38\frac{2}{3}$) is stopped by the allocations {B, AC, –, –} entering the competitive allocation set because the slope of {B, AC, –, –} is $\frac{5}{7}$ that is larger than $\frac{3}{7}$, the slope of the competitive allocations. The effect of this event is clearly visible in Figure 4.5. Even though the frequency of the allocation {B, AC, –, –} being selected is zero in Step 13, it is still competitive because all the members in the allocation bid on the corresponding bundles.

At Step 14 ($t = 65\frac{29}{32}$), we see an interesting effect when bundle A enters Agent 3's demand set again because it was in Agent 3's demand set and left at Step 9. The price of C at the time is $15\frac{79}{96}$, so when Agent 3 joins the bidding, the allocations {B, C, A, –} and {B, –, A, C} become competitive, as seen by the dramatic rise in their values in Figure 4.5. As a side effect, Agent 1 finds that it is always winning, so it puts 0 attention on B and ABC. After Step 14, Agent 2 drops out and only competitive allocation {B, –, A, C} remains. Note that every remaining active agent receives a bundle in this allocation, and thus does not need to bid further.

## 4.6 Multi-Stage Proxy Auction

In proxy auctions, bidders give their bundle valuations to their agents and let the agents attend the auction. After the auction finishes, the agents report the final winners and bundle prices to their bidders. Now, each bidder has two choices: either update the bundle valuations to higher values, or keep the values, and lets the agent attend the auction again. We call this a multi-stage auction because of the interaction between the bidder and his agent.

### 4.6.1 Interaction between Bidder and Agent

Proxy bidding means that a bidder gives his valuation on bundles to his agent. Then the agent takes the role of this bidder and participates in the iterative auction. In general, the agent starts from a bid that is lower than the bidder's valuation and increases the bid in the following rounds. The maximum bid from the agent cannot be greater than the valuation from the bidder.

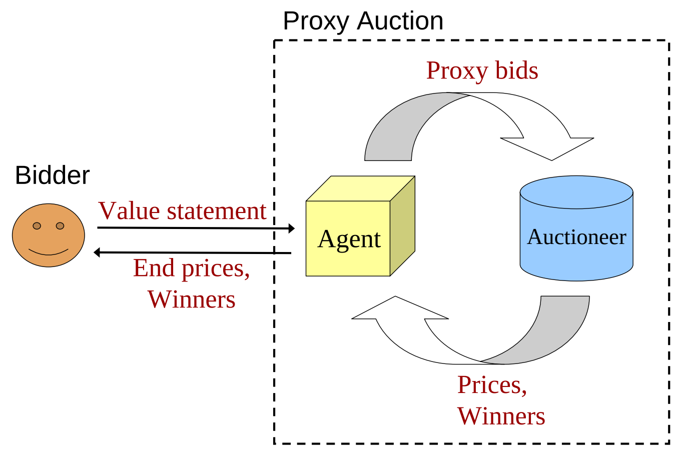


Figure 4.6: Framework of multi-stage auction.

In a multi-stage proxy auction [46], each stage includes one proxy auction. After the proxy auction finishes, the agent reports the final competitive allocation and payments to the bidder. If the bidder wants to increase his valuation and let his agent go back to auction, a new stage starts and the proxy auction starts to run again. The bundle prices start from \$0 in the auction of the first stage. In stage $n > 1$, the bundle prices always start at the final bundle prices announced in stage $n - 1$. Figure 4.6 show how the multi-stage auction works.

### 4.6.2 An Example of Multi-Stage Proxy Auction

Consider a multi-stage variation of the example in Table 2.1. In the following, the choices of statements made to the proxy agents were chosen to illustrate the algorithm and not to achieve strategic goals. The buyers are not required to express values for all bundles as they communicate with their respective proxy agents, nor are they required to reveal their true valuations for bundles. Suppose that going into the auction, Buyer 1 believes

Table 4.3: Multi-stage auction example.

| | A | B | AB | C | AC | BC | ABC |
|---|---|---|---|---|---|---|---|
| | Stage 1 | | | | | | |
| Buyer 1 | 8 | | 8 | | 8 | | 8 |
| Buyer 2 | | | 14 | | | 14 | 14 |
| Buyer 3 | | | | 8 | 15 | 15 | 15 |
| Buyer 4 | | | | | | | 18 |
| | Stage 2 | | | | | | |
| Buyer 1 | 8 | | 8 | | 8 | | 8 |
| Buyer 2 | | 9 | 15 | | | 18 | 20 |
| Buyer 3 | | | | 8 | 15 | 15 | 15 |
| Buyer 4 | | 7 | 16 | | 16 | 16 | 20 |
| | Stage 2 | | | | | | |
| Buyer 1 | 8 | | 8 | | 8 | | 8 |
| Buyer 2 | | 9 | 15 | | | 18 | 20 |
| Buyer 3 | | | | 9 | 15 | 17 | 25 |
| Buyer 4 | | 7 | 16 | | 16 | 16 | 20 |

that he will end up with A, Buyer 2 expects to get B and one other item, Buyer 3 expects C, and Buyer 4 anticipates getting the combination ABC. Based on these expectations, the buyers reveal the information shown in Stage 1 of Table 4.3. The $\text{PTA}_{\text{SCPA}}$ computes the allocation {A, -, BC, -} with supporting prices {7.833, 0, 14, 7, 14, 14, 18}. Because Buyers 2 and 4 are not winning anything after the first stage, they communicate revised bids to their respective proxy agents, as shown in Stage 2 of Table 4.3. Based on the stage 2 bids, the $\text{PTA}_{\text{SCPA}}$ computes the allocation {A, BC, -, -} with supporting prices {8, 7, 16, 8, 16, 16, 20}. Now Buyers 3 and 4 are not winning and choose to send the revised bids shown in Stage 3 to their respective proxy agents. Based on the Stage 3 bids, the auctioneer computes that both allocations {A, BC, -, -} and {A, B, C, -} produce the same value with supporting prices {8, 8, 16, 9, 16, 17, 25}. The trajectories of the bundle prices across all three stages are shown in Figure 4.7.

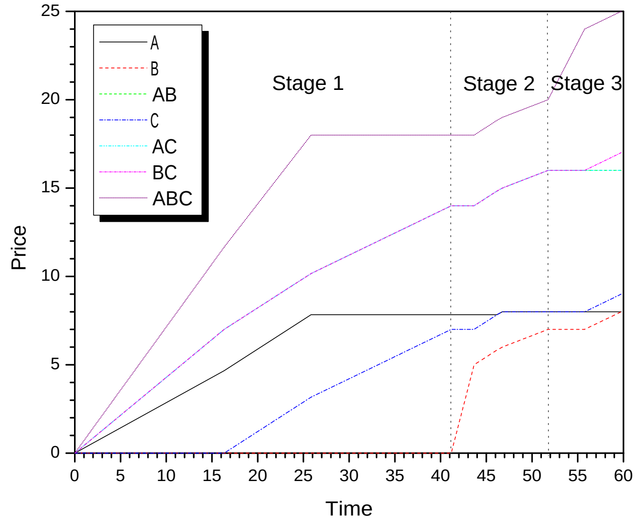


Figure 4.7: SCPA bundle prices of solving the multi-stage auction example in Table 4.3 by the PTA.

# Chapter 5

# Correctness of the Price Trajectory Algorithm and Computational Results

In this chapter, I show that the Simple Combinatorial Proxy Auction (SCPA) version of the Price Trajectory Algorithm ($\mathrm{PTA_{SCPA}}$) is a correct implementation of straightforward bidding as the increment $\delta$ goes to zero. In Section 5.2, I discuss the the computational complexity of $\mathrm{PTA_{SCPA}}$ and present that the mixed integer linear program of attention allocation method can be simplified into a linear program when some conditions are satisfied. I compare the computation time of the ($\mathrm{PTA_{SCPA}}$) and the simulation with different problem sizes in Section 5.3. Finally, I make comparisons between $\mathrm{PTA_{SCPA}}$ and other approaches in the last of this chapter.

## 5.1 Correctness of Price Trajectory Algorithm

From the simulation method in Chapter 3, a bundle price of $\pi_b^n$ at simulation round $n$ is defined by $\pi_b^n = \max_{i\in I}\{r_{ib}^n\}$. Agent $i$'s surplus $s_i^n$ at round $n$ is defined by $s_i^n = \max_{b\in B}\{s_{ib}^n\}$. The expression $s_{ib}^n$ is agent $i$'s bundle surplus computed as $s_{ib}^n = v_{ib} - \pi_b^n$. Therefore, $s_i^n \geq s_{ib}^n$ implies $s_i^n - (v_{ib} - \pi_b^n) \geq 0$ for all $b \in B$.

Agent $i$'s demand set of $D_i^n$ at round $n$ is the collection of all bundles that maximize the agent's bundle surplus at the current bundle prices $\pi^n$. In the context of the simulation, we relax the definition of (2.1) as

$$D_i^n = \{b : v_{ib} - \pi_b^n > 0 \text{ and } s_i^n - (v_{ib} - \pi_b^n) < K\delta\},$$

where $\delta$ is the bidding increment, and $K$ is a small constant number. For example, suppose four agents are interested in bundle $b$, whose current price is $\pi_b^n = 10$. All agents value the bundle at 15 and currently bid 10, which makes their surplus equal to 5. If Agent 1 passes twice in the next two rounds and the others bid on $b$, then $\pi_b^{n+2} = 10 + 2\delta$. If Agent 1 is not winning at round $n+2$ and although $v_1(b) - \pi_b^{n+2} = 5 - 2\delta < s_1^{n+2}$, then bundle $b$ is still considered in Agent 1's demand set. Therefore, $K \geq 2$.

**Lemma 5.1.1** *Suppose bundle $b_0$ is in agent $i_0$'s demand set, $D_{i_0}^n$, at round $n$. Then agent $i_0$'s bid price on bundle $b_0$ is close to the price of bundle $b_0$.*

$$0 \leq \pi_{b_0}^n - r_{i_0b_0}^n < K\delta.$$

**Proof.** Because bundle $b_0$'s price is $\pi_{b_0}^n = \max_{i\in I}\{r_{ib_0}^n\}$, the price is greater than or equal to the bid of the particular agent, $i_0$. That is, $\pi_{b_0}^n \geq r_{i_0b_0}^n$. Therefore, the expression becomes

$$\pi_{b_0}^n - r_{i_0b_0}^n \geq 0. \tag{5.1}$$

From the preceding definition of the demand set, $D_{i_0}^n$, and $b_0 \in D_{i_0}^n$, the expression is

$$v_{i_0 b_0} - \pi_{b_0}^n > s_{i_0}^n - K\delta. \tag{5.2}$$

Because $s_{i_0}^n = \max_{b \in B}\{s_{i_0 b}^n\}$, $s_{i_0}^n$ is greater than or equal to the surplus of the particular bundle, $b_0$. That is,

$$s_{i_0}^n \geq v_{i_0 b_0} - \pi_{b_0}^n.$$

By substituting $\pi_{b_0}^n \geq r_{i_0 b_0}^n$ into the preceding expression, we get

$$s_{i_0}^n \geq v_{i_0 b_0} - r_{r_0 b_0}^n. \tag{5.3}$$

Combining (5.2) and (5.3) gives

$$v_{i_0 b_0} - \pi_{b_0}^n > v_{i_0 b_0} - r_{r_0 b_0}^n - K\delta.$$

The preceding expression implies

$$\pi_b^n - r_{ib}^n < K\delta. \tag{5.4}$$

Therefore, from (5.1) and (5.4), we get the following result:

$$0 \leq \pi_b^n - r_{ib}^n < K\delta.$$

That is, agent $i_0$'s bid price on bundle $b_0$ is close to bundle price $\pi_{b_0}$. ■

The preceding lemma states that if a bundle is in an agent's demand set, the agent's bid price is equal to the bundle price. In this sense, "equal" means that if $\delta$ goes to 0, the bid price converges to the bundle price.

The competitive allocation set $F^{*n}$ at round $n$ is the set of all allocations such that

$$F^{*n} = \left\{ f : V^n(f) = \max_{\forall f'}\{V^n(f')\} \right\}.$$

Let us look at the slopes of the bundles and competitive allocations that start from round $n$. Suppose $\Gamma$ is a small price expanse such that $\Gamma = \delta(m-n)$, where $m$ is a round number and $m > n$. In other words, a bundle price would increase by the scalar, $\Gamma$, if the bundle is bidden on $m-n$ times continuously. Consider the slopes, $\theta_b^{mn}$, between round $n$ and round $m$. It is defined by $\theta_b^{mn} = \frac{\pi_b^m - \pi_b^n}{\Gamma}$. Agent $i$'s bundle slope is defined as $\alpha_{ib}^{mn} = \theta_b^{mn}$ if bundle $b$ is in agent $i$'s demand set through round $n$ to $m$, $\alpha_{ib}^{mn} = 0$ otherwise. The slope of an allocation $f$ is defined by $\alpha_f^{mn} = \sum_{i \in I} \alpha_{if_i}^{mn}$. By substituting $\alpha_{if_i}^{mn}$, we get $\alpha_f^{mn} = \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} \theta_{f_i}^{mn}$.

**Proposition 5.1.2** *If both bundles $b_1$ and $b_2$ are in an agent $i$'s demand set at round $m$ and $n$, the slopes of bundle $b_1$ and $b_2$ satisfy the expression*

$$\left|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}\right| < \frac{2K}{m-n}.$$

**Proof.** From the definition of bundle slope, we have

$$|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}| = \left|\frac{\pi_{b_1}^m - \pi_{b_1}^n}{\Gamma} - \frac{\pi_{b_2}^m - \pi_{b_2}^n}{\Gamma}\right|.$$

The expression can be written as

$$|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}| = \frac{1}{\Gamma}|\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n|. \tag{5.5}$$

Substitute the following four zero value terms, $s_i^m - s_i^m$, $s_i^n - s_i^n$, $v_{ib_1} - v_{ib_1}$, and $v_{ib_1} - v_{ib_1}$ into $|\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n|$, we have

$$|\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n| = |\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n + s_i^m - s_i^m + s_i^n - s_i^n + v_{ib_1} - v_{ib_1} + v_{ib_1} - v_{ib_1}|.$$

Reorder the items in the preceding expression gives

$$\begin{aligned} |\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n| &= |(s_i^m - v_{ib_1} + \pi_{b_1}^m) - (s_i^n - v_{ib_1} + \pi_{b_1}^n) \\ &\quad -(s_i^m - v_{ib_2} + \pi_{b_2}^m) + (s_i^n - v_{ib_2} + \pi_{b_2}^n)|. \end{aligned} \tag{5.6}$$

Because the bundles $b_1$ and $b_2$ are in agent $i$'s demand set at round $n$ and $m$, and from the definition of the demand set, the expressions are

$$0 \leq s_i^m - v_{ib_1} + \pi_{b_1}^m < K\delta,$$

$$0 \leq s_i^n - v_{ib_1} + \pi_{b_1}^n < K\delta,$$

$$0 \leq s_i^m - v_{ib_2} + \pi_{b_2}^m < K\delta,$$

and

$$0 \leq s_i^n - v_{ib_2} + \pi_{b_2}^n < K\delta.$$

Substitute the preceding four expressions into (5.6), we get

$$|\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n| < 2K\delta. \tag{5.7}$$

Merge (5.5) and (5.7), we have

$$|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}| < \frac{2K\delta}{\Gamma}.$$

Because $\Gamma = \delta(m-n)$, the preceding expression becomes

$$|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}| < \frac{2K}{m-n}.$$

That is, the slopes of bundle $b_1$ and $b_2$ are close. ■

**Proposition 5.1.3** *If bundle $b_1$ and $b_2$ are in agent i's demand set $D_i^n$ at round $n$, and $b_1$ is in $D_i^m$ at round $m$, and $b_2$ is not in $D_i^m$, the following expression is true,*

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} < \frac{2K}{m-n} - \frac{s_i^m - v_i(b_2) + \pi_{b_2}^m}{\Gamma}.$$

**Proof.** Given the definition of the bundle slope, the expression becomes

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} = \frac{\pi_{b_1}^m - \pi_{b_1}^n}{\Gamma} - \frac{\pi_{b_2}^m - \pi_{b_2}^n}{\Gamma}.$$

The preceding expression can be written as

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} = \frac{1}{\Gamma}(\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n). \tag{5.8}$$

Following the same logic as in the proof of Proposition 5.1.2 gives

$$\begin{aligned} \pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n &= (s_i^m - v_{ib_1} + \pi_{b_1}^m) - (s_i^n - v_{ib_1} + \pi_{b_1}^n) \\ &\quad -(s_i^m - v_{ib_2} + \pi_{b_2}^m) + (s_i^n - v_{ib_2} + \pi_{b_2}^n). \end{aligned} \tag{5.9}$$

Because bundle $b_1$ and $b_2$ are in agent $i$'s demand set at round $n$, and given the definition of the demand set, the expressions are

$$0 \leq s_i^n - v_{ib_1} + \pi_{b_1}^n < K\delta$$

$$0 \leq s_i^n - v_{ib_2} + \pi_{b_2}^n < K\delta.$$

Because bundle $b_1$ is in agent $i$'s demand set at round $m$, the expression is

$$0 \leq s_i^m - v_{ib_1} + \pi_{b_1}^m < K\delta.$$

Substitute the preceding three expressions into (5.9), we get

$$\pi_{b_1}^m - \pi_{b_1}^n - \pi_{b_2}^m + \pi_{b_2}^n < 2K\delta - (s_i^m - v_{ib_2} + \pi_{b_2}^m). \tag{5.10}$$

From (5.8) and (5.10), the expression is

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} < \frac{2K\delta}{\Gamma} - \frac{s_i^m - v_{ib_2} + \pi_{b_2}^m}{\Gamma}.$$

Substitute Γ's value, $\delta(m-n)$, in the preceding expression, we get

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} < \frac{2K}{m-n} - \frac{s_i^m - v_i(b_2) + \pi_{b_2}^m}{\Gamma}.$$

Thus, the proposition is true. ■

**Proposition 5.1.4** *If both allocations $f$ and $\hat{f}$ are in the competitive allocation set at round $m$ and $n$, their slopes between $m$ and $n$ satisfy the expression*

$$\left|\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}\right| < \frac{2|I|K}{m-n},$$

*where $|I|$ is the number of agents.*

**Proof.** Because $f \in F^{*n}, \hat{f} \in F^{*n}$ and $f \in F^{*m}, and \hat{f} \in F^{*m}$, we have

$$V^n(f) = V^n(\hat{f}) \quad \text{and} \quad V^m(f) = V^m(\hat{f}). \tag{5.11}$$

Given the definition of the slope of the competitive allocation, the expression is

$$\alpha_f^{mn} = \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} \theta_{f_i}^{mn}.$$

Because $\theta_{f_i}^{mn} = \frac{\pi_{f_i}^m - \pi_{f_i}^n}{\Gamma}$, the preceding expression becomes

$$\alpha_f^{mn} = \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} \frac{\pi_{f_i}^m - \pi_{f_i}^n}{\Gamma}.$$

Therefore, the expression is

$$|\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}| = \left| \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} \frac{\pi_{f_i}^m - \pi_{f_i}^n}{\Gamma} - \sum_{i \in I, \hat{f}_i \in D_i^n, \hat{f}_i \in D_i^m} \frac{\pi_{\hat{f}_i}^m - \pi_{\hat{f}_i}^n}{\Gamma} \right|.$$

The preceding expression can be written as

$$\begin{aligned} |\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}| = \frac{1}{\Gamma} \quad & \left| \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} \pi_{f_i}^m - \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} \pi_{f_i}^n \right. \\ & \left. - \sum_{i \in I, \hat{f}_i \in D_i^n, \hat{f}_i \in D_i^m} \pi_{\hat{f}_i}^m + \sum_{i \in I, \hat{f}_i \in D_i^n, \hat{f}_i \in D_i^m} \pi_{\hat{f}_i}^n \right|. \end{aligned}$$

From Lemma 5.1.1, the preceding expression implies

$$\begin{aligned} |\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}| < \frac{1}{\Gamma} \quad & \left| \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} (r_{if_i}^m + K\delta) - \sum_{i \in I, f_i \in D_i^n, f_i \in D_i^m} r_{if_i}^n \right. \\ & \left. - \sum_{i \in I, \hat{f}_i \in D_i^n, \hat{f}_i \in D_i^m} r_{i\hat{f}_i}^m + \sum_{i \in I, \hat{f}_i \in D_i^n, \hat{f}_i \in D_i^m} (r_{i\hat{f}_i}^n + K\delta) \right|. \end{aligned}$$

Because $V^m(f) = \sum_{i\in I, f_i\in D_i^m} r_{if_i}^m$, $V^m(\hat{f}) = \sum_{i\in I,\hat{f}_i\in D_i^m} r_{i\hat{f}_i}^m$, $V^n(f) = \sum_{i\in I, f_i\in D_i^n} r_{if_i}^n$, and $V^n(\hat{f}) = \sum_{i\in I,\hat{f}_i\in D_i^n} r_{i\hat{f}_i}^n$, the preceding expression can be written as

$$|\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}| < \frac{1}{\Gamma}|V^m(f) + |I|K\delta - V^n(f) - V^m(\hat{f}) + V^n(\hat{f}) + |I|K\delta|.$$

By substituting (5.11) and $\Gamma = (m-n)\delta$ into the preceding expression, we get

$$|\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}| < \frac{2|I|K}{m-1}.$$

That is, the slope of allocation $f$ is close to the slope of allocation $\hat{f}$. ■

**Proposition 5.1.5** *Suppose allocation $f$ and $\hat{f}$ are in the competitive allocation set $F^{*n}$, $f$ is in $F^{*m}$, $\hat{f}$ is not in $F^{*m}$, and $m > n$. Then the slopes of $f$ and $\hat{f}$ satisfy the expression*

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} \geq \frac{V^m(f) - V^m(\hat{f})}{\Gamma} - \frac{2K|I|}{m-n}.$$

**Proof.** Similar to the proof of Proposition 5.1.4, we can get

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} = \frac{1}{\Gamma}\Bigg(\sum_{i\in I, f_i\in D_i^n, f_i\in D_i^m} \pi_{f_i}^m - \sum_{i\in I, f_i\in D_i^n, f_i\in D_i^m} \pi_{f_i}^n - \sum_{i\in I,\hat{f}_i\in D_i^n,\hat{f}_i\in D_i^m} \pi_{\hat{f}_i}^m + \sum_{i\in I,\hat{f}_i\in D_i^n,\hat{f}_i\in D_i^m} \pi_{\hat{f}_i}^n\Bigg).$$

From Lemma 5.1.1, the preceding expression implies

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} > \frac{1}{\Gamma}\Bigg(\sum_{i\in I, f_i\in D_i^n, f_i\in D_i^m} r_{if_i}^m - \sum_{i\in I, f_i\in D_i^n, f_i\in D_i^m} (r_{if_i}^n + K\delta) - \sum_{i\in I,\hat{f}_i\in D_i^n,\hat{f}_i\in D_i^m} (r_{i\hat{f}_i}^m + K\delta) + \sum_{i\in I,\hat{f}_i\in D_i^n,\hat{f}_i\in D_i^m} r_{i\hat{f}_i}^n\Bigg).$$

Because $V^m(f) = \sum_{i\in I, f_i\in D_i^m} r_{if_i}^m$, $V^m(\hat{f}) = \sum_{i\in I,\hat{f}_i\in D_i^m} r_{i\hat{f}_i}^m$, $V^n(f) = \sum_{i\in I, f_i\in D_i^n} r_{if_i}^n$, and $V^n(\hat{f}) = \sum_{i\in I,\hat{f}_i\in D_i^n} r_{i\hat{f}_i}^n$, the preceding expression can be written as

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} > \frac{1}{\Gamma}\left(V^m(f) - V^n(f) - |I|K\delta - V^m(\hat{f}) - |I|K\delta + V^n(\hat{f})\right). \tag{5.12}$$

Because allocations $f$ and $\hat{f}$ are competitive at round $n$, their values are equal.

$$V^n(f) = V^n(\hat{f})$$

Combining the preceding expression and (5.12) gives

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} > \frac{1}{\Gamma}\left(V^m(f) - V^m(\hat{f}) - 2|I|K\delta\right).$$

The preceding expression can be rewritten

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} > \frac{V^m(f) - V^m(\hat{f})}{\Gamma} - \frac{2|I|K\delta}{\Gamma}.$$

By substituting $\Gamma = \delta(m-n)$ in the preceding expression, we get

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} > \frac{V^m(f) - V^m(\hat{f})}{\Gamma} - \frac{2|I|K}{m-n}.$$

Thus, the proposition is true. ■

Now, we show that the simulation outcome is a feasible solution to $\mathrm{AAM}_{\mathrm{SCPA}}$ because within a small price range $\Gamma$, the slopes of bundles and competitive allocations between the simulation round $n$ and $m$ satisfy all the constraints in $\mathrm{AAM}_{\mathrm{SCPA}}$.

**Theorem 5.1.6** *Denote $\Gamma$ as a small scalar. The number of simulation rounds between round $n$ and $m$ satisfies $\Gamma = \delta(m-n)$ where $m > n$. The bundle prices are $\pi^n$ and $\pi^m$ at simulation round $n$ and $m$, respectively. Denote $y_{ib}^{mn}$ as the indicator, such that $y_{ib}^{mn} = 0$ if bundle $b$ is not in agent $i$'s demand set ($D_i^p$) at round $p$ where $n < p \leq m$, and $y_{ib}^{mn} = 1$ if bundle $b$ is in $D_i^p$ for $n \leq p \leq m$. Denote $\theta_{ib}^{mn}$ to be the number of times that agent $i$ bids on bundle $b$ within round $n$ and $m$ over the total number of rounds, $m-n$. Denote $F^{*n}$ as the competitive allocation set at round $n$. The indicator function of a competitive allocation $f \in F^{*n}$ being announced competitive within round $n$ and $m$ is $x_f$. Let $\beta_f^{mn}$ where $f \in F^{*n}$*

*be the number of times that allocation $f$ is announced from round $n$ to $m$ over the total number of rounds $m-n$. If $\delta$ goes to 0, the solution of simulation $(y_{ib}^{mn}, \theta_{ib}^{mn}, x_f^{mn}, \beta_f^{mn})$ is a feasible solution to* $\mathrm{AAM}_{\mathrm{SCPA}}$ *based on $(D_i^n, F^{*n})$.*

**Proof.** Given the definition of $\theta_{ib}^{mn}$, we get

$$0 \leq \theta_{ib}^{mn} \leq 1 \ \text{ and } \ 0 \leq \beta_f^{mn} \leq 1.$$

From round $n$ to $m$, bundle $b$'s price increases because agents bid on bundle $b$. The total price increment, $\pi_b^m - \pi_b^n$, is the product of $\delta$ and the number of all agents bidding on bundle $b$. Therefore, we get

$$\theta_b^{mn} = \sum_{i \in I} \theta_{ib}^{mn}.$$

Given the definition of $y_{ib}^{mn}$, the constraint (4.11) is satisfied automatically.

Because active agent's actual demand set cannot be empty from round $n$ to $m$, we suppose bundle $b_1$ is in agent $i$'s actual demand set. Then to bundle $b_2$, there are two cases.

Case (i): If $y_{ib_1}^{mn} = 1$ and $y_{ib_2}^{mn} = 1$, that is, both bundle $b_1$ and $b_2$ are in agent $i$'s demand set at round $n$ and $m$, then Proposition 5.1.2 is

$$\left|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}\right| < \frac{2K}{m-n}. \tag{5.13}$$

Because the increment amount goes to zero, $\delta \to 0$, and $\Gamma$ is a small scalar, the number of rounds between $m$ and $n$ goes to infinity. That is, $m - n = \frac{\Gamma}{\delta} \to \infty$, which leads

$$\frac{2K}{m-n} \to 0. \tag{5.14}$$

Thus, (5.13) can be written as

$$\left|\theta_{b_1}^{mn} - \theta_{b_2}^{mn}\right| \leq 0.$$

The preceding expression becomes

$$\theta_{b_1}^{mn} = \theta_{b_2}^{mn}.$$

Case (ii): If $y_{ib_1}^{mn} = 1$ and $y_{ib_2}^{mn} = 0$, that is, bundle $b_1$ and $b_2$ are in agent $i$'s demand set at round $n$, and $b_1$ is in agent $i$'s demand set at round $m$ but $b_2$ is not at round $m$, then Proposition 5.1.3 is

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} < \frac{2K}{m-n} - \frac{s_i^m - v_i(b_2) + \pi_{b_2}^m}{\Gamma}.$$

When $\delta$ goes to zero, (5.14) holds. The preceding expression can be written as

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} \leq -\frac{s_i^m - v_{ib_2} + \pi_{b_2}^m}{\Gamma}.$$

Because bundle $b_2$ is not in agent $i$'s demand set at round $m$, $s_i^m \geq v_{ib_2} - \pi_{b_2}^m$, we get

$$\theta_{b_1}^{mn} - \theta_{b_2}^{mn} < 0.$$

Therefore, from case (i) and (ii), we show that the constraint (4.12) is satisfied.

Define $\alpha_{ib}^{mn} = \sum_{b \in D_i^{mn}} \theta_b^{mn}$ and $\alpha_f^{mn} = \sum_{i \in I} \alpha_{if_i}^{mn}$ for $f \in F^{*n}$. Then it is trivial that the constraints (4.13)–(4.15) are satisfied.

Suppose $f$ and $\hat{f}$ are two competitive allocations at round $n$. Without loss of generality, suppose allocation $f$ is competitive from round $n$ to $m$. To allocation $\hat{f}$, we have the following two cases.

Case (1): If $x_f = 1$ and $x_{\hat{f}} = 1$, which means that both competitive allocation $f$ and $\hat{f}$ are competitive at round $m$, then Proposition 5.1.4 is

$$\left| \alpha_f^{mn} - \alpha_{\hat{f}}^{mn} \right| < \frac{2|I|K}{m-n}. \tag{5.15}$$

Since the increment goes to zero, $\delta \to 0$, and $\Gamma$ is a small scalar, we have that the number of rounds between $n$ and $m$ goes to infinity. Because $|I|$ and $K$ are two constants,

$$\frac{2|I|K}{m-n} \to 0. \tag{5.16}$$

Combining (5.15) and (5.16) gives

$$\left|\alpha_f^{mn} - \alpha_{\hat{f}}^{mn}\right| \leq 0.$$

The preceding expression implies

$$\alpha_f^{mn} = \alpha_{\hat{f}}^{mn}.$$

Case (2): If $x_f = 1$ and $x_{\hat{f}} = 0$, which means competitive allocation $f$ is competitive at round $m$ and $\hat{f}$ is not competitive at round $m$, then Proposition 5.1.5 is

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} \geq \frac{V^m(f) - V^m(\hat{f})}{\Gamma} - \frac{2K|I|}{m-n}.$$

From (5.16), the preceding expression becomes

$$\alpha_f^{mn} - \alpha_{\hat{f}}^{mn} \geq \frac{V^m(f) - V^m(\hat{f})}{\Gamma}. \tag{5.17}$$

Because $f$ is competitive and $\hat{f}$ is not competitive at round $m$, we have

$$V^m(f) > V^m(\hat{f}). \tag{5.18}$$

Combine (5.17) and (5.18), we get

$$\alpha_f^{mn} > \alpha_{\hat{f}}^{mn}.$$

Therefore, case (1) and (2) prove that the simulation method satisfies constraint (4.16).

Given the definition of $x_f$, the constraint (4.17) is satisfied automatically.

Because only one allocation can be announced as the competitive allocation in each round, the constraint (4.17) is true.

The auction rules require that if an agent wins a bundle, the agent passes in the next round. Otherwise, the agent bids on a bundle in the agent's demand set if the agent is active. Therefore, the constraint (4.19) is satisfied.

Therefore, the simulation result $(y_{ib}^{mn}, \theta_{ib}^{mn}, x_f^{mn}, \beta_f^{mn})$ is a feasible solution to the Attention Allocation Method ($\text{AAM}_{\text{SCPA}}$). ■

Theorem 5.1.6 states that the result of simulation is feasible to $\text{AAM}_{\text{SCPA}}$. In other words, the simulation solution always satisfies $\text{AAM}_{\text{SCPA}}$. However, the theorem does not guarantee that the solution of $\text{AAM}_{\text{SCPA}}$ satisfies all constraints in the simulation. The following conjecture states that $\text{AAM}_{\text{SCPA}}$ gives a unique solution, where "unique" means the same bundle slope.

**Conjecture 5.1.7** *If $s_1 = (y_{ib}, \theta_{ib}, x_f, \beta_f, \alpha_{ib}, \beta_f)$ and $s_2 = (\hat{y}_{ib}, \hat{\theta}_{ib}, \hat{x}_f, \hat{\beta}_f, \hat{\alpha}_{ib}, \hat{\beta}_f)$ are two feasible solutions of* $\text{AAM}_{\text{SCPA}}$*, then*

$$\sum_{i \in I} \theta_{ib} = \sum_{i \in I} \hat{\theta}_{ib} \quad \forall b \in B. \tag{5.19}$$

## 5.2 Computational Complexity of the Price Trajectory Algorithm

To consider the complexity of the price trajectory algorithm, first we look at the agent's computation. In each step, agent $i$ computes the following:

1. the agent's potential demand set $\hat{D}_i$;
2. the time at which the agent's demand set changes.

From expression (2.1), the complexity of computing $\hat{D}_i$ is $O(|B|)$, where $|B|$ is the number of bundles. The demand set computation is linear in the number of bundles, but exponential in the number of items. The agent changes the demand set when (i) bundle $c \notin D_i$ becomes as attractive as bundle $b \in D_i$, and (ii) the surplus of bundle $b \in D_i$

becomes 0. From the discussion in Section 4.3 of Chapter 4, it is known that the complexity of computing the duration of the interval point is also $O(|B|)$. Therefore, the complexity of an agent's computation is $O(2^n)$, where $n$ is the number of items.

Secondly, we consider the complexity of the auctioneer's computation. In one iteration, the auctioneer solves $\text{AAM}_{\text{SCPA}}$ and $\text{IPM}_{\text{SCPA}}$. AAM is a difficult NP problem because it is a mixed integer linear program. It is easy to show that $\text{IPM}_{\text{SCPA}}$ is also a difficult NP problem because $\text{IPM}_{\text{SCPA}}$ needs to check for a collision for each non-competitive allocation that becomes competitive, and the number of possible allocations is $m^n$ where $m$ is the number of agents.

Therefore, in theory the price trajectory algorithm is not a polynomial algorithm. However, there are some special situations in which the computation of $\text{AAM}_{\text{SCPA}}$ can be simplified.

1. If there is only one bundle, $b$, in agent $i$'s potential demand set $\hat{D}_i$, that bundle must be kept in the demand set $D_i$. Therefore, we can eliminate the integer variable, $y_{ib}$, which indicates whether bundle $b$ remains in agent $i$'s demand set.

2. If a bundle is in only one agent's potential demand set, the bundle must remain in the agent's demand set.

3. If there are only two potential competitive allocations, both competitive allocations must remain competitive, and then their integer indicator $x_f$'s can be eliminated.

In these special cases, the $\text{AM}_{\text{SCPA}}$ may become a linear program. For example, at step 10 of the $\text{PTA}_{\text{SCPA}}$ that is used to solve the example in Table 4.1, the two potential competitive allocations are $f$={–, –, AB, C} and $\bar{f}$={ABC, –, –, –}, and Agent 1's demand set is {B, ABC}, Agent 2's demand set is {A, AC}, Agent 3's demand set is {AB}, and Agent 4's

Table 5.1: Solution of the example in which AAM can be simplified.

| | A | B | AB | C | AC | BC | ABC | pass |
|---|---|---|---|---|---|---|---|---|
| Agent 1 | | 0.375 | | | | | 0.375 | 0.25 |
| Agent 2 | 0.5 | | | | 0.5 | | | |
| Agent 3 | | | 0.25 | | | | | 0.75 |
| Agent 4 | | | | 0.125 | | 0.125 | | 0.75 |
| $\beta_f$=0.75 $\beta_{\bar{f}}$=0.25 | | | | | | | | |

demand set is {C, BC}. It is easy to check that these conditions satisfy the preceding situations, and therefore all of the binary variables in $\text{AAM}_{\text{SCPA}}$ can be removed. The program reduces to the following linear constraints:

$$\begin{cases} \theta_{1\text{B}} = \theta_{1\text{ABC}} \\ \theta_{2\text{A}} = \theta_{2\text{AC}} \\ \theta_{4\text{C}} = \theta_{4\text{BC}} \\ \theta_{3\text{AB}} + \theta_{4\text{C}} = \theta_{1\text{ABC}} \\ \beta_f + \beta_{\bar{f}} = 1 \\ \beta_{\bar{f}} + \theta_{1\text{B}} + \theta_{1\text{ABC}} = 1 \\ \theta_{2\text{A}} + \theta_{2\text{AC}} = 1 \\ \beta_f + \theta_{3\text{AB}} = 1 \\ \beta_f + \theta_{4\text{C}} + \theta_{4\text{BC}} = 1 \end{cases}$$

The linear program has nine variables and nine independent equality constraints. The solution is shown in Table 5.1

## 5.3 Empirical Results

To compare the computational costs against the simulation, the computational results of the $\text{PTA}_{\text{SCPA}}$ and the simulation method are provided. Adding a constant function as the objective changes $\text{AAM}_{\text{SCPA}}$ to a Mixed Integer Linear Program (MILP), which enables the use of CPLEX to solve the MILP.

I generate random problems in which the agents are assigned values based on all bundles according to the following algorithm [41], and are parameterized by $l$ and $\beta > 0$.

1. Assign values to individual items from the uniform distribution of integers between $[1, l]$.

2. Starting with bundles of size 2, and progressively increasing the bundle size,

   Let $\underline{v}_b = \max_{c \subset b} v_i(c)$.

   Let $\bar{v}_b = \max_{c \subset b} v_i(c) + v_i(b \backslash c)$.

   Assign a value to $v_i(b)$ selected from a uniform distribution of integers between $[\underline{v}_b, \underline{v}_b + \beta(\bar{v}_b - \underline{v}_b)]$.

Twenty-one problem sizes are selected, and ten problems are generated at each size. The parameter settings are $l = 50$ and $\beta = 1.5$. Figure 5.1 shows the average computing time over the 10 instances at each problem size. The x-axis indicates the size of the problems in the order of their difficulty, and the y-axis is the computing time. Note that the method of generating problems creates a value for every agent for every bundle. The largest problem size of 13 agents and 6 items that is tested represents 819 agent values and 4,826,809 possible allocations.

In Figure 5.1, there are 6 lines, which represent the run time of $\text{PTA}_{\text{SCPA}}$ and the run time of simulations with varying bid increments. The graph shows the expected result that running the simulation with smaller bid increments takes more computation time. Also, it is clear that the computation time that uses simulation increases faster than that of $\text{PTA}_{\text{SCPA}}$ as the problem size increases. The $\text{PTA}_{\text{SCPA}}$ has behavior similar to

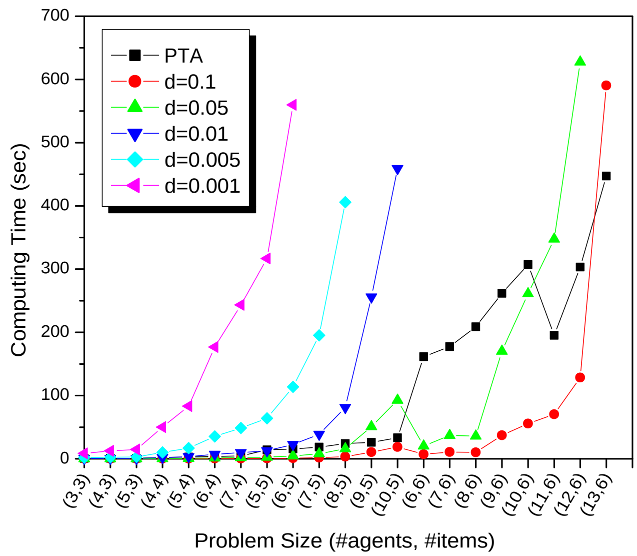


Figure 5.1: Comparison between $\mathrm{PTA}_{\mathrm{SCPA}}$ and the simulation with varying bid increments over a variety of problems.

the simulation with the bid increment taking the value of 0.1. The exponential explosion inherent in the problem has not been avoided. However, these results suggest that exact solutions that are independent of bid magnitude with similar computational costs can be generated.

## 5.4 Comparison with Alternatives

Hoffman et al. [15] gave six examples in their paper and gave the computation comparison among Pure Simulation Method (PSM), Safe Start Simulation (SSS), Increment Scaling Simulation (ISS), Increment Scaling with Safe Start Simulation (ISSSS), and Vickrey-Clarke-Groves (VCG) mechanism. We borrow these results, and add the $\mathrm{PTA}_{\mathrm{SCPA}}$ results to the following tables. From the results, we can see that the $\mathrm{PTA}_{\mathrm{SCPA}}$ takes the fewest steps among these simulation algorithms.

Table 5.2: Data and results of Hoffman et al.'s first example.

| Agent | 1 | 2 | | 3 | 4 | |
|---|---|---|---|---|---|---|
| Package | AB | AB | C | AB | AB | C |
| Value | 15 | 14 | 5 | 9 | 10 | 4 |

| Method | Rounds | Value | Winning Agent's Payment | |
|---|---|---|---|---|
| | | | Agent 1, AB | Agent 2, C |
| PSM | 2450 | 17.02 | 13.01 | 4.01 |
| SSS | 300 | 17.01 | 13 | 4.01 |
| ISS | 31 | 17.02 | 13.01 | 4.01 |
| ISSSS | 19 | 17.02 | 13.01 | 4.01 |
| VCG | – | 17 | 13 | 4 |
| $\text{PTA}_{\text{SCPA}}$ | 2 | 17 | 13 | 4 |

In all simulation approaches, the price increment, $\delta$, has the value of 0.01. The final allocations of all approaches including the $\text{PTA}_{\text{SCPA}}$ are efficient, and are the same in these six examples. However, the payments are not always the same even if the bidding relative error is ignored. We focus on the comparison between the $\text{PTA}_{\text{SCPA}}$ and the other methods. The $\text{PTA}_{\text{SCPA}}$ and VCG methods produce the same payments in the first two examples. In the other four examples, the payments of $\text{PTA}_{\text{SCPA}}$ are higher than that of the VCG method. The $\text{PTA}_{\text{SCPA}}$, SSS, ISS, and ISSSS methods generate the same value of the final allocation in five examples. In four of these five examples, the winners' payments are the same. The payments of PSM are higher than that of $\text{PTA}_{\text{SCPA}}$ in three examples, and the payments are equal in the other examples.

Table 5.3: Data and results of Hoffman et al.'s second example.

| Agent | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Package | AB | BC | C | C | AB |
| Value | 21 | 35 | 14 | 20 | 22 |

| Method | Rounds | Value | Winning Agent's Payment | |
|---|---|---|---|---|
| | | | Agent 4, C | Agent 5, AB |
| PSM | 4025 | 36.76 | 15.75 | 21.01 |
| SSS | 1 | 35.02 | 14.01 | 21.01 |
| ISS | 38 | 35.01 | 14 | 21.01 |
| ISSSS | 5 | 35.02 | 14.01 | 21.01 |
| VCG | – | 35 | 14 | 21 |
| $\text{PTA}_{\text{SCPA}}$ | 3 | 35 | 14 | 21 |

Table 5.4: Data and results of Hoffman et al.'s third example.

| Agent | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Package | AB | CD | CD | BC | AC |
| Value | 10 | 20 | 25 | 10 | 10 |

| Method | Rounds | Value | Winning Agent's Payment | |
|---|---|---|---|---|
| | | | Agent 1, AB | Agent 3, CD |
| PSM | 3250 | 27.52 | 7.51 | 20.01 |
| SSS | 1 | 20.02 | 0.01 | 20.01 |
| ISS | 18 | 20.02 | 0.01 | 20.01 |
| ISSSS | 7 | 20.02 | 0.01 | 20.01 |
| VCG | – | 20 | 0 | 20 |
| $\text{PTA}_{\text{SCPA}}$ | 2 | 23.33 | 3.33 | 20 |

Table 5.5: Data and results of Hoffman et al.'s fourth example.

| Agent | 1 | | 2 | | 3 |
|---|---|---|---|---|---|
| Package | A | B | A | B | AB |
| Value | 16 | 16 | 8 | 8 | 10 |

| Method | Rounds | Value | Winning Agent's Payment | |
|---|---|---|---|---|
| | | | Agent 1, A | Agent 2, B |
| PSM | 1500 | 10.02 | 5.01 | 5.01 |
| SSS | 501 | 10.02 | 5.01 | 5.01 |
| ISS | 19 | 10.02 | 5.01 | 5.01 |
| ISSSS | 10 | 10.02 | 5.01 | 5.01 |
| VCG | – | 2 | 2 | 0 |
| $\text{PTA}_{\text{SCPA}}$ | 1 | 10 | 5 | 5 |

Table 5.6: Data and results of Hoffman et al.'s fifth example.

| Agent | 1 | | 2 | | 3 | | 4 | 5 |
|---|---|---|---|---|---|---|---|---|
| Package | AB | C | BC | B | AC | C | AB | C |
| Value | 15 | 5 | 15 | 5 | 12 | 3 | 12 | 6 |

| Method | Rounds | Value | Winning Agent's Payment | |
|---|---|---|---|---|
| | | | Agent 1, AB | Agent 5, C |
| PSM | 1890 | 17.12 | 12.01 | 5.01 |
| SSS | 100 | 17 | 13 | 4 |
| ISS | 40 | 17.01 | 12 | 5.01 |
| ISSSS | 6 | 17.02 | 13.01 | 4.01 |
| VCG | – | 14 | 11 | 3 |
| $\mathrm{PTA}_{\mathrm{SCPA}}$ | 7 | 17 | 13 | 4 |

Table 5.7: Data and results of Hoffman et al.'s sixth example.

| Agent | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Package | AB | BC | AC | A |
| Value | 20 | 26 | 24 | 16 |

| Method | Rounds | Value | Winning Agent's Payment | |
|---|---|---|---|---|
| | | | Agent 1, BC | Agent 2, A |
| PSM | 3100 | 24.02 | 12.01 | 12.01 |
| SSS | 1201 | 24.02 | 12.01 | 12.01 |
| ISS | 20 | 24.02 | 17.01 | 7.01 |
| ISSSS | 15 | 24.02 | 17.01 | 7.01 |
| VCG | – | 8 | 8 | 0 |
| $\mathrm{PTA}_{\mathrm{SCPA}}$ | 2 | 24 | 12 | 12 |

# Chapter 6

# Application to Ascending Package Auction

In this chapter, the Price Trajectory Algorithm (PTA) is used to solve the Ascending Package Auction (APA) problems. Like the SCPA version of PTA, the Attention Allocation Method of APA ($\text{AAM}_{\text{APA}}$) and the Inflection Point Method of APA ($\text{IPM}_{\text{APA}}$) are presented in the fist two sections. In the third section, an example is used to show how the PTA works for APA. Some of the computational results are provided at the end of this chapter.

## 6.1 Attention Allocation Method (AAM)

From the auction rules and the straightforward bidding strategy that are described in Section 2.3.2, it is known that each agent bids on all bundles in its demand set if the agent does not win in the previous round. For example, if bundle $b$ and $c$ are in agent $i$'s demand set, the agent bids on $b$ and $c$ at the same time.

There are no announced bundle prices in APA. All agents keep track of their own bids on a bundle. To one agent, the slopes of bundle $b$ and bundle $c$ are always the same if both $b$ and $c$ are in the agent's demand set because it bids both bundles equally.

**Proposition 6.1.1** *If bundle $b$ and $c$ are in agent $i$'s demand set at time $t_0$, the slope of bundle $b$ is equal to the slope of bundle $c$ for any $t > t_0$:*

$$\theta_{ib}^{t_0 t} = \theta_{ic}^{t_0 t} \tag{6.1}$$

Given the auction rules, an agent either bids or passes in each iteration. Therefore, an active agent focuses on two parts: $\theta_{ib}$ for all $b$ in $D_i$, and $\theta_{i\text{pass}}$. Therefore, the expression is

$$\theta_{ib} + \theta_{i\text{pass}} = 1, \quad \forall b \in D_i. \tag{6.2}$$

Moreover, it follows from the auction description that a bundle never leaves the demand set of an agent once the bundle becomes a member of the demand set. Thus, the potential demand set is the actual demand set. The bids on all bundles in the demand set increase at the same rate, which means that the surpluses of these bundles are maximized and stay the same until the end of the auction.

An allocation that is competitive at a particular time $t$, may not be competitive at time $t + \varepsilon$ even when $\varepsilon$ is very small. In the example in Figure 2.2, allocation $f$={ABC, –, –, –} is competitive until the time at which Agent 1's bid on ABC reaches 12. But after this point, $f$ is not competitive any more. By definition, the slope $\gamma_f^{t_1 t_2}$ of an allocation $f$, which is written as $\gamma_f$ (if there is no ambiguity), within $[t_1, t_2]$ is

$$\gamma_f^{t_1 t_2} = \frac{V^{t_2}(f) - V^{t_1}(f)}{t_2 - t_1}.$$

**Proposition 6.1.2** *If both allocation $f$ and $\hat{f}$ are in the competitive allocation set $F^*$ at time $t_1$, and if $f$ is in $F^*$ at time $t_2$ but $\hat{f}$ is not, then the slope of allocation $f$ is larger than the slope of $\hat{f}$.*

$$\gamma_f > \gamma_{\hat{f}}$$

**Proof.** At time $t_1$, because both $f$ and $\hat{f}$ are in the competitive allocation set $F^*$, the expression is

$$V^{t_1}(f) = V^{t_1}(\hat{f}).$$

At time $t_2$ $(t_2 > t_1)$, $f \in F^*$ and $\hat{f} \notin F^*$, we have

$$V^{t_2}(f) > V^{t_2}(\hat{f}).$$

Therefore, the expression becomes

$$\frac{V^{t_2}(f) - V^{t_1}(f)}{t_2 - t_1} > \frac{V^{t_2}(\hat{f}) - V^{t_1}(\hat{f})}{t_2 - t_1}.$$

Substitute $\gamma_f$ into preceding expression, we prove the claim. ■

**Proposition 6.1.3** *If both allocation $f$ and $\hat{f}$ are in the competitive allocation set $F^*$ between time $t_1$ and $t_2$, their slopes are the same,*

$$\gamma_f = \gamma_{\hat{f}}.$$

**Proof.** The proof is similar to the proof of Proposition 6.1.2. ■

In the algorithm, we compute the slope of an allocation by

$$\gamma_f = \sum_{i \in f} \theta_{if_i}.$$

An integer variable $x_f$ is introduced to indicate whether $f$ is competitive in an interval. If allocation $f$ remains competitive during the interval, $x_f$ has the value of 1, or otherwise 0. In the abstract, the Attention Allocation Method (AAM) computes

$$\text{AAM}_{\text{APA}} : \{D_i, \hat{F}^*\} \rightarrow \{D_i, F^*, \theta_{ib}, x\}.$$

Like in $\text{AAM}_{\text{SCPA}}$, we construct a mathematical model of $\text{AA}_{\text{APA}}$. The constraints (6.1) and (6.2) on the allocation of attention among bundles in an agent's demand set already are established.

Let's consider the constraints that capture the interactions of the competitive allocations. In a time period, let $\beta_f \in [0, 1]$ be the frequency with which allocation $f$ is announced as the winner, which enables the members of that allocation to pass. At every iteration, only one of the competitive allocations must be announced as the winning agents/bundles. Therefore,

$$\sum_{f \in \hat{F}^*} \beta_f = 1. \tag{6.3}$$

As discussed earlier, if allocation $f$ is not competitive, $\beta_f = 0$. In other words,

$$\beta_f \leq x_f. \tag{6.4}$$

From Proposition 6.1.2 and 6.1.3, competitive allocations increase their value at the same rate. If a potential competitive allocation turns out to be not competitive, then its slope must be less than the allocations that are competitive. The integer-linear form of this logical constraint is

$$\gamma_{\hat{f}} - \gamma_f + N x_f \leq N, \quad \forall f, \hat{f} \in \hat{F}^*, \tag{6.5}$$

where $N$ is a sufficiently large number.

Finally, we develop constraints about the relationship between the agent attention and the frequency with which an allocation is announced as the winner. Because an agent

$$
\text{AAM}_{\text{APA}} \begin{cases} \theta_{ib} = \theta_{ic}, \quad \forall i \in A,\ b,c \in D_i & (6.7) \\ \sum_{i \in \hat{f}} \theta_{i\hat{f}_i} - \sum_{i \in f} \theta_{if_i} + Nx_f \leq N, \quad \forall f, \hat{f} \in \hat{F}^* & (6.8) \\ \beta_f \leq x_f, \quad \forall f \in \hat{F}^* & (6.9) \\ \sum_{f \in \hat{F}^*} \beta_f = 1, & (6.10) \\ \sum_{f \in \hat{F}^*} K_i \cdot 1_{[i \in f]} \cdot \beta_f + \sum_{b \in B} \theta_{ib} = K_i, \quad \forall i \in A & (6.11) \\ \beta_f \geq 0, \quad x_f \in \{0,1\}, \quad \forall f \in \hat{F}^* \\ 0 \leq \theta_{ib} \leq 1, \quad \forall i \in A,\ b \in D_i \end{cases}
$$

Figure 6.1: Mathematical model of the AAM of the PTA for APA.

passes if the allocation is included in the announced allocation, and the agent bids if the allocation is not included in the announced allocation, then I have

$$\theta_{i\text{pass}} = \sum_{f \in \hat{F}^*} K_i \cdot 1_{[i \in f]} \cdot \beta_f. \tag{6.6}$$

$K_i$ indicates whether agent $i$ is active, and $1_{[i \in f]}$ indicates whether agent $i$ is included in the allocation $f$.

From these constraints, we get $\text{AAM}_{\text{APA}}$ in Figure 6.1.

## 6.2 Inflection Point Method for APA

The duration of time over which agent $i$'s demand set does not change is based on either (i) a non-active bundle becoming active or (ii) agent $i$'s demand set becoming empty. Suppose bundle $b$ is in agent $i$'s demand set $D_i$. For the first case (i),

$$\Delta t_i^{b,c} = \min_{c \notin D_i} \{ \frac{v_{ib} - r_{ib} - v_{ic}}{\theta_{ib}} \}.$$

For the second case (ii),

$$\Delta t_i^b = \frac{v_{ib} - r_{ib}}{\theta_{ib}}.$$

The duration of time over which agent $i$'s demand set does not change is

$$\Delta t_i^{DS} = \min\{\Delta t_i^{b,c}, \Delta t_i^b\}.$$

Among all agents, the duration of time until the next demand set changes is

$$\Delta t^{DS} = \min_{i \in I}\{\Delta t_i^{DS}\}.$$

If $f$ is a competitive allocation, then the point in time at which the competitive allocation set changes is exactly the same as in SCPA:

$$\Delta t^{CA} = \min_{\hat{f} \notin F^*} \left\{ \frac{V(f) - V(\hat{f})}{\gamma_f - \gamma_{\hat{f}}} \middle| \gamma_f > \gamma_{\hat{f}} \right\}.$$

Therefore, the next inflection point is when

$$\Delta t = \min\{\Delta t^{DS}, \Delta t^{CA}\}.$$

## 6.3 An Example

Figure 6.2 shows the results of applying $\text{PTA}_{\text{APA}}$ to solve the problem in Table 2.1. In the four sub-graphs, the x-axis indicates the auction time and the y-axis indicates the agent's bid price. At the beginning of the auction, the bid on each bundle is zero. Each agent puts all of its interests in bundle ABC because ABC gives the maximum surplus from the valuation in Table 2.1. When Agent 1's bid of ABC reaches 2, in the first sub-graph, we can see that Agent 1 begins to bid on bundles AB and AC because these two bundles also give the maximum surplus. At the same time, Agent 2's demand set also gets a new

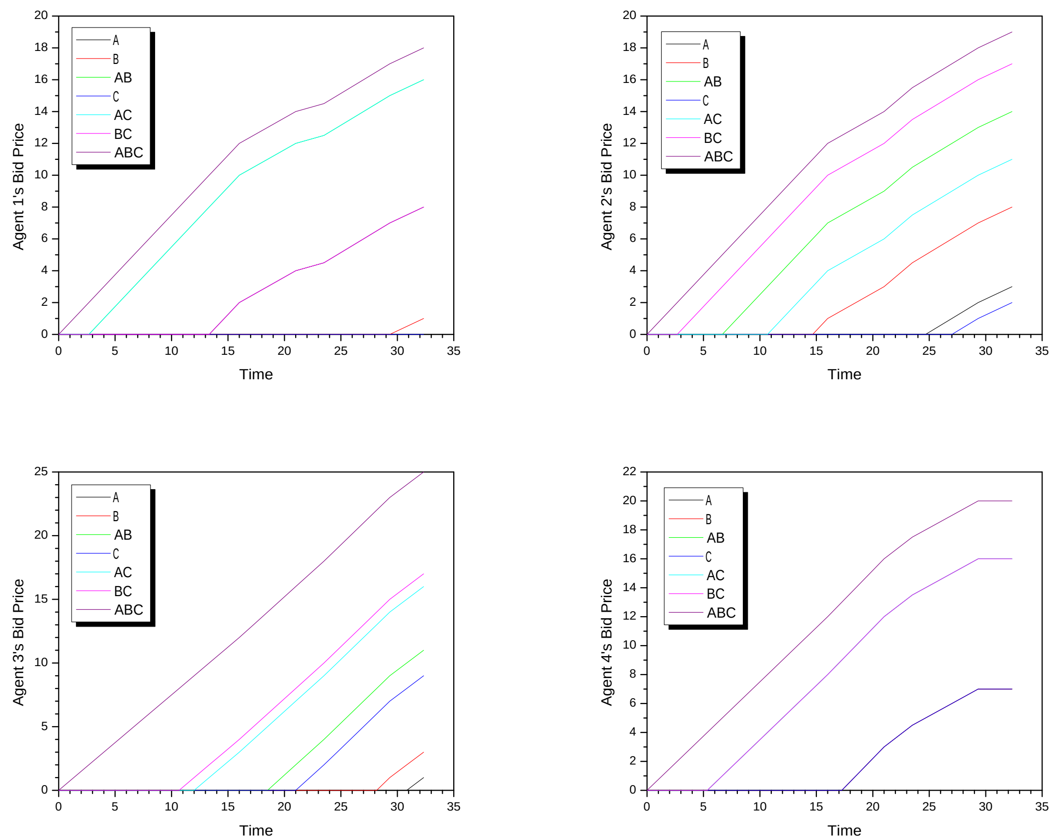


Figure 6.2: APA bundle bids of solving the example in Table 2.1 by the PTA.

member, bundle BC and it starts to bid on BC. Similar decisions cause the most of the inflection points in Figure 6.2.

The values of competitive allocations are shown in Figure 6.3 where the x-axis indicates the auction time and the y-axis is the value of the allocation. This figure contains all allocations that have chance to be competitive. The top curve represents the value of the competitive allocations. From the figure, we can see that some allocations, such as $\{$A, BC, –, –$\}$ and $\{$–, BC, –, A$\}$, are not competitive at the beginning of the auction and then become competitive. Especially allocation $\{$A, B, C, -$\}$ becomes competitive at the end of

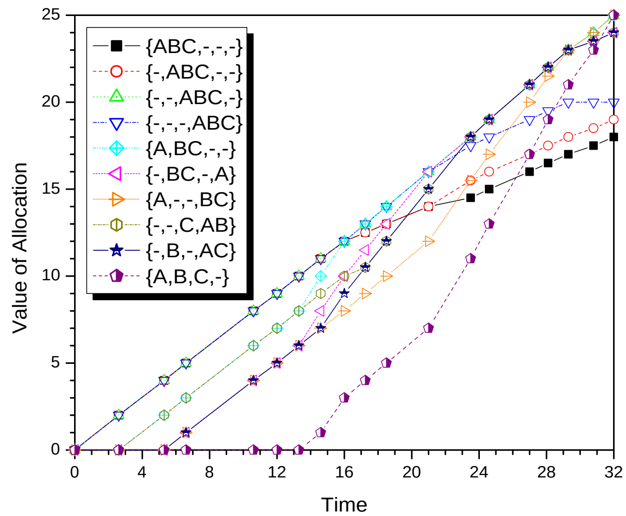


Figure 6.3: APA allocation values of solving the example in Table 2.1 by the PTA.

the auction and is one of the final allocations. Compare to the value of the competitive allocations in SCPA shown in Figure 4.1 with the same example, we can see that there are no jumps of APA competitive allocations because the bid on any bundle increases little by little due to the reason that there are no bundle prices in APA.

Tables 6.1 and 6.2 show the computations involved in six of the eighteen steps required to solve the example in Table 2.1 by using the $\text{PTA}_{\text{APA}}$. Each step shows the following:

- bids at the designated time,
- potential competitive allocations (those that remain competitive in the interval are designated with an asterisk),
- each agent's demand set,
- each agent's allocation of attention.

At $t = 0$, the bids are zero, and bundle ABC is the sole element in each agent's demand set. Because there are four competitive allocations, and no other bundles are distracting the bidders, each competitive allocation wins one-fourth of the time. At Step 2, $t = 2\frac{2}{3}$, the bundles AB and AC enter Agent 1's demand set, and BC enters Agent 2's demand set. Both agents continue bidding on ABC, while also giving attention to the new elements.

The next table shows the results of Step 9. This step is caused by the event that the allocation {A, BC, –, –} enters the competitive allocation set. The effect of this is visible in Figure 6.3. From now on, the bid prices on bundle ABC between Agents 1 and 2, and Agents 3 and 4 are different. That is, the allocations {ABC, –, –, –} and {–, ABC, –, –} are not competitive. Agents 1 and 2 are involved in another competitive allocation {A, BC, –, –}.

In Step 16, we see a complex allocation of attention across the bundles. At the end of Step 16, the surplus of Agent 4 becomes 0, which causes Agent 4 to become inactive and drop out. However, Agent 4's bids still are there and the competitive allocation {A, –, –, BC} still includes Agent 4 even though Agent 4 has stopped bidding. Let us skip to the last step, Step 18. Agent 3 puts twice the attention on bidding as Agents 1 and 2 because they are making contributions combined to allocation {A, BC, –, –}. We can see that Agent 4 does not make any new contribution on allocation {–, –, C, AB} because it is inactive, but it is involved in this allocation because it holds the bid on bundle AB. This step is stopped by two events: (1) Agent 3 becomes inactive and (2) allocation {A, B, C, –} becomes competitive. The auctioneer then selects from among the two remaining allocations: {A, B, C, –} or {A, BC, –, –}. Both Agent 1 and Agent 2 – the only remaining active agents – receive a bundle in the two candidate final allocations, and, therefore, do

Table 6.1: Some steps of solving the example in Table 2.1 by applying APA version of the PTA.

| Step 1, t=0 | | A | B | AB | C | AC | BC | ABC | |
|---|---|---|---|---|---|---|---|---|---|
| Agent | $D_i$ | Bid prices | | | | | | | |
| 1 | ABC | | | | | | | | |
| 2 | ABC | | | | | | | | |
| 3 | ABC | | | | | | | | |
| 4 | ABC | | | | | | | | |
| Competitive allocations | | Attention | | | | | | | pass |
| *{ABC, –, –, –}, *{–, ABC, –, –} | | | | | | | | 0.75 | 0.25 |
| *{–, –, ABC, –}, *{–, –, –, ABC} | | | | | | | | 0.75 | 0.25 |
| | | | | | | | | 0.75 | 0.25 |
| | | | | | | | | 0.75 | 0.25 |

| Step 2, t=$2\frac{2}{3}$ | | A | B | AB | C | AC | BC | ABC | |
|---|---|---|---|---|---|---|---|---|---|
| Agent | $D_i$ | Bid prices | | | | | | | |
| 1 | AB, AC, ABC | | | | | | | 2 | |
| 2 | BC, ABC | | | | | | | 2 | |
| 3 | ABC | | | | | | | 2 | |
| 4 | ABC | | | | | | | 2 | |
| Competitive allocations | | Attention | | | | | | | pass |
| *{ABC, –, –, –}, *{–, ABC, –, –} | | | | 0.75 | | 0.75 | | 0.75 | 0.25 |
| *{–, –, ABC, –}, *{–, –, –, ABC} | | | | | | | 0.75 | 0.75 | 0.25 |
| | | | | | | | | 0.75 | 0.25 |
| | | | | | | | | 0.75 | 0.25 |

⋮ ⋮

| Step 9, t=16 | | A | B | AB | C | AC | BC | ABC | |
|---|---|---|---|---|---|---|---|---|---|
| Agent | $D_i$ | Bid prices | | | | | | | |
| 1 | A, AB, AC, BC, ABC | 2 | | 10 | | 10 | 2 | 12 | |
| 2 | B, AB, AC, BC, ABC | | 1 | 7 | | 4 | 10 | 12 | |
| 3 | AC, BC, ABC | | | | | 3 | 4 | 12 | |
| 4 | AB, AC, BC, ABC | | | 8 | | 8 | 8 | 12 | |
| Competitive allocations | | Attention | | | | | | | pass |
| *{A, BC, -, -} | | 0.4 | | 0.4 | | 0.4 | 0.4 | 0.4 | 0.6 |
| {ABC, –, –, –}, {–, ABC, –, –} | | | 0.4 | 0.4 | | | 0.4 | 0.4 | 0.6 |
| *{–, –, ABC, –} | | | | | | 0.8 | 0.8 | 0.8 | 0.2 |
| *{–, –, –, ABC} | | | | 0.8 | | 0.8 | 0.8 | 0.8 | 0.2 |

⋮ ⋮

Table 6.2: Continuation of Table 6.1.

| Step 16, t=$28\frac{1}{6}$ | | A | B | AB | C | AC | BC | ABC | |
|---|---|---|---|---|---|---|---|---|---|
| Agent | $D_i$ | Bid prices | | | | | | | |
| 1 | A, AB, AC, BC, ABC | 6.5 | | 14.5 | | 14.5 | 6.5 | 16.5 | |
| 2 | A, B, AB, C, AC, BC, ABC | 1.5 | 6.5 | 12.5 | 0.5 | 9.5 | 15.5 | 17.5 | |
| 3 | B, AB, C, AC, BC, ABC | | | 8 | 6 | 13 | 14 | 22 | |
| 4 | A, B, AB, C, AC, BC, ABC | 6.5 | 6.5 | 15.5 | 6.5 | 15.5 | 15.5 | 19.5 | |
| Competitive allocations | | Attention | | | | | | | pass |
| *{A, BC, -, -}, *{A, -, -, BC} | | 3/7 | | 3/7 | | 3/7 | 3/7 | 3/7 | 4/7 |
| {-, BC, -, A}, *{-, B, -, AC} | | 3/7 | 3/7 | 3/7 | 3/7 | 3/7 | 3/7 | 3/7 | 4/7 |
| *{–, –, ABC, –} | | | 6/7 | 6/7 | 6/7 | 6/7 | 6/7 | 6/7 | 1/7 |
| | | 3/7 | 3/7 | 3/7 | 3/7 | 3/7 | 3/7 | 3/7 | 4/7 |

⋮ ⋮

| Step 18, t=$30\frac{5}{6}$ | | A | B | AB | C | AC | BC | ABC | |
|---|---|---|---|---|---|---|---|---|---|
| Agent | $D_i$ | Bid prices | | | | | | | |
| 1 | A, B, AB, AC, BC, ABC | 7.5 | 0.5 | 15.5 | | 15.5 | 7.5 | 17.5 | |
| 2 | A, B, AB, C, AC, BC, ABC | 2.5 | 7.5 | 13.5 | 1.5 | 10.5 | 16.5 | 18.5 | |
| 3 | A, B, AB, C, AC, BC, ABC | | 2 | 10 | 8 | 15 | 16 | 24 | |
| 4 | | 7 | 7 | 16 | 7 | 16 | 16 | 20 | |
| Competitive allocations | | Attention | | | | | | | pass |
| *{A, BC, -, -} | | 1/3 | 1/3 | 1/3 | | 1/3 | 1/3 | 1/3 | 2/3 |
| *{-, -, C, AB} | | 1/3 | 1/3 | 1/3 | 1/3 | 1/3 | 1/3 | 1/3 | 2/3 |
| *{-, -, ABC, -} | | 2/3 | 2/3 | 2/3 | 2/3 | 2/3 | 2/3 | 2/3 | 1/3 |
| | | | | | | | | | |

| Step 19, t=$32\frac{2}{3}$ | | A | B | AB | C | AC | BC | ABC |
|---|---|---|---|---|---|---|---|---|
| Agent | $D_i$ | Bid prices | | | | | | |
| 1 | A, B, AB, C, AC, BC, ABC | 8 | 1 | 16 | | 16 | 8 | 18 |
| 2 | A, B, AB, C, AC, BC, ABC | 3 | 8 | 14 | 2 | 11 | 17 | 19 |
| 3 | | 1 | 3 | 11 | 9 | 16 | 17 | 25 |
| 4 | | 7 | 7 | 16 | 7 | 16 | 16 | 20 |

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

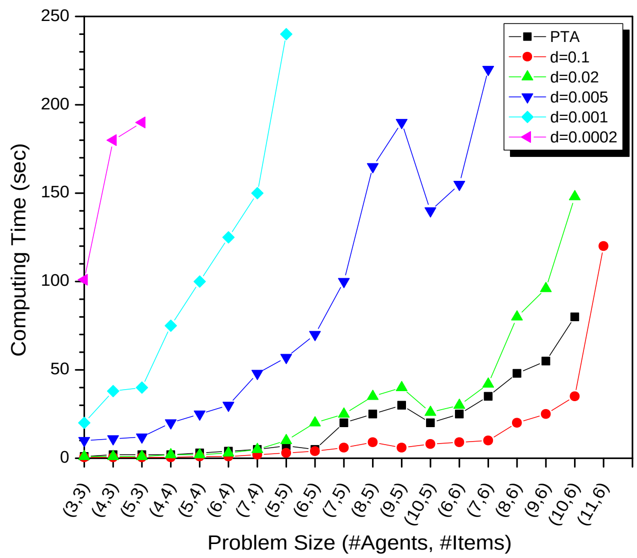


Figure 6.4: Comparison between $\text{PTA}_{\text{APA}}$ and the simulation with varying bid increments over a variety of problems.

not need to bid further. Moreover, the bidding strategies ensure that the two active agents are indifferent between the two allocations because they give the same surplus at the final bids, respectively.

## 6.4 Computational Results

In this section, the computational results of PTA and the simulation are provided. The PTA-AAM is changed to a Mixed Integer Linear Program (MILP) by adding a constant function as the objective. To solve the MILP, a well known solver, CPLEX, is employed.

By using the same random generator in Section 5.3, we construct nineteen problems at various sizes and generate ten instances for each problem size. The parameters settings are $l = 50$ and $\beta = 1.5$. The average value of the computing time is shown in Figure 6.4. The x-axis indicates the sizes of the problems and the y-axis is the computing time. In the graph, there are six lines, which represent the run time of $\mathrm{PTA}_{\mathrm{APA}}$ and the run time of a simulation with bid increments $\delta = 0.1$, $\delta = 0.02$, and so on. From the graph, $\mathrm{PTA}_{\mathrm{APA}}$ displays behavior that is similar to the simulation with a bid increment of 0.1. The exponential explosion inherent in the problem has not been avoided.

# Chapter 7

# Preserving Private Information and Detecting Fraud

In this chapter, I address the preservation of the agents' private information and the detection of fraud by the auctioneer. In the first section, some literature about multi-party computation is reviewed. Later, a secure protocol for the SCPA version of the Price Trajectory Algorithm ($\text{PTA}_{\text{SCPA}}$) is presented, through which the auctioneer can identify the next inflection point without learning each agent's time duration in which its demand set does not change. The last section discusses how the agents can detect fraud by the auctioneer in $\text{PTA}_{\text{SCPA}}$.

In many situations, the bidders would like to hide their bids from the other bidders and may not want to reveal more information than is necessary to the auctioneer. Furthermore, if some information is not publicly available, the ability to detect fraud becomes important to the bidder. In iterative auctions, the bidders submit their bids, and the auctioneer determines the potential winners and the payments in each round. From the auctioneer's point of view, only the bids from the agents in order to make the bidding process proceed to the next round are needed. We would like the PTA algorithm to collect no more information that the iterative auction that is solves.

In the $\mathrm{PTA}_{\mathrm{SCPA}}$, the auctioneer has the potential competitive allocation set and needs all agents' potential demand sets to construct the $\mathrm{AAM}_{\mathrm{SCPA}}$ model. After the auctioneer solves $\mathrm{AAM}_{\mathrm{SCPA}}$ and obtains the bundle slopes, the auctioneer announces the bundle slopes to all agents. The impetus for the algorithm to move from one inflection point to the next is a change in either the agents' demand set or the competitive allocation set. To continue the process, the auctioneer uses the $\mathrm{IPM}_{\mathrm{SCPA}}$ to get the next inflection point. As discussed in Section 4.3, the only value that the auctioneer needs from the agents is the first time that *any* of the agents' demand sets change.

If all agents tell the auctioneer when exactly their demand sets will change, the auctioneer may see more information than necessary. For the example in Table 3.1, at time $t = 0$, the prices of all bundles are \$0. The four agents' demand sets are $\{\mathrm{ABC}\}$, which they must reveal to the auctioneer. From Table 4.1, the auctioneer announces that slope of bundle ABC is 3 and the other slopes are 0. Each agent computes the time duration over which its demand set does not change and gets the following:

- Agent 1: $\frac{22}{3}$
- Agent 2: $\frac{4}{3}$
- Agent 3: 2
- Agent 4: 3

The only meaningful value necessary to the auctioneer is the minimum value (i.e., $\frac{4}{3}$). If all agents reveal their inflection times to the auctioneer, the auctioneer can determine that

Agent 1's valuation on bundle ABC is $22 more than its next best bundles, Agent 2's valuation on ABC is $4 more than its next best bundles, Agent 3's valuation on ABC is $6 more than its next best bundles, and Agent 4's valuation on ABC is $9 more than its next best bundles. This means that the auctioneer obtains unnecessary information from Agent 1, Agent 3, and Agent 4. However, it is possible for the auctioneer to learn that the next inflection is at $t = \frac{4}{3}$ without learning each individuals' next inflection.

## 7.1 Secure Multi-Party Computation

In cryptography, secure multi-party computation is a problem that was initially suggested by Yao [44] in a paper that he wrote in 1982. In that publication, the millionaire problem was introduced: Alice and Bob are two millionaires who want to find out which one is richer without revealing the precise amount of their wealth. Yao proposed a solution that allowed Alice and Bob to satisfy their curiosity while respecting the constraints. Suppose Alice has $i$ million dollars and Bob has $j$ million dollars, where $0 \leq i, j \leq 9$. We need a protocol for them to decide whether $i < j$, such that this result is the only information that they learn after computations. Let $M$ be the set of all $N$-bit nonnegative integers, and $Q_N$ be the set of all 1 to 1 functions from $M$ to $M$. Let $E_a$ be the public key, and $H_a$ be the private key of Alice that is generated by choosing a random element from $Q_N$.

The protocol proceeds as follows:

1. Bob selects a random $N$-bit integer $x$, and privately computes the value of $k = E_a(x)$.

2. Bob sends Alice the number $k - j$.

3. Alice privately computes the values of $y_u = H_a(k - j + u)$ for $u = 0, 1, \cdots, 9$.

4. Alice generates a random prime number $p$ of $N/2$ bits, and computes the values $z_u = y_u \bmod p$ for all $u$. If all $z_u$ values differ by at least 2 in the mod $p$ sense, then she stops the computation. Otherwise, Alice generates another random prime number, and repeats the process until all $z_u$ values differ by at least 2. Let $p, z_u$ denote the final set of numbers.

5. Alice sends the prime number $p$ and the following ten numbers to Bob: $z_0, z_1, \cdots, z_{i-1}$ followed by $z_i + 1, \cdots, z_9 + 1$.

6. Bob looks at the $(j+1)$-th number (not counting $p$) that was sent from Alice. If the $(j+1)$-th number is equal to $x \bmod p$, then $i \geq j$. If the $(j+1)$-th number is not equal to $x \bmod p$, then $i < j$.

7. Bob tells Alice the conclusion of who is the richer person.

The millionaire problem and result led to a generalization called multi-party computation protocols, which have been discussed by many researchers [6, 7, 11, 17]. In a multi-party computation, there is a given number of participants $p_1, p_2, \cdots, p_N$, each one having private data, $d_1, d_2, \cdots, d_N$, respectively. The participants want to compute the value of a public function $F(d_1, d_2, \cdots, d_N)$. A multi-party computation protocol is considered secure if no participant can learn more from the description of the public function and the result of the global calculation than what one can learn from their own entry.

Secure multi-party computation provides solutions to various real life problems, such as distributed voting [14], private bidding and auctions [45], and more.

Reiter and Rubin [30] introduced a system called Crowds for protecting users' anonymity on the Internet. The Crowds system is based on the idea of "blending into a crowd", i.e., hiding one's value within the values of others. The user's value to a server is first passed to a random member of the crowd. That member independently either submits the value directly to the server or forwards it to another member. Eventually, the value is submitted by a random member. This technique prevents the server from identifying the original sender. Moreover, the crowd members cannot determine the sender of a value because the receiver does not know whether the value is originated from, or forwarded by, the previous sender. The security is based on an assumption that the value itself does not contain any of the member's information.

## 7.2 Protecting Private Information in PTA

As discussed in Chapter 4, $\mathrm{PTA}_{\mathrm{SCPA}}$ divides the auction time into several time periods by the inflection points. In each duration, the auctioneer interacts with all agents. If the agent does not want to reveal more of his valuations than is necessary to the auctioneer and the other agents, then the interaction between the auctioneer and agents must be under the protection of a secure protocol.

Let $E_a$ be the public key, and $H_a$ be the private key of auctioneer. It is easy to privately communicate the potential demand set submission. Agents use the auctioneer's public key to encrypt the potential demand set $\hat{\mathbf{D}}_i = E_a(\hat{D}_i)$, and submit the potential demand set to the auctioneer. Then the auctioneer decrypts the potential demand set $\hat{D}_i = H_a(\hat{\mathbf{D}}_i)$.

The auctioneer solves the $\mathrm{AAM}_{\mathrm{SCPA}}$, and announces the bundle slopes $\theta_b$, the competitive allocation set $F^*$, the slope of competitive allocation $\alpha_f$, and the frequency of each competitive allocation $f$ that is being selected $\beta_f$. All of the information is public to every agent.

The problem of solving the $\text{IPM}_{\text{SCPA}}$ becomes complex because an agent cannot simply encrypt its time duration, $\Delta t_i^{DS}$, and submit the time duration to the auctioneer. That technique hides the value of $\Delta t_i^{DS}$ from others, but the auctioneer sees more information than just the smallest value of all agents' time durations. In the $\text{PTA}_{\text{SCPA}}$, the only value that is needed to continue the auction is the time until the next inflection, which may come from a change in the competitive allocation set or from a change in an agent's potential demand set.

Suppose agent $i$ has computed its demand set time duration value $\Delta t_i^{DS}$, and the auctioneer has computed the competitive allocation time duration value $\Delta t^{CA}$. The duration before a non-competitive allocation becomes competitive does not need to be hidden. Thus, the auctioneer makes the time duration public.

I introduce a secure protocol based on the solution of the millionaire problem and the Crowds technique, such that the minimum value $\Delta t = \min_i\{\Delta t_i^{DS}, \Delta t^{CA}\}$ is computed, and nobody knows any of the other agent's value. First of all, if agent $i$'s time duration value is greater than $\Delta t^{CA}$, let it update its time duration value by the public value, $\Delta t^{CA}$:

$$\Delta t_i^{DS} \leftarrow \Delta t^{CA} \quad \text{if } \Delta t_i^{DS} \geq \Delta t^{CA}.$$

To simplify the notation, we denote agent $i$'s value $\Delta t_i^{DS}$ as $t_i$ and the auctioneer's value $\Delta t^{CA}$ as $t_0$. Without loss of generality, we assume that $t_0, t_1, \cdots, t_m$ are one digit integers.

Then, all $m$ agents run the following protocol:

1. Agent $i$ selects a random $N$-bit integer $x$, and privately computes the value of $k = E_a(x)$.

2. Agent $i$ sends the number $k - t_i$ to the auctioneer.

3. The auctioneer privately computes the values of $y_u = H_a(k-j+u)$ for $u = 0, 1, 2, \cdots, 9$.

4. The auctioneer generates a random prime number $p$ of $N/2$ bits, and computes the values $z_u = y_u \bmod p$ for all $u$. If all $z_u$ values differ by at least $m$ in the mod $p$ sense, then stop. Otherwise, the auctioneer generates another random prime number and repeats the process until all $z_u$ values differ by a sufficiently larger number. Let $p, z_u$ denote the final set of numbers.

5. The auctioneer sends the prime number $p$ to Agent $i$.

6. The auctioneer sends the following ten numbers as a package to the Crowds system of all agents except for agent $i$: $z_0, z_1, \cdots, z_9$.

7. In the Crowds system, each agent has at least one chance to receive these ten numbers. After agent $j$ receives the numbers, agent $j$ sends the following ten numbers back to the Crowds system: $z_0, z_1, \cdots, z_{t_j-1}, z_{t_j} + 1, z_{t_j+1} + 1, \cdots, z_9$.

8. Finally, agent $i$ gets the ten numbers from the Crowds system. Agent $i$ looks at the $(t_i + 1)$-th number sent from the Crowds system, and finds that agent $i$'s value is the lowest if it is equal to $x \bmod p$. If the value is not equal to $x \bmod p$, then other agents have lower numbers.

Finally, after all agents finish running the preceding protocol, the auctioneer requires the agent who has the lowest number to send its value to the auction. If nobody among the agents has the lowest number, the auctioneer has the lowest number.

The auctioneer computes the bundle prices at the next inflection point, and the PTA$_{\text{SCPA}}$ proceeds to the next iteration.

The Figure 7.1 illustrates the secure PTA$_{\text{SCPA}}$ process.

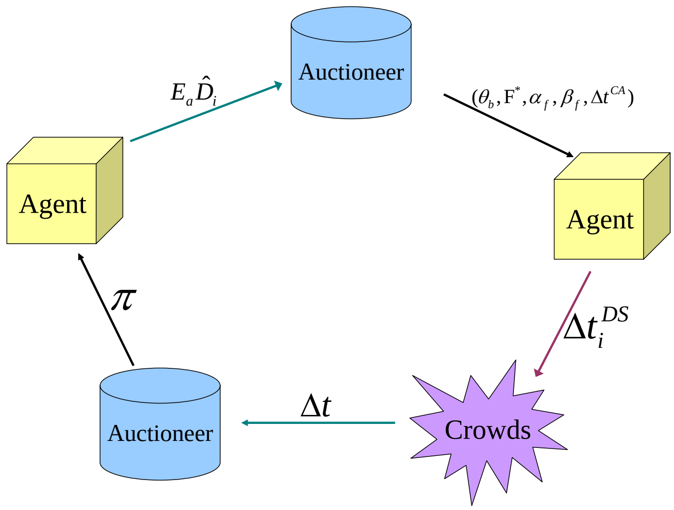


Figure 7.1: Framework of the secure PTA.

## 7.3 Fraud Detection

This section focuses on the ability of an agent to detect fraud by the auctioneer. As we know that the auctioneer solves the AAM$_{\text{SCPA}}$ and announces the bundle slopes $\theta_b$, the competitive allocation set $F^*$, the slope of competitive allocation $\alpha_f$, and the frequency of competitive allocation that is being selected $\beta_f$. Any incorrect information in the public ($\theta_b$, $F^*$, $\alpha_f$, and $\beta_f$) is seen as fraud, such as the summation of $\beta_f$'s is not equal to 1, or the slopes of the competitive allocations are not equal.

From the public data, the agents can confirm their demand set:

$$D_i = \{b \in \hat{D}_i : \theta_b \leq \theta_c,\ \forall c \in \hat{D}_i\}, \quad \forall i \in I. \tag{7.1}$$

The following theorems are useful for agents to detect fraud by the auctioneer.

**Theorem 7.3.1** *Denote $\theta_b$ as the slope of bundle $b$ and $\beta_{f^*}$ as the frequency of allocation $f^*$ being selected as a competitive allocation. Let $|f^*|$ be the number of active agents, who* *receive a bundle in allocation $f^*$. The following equation is true:*

$$\sum_{b \in B} \theta_b = m' - \sum_{f^* \in F^*} |f^*| \cdot \beta_{f^*}, \tag{7.2}$$

*where $m'$ is the number of active agents, and $F^*$ is the set of all competitive allocations.*

**Proof.** From constraint (4.19) in the $\text{AAM}_{\text{SCPA}}$, the active agent $i$'s pass value is equal to the summation of the frequencies of all competitive allocations that contain agent $i$. The expression is

$$\theta_{i,\text{pass}} = \sum_{f \in F^*} 1_{[i \in f]} \cdot \beta_f. \tag{7.3}$$

Denote $I^*$ as the set of all active agents. Then all active agents' pass value together is

$$\sum_{i \in I^*} \theta_{i,\text{pass}} = \sum_{i \in I^*} \sum_{f \in F^*} 1_{[i \in f]} \cdot \beta_f.$$

Change the summation order in the preceding expression, we get

$$\sum_{i \in I^*} \theta_{i,\text{pass}} = \sum_{f \in F^*} \left( \sum_{i \in I^*} 1_{[i \in f]} \right) \beta_f.$$

Because $\sum_{i \in I^*} 1_{[i \in f]} = |f^*|$, we get

$$\sum_{i \in I^*} \theta_{i,\text{pass}} = \sum_{f^* \in F^*} |f^*| \cdot \beta_{f^*}. \tag{7.4}$$

Note that the slope of bundle $b$ is the summation of all active agents' attention on $b$, i.e.,

$$\theta_b = \sum_{i \in I^*} \theta_{ib}. \tag{7.5}$$

Because one active agent has one unit of attention, i.e.,

$$\sum_{b \in B} \theta_{ib} + \theta_{i,\text{pass}} = 1,$$

all $m'$ active agents have $m'$ units of attention. That is

$$\sum_{i \in I^*} \left( \sum_{b \in B} \theta_{ib} + \theta_{i,\text{pass}} \right) = \sum_{b \in B} \sum_{i \in I^*} \theta_{ib} + \sum_{i \in I^*} \theta_{i,\text{pass}} = m'.$$

Substitute (7.4) and (7.5) into the preceding equation, we have

$$\sum_{b\in B}\theta_b + \sum_{f^*\in F^*} |f^*|\cdot\beta_{f^*} = m'.$$

Therefore, the theorem is proven. ■

All the values in the condition of Theorem 7.3.1 are public. Every agent can use this theorem with the public values to verify whether the auctioneer publishes correct information. The theorem states that the solution of the $\mathrm{AAM}_{\mathrm{SCPA}}$ satisfies (7.2). The theorem does not say that the value which satisfies (7.2) is the solution to the $\mathrm{AAM}_{\mathrm{SCPA}}$. Therefore, Theorem 7.3.1 is not sufficient for agents to detect fraud by the auctioneer. An individual knows is demand set from (7.1), and can use the following theorem to further detect fraud.

**Theorem 7.3.2** *Denote $\theta_b$ as the slope of bundle $b$ and $\beta_{f^*}$ as the frequency with which $f^*$ is selected as the competitive allocation. For active agent $i$, the following inequality is true:*

$$\sum_{b\in D_i}\theta_b + \sum_{f^*\in F^*, i\in f^*}\beta_{f^*} \geq 1,$$

*where $F^*$ is the set of all competitive allocations.*

**Proof.** From the constraint (4.19) of the $\mathrm{AAM}_{\mathrm{SCPA}}$, an active agent $i$ has one unit of attention,

$$\theta_{i,\text{pass}} + \sum_{b\in D_i}\theta_{ib} = 1.$$

By using (7.3), the expression can be written as

$$\sum_{b\in D_i}\theta_{ib} + \sum_{f\in F^*} 1_{[i\in f]}\cdot\beta_f = 1. \tag{7.6}$$

From (7.5), agent $i$'s attention on bundle $b$ is less than or equal to the slope of bundle $b$, that is,

$$\theta_b \geq \theta_{ib}. \tag{7.7}$$

Combine (7.7) and (7.6), we have

$$\sum_{b \in D_i} \theta_b + \sum_{f \in F^*} 1_{[i \in f]} \cdot \beta_f \geq 1.$$

Thus, we get

$$\sum_{b \in D_i} \theta_b + \sum_{f^* \in F^*, i \in f^*} \beta_{f^*} \geq 1.$$

Therefore, the theorem is proven. ■

Theorem 7.3.2 provides a necessary condition for checking the correctness of $\theta_b$, $\beta_f$, and $F^*$. By applying Theorem 7.3.2, an active agent can partially detect fraud by the auctioneer with its demand set $D_i$ and the public announced information.

The preceding two theorems are helpful to the agents. However, they do not guarantee that all types of fraud can be detected by the agents. The example in Table 7.1 shows a case in which no agent can detect the fraud that the auctioneer drops one of the competitive allocations. There are seven agents competing for four items named A, B, C, and D. Their potential demand sets and the four potential competitive allocations are shown in Table 7.1. Except for bundle CD whose price is 2 and bundle ABCD whose price is 4, the current prices of all bundles are 1. For this example, Table 7.2 shows the solution of the $\text{AAM}_{\text{SCPA}}$. The results are the following

- all agents' potential demand is their true demand set;
- the four potential competitive allocations are competitive and their slope is $\frac{4}{5}$;
- the frequency of {–, ABCD, –, –, –, –, –} is $\frac{1}{10}$, the frequency of {–, –, ABCD, –, –, –, –} is $\frac{3}{10}$, the frequency of {–, –, –, ABCD, –, –, –} is $\frac{2}{5}$, and the frequency of {–, –, –, –, A, B, CD} is $\frac{1}{5}$.

Table 7.1: Potential demand set and potential competitive allocation of the example in which the agents cannot detect fraud by the auctioneer.

| Agent | Potential Demand Set | | Potential Competitive Allocations |
|---|---|---|---|
| 1 | AB, AC | | |
| 2 | BC, ABCD | | {–, ABCD, –, –, –, –, –} |
| 3 | AB, AC, ABCD | | {–, –, ABCD, –, –, –, –} |
| 4 | ABCD | | {–, –, –, ABCD, –, –, –} |
| 5 | A, ABC, AD | | |
| 6 | B, BD, ABD | | {–, –, –, –, A, B, CD} |
| 7 | CD, ACD, BCD | | |

Table 7.2: True solution of the example in Table 7.1.

| Agent | A | B | AB | AC | BC | ABC | AD | BD | ABD | CD | ACD | BCD | ABCD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | $\frac{1}{5}$ | $\frac{4}{5}$ | | | | | | | | | |
| 2 | | | | | $\frac{4}{5}$ | | | | | | | | $\frac{1}{10}$ |
| 3 | | | $\frac{3}{5}$ | 0 | | | | | | | | | $\frac{1}{10}$ |
| 4 | | | | | | | | | | | | | $\frac{3}{5}$ |
| 5 | $\frac{4}{15}$ | | | | | $\frac{4}{15}$ | $\frac{4}{15}$ | | | | | | |
| 6 | | $\frac{4}{15}$ | | | | | | $\frac{4}{15}$ | $\frac{4}{15}$ | | | | |
| 7 | | | | | | | | | | $\frac{4}{15}$ | $\frac{4}{15}$ | $\frac{4}{15}$ | |

Suppose the auctioneer pretends the allocation {–, –, –, –, A, B, CD} is not a competitive allocation. That is, the auctioneer uses all agents' demand sets and three competitive allocations to construct the mathematical model of the $\text{AAM}_{\text{SCPA}}$. Then she solves the mixed-integer linear program of the $\text{AAM}_{\text{SCPA}}$, and gets the solution that is shown in Table 7.3. Later, the auctioneer announces the following:

- the competitive allocations are {–, ABCD, –, –, –, –, –}, {–, –, ABCD, –, –, –, –}, and {–, –, –, ABCD, –, –, –}, and their slope is $\frac{3}{4}$
- the frequency of {–, ABCD, –, –, –, –, –} is $\frac{1}{8}$, the frequency of {–, –, ABCD, –, –, –, –} is $\frac{1}{4}$, and the frequency of {–, –, –, ABCD, –, –, –} is $\frac{5}{8}$
- all slopes for the bundle are $\frac{1}{3}$, except for the slopes of AB, AC, BC, and ABCD which are $\frac{3}{4}$

Table 7.3: Fake solution of the example in Table 7.1.

| Agent | A | B | AB | AC | BC | ABC | AD | BD | ABD | CD | ACD | BCD | ABCD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | $\frac{1}{4}$ | $\frac{3}{4}$ | | | | | | | | | |
| 2 | | | | | $\frac{3}{4}$ | | | | | | | | $\frac{1}{8}$ |
| 3 | | | $\frac{1}{2}$ | 0 | | | | | | | | | $\frac{1}{4}$ |
| 4 | | | | | | | | | | | | | $\frac{3}{8}$ |
| 5 | $\frac{1}{3}$ | | | | | $\frac{1}{3}$ | $\frac{1}{3}$ | | | | | | |
| 6 | | $\frac{1}{3}$ | | | | | | $\frac{1}{3}$ | $\frac{1}{3}$ | | | | |
| 7 | | | | | | | | | | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | |

In this situation, the auctioneer removed a competitive allocation from the set of all competitive allocations. The slope of {–, –, –, –, A, B, CD} is 1, which is greater than the slope of the competitive allocations, $\frac{3}{4}$. The agent cannot detect this fraud because all agents cannot determine that bundle A, B, and CD are in different agents' demand sets. Because the auction rules require that one agent gets at most one bundle, the agents are not aware that the value of {–, –, –, –, A, B, CD} has become competitive.

However, dropping the allocation {–, –, –, –, A, B, CD} does not seem to provide any benefit to the auctioneer. So far, I have not been able to construct an example in which the auctioneer benefits by offering some incorrect information, in either the competitive allocations or the agents' demand sets. Therefore, it remains an open conjecture that all frauds that make the auctioneer profitable can be detected by at least one agent.

# Chapter 8

# Conclusion and Future Work

## 8.1 Conclusion

Auctions play an important role in the modern marketplace because many companies use auctions to buy and sell products, and millions of people shop by using auctions. However, no auction format is perfect. In this dissertation, I proposed a new type of auction called the Simple Combinatorial Proxy Auction (SCPA), which is different from existing auctions. A significant advantage of SCPA is that it has very simple auction rules.

When straightforward bidding is used, I proved that the final allocation of SCPA is the same as the existing auction type called Ascending $k$-Bundle Auction (A$k$BA). Therefore, because SCPA's bundle price computation costs less than that of A$k$BA, it is possible to speed up A$k$BA by using SCPA in the auction process. At the end of the SCPA auction, I used two linear programming models to compute the bundle prices and get the final A$k$BA payments.

The most natural algorithm to solve proxied auction problems is to simulate the incremental bidding behavior. However, this method has obvious disadvantages. In this dissertation, I presented a new approach called the Price Trajectory Algorithm (PTA) to solve iterative combinatorial auctions with proxy bidding. The PTA approach turns the thousands of rounds needed in simulation into much fewer (but more computationally costly) iterations, thus increasing the speed of computations, and resulting in a more accurate solution.

PTA computes the agents' allocation of their attention across the bundles only at "inflection points" – the points at which agents change their bidding behavior. Inflections are caused by the introduction of a new bundle into an agent's demand set, a change in the set of current competitive allocations, or the withdrawal of an agent from the set of active bidders. With the allocation of agents' attention, I can compute the slopes of price curves and increase the speed of the computation by moving from one inflection point to the next one.

I used the PTA to solve the SCPA, A$k$BA, and the Ascending Package Auction (APA). The algorithm has several advantages over the alternatives, including that PTA computes exact solutions that are independent of the bid increment or tie-breaking rules and are invariant to the magnitude of the bids. Also, I proved that solution by simulation gives a result that is a solution to the SCPA version of the PTA ($\text{PTA}_{\text{SCPA}}$).

In order to make the method appropriate for situations in which privacy is a concern, I designed a cryptographic protocol for the price trajectory algorithm. This protocol can guarantee that the auctioneer obtains only the correct and necessary information from the agents, and that there is no unnecessary information revealed between agents. Also, I provided theorems that agents can use to detect fraud by the auctioneer.

## 8.2 Future Work

In the future, I expect to extend my work in the following directions:

1. This dissertation shows that $\mathrm{PTA}_{\mathrm{SCPA}}$'s sufficiency proof is a challenging job. I have finished the necessary proofs showing that the results of a simulation method is a feasible solution to $\mathrm{PTA}_{\mathrm{SCPA}}$. My remaining work is to show that a $\mathrm{PTA}_{\mathrm{SCPA}}$ solution is one of the simulation solutions.

2. As discussed by Rakesh Vorha at the Federal Communications Commission (FCC) Combinatorial Auction Conference, a relationship between my approach and subgradient search methods exists, and is worthy of exploration and documentation.

3. To my knowledge, the PTA approach is the only algorithm that gives exact solution to SCPA, A$k$BA, and APA. I should be able to extend this approach to other auctions, such as *i*Bundle. I believe that it is an efficient way to solve all iterative auction problems.

4. In Chapter 7, I gave an example in which the agents could not detect fraud by the auctioneer. In addition, I pointed out that this fraud is not profitable to the auctioneer. Therefore, I conjecture that agents can detect fraud by an auctioneer from which the auctioneer benefits.

# Bibliography

[1] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of Fourth International Conference on Multiagent Systems*, pages 39–46. IEEE, 2000.

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

[3] Yannis Bakos and Erik Brynjolfsson. Bundling and competition on the internet. In *Marketing Science*, volume 19, pages 63–82, Linthicum, Maryland, January 2000. Institute for Operations Research and the Management Sciences (INFORMS).

[4] Sushil Bikhchandani. Ascending price vickrey auctions. In *First World Congress of the Game Theory Society (Games 2000)*, Bilbao, Spain, July 2000.

[5] Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh V. Vohra. Linear programming and vickrey auctions. In Brenda Dietrich and Rakesh V. Vohra, editors, *Mathematics of the Internet: E-Auction and Markets*, IMA Volumes in Mathematics and its Applications, pages 75–115. New York: Springer Verlag, 2002.

[6] Felix Brandt. Cryptographic protocols for secure second-price auctions. In M. Klusch

and F. Zambonelli, editors, *Cooperative Information Agents V*, volume 2182 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 154–165. Springer, 2001.

[7] Ran Canetti. *Studies in Secure Multiparty Computation and Applications*. PhD thesis, 1996.

[8] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[9] Peter Cramton. Competitive bidding behavior in uniform-price auction markets. In *Proceedings of the Hawaii International Conference on System Sciences*, January 2004.

[10] Sven de Vries and Rakesh V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on computing*, 15(3):284–309, 2003.

[11] Matthew K. Franklin and Michael K. Peiter. The design and implementation of a secure auction service. *IEEE Transactions on Software Engineering*, 22(5):302–312, 1996.

[12] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, August 1999.

[13] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[14] Ben Hardekopf, Kevin Kwiat, and Shambhu Upadhyay. Secure and fault-tolerant voting in distributed systems. In *IEEE Proceedings of Aerospace Conference*, volume 3 of *IMA Volumes in Mathematics and its Applications*, pages 1117–1126, Big Sky, Montana, 2001.

[15] Karla Hoffman, Dinesh Menon, and Thomas Wilson. Observations and near-direct implementations of the ascending proxy auction. December 2003.

[16] eBay Inc. announces third quarter 2004 financial results. *http://www.sec.gov/Archives/edgar/data/1065088/000095013404015333/f02502exv99w1.txt*, 2004.

[17] Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In *Proceedings of Asiacrypt-00*, pages 162–177, 2000.

[18] Hans Kellerer, Ulrich Pferschy, and David Pisinger. Knapsack problems. ISBN 3-540-40286-1. Springer, 2004.

[19] Terence Kelly. Generalized knapsack solvers for multi-unit combinatorial auctions. volume http://ana.lcs.mit.edu/peyman/amecvi.htm, New York, July 2004.

[20] Herman B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–479, 1983.

[21] Kevin A. McCabe, Stephen J. Rassenti, and Vernon L. Smith. Auction institutional design: Theory and behavior of simultaneous multiple-unit generalizations of the dutch and english auctions. *American Economic Review*, 80(5):1276–1283, 1990.

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

[23] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the Second ACM Conference on Electronic Commerce*, pages 1–12, 2000.

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

[25] David C. Parkes and Lyle H. Ungar. Iterative combinatorial auctions: Theory and practice. In *Seventeenth National Conference on Artificial Intelligence*, pages 74–81, 2000.

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

[27] Aleksandar Pekeč and Michael H. Rothkopf. Combinatorial auction desgin. *Mangenment Science*, 49(11):1485–1503, November 2003.

[28] S. J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13(2):402–417, 1982.

[29] Baharak Rastegari. An introduction to iterative combinatorial auctions. *Technical report*, 2005.

[30] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[31] Juan Rodríguez, Pablo Noriega, Carles Sierra, and Julian 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)*, pages 207–224, London, UK, April 1997.

[32] Alvin E. Roth and Axel 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.

[33] Michael H. Rothkopf and Ronald M. Harstad. Two models of bid-taker cheating in vickrey auctions. *Journal of Business*, 68(2):257–267, 1995.

[34] Michael H. Rothkopf, Aleksandar Pekeč, and Ronald M. Harstad. Computationally manageable combinatorial auctions. *Mangenment Science*, 44(8):1131–1147, Auguest 1998.

[35] Michael H. Rothkopf, Thomas J. Teisberg, and Edward P. Kahn. Why are Vickrey auctions rare? *Journal of Political Economy*, 98(1):94–109, 1990.

[36] Tuomas W. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, 2002.

[37] Tuomas W. Sandholm and Subhash Suri. BOB: Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence*, 145:33–58, 2003.

[38] Tuomas W. Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Seventeenth International Joint Conference on Artificial Intelligence*, pages 1102–1108, Seattle, WA, 2001.

[39] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[40] Peter R. Wurman, Gangshu Cai, Jie Zhong, and Ashish Sureka. An algorithm for computing the outcome of combinatorial auctions with proxy bidding. In Norman Sadeh,

editor, *Fifth International Conference on Electronic Commerce*, pages 1–8, Pittsburgh, PA, 2003.

[41] Peter R. Wurman and Michael P. Wellman. A$k$BA: A progressive, anonymous-price combinatorial auction. In *In the Second ACM Conference on Electronic Commerce*, pages 21–29, 2000.

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

[43] Peter R. Wurman, Jie Zhong, and Gangshu Cai. Computing price trajectories in combinatorial auctions with proxy bidding. *Electronic Commerce Research and Applications*, 3(4):329–340, 2004.

[44] Andrew C. Yao. Protocols for secure computations. In *Twenty-third Annual Symposium on Foundations of Computer Science*, volume 3-5, pages 160–164, Chicago, Illinois, November 1982. IEEE.

[45] Makoto Yokoo and Koutarou Suzuki. Secure generalized vickrey auction without third-party servers. In Ari Juels, editor, *Financial Cryptography: 8th International Conference*, volume 3110, pages 132–146, Key West, Florida, February 2004. Springer Berlin/Heidelberg.

[46] Jie Zhong and Peter R. Wurman. A framework for computing the outcome of proxied combinatorial auctions. In *IEEE Conference on Electronic Commerce*, pages 25–32, Munich, Germany, 2005.