The Dissertation Committee for Chinmayi Krishnappa
certifies that this is the approved version of the following dissertation:

# Unit-Demand Auctions:
# Bridging Theory and Practice

Committee:

C. Greg Plaxton, Supervisor

Anna Gal

Adam Klivans

Peter Stone

Sriram Vishwanath

# Unit-Demand Auctions: Bridging Theory and Practice

**by**

**Chinmayi Krishnappa, B.E.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2011

For MOM

Who taught us by example, that no situation exempts you from being cheerful, kind, and hardworking.

# Acknowledgments

I am indebted to my advisor, Greg Plaxton, for being a constant source of inspiration, guidance, and support throughout my years as a PhD student. He has been infinitely patient when I have (often) slacked off at work and has come to my rescue every time I thought I had reached a dead-end. His perseverance in battling any problem for hours has inspired me to work harder, and his exuberance when we make a breakthrough has often left me questioning my own levels of passion. Everything I know about formal writing, I have learnt from Greg. I hope this dissertation goes some way in expressing my gratitude for all that he has done for me.

I would like to thank my committee members — Anna Gal, Adam Klivans, Peter Stone, and Sriram Viswanath, for their valuable comments and feedback. I am also grateful to the CS department staff — Gloria, Lydia, and Katherine, for being extremely nice to me, and for going out of their way to ensure that I was meeting every departmental deadline.

I am blessed with a loving family who tell me that they are proud of me regardless of my achievements. Thank you Dad, for your encouragement and unshakeable (sometimes irrational?) faith in my abilities! Mom, I owe everything to you. You are without doubt the strongest woman I've known to date. Thank you for all the effort you put into providing us with the best of education and for encouraging us to dream big. I love you. You are my strongest source of inspiration and the best role model I could ask for.

A special thank you to my brother Chirayu. To me, you are more than any super-hero! I can't thank you enough for all the sound advice that you have generously doled out over the years. You always have a logically sound solution to any problem I may have. I guess I hit the jackpot with you – I am truly lucky to have you in my life and in a role that has no ifs and buts, or an expiry date.

My life in Austin has been enriched by a wonderful set of friends who ensured that I enjoyed myself despite any setbacks at work. I will miss you all greatly and I hope we remain friends for a long time. I will fondly remember the hot cups of coffee, the long walks, and the fun conversations, both professional and social, that we have shared over the years.

Finally, I am grateful for the PhD experience. I am happy to have lived it in my own unique way. It has brought the realization that as daunting as it may seem, a dissertation is but a modest contribution. I hope the future has only bigger and better things in store.

# Unit-Demand Auctions: Bridging Theory and Practice

Publication No. ____________________

Chinmayi Krishnappa, Ph.D.
The University of Texas at Austin, 2011

Supervisor: C. Greg Plaxton

Unit-demand auctions have been well studied with applications in several areas. In this dissertation, we discuss new variants of the unit-demand auction that are motivated by practical applications. We design mechanisms for these variants that have strong properties related to truthfulness, efficiency, scalability, and privacy. The main contributions of this dissertation can be divided into two parts.

In the first part, we introduce a new variant of the classic sealed-bid unit-demand auction in which each item is associated with a put option; the put option of an item gives the seller the right to sell the item at a specified strike price to a specified bidder, regardless of market conditions. We motivate our unit-demand auction setting by discussing applications to the reassignment of leases, and to the design of multi-round auctions. For the classic sealed-bid unit-demand framework, the VCG mechanism provides a truthful auction with strong associated guarantees, including efficiency and envy-freedom. For an item in our auction, the strike price of the associated put imposes a lower bound on the auction price. Due to these lower bound constraints on auction prices, we find that the VCG mechanism is not suitable for our setting. Instead, our work draws on two fundamental techniques, one from the realm of mechanism design for numerical preferences — the dynamic unit-demand approximate auction of Demange, Gale, and Sotomayor — and one from the realm of mechanism design for ordinal preferences — the Top Trading Cycles algorithm — to obtain a natural auction that satisfies the lower bound constraints on auction prices. While we cannot, in general, achieve either efficiency or envy-freedom in our setting, our auction achieves suitably relaxed versions of these properties. For example, this auction is envy-free for all bidders who do not acquire an item via the exercise of a put. We provide a polynomial time implementation of this auction. By breaking ties in an appropriate manner, we are able to prove that this auction is truthful.

In the second part, we specify rules for a dynamic unit-demand auction that supports arbitrary bid revision. In each round, the dynamic auction takes a tentative allocation and pricing as part of the input, and allows each bidder — including a tentatively allocated bidder — to submit an arbitrary unit-demand bid. Each round of our dynamic auction is implemented via a single application of the sealed-bid unit-demand auction proposed in the first part. We show that our dynamic auction satisfies strong properties related to truthfulness and efficiency. Using a certain privacy preservation property of each round of the auction, we show that the overall dynamic auction is highly resistant to shilling. We present a fast algorithm for implementing the proposed auction. Using this algorithm, the amortized cost of processing each bidding operation is upper bounded by the complexity of solving a single-source shortest paths problem on a graph with nonnegative edge weights and a node for each item in the auction. We also propose a dynamic price adjustment scheme that discourages sniping by providing bidders with incentives to bid early in the auction.

# Table of Contents

# Chapter 1

# Introduction

Online auctions have become increasingly popular over the last decade. The overwhelming majority of online auctions are single-item auctions: multiple bidding agents bid to win a single item. In contrast, the field of combinatorial auction design, which enjoys a rich history in the academic literature, is concerned with the sale of multiple items in a single auction. In such auctions, agents are not limited to submitting single-item bids; for example, some combinatorial auctions permit bids on bundles of items. Many different combinatorial auctions have been studied in the literature. However, auctions involving multiple items, and especially auctions involving multiple distinct items, are not widely encountered in practice. Significant deterrents to the widespread adoption of combinatorial auctions include increased bid complexity and the computational difficulty of determining a suitable allocation and pricing of the items.

## 1.1 Overview

In this dissertation, we study unit-demand auctions. A unit-demand auction is a restricted kind of combinatorial auction in which an agent is allowed to make a separate offer on each of a number of items, with the guarantee that at most one of these offers will be accepted.

Consider the following concrete example of a real-world auction scenario. The developer of a new high-rise condominium project wishes to sell all of its units to the public. In this setting, each agent may assign a different value to each unit, depending on factors such as floor plan, elevation, and view. An agent in this auction is said to have unit-demand preference if the agent is seeking to purchase at most one unit. In a unit-demand auction, the bid of an agent takes the form of a unit-demand preference function: The agent specifies an offer for each of a subset of items, with the understanding that the bid can win at most one item. Typical online auction houses do not support such unit-demand bids. Instead, if many items are to be sold, each is sold in a separate auction. The resulting sequence of single-item auctions forces an agent with unit-demand preferences to guess whether or not to bid on each successive item, since the agent does not know the eventual selling prices of the items. This guesswork degrades the efficiency of the allocation of items to agents, where the efficiency of an allocation is defined as the sum, over all items $v$, of the value assigned to $v$ by the agent to which $v$ is allocated. The main reason to contemplate selling many items within a single unit-demand auction, or within any form of combinatorial auction, is to reduce the need for such guesswork, thereby enhancing efficiency. By improving efficiency, one has the potential to improve the quality of the outcome for both buyers and sellers alike.

Most online auction sites are dynamic. A dynamic auction proceeds in rounds. At the beginning of each round, new bid data (bid revision requests and new bids) is received, and an update rule is applied to adjust the tentative outcome (allocation and pricing). The tentative outcome is made public at the end of each round. Dynamic auctions facilitate value discovery — agents use their observations of the tentative outcomes published by the auction thus far to guide their bidding behavior in the following rounds. By enhancing value discovery, dynamic auctions make it easier to allocate items to the agents who value them the most. Several important considerations are associated with designing a dynamic auction. It is important that agents in a dynamic auction are allowed to flexibly change their bids across rounds. If a dynamic auction is overly restrictive with respect to bid revision, then there is a significant chance that agents will limit their participation to the last round of the auction. It is also desirable to provide agents in a dynamic auction with some strategy-proof guarantees. As with restricted bid revision, strategic bid considerations can have the potential to discourage agents from bidding early in the auction.

In the standard sealed-bid unit-demand context, one can apply the well-known Vickrey-Clarke-Groves (VCG) mechanism [41, 7, 16] to obtain an auction that is truthful, efficient, and envy-free [42]. In this dissertation, we analyze variants of the unit-demand auction in both the sealed-bid and dynamic settings. The first contribution of this dissertation is a variant of the standard sealed-bid unit-demand auction in which each item is associated with a "put" option; a put option of an item models an initial tentative allocation and pricing of the item. For the second contribution of this dissertation, we use the results on the sealed-bid unit-demand auction from the first part to design a dynamic unit-demand auction that supports arbitrary bid revision.

Below we provide an informal sketch of our proposed auctions. A formal presentation of the auctions is provided in the following chapters of this dissertation.

## 1.2 A Sealed-Bid Unit-Demand Auction with Put Options

In this section, we provide an overview of our proposed sealed-bid unit-demand auction. A formal presentation of this auction is available starting at Chapter 3 and continuing upto Chapter 8 wherein we describe the auction and establish its properties.

A put option is a commitment between two parties — the "holder" of the put and the "target" of the put. The holder of the put possesses the right to sell a specified asset to the target of the put at a specified "strike price", regardless of the current market prices.

For the first contribution of this dissertation, we consider a new variant of the classic sealed-bid unit-demand auction in which each item is associated with a predetermined put option that expires when the auction terminates. The holder of an item's put is the seller of the item and the target is an agent in the auction under the constraint that no agent is the target of more than one put. We restrict attention to the case of no side-payments — the outcome of the auction consists of an allocation and pricing of the items, and each agent who is allocated an item pays the item's price to the seller of the item.

This auction setting finds motivation in several applications, some of which are described below. As a first application, we introduce the following "Lease Exchange" problem. Consider a number of leased apartments and a number of agents who have unit demand preferences for renting the apartments. The lessees of some apartments seek to break their current leases. We would like to reallocate and reprice the apartments such that each lessor receives at least the monthly rent being paid by the current lessee for the remainder of the lease term. The lease exchange problem can be modeled as an instance of a unit-demand auction with put options in the following way. Each apartment is an item in our auction. The lessor of each apartment holds a put of the apartment whose target is the current lessee and whose strike price is equal to the current monthly rent of the apartment. In practice, a lease involves many other important considerations including varying lease terms and lessee specific adjustments that have been ignored in our simple example. Such factors are easily handled by allowing lessors to specify additive amounts for each option and incorporating these amounts into the bidding based on the options chosen by each agent. We formulate a suitable solution concept for this auction setting and we design a truthful auction that implements this solution concept.

In combinatorial auction design, it is often useful to follow a two-phase approach e.g. the clock-proxy auction proposed by Ausubel et al. [5]. A second application of our auction is in implementing the second phase of a two-phase combinatorial auction. In general, our auction is a suitable candidate for implementing the last round of any dynamic unit-demand auction.

A natural third application of our auction is in the design of a dynamic unit-demand auction in which each round is resolved using our proposed sealed-bid unit-demand auction. In the first round of the dynamic auction, the seller of each item holds a put whose target is a dummy agent and whose strike price is equal to the reserve price of the item. In each subsequent round, each item is associated with a put whose target is the agent who is tentatively allocated to the item from the previous round, and whose strike price is the price of the item determined by the auction in the previous round. Such a dynamic auction generalizes the eBay auction to the unit-demand setting.

As previously discussed, in the standard sealed-bid unit-demand context, the VCG mechanism provides an auction that is truthful, efficient, and envy-free [42]. For an item in our setting, the strike price of the item imposes a lower bound on the auction price of the item — by exercising the item's put, the seller of the item can ensure that the auction price is at least as high as the strike price. Due to these lower bound constraints on auction prices, we find that the VCG mechanism is not well-suited for our setting.

Moreover, in our setting, we cannot guarantee the strong properties that are achieved by the VCG mechanism in the classic setting. For example, consider an auction instance in which no agent bids on a particular item. The auction would be forced to allocate the item to the target of its put at its associated strike price even if such an allocation violates the envy-freedom property of the target. Accordingly, we formulate a new solution concept that is appropriate for our setting (see Chapter 4).

### 1.2.1 A two-phase approach.

We construct a two-phase auction that draws on two fundamental techniques, one from the realm of mechanism design for numerical preferences — the dynamic unit-demand approximate auction of Demange et al. [10] — and one from the realm of mechanism design for ordinal preferences — the Top Trading Cycles (TTC) algorithm [37]. In what follows, we will refer to the dynamic unit-demand approximate auction of Demange et al. as DGS-approximate. The DGS-approximate auction is an ascending-price auction that proceeds in rounds. In each round, agents that are not tentatively allocated are consulted in round-robin order and given the opportunity to either select an item, or pass. If an unallocated agent $u$ selects an item $v$, the tentative price of item $v$ is increased by a parameter $\delta$, and the tentative allocation is updated to reflect that item $v$ is allocated to agent $u$. The DGS-approximate algorithm terminates when all of the unallocated agents pass.

Informally, the first phase of our auction corresponds to the following proxy version of DGS-approximate. We fix an initial tentative allocation and pricing of the items as follows: each item is allocated to the target of its put and has a price equal to the strike price of its put. We associate with each agent $u$, a proxy agent $u'$ who employs the following strategy to bid on behalf of $u$ in each round of DGS-approximate: if the tentative price on every item exceeds $u$'s offer on the item, then $u'$ passes; otherwise, $u'$ selects an item with the highest utility for $u$ (difference between $u$'s offer on the item and the tentative price of the item). On termination of the auction, we identify as "unhappy" each agent who is allocated to an item, but strictly prefers some other item at the current prices. It is easy to see that the set of unhappy agents are a subset of the agents in the initial tentative allocation. We note that in the limit as $\delta$ approaches zero, the proxy based DGS-approximate auction achieves a relaxed form of efficiency: the auction is efficient if the unit-demand bid of each unhappy agent is replaced with a single offer on its allocated item equal to the strike price of the item. The proxy based DGS-approximate auction also achieves a relaxed form of envy-freedom — the auction is envy-free for all agents other than the set of unhappy agents.

The second phase of our auction corresponds to a single application of the TTC algorithm on a suitably defined instance of the house allocation problem [37, 36]. The second phase of our auction affects only the allocation, and keeps the item prices unchanged. The proposed two-phase auction computes an outcome in the weak core and achieves the relaxed forms of efficiency and envy-freedom that are described with respect to the first phase. Alternatively, by employing the $\text{TC}^{\prec}$ algorithm of Jaramillo and Manjunath [19] in the second phase, we achieve Pareto-efficiency of our two-phase auction. The $\text{TC}^{\prec}$ algorithm runs in polynomial time and produces a strategy-proof and Pareto-efficient outcome for the house allocation problem in the absence of strict preferences.

The two-phase approach proposed above has two shortcomings. Firstly, the auction is not truthful for any positive value of $\delta$. Secondly, the $\delta$ parameter associated with the DGS-approximate auction leads to a trade-off between speed and efficiency in the first phase. The running time of the first phase increases as $\delta$ diminishes (it takes $O(1/\delta)$ time to increase the item prices by a constant). Furthermore, for large values of $\delta$, the auction is not efficient, even in the relaxed form discussed above. We address these shortcomings in our work. We provide a polynomial time implementation of the first phase auction (see Section 8.7). By carefully breaking ties, we successfully obtain a truthful first phase auction (see Lemma 8.4.9). The composition of two truthful auctions is not necessarily truthful. We successfully show that the two-phase auction obtained by composing the truthful first and second phases is also truthful (see Lemma 8.4.10).

## 1.3 A Dynamic Unit-Demand Auction Supporting Arbitrary Bid Revision

In this section, we provide an overview of our proposed dynamic unit-demand auction. We use the results of the sealed-bid auction referred to in Section 1.2 to provide a formal presentation of our dynamic auction in Chapter 9.

We noted earlier, that in the standard sealed-bid framework, the VCG mechanism yields a truthful auction with an efficient allocation and envy-free pricing [42]. However, the majority of auction sites, including the popular site eBay, are dynamic. In a dynamic auction, bidding takes place in multiple rounds. In each round, new bid data (bid revision requests and new bids) is received, and an update rule is applied to adjust the tentative outcome (allocation and pricing). The tentative outcome is made public at the end of each round. This dynamic price feedback enables agents to concentrate their value discovery efforts on the most relevant items.

Unit-demand bids are much more expressive than traditional single-item bids, and bid formulation is correspondingly more complex. Accordingly, in a dynamic auction, there is a significant chance that a tentatively allocated agent may wish to revise one or more bid components. If a unit-demand auction imposes undue constraints on bid revision, or if the semantics of bid revision introduce additional strategic considerations, then agents may be reluctant to submit unit-demand bids or may only choose to submit bids in the last round of the auction. Such an artificial reduction in the number of bids directly undercuts the main value propositions of dynamic auctions, namely value discovery and improved efficiency.

For the second contribution of this dissertation, we specify rules for a dynamic unit-demand auction that supports arbitrary bid revision. We note that each round of a dynamic auction is essentially a sealed-bid auction. A guiding principle that we follow in the design of our auction is to use the same sealed-bid auction to resolve each round of the dynamic auction. This principle is natural and also rules out certain trivial solutions, e.g, an auction that postpones all of its processing to the last round. In order to motivate the design of our auction, we analyze the special case of our auction setting for a single item. In what follows, we discuss the design of a dynamic single-item auction supporting arbitrary bid revision.

As a natural first approach, we consider using the well-known Vickrey auction to resolve each round. However, it is easy to see that such a dynamic auction discards information on the tentative outcome of each round and is essentially equivalent to running the Vickrey auction exactly once in the last round, after all of the bids have been received. Thus, this approach destroys value discovery, a key feature of dynamic auctions.

Next we consider resolving each round of a dynamic single-item auction in the style of the California auction [38] formulated by Steiglitz. The California auction is a dynamic single-item auction that in each round, allocates the item to the highest bidding agent and posts the second highest bid seen up to that round as the tentative price of the item. The California auction is efficient, satisfies envy-freedom, and retains straightforward bidding in an ex-post Nash equilibrium. The eBay auction is an example of the California auction. We note that each round of the California auction can be viewed as an instance of the Vickrey auction with a reserve price, where in each round, the reserve price of the item is set equal to the tentative price of the item from the previous round. In order to support arbitrary bid revision, in each round, we associate the item with the tentatively allocated agent of the previous round as a reserve agent: if every agent bids less than the tentative price of the item in a round, then the item remains allocated to the reserve agent in that round.

It is straightforward to see that when arbitrary bid revision is allowed, the California auction ceases to be efficient. Additionally, truthful bidding is no longer an ex-post Nash equilibrium of the auction. For example, consider an instance of the California auction with an item $v$ and agents $u_0$ and $u_1$. Agent $u_0$ values item $v$ at 20 units in round $i$ and at 10 units in a later round $j$. Agent $u_1$ values item $v$ at 19 units in round $i$ and at 20 units in round $j$. If agent $u_1$ bids truthfully, then agent $u_1$ wins item $v$ for 19 units in round $j$. However, by choosing not to bid in round $i$ and by submitting a bid of 20 units in round $j$, agent $u_1$ stands to win item $v$ for a lower price of 10 units in round $j$. Nonetheless, while the overall auction is not truthful, it can be shown that each individual round of this auction is truthful.

Sealed-bid auctions exhibit strong properties related to truthfulness and efficiency; yet, the overwhelming majority of online auctions continue to be dynamic. The popularity of dynamic auctions suggests that value discovery is one of the most important requirements of online auctions. As we discussed earlier, value discovery becomes increasingly important with increased bid complexity. From our discussion of the dynamic single-item auction setting above, it follows that there is an inherent tradeoff between flexibility of bid revision and the attainability of desirable properties such as truthfulness and efficiency. We find that the strong properties of truthfulness and efficiency are lost when arbitrary bid revision is introduced, even in the restricted single-item case. We generalize this trade-off to the unit-demand setting in the design of our proposed dynamic auction.

In keeping with our approach of using the same sealed-bid auction to resolve each round of the dynamic auction, we seek to identify a suitable sealed-bid unit-demand auction that generalizes the California auction to the unit-demand setting. In our discussion of the California auction, we observed that each round of the California auction with arbitrary bid revision is equivalent to an instance of the Vickrey auction where each item is associated with a reserve price and a reserve agent. Such a Vickrey auction with reserves can be viewed as a Vickrey auction in which the item is associated with a put option held by the item's seller, with the reserve agent being the target of the put and the reserve price being the strike price of the put. The put option of the item gives the seller of the item the right to sell the item to the reserve agent at the reserve price, regardless of market conditions.

Our proposed dynamic auction proceeds as follows. Each round of our dynamic auction is resolved using an instance of the sealed-bid unit-demand auction with put options proposed as a first contribution of this dissertation. Each item is associated with a put option held by the item's seller. In the first round, the target of each item's put is the seller of the item and the strike price is the reserve price of the item. In each subsequent round, the target and strike price of an item's put are given by the tentatively allocated agent and the tentative price of the item from the previous round.

Since each round of our dynamic auction is resolved using the sealed-bid unit-demand auction with put options, the outcome of each round satisfies all of the equilibrium properties associated with the sealed-bid auction (see Section 4.3). For example, like the sealed-bid auction, each round of our dynamic auction is truthful (see Section 9.3.1). In Section 9.3 we establish various properties of our dynamic auction that hold over multiple rounds of the auction. In the description of our auction, we resolve each round using the same sealed-bid auction. However, the technical claims of Section 9.3 hold more generally for dynamic auctions in which each round is resolved using any sealed-bid auction that satisfies certain subsets of the equilibrium properties detailed in Section 4.3.

With regard to efficiency, recall that in the absence of bid revision, the California auction produces an efficient allocation in each round. In the absence of bid revision, our dynamic auction mimics the behavior of the California auction, and hence achieves the same efficiency guarantee. When agents are allowed to revise their bids in an arbitrary manner, such an efficiency guarantee cannot be achieved without sacrificing other key properties. Since we do not wish to sacrifice these properties, we instead achieve the following relaxed form of efficiency achieved by the sealed-bid unit-demand auction with put options — while the current allocation need not be efficient with respect to the current revision of each bid, it is guaranteed to be efficient with respect to a suitable combination of previous and current revisions. We derive additional efficiency related properties pertaining to multiple rounds of our auction in Section 9.3. We show that if an agent is envy-free in a round of the auction, then for any sequence of subsequent rounds in which the agent's preference does not change and the agent does not submit a bid revision, the agent continues to remain envy-free (see Theorem 9.3.1). We also show that for any sequence of rounds in which an agent is not envy-free, does not submit a bid revision, and does not change its preference, the auction can only make progress towards achieving efficiency with respect to the most recent bid revision of the agent (see Theorem 9.3.2). We believe that our efficiency-related guarantees are essentially the strongest that can be achieved without sacrificing other properties.

An important consideration in the design on a dynamic auction is its vulnerability to "shill" bidding. If the seller of an item can deduce the maximum price that the agent who is tentatively allocated to the item is willing to pay for the item, then the seller can extract this price without forfeiting sale of the item by submitting a shill offer just below the agent's offer. Dynamic auctions are known to be particularly susceptible to shill bidding [38]. Thus, a goal of our dynamic auction is to ensure bid privacy for tentatively allocated agents. We establish bid privacy of an agent $u$ in our dynamic auction with respect to the grand coalition of all agents in the auction except agent $u$. We assume that in each round of our auction, the grand coalition learns the matching and allocation published in the round, the bid of every agent in the round except the bid of agent $u$, and whether agent $u$ submitted a bid in the round. We show that our dynamic auction is resistant to shilling by such a grand coalition of agents. Specifically, we show that for any sequence of rounds in which agent $u$ does not submit a bid revision, the grand coalition of agents cannot shill agent $u$ by more than one unit without risking forfeiture of sale in one of the rounds (see Theorem 9.3.3). Since the running time of our auction is independent of the monetary units used, each unit can be considered to be as low as one cent, thus making our auction highly resistant to shilling.

With respect to scalability, our fast implementation of the proposed dynamic auction (see Section 9.4) processes each bidding operation (i.e., new bid or bid revision) using an amortized constant number of Hungarian [24] augmentations, thereby matching the asymptotic complexity associated with the sealed-bid auction, which uses a single augmentation to process each new bid. The worst-case time complexity of such an augmentation is upper bounded by the cost of running a single-source shortest paths computation on a graph where the number of nodes is proportional to the number of items, and where the number of edges is proportional to the total number of "active" bid components of the tentatively allocated agents. (A component of a unit-demand bid is considered active if the associated offer is at least the current price of the associated item.)

In supporting arbitrary bid revision in the unit-demand setting, our dynamic auction successfully achieves the properties that the California auction achieves with arbitrary revision. For any bid revision operation, our dynamic auction immediately admits a closest approximation to the revised bid, and as prices change over rounds, our auction continually admits closer and closer approximations to the revised bid. An important feature of our auction is that in the special case where bid revisions are "consistent" – such revisions involve raising all components of the unit-demand bid by the same amount – the outcome of our auction is equivalent to the celebrated VCG outcome. For the above mentioned reasons, we believe that our proposed dynamic auction is an appropriate generalization of the California auction to the unit-demand setting.

An issue of practical concern in dynamic auctions is "sniping". Sniping refers to agents holding off on submitting bids until close to the end of the auction. Such late bidding impedes the value discovery process, thereby degrading efficiency. We propose a dynamic price adjustment scheme that encourages agents to bid early in the auction, thus discouraging sniping (see Chapter 10). With this proposed price adjustment scheme, our auction continues to satisfy all of the strong theoretical properties established for the basic version of our auction.

## 1.4 Related Work

The theory of two-sided matchings has a rich history. In recent work, Fujishige and Tamura [13] show the existence of a stable matching in a generalized many to many matching model with upper and lower bounds on payments. The model proposed by Fujishige and Tamura generalizes various previous two-sided matching models; see [13] for a discussion of the relevant literature. Like much of the prior work in this line of research, the work of Fujishige and Tamura does not address issues related to incentive compatibility. In applying the theory of two-sided matchings to the design of auctions, a fundamental challenge is to identify two-sided matching models where truthfulness is achievable. Aggarwal et al. [1] address this challenge for a special case of Fujishige and Tamura's model with applications to sponsored search auctions. Specifically, for the unit-demand auction setting with agent and item specific minimum and maximum prices, Aggarwal et al. provide a truthful auction that computes an agent optimal stable matching. Similarly, for the setting considered in the present work, our central focus is to obtain a truthful auction.

In the auction of Aggarwal et al., each agent submits a value and a maximum price that the agent is willing to pay for each item. We observe that the algorithm of Aggarwal et al. can be used to implement the first phase of our auction in the special case where each agent that is the target of a put values the associated item strictly below than the strike price of the put. This special case of our auction can be modeled in the framework of Aggarwal et al. as follows: For each item and agent pair where the agent is the target of the item's put, we submit a value of infinity and a maximum price equal to the strike price of the put; for every other agent and item pair, we submit the agent's offer for the item as the value and maximum price; we set the reserve price of each item to the strike price of its put.

Recall the lease exchange problem that was discussed in Section 1.2. A lessee may sometimes value his leased apartment below its current rent. In such cases, the first phase of our auction can be implemented using the algorithm of Aggarwal et al. [1] as described above. However, it is not difficult to see that a lessee's value for his leased apartment may in certain cases be higher than the current rent. Note that the decision to put the apartment in the auction may not rest with the lessee; in such situations, the lessee may be willing to pay a higher rent to retain his current apartment. Even in situations where the lessee decides to put the apartment in the auction, it is not uncommon to have lessees who are willing to risk winning their current apartments at higher monthly rates.

Demange et al. [10] present two dynamic unit-demand auctions: an "exact" auction, which we refer to as DGS-exact, and an "approximate" auction, which we refer to as DGS-approximate. In each round, the DGS-exact auction elicits the demand (i.e., set of preferred items at the current prices) of each agent. If there is an overdemanded set of items, a minimal overdemanded set is found, and the prices of all items in the set are incremented by one. If no overdemanded set can be found, the DGS-exact auction terminates and each item is allocated to an agent who demands it. Observe that the DGS-exact auction implicitly supports a limited form of bid revision: An agent is free to revise its unit-demand bid as long as the demands specified in all preceding rounds remain consistent with the revision.

Recognizing the highly restrictive nature of the form of bid revision permitted by the DGS-exact auction, Demange et al. propose the DGS-approximate auction. Like DGS-exact, DGS-approximate is an ascending-price auction. (We remark that Mishra and Parkes [29] describe exact and approximate descending price auctions corresponding to DGS-exact and DGS-approximate.)

Agents that are not tentatively allocated are consulted in round-robin order and given the opportunity to either select an item, or pass. If an unallocated agent $u$ selects an item $v$, the tentative price of item $v$ is increased by a parameter $\delta$, and the tentative allocation is updated to reflect that item $v$ is allocated to agent $u$. The DGS-approximate algorithm terminates when all of the unallocated agents pass. The DGS-approximate auction has several shortcomings in comparison with our dynamic unit-demand auction: the auctioneer is required to specify a value for the parameter $\delta$; the outcome is guaranteed to be approximately efficient/truthful, even in the absence of bid revision; there is a tradeoff between the quality of the approximation and the running time of the algorithm; and the bid revision framework is restrictive, since it does not allow for trading of items between tentatively allocated agents.

Gul and Stacchetti [18] present a dynamic auction that generalizes the DGS-exact auction for the setting in which agents demand bundles of items. Gul and Stacchetti show that their auction converges to the smallest Walrasian prices, and that their auction is strategy-proof if the smallest Walrasian prices correspond to the VCG payments. Gul and Stacchetti's auction, like the DGS-exact auction, supports a limited form of bid revision: An agent is free to revise its bid on a bundle as long as the demands on the bundle specified in all preceding rounds remain consistent with the revision.

General combinatorial auctions support more complex preferences than unit-demand preferences, such as preferences for bundles of items. Unfortunately, for many combinatorial auctions, the problem of finding an efficient allocation is NP-hard. The computational intractability of general combinatorial auctions motivates the study of specialized combinatorial auctions. Rothkopf et al. discuss special cases (including unit-demand) of combinatorial auctions where the problem of finding an efficient allocation can be solved in polynomial time [35]. Various generalizations of unit-demand have been considered in the literature, including work on dynamic auctions for homogeneous [3, 6] and heterogeneous [4, 9, 28, 34] commodities.

## 1.5 Organization

We present and analyze our auctions in layers. In the following chapters, we present five unit-demand auctions — the bottom-level auction, the mid-level auction, the top-level auction, our proposed sealed-bid unit-demand auction with put options, and our proposed dynamic unit-demand auction. Each round of the dynamic auction is resolved as an instance of our proposed sealed-bid unit-demand auction. As discussed above, the sealed-bid unit-demand auction consists of two phases. The first phase, which affects both the allocation and pricing, is defined in terms of the mid-level auction. The second phase affects only the tentative allocation, and involves resolving a single instance of a suitably designed house allocation problem to exchange items within a certain subset of the tentatively allocated agents.

The design of the first phase of our sealed-bid unit-demand auction constitutes a key technical component of this dissertation and the associated analysis is quite involved. For example, it is a major challenge to prove that the first phase of our sealed-bid auction is truthful. A second major challenge is to justify the correctness of our fast implementation. In order to overcome these challenges, we carry out a layered analysis of our sealed-bid auction. We first present and analyze the top-level auction, which is a basic, slow variant of our sealed-bid auction. The top-level auction, like our sealed-bid auction, is defined in terms of the mid-level auction. The mid-level auction is a determinized, proxy version of the bottom-level auction. The bottom-level auction is dynamic with each round of the auction corresponding to an agent raising all of its offers by exactly one unit.

Our proposed sealed-bid unit-demand auction and our proposed dynamic unit-demand auction are presented in [23] and [22] respectively.

The remainder of this dissertation is organized as follows. In Chapter 2, we discuss some background material. In Chapter 3, we provide a foundation for the technical presentation to follow. In Chapter 4, we propose a novel solution concept for our sealed-bid unit-demand auction. In Chapters 5, 6, and 7, we present the bottom-level, mid-level, and top-level auctions respectively. In Chapter 8, we present our proposed sealed-bid unit-demand auction with put options, establish various properties related to truthfulness, efficiency, and bid privacy, and discuss a polynomial-time implementation. In Chapter 9, we present our proposed dynamic auction, establish various properties of the auction related to truthfulness, efficiency, and shill-resistance, and discuss a fast implementation. In Chapter 10 we present a scheme that discourages sniping by encouraging early bidding. In Chapter 11 we offer some concluding remarks and discuss extensions of our dynamic auction.

# Chapter 2

# Background

Game theory [12, 33] is the branch of economics that deals with the study of agents and their interactions in strategic settings. Game theory has numerous applications in a wide variety of fields including finance, computer science, and politics. For instance, in computer science, game theoretic models are used to solve problems in network congestion, routing, and resource sharing.

Mechanism design [26, 40] is that branch of game theory that is concerned with implementing a system-wide common goal in the presence of self-interested agents. In mechanism design, the rules of a game are designed such that agents are incentivized to not strategize but to play in accordance with their true preferences. In general, a mechanisms may not be computationally feasible. For instance, in a combinatorial auction with multiple items being sold concurrently, the range of possible outcomes is huge, and finding a welfare maximizing outcome can be NP-hard.

Algorithmic mechanism design is concerned with the tractability of mechanisms and the design of computationally feasible mechanisms. Algorithmic mechanism design techniques are routinely used in the design of combinatorial auctions [27, 44]. An auction for a set of items can be viewed as a game in which bidding agents strategize to win their favorite items at the lowest prices. Each bidding agent in the auction has some private value for the items which may not be equal to the public bid that the agent submits to the auction. A common goal of auction design is to compute a value maximizing outcome; such an outcome corresponds to an allocation of items to agents who value them the most.

Below we discuss some common mechanism design terms and definitions that are used in this dissertation.

## 2.1 Basic Definitions

Let $U = \cup_{0 \leq i \leq N} u_i$ be the set of agents in a game, and let $\mathcal{O}$ be the set of outcomes associated with the game. For any agent $u_i$ in $U$ and any outcome $o$ in $\mathcal{O}$, agent $u_i$'s preference for outcome $o$ is given by a *utility* function *utility*$(o, u_i)$. For any pair of outcomes $o$ and $o'$ in $\mathcal{O}$, agent $u_i$ prefers outcome $o$ over outcome $o'$ if *utility*$(o, u_i) >$ *utility*$(o', u_i)$.

A *strategy* of an agent $u_i$ in $U$ is a complete specification of the action taken by agent $u_i$ at every point in the game. For any agent $u_i$ in $U$, the set $\Sigma_i$ represents the set of all strategies that are available to agent $u_i$. A *strategy profile* $s$ of the agents in $U$ is a joint strategy $(s_0, \ldots, s_N)$ where $s_i$ is a strategy in the set $\Sigma_i$. For any strategy profile $s = (s_0, \ldots, s_N)$, we define $s_{-i}$ as the strategy profile $(s_0, \ldots, s_{i-1}, s_{i+1}, \ldots, s_N)$ representing the joint strategy of all agents except the agent $u_i$.

A mechanism defines a mapping from the set of strategy profiles $\Sigma_0 \times \ldots \times \Sigma_N$ to the set of outcomes $\mathcal{O}$. Recall that we defined the utility of an agent in a game as a function of the outcome of the game. Alternately, we can define the utility of an agent in a game as a function of the strategy profile that determines the outcome. Thus, for any agent $u_i$ in $U$, and any strategy profile $s$ of the game, we define $\mathit{utility}(s, u_i)$ to be equal to $\mathit{utility}(o, u_i)$, where $o$ is the outcome of the mechanism with strategy profile $s$.

## 2.2 Solution Concept

A *solution concept* of a game is a formal prediction of the strategies that will be played by each agent in the game. Thus, a solution concept predicts the final outcome of a game. Equilibrium solution concepts are the most commonly studied solution concepts in mechanism design. An equilibrium solution concept predicts the strategies that will be adopted by agents in equilibrium. Some well known equilibrium solution concepts are discussed below. For the auctions described in this dissertation, our goal is to find dominant strategy equilibria (see Section 2.2.3 below).

### 2.2.1 Nash equilibrium

One of the most popular solution concepts is that of a Nash equilibrium [32]. A strategy profile is said to be a Nash equilibrium of a game if each agent maximizes his utility by playing his equilibrium strategy, given that every other agent in the game is also playing his equilibrium strategy.

Let $U$ represent the set of agents in a game. A strategy profile $s = (s_0, \ldots, s_i, \ldots, s_{|U|}) \in \prod_{0 \leq i \leq |U|} \Sigma_i$ is a Nash equilibrium of the game if for any agent $u_i$ in $U$, and any strategy profile $s' = (s_0, \ldots, s'_i, \ldots, s_{|U|})$ we have

$$\mathit{utility}(s, u_i) \geq \mathit{utility}(s', u_i).$$

Finding a Nash equilibrium is associated with making strong assumptions on the common knowledge shared amongst agents in the game — each agent is assumed to know the Nash equilibrium strategies of every other agent in the game. It was shown by Nash that every finite game is associated with at least one Nash equilibrium. A game that is associated with more than one Nash equilibrium runs the risk of agents choosing different equilibria. In a game with multple Nash equilibria, agents must co-ordinate in order to choose the same Nash equilibrium.

### 2.2.2 Bayesian-Nash equilibrium

In a Bayesian-Nash equilibrium, an agent is assumed to know the distribution of preferences for all other agents. With these assumptions on agent preferences, in a Bayesian-Nash equilibrium, an agent maximizes his expected utility by playing his equilibrium strategy. A Bayesian-Nash equilibrium makes fewer assumptions than a Nash equilibrium on the common knowledge shared amongst agents, but is less robust than the dominant strategy equilibrium described in the following section.

### 2.2.3 Dominant strategy equilibrium

In a dominant strategy equilibrium, each agent maximizes his utility by playing his dominant equilibrium strategy, irrespective of the strategies of the other agents.

Let $U$ be the set of agents in a game and let $s = (s_0, \ldots, s_i, \ldots, s_{|U|})$ be a strategy profile in $\prod_{0 \leq i \leq |U|} \Sigma_i$. We say $s_i$ is a dominant strategy for agent $u_i$ if for any strategy profile $s' = (s'_0, \ldots, s_i, \ldots, s'_{|U|})$ in $\prod_{0 \leq i \leq |U|} \Sigma_i$, we have

$$\mathit{utility}(s, u_i) \geq \mathit{utility}(s', u_i)$$

Unlike a Nash equilibrium, a dominant strategy equilibrium makes no assumptions regarding an agent's knowledge of the strategies of the other agents in a game. Thus, a dominant strategy equilibrium is a stronger solution concept and is more desirable in practice than a Nash equilibrium.

## 2.3 Quasi-Linear Mechanism

An agent is said to have *quasi-linear* preferences if its utility can be expressed as $v(o, u_i) - p(o, u_i)$, where $v(o, u_i)$ represents agent $u_i$'s valuation for outcome $o$, and $p(o, u_i)$ represents the payment made by the agent to the mechanism. Thus, a quasi-linear mechanism defines an outcome rule and a payment rule. A mechanism where agents have quasi-linear preferences is a quasi-linear mechanism.

## 2.4 Social Choice Function

As discussed earlier, mechanism design seeks to compute a common social choice in the presence of self-interested agents. By definition, a mechanism maps a strategy profile to an outcome. A *social choice* function maps the private preferences of agents to an outcome. We say that a mechanism implements a social choice if, in equilibrium (Nash equilibrium, dominant strategy equilibrium, etc), the mechanism computes the outcome corresponding to the social choice function. In general, designing a mechanism that computes a desired social choice is difficult because agents can choose to misrepresent their preferences. Thus, the goal of mechanism design is to design strategies and an outcome rule such that, despite agents being self-interested, the mechanism is successful in computing the desired social choice. Below we discuss some important properties of social choice functions.

### 2.4.1 Allocative efficiency

A social choice function is allocatively efficient if it produces a value maximizing outcome; i.e. an outcome with value

$$\max_{o \in \mathcal{O}} \sum_{u_i \in U} v(o, u_i)$$

A mechanism is allocatively efficient if it implements an efficient social choice function.

### 2.4.2 Pareto efficiency

A social choice function is Pareto efficient if its outcome is such that no other outcome is strictly preferred (in terms of utility) by at least one agent and weakly preferred by all other agents. Formally, a social choice function is Pareto efficient if there does not exists an agent $u_i$ and an outcome $o$ in the set of outcomes $\mathcal{O}$ such that $\mathit{utility}(o, u_i) > \mathit{utility}(o^*, u_i)$ and

$$\mathit{utility}(o, u_j) \geq \mathit{utility}(o^*, u_j)\ \forall u_j \in U \setminus u_i$$

where $o^*$ is the outcome associated with the social choice function. A mechanism is said to be Pareto efficient if it implements a Pareto efficient social choice function.

#### 2.4.2.1 Weak Pareto efficiency

A social choice function is weakly Pareto efficient if there does not exists an agent $u_i$ and an outcome $o$ in the set of outcomes $\mathcal{O}$ such that

$$\mathit{utility}(o, u_i) > \mathit{utility}(o^*, u_i)\ \forall u_i \in U$$

where $o^*$ is the outcome associated with the social choice function. A mechanism is said to be weakly Pareto efficient if it implements a weakly Pareto efficient social choice function. In other words, a mechanism is weakly Pareto efficient its outcome is such that no other outcome is strictly preferred by all agents.

#### 2.4.2.2 The core and the weak core

The *core* of a game is the set of all Pareto-efficient outcomes. Consider a game with a set of agents $U$ and a set of outcomes $\mathcal{O}$.

The *weak core* is the set of all outcomes $o^*$ in $\mathcal{O}$ such that for any subset of agents $U_0$ of $U$, there does not exists an outcome $o$ in $\mathcal{O}$ for which

$$utility(o, u_i) > utility(o^*, u_i) \forall u_i \in U_0$$

In other words, an outcome is in the weak core if no other outcome is strictly preferred by all members of any coalition.

### 2.4.3 Budget balance

A social choice function is budget balanced if there are no net transfers either out of or into the system; that is, if the social choice function computes an outcome $o$, then $\sum_{u_i \in U} p(o, u_i) = 0$. A social choice function is weakly budget balanced if there are no net transfers out of the system, i.e $\sum_{u_i \in U} p(o, u_i) \geq 0$. A mechanism is (weakly) budget balanced if it implements a (weakly) budget balanced social choice function.

We do not discuss the budget balance property in the following chapters of this dissertation. However, we note that all of the auctions presented in this dissertation are weakly budget balanced.

## 2.5 Direct-Revelation Mechanism

In a direct-revelation mechanism, agent strategies are limited to directly reporting their preferences to the mechanism. A single-item auction in which agents submit bids to win the item is an example of a direct-revelation mechanism. Note however, that in a direct-revelation mechanism, agents can choose to strategically misreport their preferences.

## 2.6 Incentive-Compatibility

A direct-revelation mechanism is incentive-compatible if the equilibrium strategy (Nash, Bayesian-Nash, dominant etc.) of each agent is to report their true preferences. A *strategy proof* or *truthful* mechanism is a mechanism that is incentive-compatible in dominant strategy equilibrium.

## 2.7 Revelation Principle

According to the revelation principle [14, 15, 30, 31], any mechanism that implements a particular social choice function under equilibrium has an equivalent incentive-compatible direct-revelation mechanism that implements the same social choice function.

The revelation principle has powerful implications in mechanism design — it is usually sufficient to restrict attention to the study of incentive-compatible direct-revelation mechanisms.

## 2.8 Voluntary Participation

A mechanism satisfies *voluntary participation* or *individual-rationality* if the expected utility of an agent from participating in the mechanism is at least as high as the agent's utility from not participating in the mechanism.

## 2.9 Vickrey-Clarke-Groves Mechanisms

The Vickrey-Clarke-Groves mechanisms, also referred to as the VCG mechanisms, are a general class of direct-revelation mechanisms proposed by Vickrey [41], Clarke [7], and Groves [16] for agents with quasi-linear preferences. The VCG mechanisms are the only direct-revelation mechanisms that are strategy-proof and efficient [15], and the only efficient, Bayesian Nash mechanisms [21, 43].

Consider a direct-revelation mechanism with a set of agents $U$. Let $s$ denote the strategy profile reported by the agents in $U$, and let $o$ be the outcome of the mechanism in response to strategy profile $s$. We know that the agents in $U$ have quasi-linear preferences; it follows that the utility of any agent $u_i$ in $U$ is of the following form

$$\mathit{utility}(o, u_i) = v(o, u_i) - p(o, u_i)$$

where $v(o, u_i)$ is the valuation of agent $u_i$ for outcome $o$, and $p(o, u_i)$ is the payment made by agent $u_i$ to the mechanism.

The VCG outcome is the outcome $o^*$ in $\mathcal{O}$ such that

$$\sum_{u_i \in U} v(o^*, u_i)$$

is maximized.

The VCG price paid by an agent $u_i$ is given by

$$p(o^*, u_i) = h_i(s_{-i}) - \sum_{,j \neq i, u_j \in U} v(o^*, u_j)$$

where $o^*$ is the VCG outcome, and $h_i$ is a function of the reported strategies of all agents except the agent $u_i$. The family of VCG mechanisms arise from the different choices of function $h_i$.

### 2.9.1 The Clarke pivot pricing rule

The Clarke pivot pricing rule is a specific VCG pricing rule that achieves individual rationality and weak budget-balance under quite general conditions. A VCG mechanism with the Clarke pivot pricing rule is referred to as the *pivotal* mechanism.

In a pivotal mechanism, the term $h_i(s_{-i})$ in the VCG pricing for an agent $u_i$ is given by

$$h_i(s_{-i}) = \sum_{u_j \in U \setminus u_i} v(o^*_{-i}, u_j)$$

where $o^*_{-i}$ is the VCG outcome when agent $u_i$ is removed from the mechanism. Informally, each agent in the Pivotal mechanism is charged the opportunity cost introduced by the agent to the remaining agents in the mechanism. exce

## 2.10 The Vickrey auction

The Vickrey auction [41] is a well-known sealed-bid single-item auction. The Vickrey auction is the special case of the pivotal mechanism for a single item. If $u_i$ and $u_j$ are agents with the highest and second highest bids of $b_i$ and $b_j$ respectively, then in the Pivotal mechanism, the item is sold to agent $u_i$ for a price equal to $b_j$. Since the price of the item is equal to the second highest bid, the Vickrey auction is also referred to as the the "second-price" auction.

# Chapter 3

# Preliminaries

In this chapter, we present definitions and notation that will be useful through the remainder of this dissertation. We introduce a number of basic types and their auxiliary functions, and present a number of associated preliminary lemmas. The results of Sections 3.5 and 3.6 follow from standard results in the literature [17, 42]; the purpose of including these sections is to provide a self-contained presentation.

## 3.1 Agents and Items

We refer to the bidders in our auction as agents. In order to break ties among agents, we identify each agent with a binary string. We define the maximum over an empty set of agents as the empty agent $\epsilon$.

An item $v$ in our auction is a pair where the first component is a binary string identifier, denoted $id(v)$, and the second component is an integer lower bound on the price of $v$, denoted $min(v)$. We allow the price of an item in our auction to be negative in order to support procurement-type auctions.

## 3.2 Bid-Graphs

A *bid-graph* encapsulates a set of items and a set of agents having unit-demand bids on the items.

Formally, a bid-graph is an edge-weighted complete bipartite graph $G = (U, V, w)$, where $U$ is a set of agents, $V$ is a set of items, $w$ is a function from the set $U \times V$ to the set of integers, and the following conditions are satisfied:

1. the cardinality of $U$ is at least the cardinality of $V$
2. for any agent $u$ in $U$, agent $u$ is nonempty
3. for any pair of distinct items $v$ and $v'$ in $V$, we have $id(v) \neq id(v')$.

For any bid-graph $(U, V, w)$, the function $w$ encodes the unit-demand bids of the agents in $U$. In general, an agent's unit-demand bid may not include an offer for every item in the bid-graph. In this dissertation, we assume that an agent's unit-demand bid includes an integer offer for every item in the big-graph, and we choose to represent the absence of an offer by a negative integer that is sufficiently large in magnitude.

For any set of items $V$, we define a (unit-demand) *bid* on $V$ as a function that maps each item in $V$ to an integer. In the definitions that follow, let $G = (U, V, w)$ be a bid-graph. We define $bids(G)$ as the set of all possible bids on the set $V$. For any agent $u$ in $U$, we define $bid(G, u)$ as the bid $\beta$ in $bids(G)$ such that $\beta(v) = w(u, v)$ for any item $v$ in $V$.

For any nonempty agent $u$ not in $U$, and any bid $\beta$ in $\mathit{bids}(G)$, we define $\mathit{add}(G, u, \beta)$ as the bid-graph $G' = (U + u, V, w')$ where $\mathit{bid}(G', u) = \beta$ and $\mathit{bid}(G', u') = \mathit{bid}(G, u')$ for any agent $u'$ in $U$. For any nonempty agent $u$ not in $U$, any item $v$ in $V$, and any integer $z$, we define $\mathit{add}(G, u, v, z)$ as $\mathit{add}(G, u, \beta)$, where $\beta$ is the bid in $\mathit{bids}(G)$ such that $\beta(v) = z$ and $\beta(v') = \mathit{min}(v') - 1$ for any item $v'$ in $V - v$.

For any any agent $u$ in $U$, and any integer $z$, we define $\mathit{shift}(G, u, z)$ as the bid-graph $(U, V, w')$ where $w'(u, v) = w(u, v) + z$ for any item $v$ in $V$, and $w'(u', v) = w(u', v)$ for any agent $u'$ in $U - u$ and any item $v$ in $V$.

For any any agent $u$ in $U$, and any bid $\beta$ in $\mathit{bids}(G)$, we define $\mathit{subst}(G, u, \beta)$ as the bid-graph $G' = (U, V, w')$ where $\mathit{bid}(G', u) = \beta$, and $\mathit{bid}(G', u') = \mathit{bid}(G, u')$ for any agent $u'$ in $U - u$.

## 3.3 Configurations

A *configuration* encapsulates a bid-graph along with an associated outcome (allocation and pricing of the items).

A configuration $\chi$ is a triple $(G, M, \Phi)$, where $G = (U, V, w)$ is a bid-graph, $M$ is a maximum cardinality matching (MCM) of $G$, and $\Phi$ is a potential function that maps each item $v$ in $V$ to an integer $\Phi(v)$ such that $\Phi(v) \geq \mathit{min}(v)$. In the definitions that follow, let $\chi = (G, M, \Phi)$ be a configuration where $G = (U, V, w)$.

The function $\mathit{agents}(\chi)$ is the set $U$ and the function $\mathit{items}(\chi)$ is the set $V$. We say $\chi$ is *efficient* if $M$ is a maximum weight MCM (MWMCM) of $G$.

We define $\mathit{matched}(\chi)$ as the subset of agents in $U$ that are matched in $M$, and we define $\mathit{unmatched}(\chi)$ as the set of agents in $U \setminus \mathit{matched}(\chi)$. For any item $v$ in $V$, we define $\mathit{match}(\chi, v)$ as the agent $u$ in $U$ such that the edge $(u, v)$ belongs to $M$. For any agent $u$ in $U$, we define $\mathit{gap}(\chi, u)$ as $w(u, v) - \Phi(v)$ if $\mathit{match}(\chi, v) = u$, and as zero otherwise.

We say that an agent $u$ in $U$ satisfies *voluntary participation* if $\mathit{gap}(\chi, u)$ is nonnegative, and that $u$ satisfies *envy-freedom* if $\mathit{gap}(\chi, u) \geq w(u, v) - \Phi(v)$ for all items $v$ in $V$. We say $\chi$ is *Walrasian* if every agent $u$ in $U$ satisfies voluntary participation and envy-freedom.

For any item $v$ in $V$, we define $\mathit{amount}(\chi, v)$ as $w(\mathit{match}(\chi, v), v)$, and we define $\mathit{amount}(\chi)$ as the function that maps each item $v$ in $V$ to $\mathit{amount}(\chi, v)$. We define $\mathit{positive}(\chi)$ as the set of agents $u$ in $U$ such that $\mathit{gap}(\chi, u) > 0$ and we define the set $\mathit{nonpositive}(\chi)$ as $U \setminus \mathit{positive}(\chi)$.

For any bid $\beta$ in $\mathit{bids}(G)$, we define $\mathit{max\text{-}gap}(\chi, \beta)$ as the maximum over all items $v$ in $V$, of $\beta(v) - \Phi(v)$. For any bid $\beta$ in $\mathit{bids}(G)$, we define $\mathit{pseudo\text{-}demand}(\chi, \beta)$ as the set of all items $v$ in $V$ such that $\beta(v) - \Phi(v) = \mathit{max\text{-}gap}(\chi, \beta)$.

For any bid $\beta$ in $\mathit{bids}(G)$, we define $\mathit{demand}(\chi, \beta)$ as $\mathit{pseudo\text{-}demand}(\chi, \beta)$ if $\mathit{max\text{-}gap}(\chi, \beta) \geq 0$, and as the empty set $\emptyset$ otherwise. For any item $v$ in $V$, we define $\mathit{bids}(\chi, v)$ as the set of all bids $\beta$ in $\mathit{bids}(G)$ such that $v$ belongs to *demand*$(\chi, \beta)$.

For any agent $u$ in $U$, we define *max-gap*$(\chi, u)$ as *max-gap*$(\chi, \mathit{bid}(G, u))$, we define *pseudo-demand*$(\chi, u)$ as *pseudo-demand*$(\chi, \mathit{bid}(G, u))$, and we define *demand*$(\chi, u)$ as *demand*$(\chi, \mathit{bid}(G, u))$.

For any nonempty agent $u$ not in $U$, and any bid $\beta$ in *bids*$(G)$, we define *add*$(\chi, u, \beta)$ as the configuration $(\mathit{add}(G, u, \beta), M, \Phi)$. Similarly, for any nonempty agent $u$ not in $U$, any item $v$ in $V$, and any integer $z$, we define *add*$(\chi, u, v, z)$ as the configuration $(\mathit{add}(G, u, v, z), M, \Phi)$.

For any agent $u$ in $U$ and any bid $\beta$ in *bids*$(G)$, we denote the configuration $(\mathit{subst}(G, u, \beta), M, \Phi)$ by *subst*$(\chi, u, \beta)$. For any agent $u$ in $U$ and any nonempty agent $u'$ not in $U$, we define *subst*$(\chi, u, u')$ as the configuration obtained from $\chi$ by replacing all occurrences of agent $u$ with agent $u'$.

For any agent $u$ in $U$ and any integer $z$, we define *shift*$(\chi, u, z)$ as the configuration *subst*$(\chi, u, \beta)$ where $\beta$ is the bid in *bids*$(G)$ such that $\beta(v) = w(u, v) + 1$ for any item $v$ in $V$.

We now characterize a suitable directed graph on $\chi$ and formulate a reachability condition on this directed graph. We define *digraph*$(\chi)$ as the directed graph $(U \cup V, A)$, where $A$ is the set of arcs that includes for each edge $(u, v)$ in $U \times V$ such that $w(u, v) \geq w(u, v')$ for every item $v'$ in $V - v$:

1. an arc $(v, u)$ if edge $(u, v)$ is in $M$
2. an arc $(u, v)$ if edge $(u, v)$ is not in $M$.

For any agent $u$ in *unmatched*$(\chi)$, we define *items*$(\chi, u)$ as the set of items $v$ in $V$ such that there exists a directed path from agent $u$ to item $v$ in *digraph*$(\chi)$. In the terminology of the Hungarian algorithm, the set *items*$(\chi, u)$ is the set of items reachable from agent $u$ in the Hungarian tree rooted at $u$.

## 3.4 Agent Colors

We identify special classes of configurations (e.g. Walrasian configurations) by adopting a suitable coloring scheme of the agents. Every agent in a configuration is colored white, gray, or black according to certain rules. Informally, a non-black agent satisfies voluntary participation and envy-freedom, and a white agent satisfies a certain tie-breaking convention described below.

For any configuration $\chi$, the color of any agent $u$ in *agents*$(\chi)$ is determined as follows. We first consider the case where agent $u$ belongs to *matched*$(\chi)$. In this case, let $v$ be the item such that *match*$(\chi, v) = u$. If $v$ does not belong to *pseudo-demand*$(\chi, u)$, then agent $u$ is black. If $v$ belongs to *demand*$(\chi, u)$, then agent $u$ is white. Otherwise, agent $u$ is gray. Next, we consider the case where agent $u$ belongs to *unmatched*$(\chi)$. In this case, if *max-gap*$(\chi, u) > 0$, then agent $u$ is black. If *max-gap*$(\chi, u) = 0$, and there exists some item $v$ in *items*$(\chi, u)$ such that either *match*$(\chi, v)$ is non-white, or *match*$(\chi, v) < u$ and *gap*$(\chi,$ *match*$(\chi, v)) = 0$, then agent $u$ is gray. Otherwise, agent $u$ is white.

We define *white*$(\chi)$ as the set of white agents in $\chi$. The sets *gray*$(\chi)$, *black*$(\chi)$, *nonblack*$(\chi)$, and *nonwhite*$(\chi)$ are defined similarly.

1. $u$ belongs to *white*$(\chi)$, or

2. $u$ belongs to *nonwhite*$(\chi)$ and for all items $v$ in *items*$(\chi)$, we have $\beta(v) < \Phi(v) - 1$, where $\beta = bid(G, u)$.

## 3.5 Walrasian Configurations

One of the most common notions of equilibrium in auctions is the *Walrasian* equilibrium. In Section 3.3, we defined a Walrasian configurations as a configuration in which every agent satisfies voluntary participation and envy-freedom. In this section, we formalize the notion of Walrasian configurations in terms of agent colors.

A configuration $\chi = (G, M, \Phi)$ is *Walrasian* if *matched*$(\chi) \subseteq$ *white*$(\chi)$ and *unmatched*$(\chi) \subseteq$ *nonblack*$(\chi)$. A bid-graph $G$ is Walrasian if it admits a Walrasian configuration of the form $(G, M, \Phi)$.

The following is a list of definitions and lemmas related to Walrasian configurations.

**Lemma 3.5.1.** *For any bid-graph $G'$ of the form* $\mathrm{add}(G, u, \beta)$*, if bid-graph $G$ is Walrasian then the bid-graph $G'$ is Walrasian.*

*Proof.* Since $G$ is Walrasian, there exists some Walrasian configuration of the form $(G, M, \Phi)$. It follows that $M$ is an MWMCM of $G$, and thus there exists some MWMCM $M'$ of $G'$. Koopmans and Beckmann [20] and Leonard [25] show the existence of prices satisfying the Walrasian properties in the unit-demand setting; thus, there exists some Walrasian configuration of the form $(G', M', \Phi')$. □

**Lemma 3.5.2.** *If $\chi = (G, M, \Phi)$ is a Walrasian configuration, then $M$ is an MWMCM of $G$.*

*Proof.* By definition, every agent in *matched*$(\chi)$ is white; thus for every item $v$ in *items*$(\chi)$, we have $w(\mathit{match}(\chi, v), v) \geq \Phi(v)$. Similarly, for every agent $u$ in *unmatched*$(\chi)$ and every item $v$ in *items*$(\chi)$, we have $w((u, v)) \leq \Phi(v)$. It follows that any MCM of $G$ with higher weight than $M$ must match the set of agents in *matched*$(\chi)$.

Suppose there exists an MCM $M'$ of $G$ that matches agents in *matched*$(\chi)$ and has higher weight than $M$. Since every agent in *matched*$(\chi)$ is white, it follows that $w((u, v)) - \Phi(v) \geq w((u, v')) - \Phi(v')$, where $v$ and $v'$ are the items matched to $u$ in $M$ and $M'$ respectively. Thus, $\sum_{(u,v) \in M} w((u, v)) - \Phi(v) \geq \sum_{(u',v) \in M'} w((u', v)) - \Phi(v)$. Thus, $M'$ cannot be of higher weight than $M$. □

[Definition] For any Walrasian bid-graph $G$, we define *potentials*$(G)$ as the set of all potential functions $\Phi$ such that there exists a Walrasian configuration of the form $(G, M, \Phi)$.

**Lemma 3.5.3.** *Let $(G, M, \Phi)$ be a Walrasian configuration where bid-graph $G = (U, V, w)$, and let $M^*$ be an MWMCM of $G$. Let $P$ be a path in the undirected graph $(U \cup V, M \oplus M^*)$. Let $A$ denote the set of edges of $P$ that* *are in $M$ and let $B$ denote the set of edges of $P$ that are in $M^*$. Then the configuration $(G, M - A + B, \Phi)$ is Walrasian.*

*Proof.* Let path $P$ be defined by the sequence of vertices $u_0, v_0, u_1, v_1, \cdots v_{k-1}, u_k$. We use the fact that $(G, M, \Phi)$ is Walrasian to derive the following equations.

$$w((u_k, v_{k-1})) \geq \Phi(v_{k-1}), \tag{3.1}$$

$$w((u_0, v_0)) \leq \Phi(v_0), \tag{3.2}$$

$$w((u_i, v_{i-1})) - \Phi(v_{i-1}) \geq w((u_i, v_i)) - \Phi(v_i) \tag{3.3}$$

for all $i$ such that $0 < i < k$. Rewriting inequalities 3.1 and 3.2 as $w((u_k, v_{k-1})) - \Phi(v_{k-1}) \geq 0$ and $0 \geq w((u_0, v_0)) - \Phi(v_0)$, respectively, and then adding the latter inequalities to those obtained from 3.3, we find that the potential terms all cancel out, and we are left with the inequality $\sum_{0<i\leq k} w((u_i, v_{i-1})) \geq \sum_{0\leq i<k} w((u_i, v_i))$. Since $M$ and $M^*$ are both MWMCMs, the above inequality is tight. It follows that all of the inequalities we summed in order to obtain the above inequality are also tight. In other words, we have

$$w((u_k, v_{k-1})) = \Phi(v_{k-1}), \tag{3.4}$$

$$w((u_0, v_0)) = \Phi(v_0), \tag{3.5}$$

and

$$w((u_i, v_{i-1})) - \Phi(v_{i-1}) = w((u_i, v_i)) - \Phi(v_i) \tag{3.6}$$

for all $i$ such that $0 < i < k$. Armed with the above equations, we are now ready to establish that $(G, M', \Phi)$ is Walrasian.

First we show that for all edges $e = (u, v)$ in $M' \setminus M$, $w(e) \geq \Phi(v)$. The latter claim follows immediately from Equations 3.5 and 3.6. Next we show that for all nodes $v$ such that $(u_k, v)$ belongs to $M \oplus M^*$, $w((u_k, v)) \leq \Phi(v)$. (Notice that $u_k$ is the only node that is matched under $M$ and unmatched under $M'$.) For $v = v_{k-1}$, the desired inequality holds tightly by Equation 3.4. For $v \neq v_{k-1}$, Walrasian property of $(G, M, \Phi)$ implies $w((u_k, v_{k-1})) - \Phi(v_{k-1}) \geq w((u_k, v)) - \Phi(v)$, so the desired inequality follows from Equation 3.4.

Finally, we show that for any edges $e = (u, v)$ and $e' = (u, v')$ such that $(u, v)$ belongs to $M'$, $w(e) - \Phi(v) \geq w(e') - \Phi(v')$. We consider three subcases. In the first subcase, assume that $u$ is not equal to one of the $u_i$'s, $0 \leq i \leq k$. In this subcase, the claim follows from the Walrasian property of $(G, M, \Phi)$. In the second subcase, assume that $u = u_0$, which is unmatched in $M$ and matched via edge $e = (u_0, v_0)$ in $M'$. Let edge $e' = (u_0, v')$ belong to $E$, where $v' \neq v_0$. Then Equation 3.5 implies that $w(e) - \Phi(v_0) = 0$, and the Walrasian property of $(G, M, \Phi)$ implies that $w(e') \leq \Phi(v')$; hence $w(e) - \Phi(v_0) \geq w(e') - \Phi(v')$, as required. In the third subcase, assume that $u = u_i$ for some $i$ such that $0 < i < k$. (We do not need to consider $u = u_k$ because $u_k$ is unmatched in $M'$.) Notice that $u_i$ is matched in $M'$ via edge $e = (u_i, v_i)$. Let edge $e' = (u_i, v')$ belong to $E$, where $v' \neq v_i$. If $v' = v_{i-1}$, the required inequality $w(e) - \Phi(v_i) \geq w(e') - \Phi(v')$ holds tightly by Equation 3.6. Otherwise, Equation 3.6 implies that $w(e) - \Phi(v_i) = w((u_i, v_{i-1})) - \Phi(v_{i-1})$, the Walrasian property of $(G, M, \Phi)$ implies that $w((u_i, v_{i-1})) - \Phi(v_{i-1}) \geq$

$w(e') - \Phi(v')$, and hence the required inequality $w(e) - \Phi(v_i) \geq w(e') - \Phi(v')$ holds. $\square$

**Lemma 3.5.4.** *Let $(G, M, \Phi)$ be a Walrasian configuration where bid-graph $G = (U, V, w)$, and let $M^*$ be an MWMCM of $G$. Let $C$ be a cycle in the undirected graph $(U \cup V, M \oplus M^*)$. Let $A$ denote the set of edges of $C$ that are in $M$ and let $B$ denote the set of edges of $C$ that are in $M^*$. Then the configuration $(G, M - A + B, \Phi)$ is Walrasian.*

*Proof.* The proof is similar to that of Lemma 3.5.3. $\square$

**Lemma 3.5.5.** *For any Walrasian bid-graph $G$, any MWMCM $M$ of $G$, and any potential function $\Phi$ in* potentials$(G)$*, the configuration $(G, M, \Phi)$ is Walrasian.*

*Proof.* Let $\mathcal{M}$ denote the set of all MWMCMs $M'$ of $G$ such that $(G, M', \Phi)$ is Walrasian. Since $\Phi$ is in *potentials*$(G)$, the set $\mathcal{M}$ is guaranteed to be nonempty. Fix an MWMCM $M^*$ in $\mathcal{M}$ such that $|M \setminus M^*|$ is minimized. It is sufficient to prove that $M = M^*$. We prove this by contradiction. Assume that $|M \setminus M^*|$ is equal to some positive integer $k$. Consider the undirected graph $G' = (U \cup V, M \oplus M^*)$. Since no vertex in $G'$ has degree greater than 2, it can be partitioned into isolated vertices, simple paths of positive length, and simple cycles of positive length. Since $k > 0$, we are assured that $G'$ contains either a simple path of positive length or a simple cycle of positive length. We consider these two cases separately.

- Graph $G'$ contains a simple path $P$ of positive length

  Let $A$ and $B$ denote the sets of edges of $P$ that are in $M^*$ and $M$ respectively. By Lemma3.5.3, the configuration $(G, M', \Phi)$ is Walrasian, where $M' = M^* - A + B$. Furthermore, $|M \setminus M'| = k - |A| < k$, contradicting the definition of $M^*$.

- Graph $G'$ contains a simple cycle $C$ of positive length

  Let $A$ and $B$ denote the sets of edges of $P$ that are in $M^*$ and $M$ respectively. By Lemma 3.5.4, the configuration $(G, M', \Phi)$ is Walrasian, where $M' = M^* - A + B$. Furthermore, $|M \setminus M'| = k - |A| < k$, contradicting the definition of $M^*$.

$\square$

**Lemma 3.5.6.** *For any Walrasian bid-graph $G$, the functions in* potentials$(G)$ *form a lattice with meet and join operations given by pointwise minimum and maximum, respectively.*

*Proof.* Let $G = (U, V, w)$ and let $\Phi_0$ and $\Phi_1$ be potential functions in *potentials*$(G)$. Let $\Phi$ and $\Phi'$ be potential functions such that for any item $v$ in $V$, $\Phi(v) = \min(\Phi_0(v), \Phi_1(v))$ and $\Phi'(v) = \max(\Phi_0(v), \Phi_1(v))$. We are required to show that configurations $(G, M, \Phi)$ and $(G, M, \Phi')$ are Walrasian. In what follows, we will show that $(G, M, \Phi)$ is Walrasian. By a similar argument, it follows that $(G, M, \Phi')$ is also Walrasian.

Observe that $(G, M, \Phi_0)$ and $(G, M, \Phi_1)$ are Walrasian; thus, for any edge $(u, v)$ in $M$, we have $w((u, v)) \geq \Phi_0(v)$ and $w((u, v)) \geq \Phi_1(v)$; thus $w((u, v)) \geq \Phi(v)$. Similarly, for any agent $u$ unmatched in $M$ and any item $v$ in $V$, we have $w((u, v)) \leq \Phi_0(v)$ and $w((u, v)) \leq \Phi_1(v)$; thus $w((u, v)) \leq \Phi(v)$. It now remains to be shown the following condition: for any edge $(u, v)$ in $M$ and any item $v'$ in $V - v$, we have $w((u, v)) - \Phi(v) \geq w((u, v')) - \Phi(v')$. We accomplish this by showing that the condition holds when $\Phi(v) = \Phi_0(v)$. It follows by symmetry that the condition also holds when $\Phi(v) = \Phi_1(v)$. Fix an arbitrary item $v'$ in $V$, and consider the following two cases.

- $\Phi(v') = \Phi_0(v')$.

  Since $(G, M, \Phi_0)$ is Walrasian, we have $w((u, v)) - \Phi_0(v) \geq w((u, v')) - \Phi_0(v')$; using $\Phi(v) = \Phi_0(v)$ and $\Phi(v') = \Phi_0(v')$, we obtain the desired inequality $w((u, v)) - \Phi(v) \geq w((u, v')) - \Phi(v')$.

- $\Phi(v') = \Phi_1(v')$

  By the Walrasian property of configuration $(G, M, \Phi_1)$, we have $w((u, v)) - \Phi_1(v) \geq w((u, v')) - \Phi_1(v')$. Since $\Phi(v) = \Phi_0(v)$, we have $\Phi_0(v) \leq \Phi_1(v)$. Thus $w((u, v)) - \Phi_0(v) \geq w((u, v')) - \Phi_1(v')$; using $\Phi(v) = \Phi_0(v)$ and $\Phi(v') = \Phi_1(v')$, we obtain the desired inequality $w((u, v)) - \Phi(v) \geq w((u, v')) - \Phi(v')$.

$\square$

[Definition] For any Walrasian bid-graph $G$, we define *max-potential*$(G)$ as the maximum function in *potentials*$(G)$, and we define *min-potential*$(G)$ as the maximum function in *potentials*$(G)$; the existence of these functions is guaranteed by Lemma 3.5.6.

**Lemma 3.5.7.** *For any bid-graph $G'$ of the form* $\mathrm{add}(G, u, v, z)$ *where bid-graph $G$ is Walrasian, there exists a unique integer $z_0$ such that the following conditions hold:*

- *If $z > z_0$ and configuration $\chi = (G', M, \Phi)$ is Walrasian, then agent $u$ belongs to* $\mathrm{matched}(\chi)$.
- *If $z < z_0$ and configuration $\chi = (G', M, \Phi)$ is Walrasian, then agent $u$ belongs to* $\mathrm{unmatched}(\chi)$.
- *If $z = z_0$, then there exist Walrasian configurations $\chi = (G', M, \Phi)$ and $\chi' = (G', M', \Phi)$ such that agent $u$ belongs to* $\mathrm{matched}(\chi) \cap \mathrm{unmatched}(\chi')$.

*Proof.* Let $\Phi = \textit{max-potential}(G)$ and let $M$ be some MWMCM of $G$. Since $G$ is Walrasian, $(G, M, \Phi)$ is Walrasian. Thus the weight of any MWMCM of $G$ is at least equal to $\sum_{v \in G} \Phi(v)$. By Lemma 3.5.1, $G'$ is Walrasian and by Lemma 3.5.5, any configuration of the form $(G', M', \Phi')$ is Walrasian, where $M'$ is an MWMCM of $G'$ and $\Phi'$ is in *potentials*$(G')$. It is easy to see that when $z < \Phi(v)$, $(G', M, \Phi)$ is Walrasian and thus $u$ is unmatched in every Walrasian configuration of $G'$. We now consider the case when $z = \Phi(v)$. There exists some item $v'$ in *items*$(\chi, u)$ such that *gap*$(\chi, \textit{match}(\chi, v')) = 0$ as otherwise $\Phi(v'')$ can be incremented for each item $v''$ in *items*$(\chi, u)$. In this case, by Lemma 3.5.3, $u$ is matched in some Walrasian configuration of $G'$. Further, it is easy to see that if $u$ is matched in some Walrasian configuration $\chi$ of $G'$, then $u$ is matched in every Walrasian configuration of *shift*$(\chi, u, 1)$. Thus, there exists a unique $z_0 = \textit{max-potential}(v)$ with the desired property. $\square$

[Definition] For any Walrasian bid-graph $G = (U, V, w)$ and any item $v$ in $V$, we define *threshold*$(G, v)$ as the unique integer $z_0$ of Lemma 3.5.7, and we define *threshold*$(G)$ as the function that maps each item $v$ in $V$ to *threshold*$(G, v)$.

**Lemma 3.5.8.** *For any Walrasian bid-graph $G$, we have* threshold$(G)$ = max-potential$(G)$.

*Proof.* In the proof of Lemma 3.5.7, we showed that for any item $v$ in $G$, *threshold*$(G, v) = \textit{max-potential}(v)$. $\square$

For any Walrasian bid-graph $G = (U, V, w)$, we define *price*$(G)$ as *min-potential*$(G)$, and for any item $v$ in $V$, we define *price*$(G, v)$ as $\Phi(v)$, where $\Phi$ is equal to *min-potential*$(G)$.

**Lemma 3.5.9.** *For any bid-graph $G'$ of the form* add$(G, u, \beta)$ *where bid-graph $G = (U, V, w)$ is Walrasian, if $\beta(v) \leq$* price$(G, v)$ *for every item $v$ in $V$, then* price$(G')$ = price$(G)$.

*Proof.* Let $M$ be an MWMCM of $G$. By Lemma 3.5.5, $\chi = (G, M, \textit{price}(G))$ is Walrasian and thus the weight of $M$ is at least $\sum_{v \in G} \textit{price}(G, v)$. Since $\beta(v) \leq \textit{price}(G, v)$ for every item $v$ in $V$, $M$ is an MWMCM of $G'$. It follows that $(G', M, \textit{price}(G))$ is a Walrasian configuration. Further, if there exists a potential function $\Phi$ in *potentials*$(G')$ such that $\Phi < \textit{price}(G)$, then $(G, M, \Phi)$ is Walrasian; this contradicts the definition of *price*$(G)$. Thus, *price*$(G') =$ *price*$(G)$. $\square$

**Lemma 3.5.10.** *For any Walrasian configuration $\chi = (G, M, \Phi)$, we have* $\text{price}(G) \leq \text{threshold}(G) \leq \text{amount}(\chi)$.

*Proof.* By definition, *price*$(G) =$ *min-potential*$(G)$, and by Lemma 3.5.8, we have *threshold*$(G) =$ *max-potential*$(G)$; thus *price*$(G) \leq$ *threshold*$(G)$. Since *threshold*$(G) =$ *max-potential*$(G)$, it follows from Lemma 3.5.5 that $(G, M, \textit{threshold}(G))$ is Walrasian. By the definition of Walrasian configurations, it follows that *amount*$(\chi) \geq$ *threshold*$(G)$. $\square$

**Lemma 3.5.11.** *Let $G'$ be a bid-graph of the form* $\text{add}(G, u, \beta)$ *where bid-graph $G = (U, V, w)$ is Walrasian. Let $\Delta$ denote the maximum over all items $v$ in $V$, of $\beta(v) - \text{threshold}(G, v)$, and let $V'$ denote the set of all items $v$ in $V$ such that $\beta(v) - \text{threshold}(G, v) = \Delta$. Then the following conditions hold:*

- *If $\Delta > 0$ and configuration $\chi = (G', M, \Phi)$ is Walrasian, then* $\text{match}(\chi, v) = u$ *for some item $v$ in $V'$, and* $\text{price}(G', v) = \text{threshold}(G, v)$ *for for every item $v$ in $V'$.*
- *If $\Delta < 0$ and configuration $\chi = (G', M, \Phi)$ is Walrasian, then agent $u$ is unmatched in $M$.*
- *If $\Delta = 0$, then there exist Walrasian configurations $\chi = (G', M, \Phi)$ and $\chi' = (G', M', \Phi)$ such that agent $u$ belongs to* $\text{matched}(\chi) \cap \text{unmatched}(\chi')$.

- *If* $\Delta \leq 0$*, then* $\text{threshold}(G') = \text{threshold}(G)$.

*Proof.* Let $M$ be an MWMCM of $G$ and let $\Phi = \textit{max-potential}(G)$. By Lemma 3.5.5, $\chi_0 = (G, M, \Phi)$ is Walrasian; thus, by definition, the weight of $M$ is at least $\sum_{v \in G} \Phi(v)$. We consider the following cases:

- $\Delta < 0$

  In this case, $\beta(v) < \Phi(v)$ for each item $v$ in $G$, and thus, $M$ is an MWMCM of $G'$. Thus, $(G', M, \Phi)$ is Walrasian, and it follows by the definition of Walrasian configurations that $u$ is unmatched in $M$.

- $\Delta \leq 0$

  By the same argument as in the previous case, $(G', M, \Phi)$ is Walrasian. Further, if there exists a potential function $\Phi'$ in $\textit{potentials}(G')$ such that $\Phi' > \textit{max-potential}(G)$, then $(G, M, \Phi')$ is Walrasian; this contradicts the definition of $\textit{max-potential}(G)$. Thus, $\textit{max-potential}(G') = \textit{max-potential}(G)$. By Lemma 3.5.8, $\textit{max-potential}(G) = \textit{threshold}(G)$ and $\textit{max-potential}(G') = \textit{threshold}(G')$; thus $\textit{threshold}(G) = \textit{threshold}(G')$.

- $\Delta \geq 0$

  It is easy to see that there exists at least one item $v$ in the set $\textit{items}(\chi_0, u)$ for which $\textit{gap}(\chi_0, \textit{match}(\chi_0, v)) = 0$ as otherwise the potential associated with each item in $\textit{items}(\chi_0, u)$ can be incremented while $\chi$ remains Walrasian, violating the definition of $\textit{max-potential}(G)$. When

$\Delta \geq 0$, $v$ belongs to $\mathit{items}(\chi_0, u)$; thus there exists an augmenting path in $\mathit{digraph}(\chi_0)$. It follows that $u$ belongs to some MWMCM of $G'$.

When $\Delta = 0$, it is easy to see that $(G', M, \Phi)$ is Walrasian; thus, $\Phi$ is in $\mathit{potentials}(G')$. By Lemma 3.5.5, $(G', M, \mathit{price}(G'))$ is Walrasian. However, we know that $u$ is not matched in $M$. It follows that $\mathit{price}(G', v) \geq \Phi(v)$ for each item $v$ in $V'$ in order to satisfy the Walrasian property of $(G', M, \mathit{price}(G'))$. Thus $\mathit{price}(G', v) = \Phi(v) = \mathit{threshold}(G, v)$ for each item $v$ in $V'$. Above we showed that when $\Delta = 0$, $u$ is matched in some MWMCM $M'$ of $G'$; thus $(G', M', \mathit{price}(G'))$ is Walrasian. It is easy to see that when $\Delta > 0$, $(G', M', \mathit{price}(G'))$ remains Walrasian and thus, $\mathit{price}(G', v) = \mathit{threshold}(G, v)$ for each item $v$ in $V'$.

□

## 3.6 White Configurations

In designing our auction, we are interested in configurations, and particularly Walrasian configurations, that adhere to a specific tie-breaking scheme that ensures that a unique set of agents is allocated in the event of ties. We characterize as white configurations, all Walrasian configurations whose outcomes adhere to this tie-breaking scheme.

A configuration $\chi$ is *white* if $\mathit{agents}(\chi) = \mathit{white}(\chi)$.

The following is a set of definitions and lemmas related to white configurations. The proofs of these lemmas are similar to those for the corresponding results established in Section 3.5 for Walrasian configurations.

**Lemma 3.6.1.** *For any Walrasian bid-graph $G$, there exists a white configuration of the form $(G, M, \Phi)$, and for any white configuration of the form $(G, M, \Phi)$, the bid-graph $G$ is Walrasian.*

*Proof.* By definition, every white configuration is Walrasian. Thus, for any white configuration of the form $(G, M, \Phi)$, the bid-graph $G$ is Walrasian. Consider any Walrasian configuration $\chi = (G, M, \Phi)$ that is not white. Then there exists at least one agent $u$ in *unmatched*$(\chi)$ such that for some item $v$ in *items*$(\chi, u)$, *gap*$(\chi,$ *match*$(\chi, v)) = 0$ and *match*$(\chi, v) < u$. By repeated application of Lemma 3.5.3, $\chi$ can be transformed to a white configuration. □

**Lemma 3.6.2.** *For any Walrasian bid-graph $G$ and any potential function $\Phi$ in* potentials$(G)$*, there exists a white configuration of the form $(G, M, \Phi)$.*

*Proof.* By Lemma 3.5.5, any configuration $\chi = (G, M', \Phi)$ is Walrasian where $M'$ is an MWMCM of $G$. Then there exists at least one agent $u$ in *unmatched*$(\chi)$ such that *gap*$(\chi,$ *match*$(\chi, v)) = 0$ and *match*$(\chi, v) < u$ for some item $v$ in *items*$(\chi, u)$. By repeated application of Lemma 3.5.3, $\chi$ can be transformed to a white configuration. Thus, there exists some MWMCM $M$ of $G$ such that $(G, M, \Phi)$ is white. □

**Lemma 3.6.3.** *For any Walrasian bid-graph $G$ and any pair of white configurations $\chi = (G, M, \Phi)$ and $\chi' = (G, M', \Phi')$, we have* matched$(\chi) =$ matched$(\chi')$.

*Proof.* Let $\chi_0 = (G, M, \textit{max-potential}(G))$ and let $\chi_1 = (G, M', \textit{max-potential}(G))$. By Lemma 3.5.5, $\chi_0$ and $\chi_1$ are Walrasian. Further, since $\textit{max-potential}(G) \geq \Phi$ and $\textit{max-potential}(G) \geq \Phi'$, it follows that $\textit{white}(\chi) \cap \textit{unmatched}(\chi) = \textit{white}(\chi_0) \cap \textit{unmatched}(\chi_0)$ and $\textit{white}(\chi') \cap \textit{unmatched}(\chi') = \textit{white}(\chi_1) \cap \textit{unmatched}(\chi_1)$; thus $\chi_0$ and $\chi_1$ are white configurations. If $M \oplus M'$ consists of only cycles and no paths, then it follows that $\textit{matched}(\chi) = \textit{matched}(\chi')$. Suppose there exists some path $P$ in $M \oplus M'$ with endpoints $u$ in $\textit{matched}(\chi_0) \cap \textit{unmatched}(\chi_1)$ and $u'$ in $\textit{matched}(\chi_1) \cap \textit{unmatched}(\chi_0)$. By Lemma 3.5.3, $u$ belongs to $\textit{agents}(\chi_0, u')$ and $u'$ belongs to $\textit{agents}(\chi_1, u)$. Since $u < u'$ or $u' < u$, this violates the assumption that $\chi_0$ and $\chi_1$ are both white. Thus, there is no such path $P$ and $\textit{matched}(\chi) = \textit{matched}(\chi')$. □

[Definition] By Lemmas 3.6.1 and 3.6.3, we can conclude that for any Walrasian bid-graph $G$, there exists a unique set of matched agents in any white configuration of the form $(G, M, \Phi)$. We denote this unique set of matched agents by $\textit{matched}(G)$.

**Lemma 3.6.4.** *For any white configuration* $(G, M, \Phi)$*, and for any potential function* $\Phi'$ *in* potentials$(G)$*, the configuration* $(G, M, \Phi')$ *is white.*

*Proof.* By Lemma 3.6.2, there exists an MWMCM $M'$ of $G$ such that the configuration $\chi' = (G, M', \Phi')$ is white. Let $\chi = (G, M, \Phi)$ and let $\chi'' = (G, M, \Phi')$. By Lemma 3.5.5, $\chi''$ is Walrasian and by Lemma 3.6.3, $\textit{matched}(\chi) = \textit{matched}(\chi')$. Thus, $\textit{matched}(\chi) = \textit{matched}(\chi'')$ and $\textit{matched}(\chi'') \subseteq \textit{white}(\chi'')$. Since $\chi''$ and $\chi$ are Walrasian, for any agent $u$ in $\textit{unmatched}(\chi'')$, we have

$agents(\chi'', u) = agents(\chi, u)$; since $u$ is white in $\chi$, it follows that $u$ is white in $\chi''$. $\square$

In what follows, we sometimes compare amount-agent pairs. Such comparisons are resolved lexicographically.

**Lemma 3.6.5.** *For any Walrasian bid-graph $G = (U, V, w)$, any item $v$ in $V$, and any white configurations $\chi = (G, M, \Phi)$ and $\chi' = (G, M', \Phi)$, we have* $\text{agents}(\chi, v) = \text{agents}(\chi', v)$.

*Proof.* By Lemma 3.6.3, we have $matched(\chi) = matched(\chi')$; thus $unmatched(\chi) = unmatched(\chi')$. Since $\chi$ and $\chi'$ are Walrasian, for any item $v$, $gap(\chi, match(\chi, v)) = \max_{v \in V} w(match(\chi, v), v) - \Phi(v)$. Further, by the definition of $digraph(\chi)$, arc $(match(\chi, v), v')$ belongs to $digraph(\chi)$ for every item $v'$ in $demand(\chi, match(\chi, v))$. By a similar argument, arc $(match(\chi', v), v')$ belongs to $digraph(\chi')$ for every item $v'$ in $demand(\chi', match(\chi', v))$. Thus it follows that $agents(\chi, v) = agents(\chi', v)$. $\square$

[Definition] For any Walrasian bid-graph $G = (U, V, w)$, any potential function $\Phi$ in $potentials(G)$, and any item $v$ in $V$, we define $agents(G, \Phi, v)$ as the unique set $agents(\chi, v)$ of Lemma 3.6.5, where $\chi = (G, M, \Phi)$ is a white configuration whose existence is guaranteed by Lemma 3.6.2. For any Walrasian bid-graph $G = (U, V, w)$, and any item $v$ in $V$, we define $agents(G, v)$ as $agents(G, price(G), v)$.

[Definition] For any Walrasian bid-graph $G = (U, V, w)$, and any item $v$ in $V$, we define $price^*(G, v)$ as $(price(G), u_0)$, where $u_0$ is the maximum agent in *agents*$(G, v)$. Recall that the maximum agent over an empty set is defined as $\epsilon$. In addition, we define *price*$^*(G)$ as the function that maps each item $v$ in $V$ to *price*$^*(G, v)$.

For the following lemmas, we view bids and prices as pairs — if $u$ has an offer of $z$ on item $v$, we view the offer as the pair $(z, u)$.

**Lemma 3.6.6.** *For any bid-graph $G'$ of the form* $\text{add}(G, u, v, z)$ *where bid-graph $G = (U, V, w)$ is Walrasian, there exists a unique agent $u_0$ in $U$ such that agent $u$ belongs to* $\text{matched}(G')$ *if and only if* $(z, u) > (\text{threshold}(G, v), u_0)$.

*Proof.* The proof is similar to that of Lemma 3.5.9 when bids and prices are viewed as pairs. □

For any Walrasian bid-graph $G = (U, V, w)$ and any item $v$ in $V$, we define *threshold*$^*(G, v)$ as the unique pair $(\textit{threshold}(G, v), u_0)$ of Lemma 3.6.6.

**Lemma 3.6.7.** *For any bid-graph $G'$ of the form* $\text{add}(G, u, \beta)$ *where bid-graph $G$ is Walrasian, if the pair $(\beta(v), u) < \text{price}^*(G, v)$ for all items $v$ in $V$, then* $\text{price}^*(G') = \text{price}^*(G)$.

*Proof.* The proof is similar to that of Lemma 3.5.9 when bids and prices are viewed as pairs. □

For any configuration $\chi = (G, M, \Phi)$ where $G = (U, V, w)$, and any item $v$ in $V$, we define *amount*$^*(\chi, v)$ as the pair $(\textit{amount}(\chi, v), \textit{match}(\chi, v))$, and we define *amount*$^*(\chi)$ as the function that maps each item $v$ in $V$ to *amount*$^*(\chi, v)$.

**Lemma 3.6.8.** *For any white configuration* $\chi = (G, M, \Phi)$*, we have* $\text{price}^*(G) \leq \text{threshold}^*(G) \leq \text{amount}^*(\chi)$.

*Proof.* The proof is similar to that of Lemma 3.5.10 when bids and prices are viewed as pairs. □

**Lemma 3.6.9.** *Let* $G'$ *be a bid-graph of the form* $\text{add}(G, u, \beta)$ *where bid-graph* $G = (U, V, w)$ *is Walrasian. Let* $\Delta$ *denote the maximum, over all items* $v$ *in* $V$*, of* $\beta(v) - \text{threshold}(G, v)$*, and let* $V'$ *denote the set of all items* $v$ *in* $V$ *such that* $\beta(v) - \text{threshold}(G, v) = \Delta$*. Let* $u_0$ *denote the minimum, over all items* $v$ *in* $V'$*, of the second component of the pair* $\text{threshold}^*(G, v)$*. Then the following conditions hold:*

- *If the pair* $(\Delta, u) > (0, u_0)$ *and configuration* $\chi = (G', M, \Phi)$ *is white, then* $\text{match}(\chi, v) = u$ *for some item* $v$ *in* $V'$*, and* $\text{price}(G', v) = \text{threshold}(G, v)$ *for every item* $v$ *in* $V'$.
- *If the pair* $(\Delta, u) < (0, u_0)$ *and configuration* $\chi = (G', M, \Phi)$ *is white, then agent* $u$ *is unmatched in* $M$ *and* $\text{threshold}^*(G') = \text{threshold}^*(G)$.

*Proof.* The proof is similar to that of Lemma 3.5.11 when bids and prices are viewed as pairs. □

## 3.7 Quiescent configurations

The inputs and outputs of the bottom-level auction of Section 5 are *quiescent* configurations. A configuration $\chi = (G, M, \Phi)$ is quiescent if *unmatched*$(\chi)$

is a subset of $\mathit{white}(\chi)$, and for any agent $u$ in $\mathit{black}(\chi)$ where $\beta = \mathit{bid}(G, u)$, we have $\beta(v) < \Phi(v)$ for all items $v$ in $\mathit{items}(\chi)$.

For any configuration $\chi = (G, M, \Phi)$ where $G = (U, V, w)$, and any agent $u$ in $U$, we say $\chi$ is $u$-quiescent if either

1. $u$ belongs to $\mathit{unmatched}(\chi) \cap \mathit{gray}(\chi)$ and $(G', M, \Phi)$ is quiescent, where $G' = (U - u, V, w)$, or

2. $u$ belongs to $\mathit{matched}(\chi)$ and $\mathit{shift}(\chi, u, 1)$ is quiescent.

## 3.8 ECCs

We use tie-breaking to handle degeneracy in the bottom-level auction of Section 5; in each round, we break ties such that the set of allocated agents is uniquely determined. Below we identify equivalence classes of configurations (ECCs) that adhere to this tie breaking scheme.

For any pair of configurations $\chi = (G, M, \Phi)$ and $\chi' = (G, M', \Phi)$, we write $\chi \sim \chi'$ if $\mathit{matched}(\chi) = \mathit{matched}(\chi')$, $\mathit{nonwhite}(\chi) = \mathit{nonwhite}(\chi')$, and for any item $v$ in $\mathit{items}(\chi)$ such that $\mathit{match}(\chi, v)$ is non-white, we have $\mathit{match}(\chi, v) = \mathit{match}(\chi', v)$. Observe that $\sim$ is an equivalence relation and thus partitions the set of all configurations into equivalence classes. We refer to an equivalence class of configurations as an *ECC*, and we use the notation $[\chi]$ to refer to the ECC of a given configuration $\chi$.

By definition, for any ECC $X$, there exists a unique bid-graph $G_0$ and a unique potential function $\Phi_0$ such that every configuration in $X$ is of the form $(G_0, M, \Phi_0)$. We define *bid-graph*$(X)$ and *potential*$(X)$ as $G_0$ and $\Phi_0$ respectively. We define *potential*$(X, v)$ as $\Phi_0(v)$, for any item $v$ in $V$, where $G_0 = (U, V, w)$. An ECC $X$ is *quiescent* if every configuration $\chi$ in $X$ is quiescent. We define $u$-quiescent ECCs similarly.

It follows by definition that for any ECC $X$, every configuration $\chi$ in $X$ is associated with the same set of agents; we define *agents*$(X)$ to be this unique set of agents. We define the following similarly: *items*$(X)$, *matched*$(X)$, and *unmatched*$(X)$. For any ECC $X$ and any agent $u$ in *unmatched*$(X)$, we define *items*$(X, u)$ to be the unique set of items given by Lemma 3.8.1. We define the following functions similarly: *gray*$(X)$, *white*$(X)$, *black*$(X)$, *nonwhite*$(X)$, *nonblack*$(X)$, *enabled*$(X)$, *positive*$(X)$, *nonpositive*$(X)$, *gap*$(X, u)$, *bids*$(X, v)$, *max-gap*$(X, u)$, *demand*$(X, u)$, *pseudo-demand*$(X, u)$, *items*$(X, u)$, *agents*$(X, u)$, and *agents*$(X, v)$.

For any ECC $X$, any agent $u$ in *agents*$(X)$ and any integer $z$ such that either $u$ belongs to *unmatched*$(X)$ or $z \geq 0$, we define *shift*$(X, u, z)$ as $[$*shift*$(\chi, z, u)]$ where $\chi$ is any configuration in $X$. For any ECC $X$ and any agent $u$ in *agents*$(X)$ and any agent $u'$ not in *agents*$(X)$, we define *subst*$(X, u, u')$ as the ECC $\cup_{\chi \in X}[$*subst*$(\chi, u, u')]$ given by Lemma 3.8.3. We define the following similarly: *add*$(X, u, \beta)$, and *add*$(X, u, v, z)$. Below we establish some basic properties of ECCs.

**Lemma 3.8.1.** *For any ECC $X$, any agent $u$ in* unmatched$(X)$*, and any pair of configurations $\chi$ and $\chi'$ in $X$, we have* items$(\chi, u) = $ items$(\chi', u)$.

*Proof.* Configurations $\chi$ and $\chi'$ are associated with the same potential function. By definition, $\mathit{matched}(\chi) = \mathit{matched}(\chi')$, $\mathit{nonwhite}(\chi) = \mathit{nonwhite}(\chi')$, and for any agent $u''$ in $\mathit{matched}(\chi) \cap \mathit{nonwhite}(\chi)$, agent $u$ is matched to the same item in $\chi$ and $\chi'$. Thus, for any item $v$, $\mathit{gap}(\chi, \mathit{match}(\chi, v)) = \max_{v \in V} w(\mathit{match}(\chi, v), v) - \Phi(v)$. Further, by the definition of $\mathit{digraph}(\chi)$, arc $(\mathit{match}(\chi, v), v')$ belongs to $\mathit{digraph}(\chi)$ for every item $v'$ in $\mathit{demand}(\chi, \mathit{match}(\chi, v))$. By a similar argument, arc $(\mathit{match}(\chi', v), v')$ belongs to $\mathit{digraph}(\chi')$ for every item $v'$ in $\mathit{demand}(\chi', \mathit{match}(\chi', v))$. It follows that $\mathit{items}(\chi, u) = \mathit{items}(\chi', u)$. □

**Lemma 3.8.2.** *For any quiescent configuration $\chi$, the ECC $[\chi]$ is quiescent.*

*Proof.* Let $\chi = (G, M, \Phi)$ and let $\chi'$ be any configuration in $[\chi]$. By definition, $\mathit{potential}([\chi']) = \Phi$, $\mathit{unmatched}(\chi') = \mathit{unmatched}(\chi)$, and for every agent $u$ in $\mathit{matched}(\chi) \cap \mathit{nonwhite}(\chi)$, there exists an item $v$ in $\mathit{items}(\chi)$ such that $\mathit{match}(\chi, v) = \mathit{match}(\chi', v) = u$. It follows that since $\chi$ is quiescent, $\chi'$ is quiescent. □

**Lemma 3.8.3.** *For any ECC $X$, any agent $u$ in* agents$(X)$*, and any agent $u'$ not in* agents$(X)$*, the set of configurations given by $\cup_{\chi \in X}[\mathrm{subst}(\chi, u, u')]$ is an ECC.*

*Proof.* The proof follows from the definition of ECCs and the definition of $\mathit{subst}(\chi, u, u')$ for any configuration $\chi$ in $X$. □

# Chapter 4

# Solution Concept

In this chapter, we present a novel solution concept for our proposed sealed-bid unit-demand auction. Our sealed-bid unit-demand auction is used to implement each round of our proposed dynamic unit-demand auction; it follows that each round of our dynamic auction implements the solution concept described in this chapter.

Recall from Section 1.2 that for an item in our proposed sealed-bid unit-demand auction, the strike price of the item imposes a lower bound on the auction price of the item — by exercising the item's put, the seller of the item can ensure that the auction price is at least as high as the strike price. Due to these lower bound constraints on prices, we find that the VCG mechanism is not well-suited for our auction.

Additionally, in our proposed sealed-bid setting, we cannot guarantee the strong properties that are achieved by VCG in the classic setting. For example, consider an auction instance in which no agent bids on a particular item. The auction would be forced to allocate the item to the target of its put at its associated strike price even if such an allocation violates the envy-freedom property of the target. Consequently, we formulate a solution concept that is appropriate for our work.

The rest of this chapter is organized as follows. In Section 4.1, we present an informal discussion of the proposed solution concept for our sealed-bid unit-demand auction. In Section 4.2, we provide a formal definition of semi-Walrasian configurations. In Section 4.3, we formalize the equilibrium conditions associated with the proposed solution concept.

## 4.1 Informal Description

We say that an agent $u$ is "satisfied" in an outcome if $u$ satisfies the standard properties of voluntary participation and envy-freedom (see Section 3.3 for the formal definitions). For the classic sealed-bid unit-demand auction, a solution is said to be Walrasian if all of the agents are satisfied. Moreover, the VCG mechanism returns a Walrasian solution where the pricing is given by the unique minimum price vector over all Walrasian solutions.

For the present problem, we relax the Walrasian conditions by requiring only a certain subset of the agents in an outcome to be satisfied. For example, we enforce the natural requirement that if an agent $u$ is not allocated, then $u$ is satisfied; equivalently, each component of the unit-demand bid of $u$ is required to be less than or equal to the price of the corresponding item. Additionally, we require that if a non-allocated agent $u$ is indifferent to being allocated to an item $v$ that is allocated to some agent $u'$, then $u'$ is satisfied. Continuing in this manner, we require that if a non-allocated agent $u$ is indifferent to being allocated to an item $v$ that is allocated to agent $u'$, and $u'$ is indifferent to being allocated to an item $v'$ (not equal to $v$) that is allocated to agent $u''$, then $u''$ is satisfied, and so on.

In the terminology of the well-known Hungarian algorithm [24] for weighted bipartite matching, the aforementioned sequence of requirements may be stated more concisely as follows: If an agent $u$ belongs to the Hungarian tree rooted at some non-allocated agent, then $u$ is satisfied. (In Section 3.3, we formalize this requirement as a reachability condition in a suitably defined digraph.) The class of solutions meeting the latter requirement — which clearly includes all Walrasian solutions — plays a central role in our work. We refer to such solutions as "semi-Walrasian" (see Condition 1 in Section 4.3).

A key observation underlying the design of our solution concept is that a semi-Walrasian solution implicitly partitions the items into two sets: the set of all items $v$ such that any positive decrease in the price of $v$ (while leaving the prices of all other items unchanged) yields a solution that is no longer semi-Walrasian, and the remaining items. In Section 4.3, the former items are defined to be "priced at market", and the latter items are defined to be "priced above market". For an item that is priced at market, the associated put need not be exercised in order to justify the price. For such an item $v$, the price is required to be at least the strike price (see Condition 2 in Section 4.3); otherwise, the seller of item $v$ would prefer to exercise the put associated with $v$. For an item that is priced above market, the price can only be justified via exercise of the associated put; for such an item we require the price to be equal to the strike price (see Condition 3 in Section 4.3).

We require that the set of items $V'$ priced above market be purchased by the set of agents $U'$ who are targets of the associated puts (see Condition 4(a) in Section 4.3); the motivation for this requirement is that the items in $V'$ are too expensive to be of interest to any of the remaining agents. The problem of determining a suitable allocation of $V'$ to $U'$ may be viewed as an instance of the house allocation problem [37]; accordingly, we enforce standard desiderata related to voluntary participation (see Condition 4(b) in Section 4.3) and Pareto-efficiency (see Condition 5 in Section 4.3).

## 4.2 Semi-Walrasian Configurations

We now formally introduce *semi-Walrasian* configurations that we referred to during the discussion of the solution concept in Section 4.1. A configuration $\chi$ is *semi-Walrasian* if for every agent $u$ in $\mathit{unmatched}(\chi)$ and every item $v$ in $\mathit{items}(\chi, u)$, the agent $\mathit{match}(\chi, v)$ satisfies voluntary participation and envy-freedom.

A semi-Walrasian configuration $\chi$ induces a partition of the items into two sets: the set of items that belong to $\mathit{items}(\chi, u)$ for some agent $u$ in $\mathit{unmatched}(\chi)$, and the remaining items. We say that the items in the former set are *priced at market*, and that the remaining items are *priced above market*. For the standard sealed-bid unit-demand auction, the VCG mechanism yields a Walrasian configuration in which every item is priced at market.

## 4.3 Equilibrium Conditions

Given a configuration $\chi_0 = (G, M_0, \Phi_0)$ as input where $G = (U, V, w)$, we seek to devise a truthful mechanism that computes a configuration $\chi = (G, M, \Phi)$ satisfying the equilibrium conditions listed below.

1. The configuration $\chi$ is semi-Walrasian.
2. For any item $v$ in $V$ that is priced at market, we have $\Phi(v) \geq \Phi_0(v)$.
3. For any item $v$ in $V$ that is priced above market, we have $\Phi(v) = \Phi_0(v)$.
4. Let $V'$ denote the set of all items in $V$ that are priced above market. Then there is a permutation $\pi$ of $V'$ such that the following conditions hold.
   (a) For any item $v$ in $V'$, $\mathit{match}(\chi, \pi(v)) = \mathit{match}(\chi_0, v)$.
   (b) For any item $v$ in $V'$ having $\mathit{match}(\chi_0, v) = u$, $\mathit{gap}(\chi, u) \geq \mathit{gap}(\chi_0, u)$.
5. For any configuration $\chi' = (G, M', \Phi)$, if there exists an agent $u$ in $U$ such that $\mathit{gap}(\chi, u) < \mathit{gap}(\chi', u)$, then there exists an agent $u'$ in $U$ such that: (strong version) $\mathit{gap}(\chi', u') < \mathit{gap}(\chi, u')$; (weak version) $u'$ is matched differently in $M$ and $M'$, and $\mathit{gap}(\chi', u') \leq \mathit{gap}(\chi, u')$.

The reader will note that above conditions are stated in terms of an agent's *gap* rather than the utility. For a unit-demand auction where agents bid truthfully, the *gap* of an agent is equal to its utility, and (the weak version of) Condition 5 corresponds to a solution in the (weak) core. Our reference to the (weak) core is in the sense defined by Jaramillo and Manjunath [19]. Consequently, for a truthful auction, a solution in the core satisfies Pareto-efficiency, and a solution in the weak core satisfies the following property: no subset of agents can exchange their allocated items amongst themselves such that every agent in the subset experiences a strict improvement in utility.

# Chapter 5

# Bottom-Level Auction

In this chapter, we present the bottom-level auction. The bottom-level auction is a dynamic unit-demand auction, and is a building block of the mid-level auction described in Chapter 6.

The input to the bottom-level auction is a quiescent ECC (see Sections 3.7 and 3.8 for definition). It is easy to see that quiescent ECCs satisfy equilibrium conditions 1, 2, and 3 of Section 4.3. In each round of the bottom-level auction, a single agent increments its offers on all items by one unit, and the round is resolved by incorporating the bid increment while continuing to satisfy equilibrium conditions 1, 2, and 3.

The rest of this chapter is organized as follows. In Section 5.1, we introduce some preliminary definitions. In Section 5.2, we describe the bottom-level auction. In Section 5.3, we develop formalism leading to the definition of the *raise* operation. In Section 5.4, we establish some basic properties of the bottom-level auction. In Section 5.5, we establish commutativity of *raise* invocations.

## 5.1 Preliminaries

For any configuration $\chi$, we define *enabled*$(\chi)$ as the set of agents $u$ in *agents*$(\chi)$ such that either

1. $u$ belongs to *white*$(\chi)$, or
2. $u$ belongs to *nonwhite*$(\chi)$ and for all items $v$ in *items*$(\chi)$, we have $\beta(v) < \Phi(v) - 1$, where $\beta = bid(G, u)$.

We define *enabled*$(X)$ similarly.

## 5.2 Description

The bottom-level auction is dynamic. The auction takes a quiescent ECC as input and updates the ECC over a sequence of rounds. In a general round of the bottom-level auction, a single enabled agent in the ECC invokes the function *raise* defined in Section 5.3 below. Informally, an invocation of *raise* by an agent corresponds to the agent incrementing all components of its bid by one unit. If two or more enabled agents wish to invoke *raise* in a round, then the auction chooses from amongst them arbitrarily. The auction terminates when no agent invokes *raise* in a round.

## 5.3 The *Raise* Operation

In this section, we develop formalism leading to the definition of the function *raise*.

For any ECC $X$ and any agent $u$ in *unmatched*$(X)$, we define the predicate $P_0(X, u)$ to hold if $X$ is either quiescent or $u$-quiescent.

We now define *victim*$(X, u, z)$ for any ECC $X$, any integer $z$ in $\{0, 1\}$, and any agent $u$ in *unmatched*$(X)$ such that the predicate $P_0(X, u)$ holds. Let set $U_0$ denote *white*$(X)$ and let set $U_1$ denote *agents*$(X, u) \cup \{u\} \cap$ *nonpositive*$(X)$. Note that set $U_1$ is nonempty as it contains agent $u$. If $U_1 \setminus U_0 \neq \emptyset$, we define *victim*$(X, u, z)$ as the minimum agent in $U_1 \setminus U_0$. If $U_1 \setminus U_0 = \emptyset$, $z = 1$, and $U_1 - u \neq \emptyset$, then we define *victim*$(X, u, z)$ as the minimum agent in $U_1 - u$. Otherwise, we define *victim*$(X, u, z)$ as the minimum agent in $U_1$.

For any ECC $X$ and any agent $u$ in *enabled*$(X)$, we define the predicate $P_1(X, u)$ to hold if either

1. agent $u$ belongs to *matched*$(X)$ and $X$ is quiescent, or
2. agent $u$ belongs to *unmatched*$(X)$ and the predicate $P_0(X, u)$ holds.

We now define augment$(X, u, z)$ for any ECC $X$, any integer $z$ in $\{0, 1\}$, and any agent $u$ in *enabled*$(X)$ such that the predicate $P_1(X, u)$ holds. If agent $u$ belongs to *matched*$(X)$, then augment$(X, u, z)$ is the ECC $X$. Otherwise, augment$(X, u, z)$ is the ECC $[\chi']$, where $\chi'$ is constructed as follows: Let $\chi$ be an arbitrary configuration in $X$ and let $P$ be an arbitrary simple directed path from $u$ to *victim*$(X, u, z)$ in *digraph*$(\chi)$; for every item $v'$ such that there exists an arc of the form $(u', v')$ on path $P$, we set *match*$(\chi', v') = u'$, and for every item $v'$ that is not on path $P$, we set *match*$(\chi', v') =$ *match*$(\chi, v')$. By Lemma 5.3.1, it follows that augment$(X, u, z)$ is well defined.

For any ECC $X$ and any agent $u$ in $\mathit{enabled}(X)$ such that either

1. $X$ is quiescent and $\mathit{agents}(X,u) \cap \mathit{nonpositive}(X) = \emptyset$, or
2. $X$ is $u$-quiescent and $u$ belongs to $\mathit{matched}(X)$

we define $\mathrm{inc}(X,u)$ as

$$\cup_{(G,M,\Phi)\in X}[(G', M, \Phi')]$$

where $G' = \mathit{shift}(\mathit{bid\text{-}graph}(X), u, 1)$, and $\Phi'$ is defined as follows: if agent $u$ belongs to $\mathit{matched}(\chi)$, then $\Phi' = \Phi$; otherwise $\Phi'(v) = \Phi(v) + 1$ for any item $v$ in $\mathit{items}(\chi, u)$ and $\Phi'(v) = \Phi(v)$ for any item $v$ in $\mathit{items}(\chi) \setminus \mathit{items}(\chi, u)$. The existence of such an ECC is established by Lemma 5.3.2.

For any quiescent ECC $X$ and any agent $u$ in $\mathit{enabled}(X)$, we define $\mathit{raise}'(X,u)$ as $\mathrm{augment}(X, u, 1)$. For any ECC $X$ and any agent $u$ in $\mathit{enabled}(X)$ such that either $X$ is quiescent, or $X$ is $u$-quiescent and $u$ belongs to $\mathit{matched}(X)$, we define $\mathit{raise}''(X,u)$ as $\mathrm{augment}(\mathrm{inc}(X,u), u, 0)$ For any quiescent ECC $X$ and any agent $u$ in $\mathit{enabled}(X)$, the function $\mathit{raise}(X,u)$ is defined as $\mathit{raise}''(\mathit{raise}'(X,u), u)$.

For any quiescent ECC $X$ and any agent $u$ in $\mathit{unmatched}(X)$, we define $\mathit{victim}(X,u)$ as follows: if $\mathit{matched}(X) \cap \mathit{unmatched}(\mathit{raise}(X,u)) = \{u'\}$, then $\mathit{victim}(X,u) = u'$; otherwise, $\mathit{victim}(X,u) = \emptyset$. Recall that by Fact 5.3.4, $\mathit{matched}(X) \cap \mathit{unmatched}(X)$ has a cardinality of at most 1.

The facts below follow from the definition of the function *raise*.

**Fact 5.3.1.** *For any quiescent ECC $X$ and any agent $u$ in* $\text{enabled}(X) \cap \text{matched}(X)$*, we have* $\text{raise}(X, u) = \text{shift}(X, u, 1)$.

**Fact 5.3.2.** *For any quiescent ECC $X$ and any agent $u$ in* $\text{enabled}(X)$*, we have* $\text{potential}(\text{raise}(X, u)) \geq \text{potential}(X)$.

**Fact 5.3.3.** *For any quiescent ECC $X$ and any agent $u$ in* $\text{enabled}(X)$ *such that* $\text{bid}(\text{bid-graph}(X), u) < \text{potential}(X)$*, we have* $\text{potential}(\text{raise}(X, u)) = \text{potential}(X)$.

**Fact 5.3.4.** *For any ECC $X'$ of the form* $\text{raise}(X, u)$*, we have $|S| \leq 1$, where* $S = \text{matched}(X) \setminus \text{matched}(X')$.

The following lemmas establish that the output of the bottom-level auction is a quiescent ECC.

**Lemma 5.3.1.** *For any ECC $X$, any integer $z$ in $\{0, 1\}$, and any agent $u$ in* $\text{enabled}(X)$ *such that the predicate $P_1(X, u)$ holds, there is a unique ECC of the form $augment(X, u, z)$. $augment(X, u, z)$.*

*Proof.* If $u$ belongs to $matched(X)$, then by definition $\text{augment}(X, u, z) = X$. We now consider the case where $u$ belongs to $unmatched(X)$. Let $\chi$ be any configuration in $X$. By definition, irrespective of the choice of $\chi$ and the path $P$ used, the agent $victim(\chi, u, z)$ is unmatched in $\text{augment}(X, u, z)$ and each agent in $matched(\chi) \cap nonwhite(\chi) \setminus victim(\chi, u, z)$ is matched to the same item in $\chi$ and $\text{augment}(X, u, z)$. Thus it follows that $\text{augment}(X, u, z)$ is an ECC. □

**Lemma 5.3.2.** *Any set of configurations of the form inc*$(X, u)$ *is an ECC.*

*Proof.* If $u$ belongs to *matched*$(X)$, then by definition $\text{inc}(X, u) = \textit{shift}(X, u, 1)$. We now consider the case where $u$ belongs to *unmatched*$(X)$. By the preconditions on $X$ required by $\text{inc}(X, u)$, it follows that $X$ is quiescent and there exists no agent $u'$ in *agents*$(X, u)$ such that $\textit{gap}(X, u') \leq 0$. Let $(G, M, \Phi)$ be any configuration in $X$. By definition, $\text{inc}(X, u)$ includes the configuration $(G', M, \Phi')$ where $G' = \textit{shift}(\textit{bid-graph}(X), u, 1)$ and $\Phi'(v) = \Phi(v) + 1$ for each item $v$ in *items*$(\chi, u)$. Thus, every agent in *nonwhite*$(\chi)$ is matched to the same item in $\chi$ and $\chi'$. It follows that the set of configurations given by $\text{inc}(X, u)$ is an ECC. □

**Lemma 5.3.3.** *For any quiescent ECC $X$ and any agent $u$ in* enabled$(X)$*, the predicate $P_1(X, u)$ holds.*

*Proof.* Since $X$ is quiescent, by definition, $P_1(X, u)$ holds if $u$ belongs to *matched*$(X)$. Suppose $u$ belongs to *unmatched*$(X)$. Then, $P_1(X, u)$ holds if $P_0(X, u)$ holds. By definition, $P_0(X, u)$ holds when $X$ is quiescent. □

**Lemma 5.3.4.** *For any ECC $X'$ of the form inc$(X, u)$, the predicate $P_1(X', u)$ holds.*

*Proof.* By the preconditions of $\text{inc}(X, u)$, we know that either $\textit{agents}(X, u) \cap \textit{nonpositive}(X) = \emptyset$ and $X$ is quiescent, or $u$ belongs to *matched*$(X)$ and *shift*$(X, u, 1)$ is quiescent. We first consider the case when $u$ belongs to *matched*$(X)$. In this case, $\text{inc}(X,u) = \mathit{shift}(X,u,1)$. Thus $\text{inc}(X,u)$ is quiescent and $P_1(X',u)$ holds.

Next we consider the case when $u$ belongs to *unmatched*$(X)$. In this case, $\mathit{agents}(X,u) \cap \mathit{nonpositive}(X) = \emptyset$ and $X$ is quiescent. By definition, $\text{inc}(X,u)$ is an ECC $X'$ whose bid-graph $G = \mathit{shift}(\mathit{bid\text{-}graph}(X),u,1)$ and whose potential function has incremented $\mathit{potential}(X,v)$ by one for each item $v$ in $\mathit{items}(X,u)$. It is easy to see that $u$ is either white or gray in $\text{inc}(X,u)$; thus $\text{inc}(X,u)$ is either quiescent or $u$-quiescent. It follows that $P_1(\text{inc}(X,u),u)$ holds $\square$

**Lemma 5.3.5.** *For any quiescent ECC $X$ and any agent $u$ in* $\text{enabled}(X)$*, either* $\text{raise}'(X,u)$ *is quiescent, or* $\text{raise}'(X,u)$ *is $u$-quiescent and $u$ belongs to* $\text{matched}(\text{raise}'(X,u))$*.*

*Proof.* We first consider the case where $u$ belongs to *matched*$(X)$. In this case $\mathit{raise}'(X,u) = X$ and thus $\mathit{raise}'(X,u)$ is quiescent.

Next, we consider the case where $u$ belongs to *unmatched*$(X)$. Since $X$ is quiescent, $u$ belongs to *white*$(X)$. In this case, $\mathit{raise}'(X,u) = \text{augment}(X,u,1)$. Thus, either $u$ is unmatched and $u$ is white in $\mathit{raise}'(X,u)$, or $u$ belongs to $\mathit{matched}(\mathit{raise}'(X,u))$ and $u$ is gray in $\mathit{raise}'(X,u)$. Thus, $\mathit{raise}'(X,u)$ is either quiescent or $u$-quiescent. $\square$

**Lemma 5.3.6.** *Any ECC of the form* $\text{raise}(X,u)$ *is quiescent.*

*Proof.* By definition $\mathit{raise}(X,u) = \mathit{raise}''(\mathit{raise}'(X,u),u)$. By Lemma 5.3.5, $\mathit{raise}'(X,u)$ satisfies the preconditions of $\mathit{raise}''$. We consider the following two cases. First, we consider the case where $u$ belongs to $\textit{matched}(\textit{raise}'(X, u))$, In this case, by Lemma 5.3.5, $\textit{raise}'(X, u)$ is $u$-quiescent. Further, by definition, $\textit{raise}''$ for a matched agent results in incrementing the bid of the agent by one unit; thus, $\textit{raise}(X, u) = \textit{shift}(\textit{raise}'(X, u), u, 1)$ which is quiescent by definition.

Next we consider the case where $u$ belongs to $\textit{unmatched}(\textit{raise}'(X, u))$. In this case, by Lemma 5.3.5, $\textit{raise}'(X, u)$ is quiescent and thus $\textit{raise}'(X, u)$ satisfies the precondition for invoking *inc*. By Lemma 5.3.4, the predicate $P_1(\text{inc}(\textit{raise}'(X, u), u), u)$ holds; it follows that agent $u$ belongs to the set $\textit{matched}(\text{inc}(\textit{raise}'(X, u), u))$, or the ECC $\text{inc}(\textit{raise}'(X, u), u)$ is either quiescent of $u$-quiescent. In the case where $\text{inc}(\textit{raise}'(X, u), u)$ is $u$-quiescent, it is easy to see from the definition of *augment* that $\textit{raise}(X, u)$ is quiescent. For the remaining two cases, *augment* is a no-op. $\square$

## 5.4 Properties

In this section, we discuss some basic properties of the bottom-level auction that are useful in both proving lemmas of Section 5.5 and in establishing results in Chapter 6.

**Lemma 5.4.1.** *For any ECC $X'$ of the form* $\text{raise}(X, u')$ *and any agent $u$ in* $\text{nonwhite}(X)$*, either*

1. *$u$ belongs to* $\text{unmatched}(X')$*, or*

2. *$u$ belongs to* nonwhite$(X')$*, and there exists an item $v$ in* items$(X)$ *such that* potential$(X, v)$ *is equal to* potential$(X', v)$ *and* match$(\chi, v) = u$ *for any configuration $\chi$ in $X \cup X'$.*

*Proof.* Since $u$ belongs to *nonwhite*$(X)$, there exists an item $v$ in *items*$(X)$ such that for any configuration $\chi$ in $X$, we have *match*$(\chi, v) = u$. By definition, $u$ does not belong to *digraph*$(X)$ and $v$ is a leaf of *digraph*$(X)$. Since $v$ is a leaf of *digraph*$(X)$, by the definition of the function *raise*$'$ either implies that $u =$ *victim*$(X, u', 1)$ or *match*$(\chi, v) = u$ for any configuration $\chi$ in $X \cup$ *raise*$'(X, u')$. If $u =$ *victim*$(X, u', 1)$, then $u$ belongs to *unmatched*$(X')$ and the proof is complete.

We now consider the case where $u \neq$ *victim*$(X, u', 1)$; thus $v$ does not belong to *items*$(X, u')$. By the definition of the function *raise*$''$, we have *potential*$(X', v') =$ *potential*$(X, v') + 1$ for any item $v'$ in *items*$(X, u')$ and *potential*$(X', v') =$ *potential*$(X, v')$ for any item $v'$ not in *items*$(X, u')$; thus *potential*$(X', v) =$ *potential*$(X, v)$. Let $X'' = \text{inc}(\textit{raise}'(X, u'), u')$. It is easy to see that $v$ is a leaf of *digraph*$(X'')$. Thus, either $u =$ *victim*$(X'', u', 0)$ or *match*$(\chi, v) = u$ for any configuration $\chi$ in $X \cup X'$. □

**Lemma 5.4.2.** *For any quiescent ECC $X$ and any agent $u$ in* enabled$(X)$*, if $X' =$* raise$(X, u)$*, then*

$$\text{gray}(X) \subseteq \text{nonblack}(X') \wedge \text{white}(X) \subseteq \text{white}(X').$$

*Proof.* By the definition of the function *raise*, if $u$ belongs to *gray*$(X)$, then $u$ belongs to *gray*$(X')$, and if $u$ belongs to *white*$(X)$, then $u$ belongs to *white*$(X')$.

Consider any agent $u_0$ in $\mathit{agents}(X) - u$. By Lemma 5.3.6, $X'$ is quiescent, and by the definition of a quiescent ECC, $\mathit{unmatched}(X') \subseteq \mathit{white}(X')$. Thus, if $u_0$ belongs to $\mathit{unmatched}(X')$, then $u_0$ belongs to $\mathit{white}(X')$ and hence $u_0$ belongs to $\mathit{enabled}(X')$. Now suppose that $u_0$ belongs to $\mathit{matched}(X')$. We consider the following two cases.

First we consider the case where $u_0$ belongs to $\mathit{gray}(X) \cap \mathit{matched}(X')$. By Fact 5.3.2, we have $\mathit{potential}(X') \geq \mathit{potential}(X)$ and by Lemma 5.4.1, it follows that there exists an item $v_0$ in $\mathit{items}(X)$ having $\mathit{potential}(X, v_0) = \mathit{potential}(X', v_0)$ and for any configuration $\chi$ in $X \cup X'$, we have $\mathit{match}(\chi, v_0) = u_0$. It follows that $u_0$ belongs to $\mathit{gray}(X')$.

Next we consider the case where $u_0$ belongs to $\mathit{white}(X) \cap \mathit{matched}(X')$. By our assumption, $u_0$ belongs to $\mathit{matched}(X)$. By the definition of *raise*, it follows that $\mathit{gap}(X', u_0) \geq 0$. Thus, $u_0$ belongs to $\mathit{white}(X')$. $\square$

**Lemma 5.4.3.** *For any quiescent ECC $X$ and any agent $u$ in* $\text{enabled}(X)$*, we have* $\text{enabled}(X) - u \subseteq \text{enabled}(\text{raise}(X, u))$.

*Proof.* Let $X' = \mathit{raise}(X, u)$. By Lemma 5.3.6, $X'$ is quiescent. Consider any agent $u_0$ in $\mathit{enabled}(X) - u$. Suppose $u_0$ belongs to $\mathit{white}(X)$; then by Lemma 5.4.2, $u_0$ belongs to $\mathit{white}(X')$, and hence $u_0$ belongs to $\mathit{enabled}(X')$.

Suppose $u_0$ belongs to $\mathit{nonwhite}(X)$. Since $u_0$ belongs to $\mathit{enabled}(X)$, we have $\beta(v) < \mathit{potential}(X, v)$ for every item $v$ in $\mathit{items}(X)$, where $\beta = \mathit{bid}(X, u_0)$. By Fact 5.3.2, $\mathit{potential}(X') \geq \mathit{potential}(X)$ and by Lemma 5.4.1, either $u_0$ belongs to $\mathit{unmatched}(X')$ or there exists an item $v_0$ in $\mathit{items}(X)$

such that for any configuration $\chi$ in $X \cup X'$, we have $match(\chi, v_0) = u_0$. Thus, $u_0$ belongs to $enabled(X')$. $\square$

**Lemma 5.4.4.** *For any quiescent ECC $X$ and any agent $u$ in* $\mathrm{enabled}(X)$*, if there exists an item $v$ in* $\mathrm{items}(X)$ *such that* $\mathrm{potential}(X, v)$ *is equal to* $\mathrm{potential}(\mathrm{raise}(X, u), v)$*, then* $\mathrm{bids}(X, v) \subseteq \mathrm{bids}(\mathrm{raise}(X, u), v)$.

*Proof.* Let $\beta$ be any bid in in $bids(X, v)$. By definition, for any item $v'$ in $items(X) - v$, we have $\beta(v) - potential(X, v) \geq \beta(v') - potential(X, v')$. By Lemma 5.3.2, we have $potential(raise(X, u)) \geq potential(X)$. Thus, for any item $v'$ in $items(X) - v$, we have

$$\beta(v) - potential(X, v) \geq \beta(v') - potential(raise(X, u), v')$$

Thus, $\beta$ belongs to $bids(raise(X, u), v)$. $\square$

**Lemma 5.4.5.** *For any quiescent ECC $X_0$ and any quiescent ECC $X_1$ of the form* $\mathrm{subst}(X_0, u_0, u_1)$ *where $u_0$ belongs to* $\mathrm{unmatched}(X_0)$ *and $u_1 < u_0$, we have* $\mathrm{gap}(\mathrm{raise}(X_0, u_0), u_0) = \mathrm{gap}(\mathrm{raise}(X_1, u_1), u_1) = 0$*. Furthermore, either*

1. $\mathrm{raise}(X_1, u_1) = \mathrm{subst}(\mathrm{raise}(X_0, u_0), u_0, u_1)$*, or*

2. $\mathrm{raise}(\mathrm{raise}(X_1, u_1), u_1) = \mathrm{subst}(\mathrm{raise}(\mathrm{raise}(X_0, u_0), u_0), u_0, u_1)$

*Proof.* Let $\beta = bid(bid\text{-}graph(X_0), u_0)$. Let $X_0' = raise'(X_0, u_0)$ and let $X_0'' = raise''(X_0', u_0)$. Let $X_1' = raise'(X_1, u_1)$ and let $X_1'' = raise''(X_1', u_1)$. Note that $items(X_0, u_0) = items(X_1, u_1)$. Thus, by the definition of the function *raise*$'$ it follows that $X_1' = \mathit{subst}(X_0', u_0, u_1)$. If $u_0$ belongs to *matched*$(X_0')$, then it is easy to see that $X_1'' = \mathit{subst}(X_0'', u_0, u_1)$ and the proof is complete. We now consider the case where $u_0$ belongs to *unmatched*$(X_0')$. Note that *items*$(X_0', u_0) =$ *items*$(X_1', u_1)$. Since $u_1 < u_0$, it follows from the definition of the function *raise*$''$ that if $u_1$ belongs to *matched*$(X_1'')$, then $u_0$ belongs to *matched*$(X_0'')$. Similarly, if $u_0$ belongs to *unmatched*$(X_0'')$, then $u_1$ belongs to *unmatched*$(X_1'')$. Thus, either $X_1'' = \mathit{subst}(X_0'', u_0, u_1)$, or $u_0$ belongs to *matched*$(X_0'')$ and $u_1$ belongs to *unmatched*$(X_1'')$. Thus, there exists an item $v$ in *items*$(\mathrm{inc}(X_1', u_1), u_1)$ such that *match*$(\mathrm{inc}(X_1', u_1), v)$ belongs to *zero*$(\mathrm{inc}(X_1', u_1))$ and $u_1 < u' < u_0$. It is easy to see from the definition of the function *raise*$'$ that *raise*$'(X_0'', u_0) = X_0''$ and *raise*$'(X_1'', u_1) = \mathit{subst}(X_0'', u_0, u_1)$. Thus, *raise*$(X_1'', u_0) = \mathit{subst}(\mathit{raise}(X_0'', u_0), u_0, u_1)$. □

## 5.5 Commutativity of *Raise* Operations

A key property of the bottom-level auction is the commutativity of *raise* invocations. This property is formalized in Lemma 5.5.9 and is used extensively in Chapter 6 of this dissertation.

**Lemma 5.5.1.** *For any quiescent ECC $X$, any agents $u_0$ and $u_1$ in* unmatched$(X)$ *such that* agents$(X, u_0) \cap$ nonpositive$(X) = \emptyset$*, and any item $v$ in* items$(X, u_0)$*, we have $v$ belongs to* items(raise$(X, u_1), u_0)$ *if and only if*

$$\text{potential}(\text{raise}(X, u_1), v) = \text{potential}(X, v)$$

*Proof.* Since *agents*$(X, u_0) \cap$ *nonpositive*$(X) = \emptyset$, it follows that *victim*$(X, u_1, 1)$

does not belong to $\mathit{agents}(X, u_0)$, thus $\mathit{agents}(X, u_0) = \mathit{agents}(\mathit{raise}'(X, u_1), u_0)$ and $\mathit{items}(X, u_0) = \mathit{items}(\mathit{raise}'(X, u_1), u_0)$. Let $\chi = (G, M, \Phi)$ be any configuration in $X$ and let $\chi' = (G', M', \Phi')$ be any configuration in $\mathit{raise}(X, u_1)$. By Lemma 3.8.1, we have $\mathit{items}(\chi, u_0) = \mathit{items}(X, u_0)$ and $\mathit{items}(\chi', u_0) = \mathit{items}(\mathit{raise}(X, u_1), u_0)$. By definition, $v$ belongs to $\mathit{items}(\chi, u_0)$ if and only if there exists a directed path from $u_0$ to $v$ in $\mathit{digraph}(\chi)$, where every edge $(u', v')$ in $\mathit{digraph}(\chi)$ is such that $v'$ belongs to $\mathit{demand}(\chi, \mathit{bid}(\mathit{bid\text{-}graph}(X), u'))$.

It is easy to see that if $\mathit{potential}(\mathit{raise}(X, u_1), v) > \mathit{potential}(X, v)$, then there is no directed path from $u_0$ to $v$ in $\mathit{digraph}(\chi)$. We now consider the case where $\mathit{potential}(\mathit{raise}(X, u_1), v) = \mathit{potential}(X, v)$. It follows from the definition of the *raise* function that if $\Phi'(v') > \Phi(v')$ for some item $v'$ on a directed path from $u_0$ to $v$, then $\Phi'(v) > \Phi(v)$, and this would contradict our assumption that $\mathit{potential}(\mathit{raise}(X, u_1), v) = \mathit{potential}(X, v)$. Thus every item $v'$ on every directed path from $u_0$ to $v$ has $\Phi'(v') = \Phi(v')$; it follows that all such directed paths are preserved in $\mathit{digraph}(\chi')$, and thus, $v$ belongs to $\mathit{items}(\mathit{raise}(X, u_1), u_0)$. □

**Lemma 5.5.2.** *For any quiescent ECC $X$ and any agents $u_0$ in* $\text{unmatched}(X)$ *and $u_1$ in* $\text{enabled}(X)$*, if* $\text{agents}(X, u_0) \cap \text{nonpositive}(X) = \emptyset$*, then*

$$\text{agents}(X_1, u_0) \subseteq \text{agents}(X, u_0) \ \wedge \ \text{agents}(X_1, u_0) \cap \text{nonpositive}(X_1) = \emptyset$$

*where* $X_1 = \text{raise}(X, u_1)$.

*Proof.* If agent $u_1$ belongs to $\mathit{matched}(X)$, then by Fact 5.3.1, we have $X_1 =$

*shift*$(X, u_1, 1)$; in this case it is easy to see that *agents*$(X_1, u_0) \subseteq$ *agents*$(X, u_0)$ and *agents*$(X_1, u_0) \cap$ *nonpositive*$(X_1) = \emptyset$.

We now consider the case where $u_1$ belongs to *unmatched*$(X)$. By Lemma 5.5.1, we have *items*$(X_1, u_0) \subseteq$ *items*$(X, u_0)$ and *potential*$(X_1, v) =$ *potential*$(X, v)$ for any item $v$ in *items*$(X_1, u_0)$. Thus, we have *agents*$(X_1, u_0) \subseteq$ *agents*$(X, u_0)$ and *agents*$(X_1, u_0) \cap$ *nonpositive*$(X_1) = \emptyset$. □

**Lemma 5.5.3.** *Let $X_0$ and $X_1$ be quiescent ECCs such that the following conditions hold*

1. bid-graph$(X_0)$ = bid-graph$(X_1)$
2. potential$(X_0)$ = potential$(X_1)$
3. *for any agent $u$ in* nonwhite$(X) \cap$ matched$(X)$*, there exists an item $v$ such that* match$(\chi, v) = u$ *for any configuration $\chi$ in $X_0 \cup X_1$*

*For any agents $u_0$ and $u_1$ such that* matched$(X_0) \setminus$ matched$(X_1) = \{u_1\}$ *and* matched$(X_1) \setminus$ matched$(X_0) = \{u_0\}$*, if $u_1$ belongs to* agents$(X_0, u_0)$ *and $u_0$ belongs to* agents$(X_1, u_1)$*, then either* victim$(X_0, u_0, 1)$ = victim$(X_1, u_1, 1)$ *or* victim$(X_0, u_0, 0)$ = victim$(X_1, u_1, 0)$*.*

*Proof.* Let $U =$ *matched*$(X_0) - u_1 =$ *matched*$(X_1) - u_0$. We have *potential*$(X_0) =$ *potential*$(X_1)$ and for any agent $u$ in *nonwhite*$(X) \cap$ *matched*$(X)$, there exists an item $v$ such that *match*$(\chi, v) = u$ for any configuration $\chi$ in $X_0 \cup X_1$; thus we have *nonpositive*$(X_0) =$ *nonpositive*$(X_1)$ and for any agent $u$ in $U$, we have

$agents(X_0, u) = agents(X_1, u)$ and $items(X_0, u) = items(X_1, u)$. Additionally, since $u_1$ belongs to $agents(X_0, u_0)$ and $u_0$ belongs to $agents(X_1, u_1)$, we have $nonpositive(X_0) \cap agents(X_0, u_0) = nonpositive(X_1) \cap agents(X_1, u_1)$. By the definition of the function *victim*, it is easy to see that either $victim(X_0, u_0, 1) = victim(X_1, u_1, 1)$ or $victim(X_0, u_0, 0) = victim(X_1, u_1, 0)$. $\square$

**Lemma 5.5.4.** *For any quiescent ECC $X$ and any agents $u_0$ and $u_1$ in* $\text{unmatched}(X)$*, if* $\text{victim}(X, u_0) = \text{victim}(X, u_1, 1)$*, then agent $u_0$ belongs to* $\text{agents}(\text{raise}(X, u_0), u_1)$*.*

*Proof.* Let $X_0 = raise(X, u_0)$ and let $victim(X, u_0) = victim(X, u_1, 1) = u$. Since $u = victim(X, u_1, 1)$, we have $u_1$ belongs to $nonpositive(X)$ and $V \subseteq items(X, u_1)$ where $V = demand(X, u)$.

Suppose $V \cap items(X_0, u_1) = \emptyset$. Then, by Lemma 5.5.1, we have $potential(X_0, v) = potential(X, v)+1$ for every item in $V$ and $V \subseteq items(X, u_0)$; thus $u$ belongs to $agents(X, u_0)$ and by the definition of the function $raise'$, $potential(X_0) = potential(X)$, which is a contradiction. Thus, we have $V \cap items(X_0, u_1) \neq \emptyset$. Additionally, since $u = victim(X, u_0)$, we have $u_0$ belongs to $agents(X_0, u')$ for any agent $u'$ having $V \cap items(X_0, u') = \emptyset$. Thus, $u_0$ belongs to $agents(raise(X, u_0), u_1)$. $\square$

**Lemma 5.5.5.** *For any quiescent ECC $X$ and any agents $u_0$ and $u_1$ in* $\text{unmatched}(X)$ *such that* $u_1 = \text{victim}(X, u_1, 1)$*, if* $\text{victim}(X, u_0)$ *is equal to* $\text{victim}(X'_1, u_1, 0)$ *where* $X'_1 = inc(\text{raise}'(X, u_1), u_1)$*, then agent $u_0$ belongs to* $\text{agents}(X'_{01}, u_1)$ *where* $X'_{01} = inc(\text{raise}'(\text{raise}(X, u_0), u_1), u_1)$*.*

*Proof.* Let $X_0 = \mathit{raise}(X, u_0)$ and let $\mathit{victim}(X, u_0) = \mathit{victim}(X'_1, u_1, 0) = u$. Note that the case where $\mathit{victim}(X, u_0, 1) = u$ is symmetric to the case handled by Lemma 5.5.4; thus the proof of this case follows from Lemma 5.5.4.

We now focus on the case where $\mathit{victim}(X, u_0, 1) = u_0$. By the definition of the function $\mathit{raise}''$, we have $\mathit{potential}(X_0, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0)$ and $\mathit{potential}(X, v) = \mathit{potential}(X, v)$ for any item $v$ in $\mathit{items}(X) \setminus \mathit{items}(X, u_0)$. Since $\mathit{victim}(X, u_0) = u$, we have $u_0$ belongs to $\mathit{agents}(X_0, u')$ for any agent $u'$ such that $\mathit{demand}(X_0, u) \cap \mathit{items}(X_0, u') \neq \emptyset$. Since $\mathit{victim}(X, u_1, 1) = u_1$, we have $\mathit{nonpositive}(X) \cap \mathit{agents}(X, u_1) = \emptyset$; thus by Lemmas 5.5.1 and 5.5.2, we have $\mathit{potential}(X'_{01}, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0) \cup \mathit{items}(X, u_1)$. Since $\mathit{victim}(X'_1, u_1, 0) = u$, we have $V \cap \mathit{items}(X'_{01}, u_1) \neq \emptyset$. It follows that $u_0$ belongs to $\mathit{agents}(X'_{01}, u_1)$. □

**Lemma 5.5.6.** *Let $X$ be a quiescent ECC and let $u_0$ and $u_1$ be agents in* $\mathrm{unmatched}(X)$. *Let $X_0 = \mathrm{raise}(X, u_0)$ and let $X_1 = \mathrm{raise}(X, u_1)$. If* $\mathrm{victim}(X, u_0)$ *is not equal to* $\mathrm{victim}(X, u_1)$*, then we have* $\mathrm{victim}(X_0, u_1) = \mathrm{victim}(X, u_1)$ *and* $\mathrm{potential}(\mathrm{raise}(X_0, u_1), v) = \mathrm{potential}(X_1, v)$ *for any item $v$ in* $\mathrm{items}(X)$ *such that* $\mathrm{potential}(X_1, v) = \mathrm{potential}(X, v) + 1$.

*Proof.* First we consider the case where $\mathit{victim}(X, u_1, 1) \neq u_1$. In this case, we have $\mathit{victim}(X, u_1) = \mathit{victim}(X, u_1, 1)$ and $\mathit{potential}(X_1) = \mathit{potential}(X)$; thus $\mathit{victim}(X, u_1)$ belongs to $\mathit{nonpositive}(X)$. The statement of the lemma assumes that $\mathit{victim}(X, u_0) \neq \mathit{victim}(X, u_1)$; thus, by the definition of the function $\mathit{raise}$, we find that $\mathit{victim}(X, u_1)$ belongs to $\mathit{agents}(X_0, u_1) \cap \mathit{nonpositive}(X_0)$.

If $\mathit{victim}(X_0, u_1) = \mathit{victim}(X, u_1)$, then the proof is complete. Suppose that $\mathit{victim}(X_0, u_1) \neq \mathit{victim}(X, u_1)$. Then there is an agent $u'$ in $\mathit{agents}(X, u_1)$ such that $u' = \mathit{victim}(X, u_0)$, and hence $u_0$ belongs to $\mathit{agents}(X_0, u_1)$. Since $u' = \mathit{victim}(X_0, u_0)$, $u'$ belongs to $\mathit{agents}(X, u_1)$, and $u' \neq \mathit{victim}(X, u_1, 1)$, the definition of the function *victim* implies that $\mathit{victim}(X_0, u_1, 1) = \mathit{victim}(X, u_1, 1)$.

Next we consider the case where $\mathit{victim}(X, u_1, 1) = u_1$; thus, by the definition of the function *raise*, we have $\mathit{nonpositive}(X) \cap \mathit{agents}(X, u_1) = \emptyset$ and $\mathit{potential}(X_1, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_1)$. Thus, $\mathit{victim}(X, u_1) = \mathit{victim}(X_1', u_1, 0)$, where $X_1' = \mathrm{inc}(\mathit{raise}'(X, u_1), u_1)$. By Lemma 5.5.1, we have $\mathit{items}(X_0, u_1) = \mathit{items}(X, u_1) \setminus \mathit{items}(X, u_0)$ and $\mathit{potential}(X_0, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0)$. By Lemmas 5.5.1 and 5.5.2, we have $\mathit{nonpositive}(X_0) \cap \mathit{agents}(X_0, u_1) = \emptyset$; thus, we have $\mathit{potential}(\mathit{raise}(X_0, u_1), v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X_0, u_1)$. Since $\mathit{items}(X_0, u_1) = \mathit{items}(X, u_1) \setminus \mathit{items}(X, u_0)$, we have $\mathit{potential}(\mathit{raise}(X_1, u_1), v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0) \cup \mathit{items}(X, u_1)$.

Let $X_{01}' = \mathrm{inc}(\mathit{raise}'(X_0, u_1), u_1)$. If $\mathit{victim}(X_{01}', u_1, 0) = \mathit{victim}(X_1', u_1, 0)$, then the proof is complete. Suppose that $\mathit{victim}(X_{01}', u_1, 0) \neq \mathit{victim}(X_1', u_1, 0)$; then there is an agent $u'$ in $\mathit{agents}(X_1', u_1)$ such that $u' = \mathit{victim}(X, u_0)$, and hence $u_0$ belongs to $\mathit{agents}(X_{01}', u_1)$. Since $u' = \mathit{victim}(X_0, u_0)$, $u'$ belongs to $\mathit{agents}(X_1', u_1)$, and $u' \neq \mathit{victim}(X_1', u_1, 0)$. The definition of the function *victim* implies that $\mathit{victim}(X_{01}', u_1, 0) = \mathit{victim}(X_1', u_1, 0)$.

Thus, $\mathit{victim}(X_0, u_1) = \mathit{victim}(X, u_1)$ and $\mathit{potential}(\mathit{raise}(X_0, u_1), v) =$

$potential(X_1, v)$ for any item $v$ in $items(X)$ such that $potential(X_1, v) = potential(X, v) + 1$.

$\square$

**Lemma 5.5.7.** *For any quiescent ECC $X$ and any agents $u_0$ and $u_1$ in* $\text{agents}(X)$*, if* $\text{matched}(X) \cap \{u_0, u_1\} \neq \emptyset$*, then* $\text{potential}(X_{01}) = \text{potential}(X_{10})$ *and* $\text{matched}(X_{01}) = \text{matched}(X_{10})$*, where* $X_{01} = \text{raise}(\text{raise}(X, u_0), u_1)$ *and* $X_{10} = \text{raise}(\text{raise}(X, u_1), u_0)$*.*

*Proof.* Let $X_0 = raise(X, u_0)$ and let $X_1 = raise(X, u_1)$.

We first consider the case where $|\{u_0, u_1\} \cap matched(X)| = 2$; thus, $\{u_0, u_1\} \subseteq matched(X)$. By Fact 5.3.1, we have $X_{01} = shift(shift(X, u_0, 1), u_1, 1)$, and $X_{10} = shift(shift(X, u_1, 1), u_0, 1)$; thus, $X_{01} = X_{10}$.

We now focus on the case where $|\{u_0, u_1\} \cap matched(X)| = 1$. Without loss of generality, we assume that $\{u_0, u_1\} \cap matched(X) = \{u_1\}$; thus, $u_1$ belongs to $matched(X)$. Since $u_1$ belongs to $enabled(X) \cap matched(X)$ and $X_1 = shift(X, u_1, 1)$, either $u_1$ belongs to $nonwhite(X) \cap nonwhite(X_1)$ or $u_1$ belongs to $white(X) \cap white(X_1)$. If $u_1$ belongs to $nonwhite(X) \cap nonwhite(X_1)$, then for every item $v$ in $items(X)$, we have $\beta(v) < potential(X, v) - 2$, where $\beta = bid(bid\text{-}graph(X), u_1)$. Thus we have $victim(X, u_0) = victim(X_1, u_0) = u_1$ and $raise(X_0, u_1) = X_0$. Using these facts, it is straightforward to argue that $potential(X_{01}) = potential(X_{10})$ and $matched(X_{01}) = matched(X_{10})$. It remains to address the case where $u$ belongs to $white(X) \cap white(X_1)$. We proceed via the following case analysis.

- Case 1: $\mathit{victim}(X, u_0) \neq u_1$.

  - Case 1.1: $\mathit{victim}(X, u_0, 1) \neq u_0$.

    We have $\mathit{victim}(X, u_0) = \mathit{victim}(X, u_0, 1)$. In this case, $u_0$ belongs to $\mathit{matched}(\mathit{raise}'(X, u_0))$; thus, by the definition of the function $\mathit{raise}$, we have $\mathit{matched}(X_0) = \mathit{matched}(X) + u_0 - \mathit{victim}(X, u_0)$ and $\mathit{potential}(X_0) = \mathit{potential}(X)$. By Fact 5.3.1, we have $X_{01} = \mathit{shift}(X_0, u_1, 1)$; thus we have $\mathit{potential}(X_{01}) = \mathit{potential}(X)$ and $\mathit{matched}(X_{01}) = \mathit{matched}(X) + u_0 - \mathit{victim}(X, u_0)$.

    Since $\mathit{victim}(X, u_0) \neq u_1$, there exists an agent $u'$ in $\mathit{nonwhite}(X) \cap \mathit{agents}(X, u_0)$ such that $\mathit{victim}(X, u_1, 1) = u'$. By Fact 5.3.1, we have $X_1 = \mathit{shift}(X, u_1, 1)$ and thus, $\mathit{nonwhite}(X) \cap \mathit{agents}(X, u_0) - u_1 = \mathit{nonwhite}(X_1) \cap \mathit{agents}(X_1, u_0) - u_1$; it follows that $\mathit{victim}(X_1, u_0)$ is equal to $\mathit{victim}(X, u_0)$. Thus, $\mathit{matched}(X_{10}) = \mathit{matched}(X) + u_0 - \mathit{victim}(X, u_0)$. Since $X_1 = \mathit{shift}(X, u, 1)$ and $u_0$ belongs to $\mathit{matched}(\mathit{raise}'(X, u_0))$, we have $\mathit{potential}(X_{10}) = \mathit{potential}(X)$.

    Thus, we have $\mathit{matched}(X_{01}) = \mathit{matched}(X_{10}) \land \mathit{potential}(X_{01}) = \mathit{potential}(X_{10})$.

  - Case 1.2: $\mathit{victim}(X, u_0, 1) = u_0$.

    In this case, $\mathit{victim}(X, u_0) = \mathit{victim}(X_0', u_0, 0)$ where $X_0'$ is equal to $\mathrm{inc}(\mathit{raise}'(X, u_0), u_0)$. Since $\mathit{victim}(X, u_0, 1) = u_0$, it follows that $\mathit{nonwhite}(X) \cap \mathit{agents}(X, u_0) = \emptyset$; thus, $\mathit{potential}(X_0, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0)$. By Fact 5.3.1, we have $X_{01} = \mathit{shift}(X_0, u_1, 1)$; thus $\mathit{potential}(X_{01}) = \mathit{potential}(X_0)$ and $\mathit{matched}(X_{01}) = \mathit{matched}(X) + u_0 - \mathit{victim}(X, u_0)$.

We established above that $X_1 = \mathit{shift}(X, u_1, 1)$; thus $\mathit{potential}(X_1)$ is equal to $\mathit{potential}(X)$ and $\mathit{matched}(X_1) = \mathit{matched}(X)$. Further, since $\mathit{agents}(X, u_0) \cap \mathit{nonpositive}(X) = \emptyset$ and $\mathit{potential}(X_1) = \mathit{potential}(X_0)$, by Lemmas 5.5.1 and 5.5.2 can be used to argue that we have $\mathit{items}(X_1, u_0) = \mathit{items}(X, u_0)$ and $\mathit{agents}(X_1, u_0) \cap \mathit{nonpositive}(X_1) = \emptyset$; thus $\mathit{potential}(X_{10}, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0)$.

Let $X_1' = \mathrm{inc}(\mathit{raise}'(X_1, u_0), u_0)$; it is easy to see that $X_1'$ is equal to $\mathit{shift}(X_0', u_1, 1)$. Since $u_1 \neq \mathit{victim}(X_0', u_0)$ and $X_1' = \mathit{shift}(X_0', u_1, 1)$, we have $u_1 \neq \mathit{victim}(X_1', u_0)$; thus, $\mathit{victim}(X_1', u_0) = \mathit{victim}(X_0', u_0) = \mathit{victim}(X, u_0)$, and $\mathit{matched}(X_{10}) = \mathit{matched}(X) + u_0 - \mathit{victim}(X, u_0)$.

It follows that $\mathit{matched}(X_{01}) = \mathit{matched}(X_{10})$ and $\mathit{potential}(X_{01}) = \mathit{potential}(X_{10})$.

- Case 2: $\mathit{victim}(X, u_0) = u_1$.

  - Case 2.1: $\mathit{victim}(X, u_0, 1) \neq u_0$.

    In this case, $\mathit{victim}(X, u_0) = \mathit{victim}(X, u_0, 1) = u_1$; thus, we have $\mathit{potential}(X_0) = \mathit{potential}(X)$ and $\mathit{matched}(X_0) = \mathit{matched}(X) + u_0 - u_1$. By the definition of the function *raise*, we have $\mathit{gap}(X, u_1) = 0$ and $\mathit{gap}(X_0, u_0) = 1$. We consider two sub-cases.

First we consider the case where $\mathit{agents}(X_0, u_1) \cap \mathit{nonpositive}(X_0) \neq \emptyset$. In this case, we have $\mathit{potential}(X_{01}) = \mathit{potential}(X_0)$ and by Lemma 5.5.2, $\mathit{agents}(X_0, u_1) \cap \mathit{nonpositive}(X_0) = \mathit{nonpositive}(X) \cap (\mathit{agents}(X, u_1) \cup \mathit{agents}(X, u_0))$. Since $\mathit{gap}(X, u_1) = 0$ and $X_1 = \mathit{shift}(X, u_1, 1)$, we have $\mathit{gap}(X_1, u_1) = 1$; by the definition of the function $\mathit{raise}'$, we find that $u_1$ does not belong to $\mathit{nonpositive}(X_0)$ and $\mathit{victim}(X_0, u_1) \neq u_1$. Thus, $\mathit{potential}(X_{01}) = \mathit{potential}(X)$ and $\mathit{matched}(X_{01}) = \mathit{matched}(X) + u_0 + u_1 - \mathit{victim}(X_0, u_1)$.

Since $u_1$ belongs to $\mathit{agents}(X_1, u_0)$ and $X_1 = \mathit{shift}(X, u_1, 1)$, it follows that $\mathit{potential}(X_1) = \mathit{potential}(X_0)$ and $\mathit{nonpositive}(X_1) \cap \mathit{agents}(X_1, u_0) = \mathit{nonpositive}(X) \cap (\mathit{agents}(X, u_0) \cup \mathit{agents}(X, u_1)) - u_1$; thus we have $\mathit{victim}(X_1, u_0) = \mathit{victim}(X_0, u_1)$. It follows that $\mathit{potential}(X_{10}) = \mathit{potential}(X)$ and $\mathit{matched}(X_{10}) = \mathit{matched}(X) - \mathit{victim}(X_0, u_1) + u_0 + u_1$.

Next we consider the case where $\mathit{agents}(X_0, u_1) \cap \mathit{nonpositive}(X_0) = \emptyset$, In this case, we have $\mathit{potential}(X_{01}, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X_0, u_1)$ and $\mathit{victim}(X_0, u_1) = \mathit{victim}(X_0', u_1, 0)$ where $X_0' = \mathrm{inc}(\mathit{raise}'(X, u_1), u_1)$. Since $u_0$ belongs to $\mathit{agents}(X_0, u_1)$ and $\mathit{gap}(X_0, u_0) = 1$, we have $u_0$ belongs to $\mathit{nonpositive}(X_0') \cap \mathit{agents}(X_0', u_1)$; thus $u_1$ belongs to $\mathit{matched}(X_{01})$. Thus, we have $\mathit{matched}(X_{01}) = \mathit{matched}(X) + u_0 + u_1 - \mathit{victim}(X_0', u_1, 0)$ and $\mathit{potential}(X_{01}, v) = \mathit{potential}(X, v) + 1$ for any item in $\mathit{items}(X, u_1)$. Since $\mathit{potential}(X_0) = \mathit{potential}(X)$ and $\mathit{victim}(X, u_0, 1) = u_1$, and since $\mathit{agents}(X_0, u_1) \cap \mathit{nonpositive}(X_0) = \emptyset$, we have $\mathit{agents}(X, u_0) \cap \mathit{nonpositive}(X_0) = \{u_1\}$. Since $X_1 = \mathit{shift}(X, u, 1)$, it follows that $\mathit{agents}(X_1, u_0) \cap \mathit{nonpositive}(X_1) = \emptyset$; thus $\mathit{potential}(X_{10}, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X_1, u_0) = \mathit{items}(X_0, u_1)$. Let $X_1' = \mathrm{inc}(\mathit{raise}'(X_1, u_1), u_1)$; thus we have $X_1' = \mathit{shift}(X_0', u_1, 1)$, $\mathit{agents}(X_1', u_0) \cap \mathit{nonpositive}(X_1') = \mathit{agents}(X_0', u_1) \cap \mathit{nonpositive}(X_0') - u_1$, and we have $\mathit{victim}(X_1', u_0, 0) = \mathit{victim}(X_0', u_1, 0)$. Therefore, $\mathit{potential}(X_{10}) = \mathit{potential}(X)$ and $\mathit{matched}(X_{10}) = \mathit{matched}(X) + u_0 + u_1 - \mathit{victim}(X_0', u_1, 0)$.

– Case 2.2: $\mathit{victim}(X, u_0, 1) = u_0$.

  In this case, $\mathit{victim}(X, u_0) = \mathit{victim}(X_0', u_0, 0) = u_1$ where $X_0' = \mathrm{inc}(\mathit{raise}'(X, u_0), u_0)$. Since $\mathit{victim}(X, u_0, 1) = u_0$, it follows that $\mathit{nonwhite}(X) \cap \mathit{agents}(X, u_0) = \emptyset$; as a result, $\mathit{potential}(X_0, v) = \mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0)$, and by the definition of the function $\mathit{raise}''$, we have $\mathit{gap}(X_0, u_0) = 0$. Since $u_0$ belongs to $\mathit{agents}(X_0, u_1)$ and $\mathit{gap}(X_0, u_0) = 0$, we have $u_1$ belongs to $\mathit{matched}(\mathit{raise}'(X_0, u_1))$; thus $\mathit{potential}(X_{01}) = \mathit{potential}(X_0)$ and $\mathit{matched}(X_{01}) = \mathit{matched}(X) + u_0 + u_1 - \mathit{victim}(X_0, u_0, 1)$.

  We established above that $X_1 = \mathit{shift}(X, u_1, 1)$; thus $\mathit{potential}(X_1) = \mathit{potential}(X)$ and $\mathit{matched}(X_1) = \mathit{matched}(X)$. Since $\mathit{agents}(X, u_0) \cap \mathit{nonpositive}(X) = \emptyset$ and $\mathit{potential}(X_1) = \mathit{potential}(X)$, Lemmas 5.5.1 and 5.5.2 imply that $\mathit{agents}(X_1, u_0) \cap \mathit{nonpositive}(X_1) = \emptyset$ and $\mathit{items}(X_1, u_0) = \mathit{items}(X, u_0)$; thus we have $\mathit{potential}(X_1', v) =$

$\mathit{potential}(X, v) + 1$ for any item $v$ in $\mathit{items}(X, u_0)$, where $X'_1 = \text{inc}(\mathit{raise}'(X_1, u_0), u_0)$.

Since $\mathit{gap}(X_0, u_1) = 0$, $\mathit{potential}(X'_1) = \mathit{potential}(X_0)$ and $X_1 = \mathit{shift}(X, u_1, 1)$, we have $\mathit{gap}(X'_1, u_1) = 1$; thus $u_1 \neq \mathit{victim}(X'_1, u_0, 0)$. By the definition of the function *raise*, we have $X'_1 = \mathit{shift}(X_0, u_1, 1)$; thus, $\mathit{victim}(X'_1, u_0) = \mathit{victim}(X_0, u_0, 0)$.

It follows that $\mathit{matched}(X_{10}) = \mathit{matched}(X) + u_0 + u_1 - \mathit{victim}(X, u_0, 0)$ and $\mathit{potential}(X_{10}) = \mathit{potential}(X_0)$.

$\square$

**Lemma 5.5.8.** *For any quiescent ECC $X$ and any agents $u_0$ and $u_1$ in* $\text{unmatched}(X)$*, if* $\text{victim}(X, u_0) = \text{victim}(X, u_1) = u$*, then*

$$\text{potential}(X_{01}) = \text{potential}(X_{10}) \wedge \text{matched}(X_{01}) = \text{matched}(X_{10})$$

*where* $X_{01} = \text{raise}(\text{raise}(X, u_0), u_1)$ *and* $X_{10} = \text{raise}(\text{raise}(X, u_1), u_0)$*.*

*Proof.* Let $X_0 = \mathit{raise}(X, u_0)$ and let $X_1 = \mathit{raise}(X, u_1)$.
Let $X'_0 = \text{inc}(\mathit{raise}'(X, u_0), u_0)$, let $X'_{01} = \text{inc}(\mathit{raise}'(X_0, u_1), u_1)$, let $X'_1 = \text{inc}(\mathit{raise}'(X, u_1), u_1)$, and let $X'_{10} = \text{inc}(\mathit{raise}'(X_1, u_0), u_0)$. We consider the following cases.

- Case 1: $\mathit{victim}(X, u_1, 1) \neq u_1$
  - Case 1.1 $\mathit{victim}(X, u_0, 1) \neq u_0$. We begin by establishing the following sequence of claims.

1. *potential*$(X_0)$ = *potential*$(X)$. Follows from the definition of the function *raise*$''$ and the fact that *victim*$(X, u_0, 1) \neq u_0$.

2. $u_0$ belongs to *agents*$(X_0, u_1)$. Follows from Lemma 5.5.4 and the fact that *victim*$(X, u_0)$ = *victim*$(X, u_1, 1) = u$.

3. *potential*$(X_1)$ = *potential*$(X)$. Follows from the definition of the function *raise*$''$ and the fact that *victim*$(X, u_1, 1) \neq u_1$.

4. $u_1$ belongs to *agents*$(X_1, u_0)$. Follows from Lemma 5.5.4 and the fact that *victim*$(X, u_1)$ = *victim*$(X, u_0, 1) = u$.

   We now consider two sub-cases.

   (a) Case 1.1.2 *victim*$(X_0, u_1, 1) \neq u_1$.

   By claims 1 and 2, we have *potential*$(X_0)$ = *potential*$(X_1)$ and by claims 2 and 4, we have $u_0$ belongs to *agents*$(X_0, u_1)$ and $u_1$ belongs to *agents*$(X_1, u_0)$; thus *victim*$(X_1, u_0, 1) \neq u_0$.

   Since *victim*$(X, u_0)$ = *victim*$(X, u_1) = u$, it follows that *matched*$(X_0) \setminus$ *matched*$(X_1) = \{u_0\}$ and *matched*$(X_1) \setminus$ *matched*$(X_0) = \{u_1\}$. Further, by the definition of the function *raise*, we have *bid-graph*$(X_0)$ = *bid-graph*$(X_1)$ and for any agent $u'$ in *nonwhite*$(X)$, there exists an item $v'$ in *items*$(X)$ such that *match*$(\chi, v) = u'$ for any configuration $\chi$ in $X_0 \cup X_1$. Thus, by Lemma 5.5.3, we have *victim*$(X_0, u_1, 1)$ = *victim*$(X_1, u_0, 1)$.

   Further, by the definition of the function *raise*$''$, we have *potential*$(X_{01})$ = *potential*$(X_{10})$. Since *victim*$(X, u_0)$ = *victim*$(X, u_1)$ and *victim*$(X_0, u_1, 1)$ = *victim*$(X_1, u_0, 1)$, we have *matched*$(X_{01})$ = *matched*$(X_{10})$.

(b) Case 1.1.2 *victim*$(X_0, u_1, 1) = u_1$.

By claims 1 and 2, we have *potential*$(X_0)$ = *potential*$(X_1)$ and by claims 2 and 4, we have $u_0$ belongs to *agents*$(X_0, u_1)$ and $u_1$ belongs to *agents*$(X_1, u_0)$; thus *agents*$(X_0, u_1)$ = *agents*$(X_1, u_0)$ and by the definition of the function *raise*$''$, we have *potential*$(X'_{01})$ = *potential*$(X'_{10})$.

We know that *victim*$(X, u_0)$ = *victim*$(X, u_1) = u$, thus *matched*$(X'_{01}) \setminus$ *matched*$(X'_{10}) = \{u_0\}$ and *matched*$(X'_{10}) \setminus$ *matched*$(X'_{01}) = \{u_1\}$. Further, by the definition of the function *raise*, we have *bid-graph*$(X'_{01})$ = *bid-graph*$(X'_{10})$ and for any agent $u'$ in *nonwhite*$(X)$, there exists an item $v'$ in *items*$(X)$ such that *match*$(\chi, v) = u'$ for any configuration $\chi$ in $X'_{01} \cup X'_{10}$. Thus, by Lemma 5.5.3, we have *victim*$(X'_{01}, u_1, 0)$ = *victim*$(X'_{10}, u_0, 0)$. By the definition of the function *raise*$''$, we have *potential*$(X_{01})$ = *potential*$(X_{10})$.

Since *victim*$(X, u_0)$ = *victim*$(X, u_1)$ and *victim*$(X'_{01}, u_1, 0)$ = *victim*$(X_{10}, u_0, 0)$, we have *matched*$(X_{01})$ = *matched*$(X_{10})$.

– Case 1.2. *victim*$(X, u_0, 1) = u_0$. We begin by establishing the following sequence of claims.

1. $\textit{potential}(X_0, v) = \textit{potential}(X, v)+1$ for any item $v$ in $\textit{items}(X, u_0)$ and $\textit{potential}(X_0, v) = \textit{potential}(X, v)$ for any item $v$ in $\textit{items}(X)\setminus \textit{items}(X, u_0)$. Follows from the fact that $\textit{victim}(X, u_0, 1) = u_0$ and the definition of the function $\textit{raise}''$.

2. $\textit{gap}(X_0, u_0) = 0$. We have $\textit{victim}(X, u_0, 1) = u_0$ and we have $\textit{victim}(X, u_0) = u$; thus, $u_0$ is matched by a $\textit{raise}''$ invocation and $\textit{gap}(X_0, u_0) = 0$.

3. $u_0$ belongs to $\textit{agents}(X_0, u_1)$. Since $\textit{victim}(X, u_0) = \textit{victim}(X, u_1, 1)$, by Lemma 5.5.4, we have $u_0$ belongs to $\textit{agents}(X_0, u_1)$.

4. $\textit{potential}(X'_{01}) = \textit{potential}(X_0)$. By 2 and 3, we have $u_0$ belongs to $\textit{nonpositive}(X_0) \cap \textit{agents}(X_0, u_1)$; thus by the definition of the function $\textit{raise}'$, we have $\textit{potential}(X'_{01}) = \textit{potential}(X_0)$.

5. $\textit{potential}(X_1) = \textit{potential}(X)$. This follows from the fact that $\textit{victim}(X, u_1, 1) = u$ and the definition of the function $\textit{raise}'$.

6. $\textit{potential}(X'_{10}) = \textit{potential}(X_0)$. Since $\textit{victim}(X, u_0, 1) = u$, we have $\textit{nonpositive}(X) \cap \textit{agents}(X, u_0) = \emptyset$, thus by 5 and Lemma 5.5.1, we have $\textit{items}(X_1, u_0) = \textit{items}(X, u_0)$, and by Lemma 5.5.2, we have $\textit{nonpositive}(X_1) \cap \textit{agents}(X_1, u_0) = \emptyset$; thus by the definition of the function $\textit{raise}''$ and 1, we have $\textit{potential}(X'_{10}) = \textit{potential}(X_0)$.

7. $u_1$ belongs to $\textit{agents}(X'_{10}, u_0)$. Since $\textit{victim}(X, u_0, 1) = u_0$, we have $\textit{victim}(X, u_0) = \textit{victim}(X'_0, u_0, 0) = u$ where $X'_0 = \text{inc}(\textit{raise}'(X, u_0), u_0)$. We have $\textit{victim}(X, u_1, 1) = u$. Thus, by

Lemma 5.5.5, we have $u_1$ belongs to *agents*$(X'_{10}, u_0)$.

8. *victim*$(X'_{10}, u_0, 0) =$ *victim*$(X_0, u_1, 1)$. By claims 4 and 6, we have *potential*$(X'_{01}) =$ *potential*$(X'_{10})$, and by claims 3 and 7 we know that $u_0$ belongs to *agents*$(X_0, u_1)$ and $u_1$ belongs to *agents*$(X_{10'}, u_0)$. It is now easy to see that *matched*$(X_0) \setminus$ *matched*$(X'_{10}) = \{u_0\}$ and *matched*$(X'_{10}) \setminus$*matched*$(X_0) = \{u_1\}$, and by the definition of the function *raise*, for any agent in *nonwhite*$(X'_{01})$, there exists an item in *items*$(X)$ such that *match*$(\chi, v) = u$ for any configuration in $X_0 \cup X'_{10}$. Thus, it follows from Lemma 5.5.5 that *victim*$(X'_{10}, u_0, 0) =$ *victim*$(X_0, u_1, 1)$.

By claims 4 and 6, we have *potential*$(X_{01}) =$ *potential*$(X_{10})$. The statement of the lemma assumes that *victim*$(X, u_0) =$ *victim*$(X, u_1)$ and by claim 8, we have *victim*$(X'_{10}, u_0, 0) =$ *victim*$(X_0, u_1, 1)$; thus *matched*$(X_{01}) =$ *matched*$(X_{10})$.

- Case 2: *victim*$(X, u_1, 1) = u_1$

  - Case 2.1: *victim*$(X, u_0, 1) \neq u_0$

    This case is symmetric to case 1.2.

  - Case 2.1: *victim*$(X, u_0, 1) = u_0$. We begin by establishing the following sequence of claims.

    1. *potential*$(X_0, v) =$ *potential*$(X, v)+1$ for any item $v$ in *items*$(X, u_0)$ and *potential*$(X_0, v) =$ *potential*$(X, v)$ for any item $v$ in *items*$(X)\setminus$

$\mathit{items}(X, u_0)$. Follows from the fact that $\mathit{victim}(X, u_0, 1) = u_0$ and the definition of the function $\mathit{raise}''$.

2. $\mathit{potential}(X'_{01}, v) = \mathit{potential}(X, v)+1$ for any $v$ in $\mathit{items}(X, u_0) \cup \mathit{items}(X, u_1)$ and $\mathit{potential}(X'_{01}, v) = \mathit{potential}(X, v)$ for any $v$ in $\mathit{items}(X) \setminus \mathit{items}(X, u_0) \cup \mathit{items}(X, u_1)$. Since $\mathit{victim}(X, u_1, 1) = u_1$, we have $\mathit{nonpositive}(X) \cap \mathit{agents}(X, u_1) = \emptyset$; by 1 and Lemma 5.5.1, we have $\mathit{items}(X_0, u_1) = \mathit{items}(X, u_1) \setminus \mathit{items}(X, u_0)$, and by Lemma 5.5.2, we have $\mathit{nonpositive}(X_0) \cap \mathit{agents}(X_0, u_1) = \emptyset$; thus, claim 2 follows by the definition of the function $\mathit{raise}''$ and claim 1.

3. $u_0$ belongs to $\mathit{agents}(X'_{01}, u_1)$. By claim 2, $\mathit{potential}(X'_{01}) > \mathit{potential}(X_0)$; thus $\mathit{victim}(X_0, u_1, 1) = u_1$ and $\mathit{victim}(X_0, u_1) = \mathit{victim}(X'_{01}, u_1, 0)$; and by Lemma 5.5.5, we find that $u_0$ belongs to $\mathit{agents}(X'_{01}, u_1)$.

4. $\mathit{potential}(X_1, v) = \mathit{potential}(X, v)+1$ for any item $v$ in $\mathit{items}(X, u_1)$ and $\mathit{potential}(X_0, v) = \mathit{potential}(X, v)$ for any item $v$ in $\mathit{items}(X) \setminus \mathit{items}(X, u_1)$. Follows from the fact that $\mathit{victim}(X, u_0, 1) = u_0$ and the definition of the function $\mathit{raise}''$.

5. $\mathit{potential}(X'_{10}, v) = \mathit{potential}(X, v)+1$ for any $v$ in $\mathit{items}(X, u_1) \cup \mathit{items}(X, u_0)$ and $\mathit{potential}(X'_{10}, v) = \mathit{potential}(X, v)$ for any item $v$ in $\mathit{items}(X) \setminus \mathit{items}(X, u_1) \cup \mathit{items}(X, u_0)$. The analysis is similar to claim 2.

6. $u_1$ belongs to $\mathit{agents}(X'_{10}, u_0)$. By claim 5, $\mathit{potential}(X'_{10}) >$

$\mathit{potential}(X_1)$; thus $\mathit{victim}(X_1, u_0, 1) = u_0$; and $\mathit{victim}(X_1, u_0) = \mathit{victim}(X'_{10}, u_0, 0)$; and by Lemma 5.5.5, we find that $u_1$ belongs to $\mathit{agents}(X'_{10}, u_0)$.

7. $\mathit{victim}(X'_{10}, u_0, 0) = \mathit{victim}(X'_{01}, u_1, 0)$. By claims 4 and 5, we have $\mathit{potential}(X'_{01}) = \mathit{potential}(X'_{10})$ and by claims 3 and 6 we know that $u_0$ belongs to $\mathit{agents}(X'_{01}, u_1)$ and $u_1$ belongs to $\mathit{agents}(X'_{10}, u_0)$. Since $\mathit{victim}(X, u_0) = \mathit{victim}(X, u_1)$, we have $\mathit{matched}(X'_{01}) \setminus \mathit{matched}(X'_{10}) = \{u_1\}$ and $\mathit{matched}(X'_{10}) \setminus \mathit{matched}(X'_{10}) = \{u_0\}$, and by the definition of the function *raise*, for any agent in $\mathit{nonwhite}(X'_{01})$, there exists an item in $\mathit{items}(X)$ such that $\mathit{match}(\chi, v) = u$ for any configuration in $X'_{01} \cup X'_{10}$. It follows from Lemma 5.5.5 that $\mathit{victim}(X'_{10}, u_0, 0) = \mathit{victim}(X'_{01}, u_1, 0)$.

By claims 2 and 5, we have $\mathit{potential}(X_{01}) = \mathit{potential}(X_{10})$. The statement of the lemma assumes that $\mathit{victim}(X, u_0) = \mathit{victim}(X, u_1)$ and by claim 7, we have $\mathit{victim}(X'_{10}, u_0, 0) = \mathit{victim}(X'_{01}, u_1, 0)$; thus, $\mathit{matched}(X_{01}) = \mathit{matched}(X_{10})$.

$\square$

**Lemma 5.5.9.** *For any quiescent ECC $X$ and any agents $u_0$ and $u_1$ in* $\mathrm{enabled}(X)$*, we have*

$$\mathrm{raise}(\mathrm{raise}(X, u_0), u_1) = \mathrm{raise}(\mathrm{raise}(X, u_1), u_0).$$

*Proof.* Let $X_{01} = \mathit{raise}(\mathit{raise}(X, u_0), u_1)$ and let $X_{10} = \mathit{raise}(\mathit{raise}(X, u_1), u_0)$.

We first prove the following claim: $\mathit{potential}(X_{01}) = \mathit{potential}(X_{10})$ and $\mathit{matched}(X_{01}) = \mathit{matched}(X_{10})$. By Lemma 5.5.7, the claim holds when $\mathit{matched}(X) \cap \{u_0, u_1\} \neq \emptyset$. It remains to show that the claim holds when $\{u_0, u_1\} \subseteq \mathit{unmatched}(X)$. By Lemma 5.5.8, the claim holds when $\{u_0, u_1\} \subseteq \mathit{unmatched}(X)$ and $\mathit{victim}(X, u_0) = \mathit{victim}(X, u_1)$. By Lemma 5.5.6, the claim holds when $\{u_0, u_1\} \subseteq \mathit{unmatched}(X)$ and $\mathit{victim}(X, u_0) \neq \mathit{victim}(X, u_1)$.

It now remains to be shown that if $\mathit{potential}(X_{01}) = \mathit{potential}(X_{10})$ and $\mathit{matched}(X_{01}) = \mathit{matched}(X_{10})$, then $X_{01} = X_{10}$. Consider any agent $u$ in $\mathit{nonwhite}(X)$; there exists an item $v$ in $\mathit{items}(X)$ such that $\mathit{match}(\chi, v) = u$ for every configuration $\chi$ in $X$. Note that if $u$ belongs to $\mathit{unmatched}(\mathit{raise}(X, u_0))$, then by the definition of the function *raise*, it follows that $u$ belongs to $\mathit{unmatched}(X_{01})$. Using this fact and by repeated application of Lemma 5.4.1, it follows that either $u$ belongs to $\mathit{unmatched}(X_{01})$ or $\mathit{match}(\chi, v) = u$ for every configuration $\chi$ in $X \cup X_{01}$. By an identical argument, we find that either $u$ belongs to $\mathit{unmatched}(X_{10})$ or $\mathit{match}(\chi, v) = u$ for every configuration in $X \cup X_{10}$. However, since we established above that $\mathit{matched}(X_{01}) = \mathit{matched}(X_{10})$, it follows that $\mathit{match}(\chi, v) = u$ for every configuration $\chi$ in $X \cup X_{01} \cup X_{10}$, and hence, $X_{01} = X_{10}$. □

# Chapter 6

# Mid-Level Auction

In this chapter, we present the mid-level auction. The mid-level auction is a sealed-bid unit-demand auction, and corresponds to a determinized, proxy-based version of the bottom-level auction.

Recall from Chapter 5 that the bottom-level auction is dynamic. In the mid-level auction, we associate with each agent $u$, a proxy agent $u'$ who bids on behalf of $u$ in the bottom-level auction. The bid of agent $u$ in the mid-level auction restricts the number of *raise* invocations of agent $u'$ in the bottom-level auction.

The rest of this chapter is organized as follows. In Section 6.1, we analyze the bottom-level auction when each agent is associated with a restricted bidding strategy — the unit-demand bid of an agent in the mid-level auction restricts the number of *raise* invocations of the agent's proxy in the underlying bottom-level auction. In Section 6.2, we describe a particular tie-breaking scheme that determinizes the *raise* operation described in Section 5.3 to yield a unique allocation and pricing. In Section 6.3, we describe the mid-level auction and establish some basic properties.

## 6.1 A Restricted Class of Bidding Strategies

In this section, we analyze the bottom-level auction when each agent in the auction has a restricted "target" number of *raise* invocations.

We define a *target* as a function from the set of all agents to the set of nonnegative integers. For any target $s$, any agent $u$, and any integer $z$ such that $\alpha(u)+z \geq 0$, we define $\mathit{shift}(s,u,z)$ as the target $s'$ where $s'(u) = \alpha(u)+z$ and $s'(u') = \alpha(u')$ for any agent $u'$ different from $u$. For any configuration $\chi = (G, M, \Phi)$ where $G = (U, V, w)$, and any target $s$, we define $\mathit{shift}(\chi, s)$ as the configuration $(G', M, \Phi)$ where $G' = (U, V, w')$ and $w'(u,v) = w(u,v) + \alpha(u)$ for any agent $u$ in $U$ and any item $v$ in $V$. For any ECC $X$ and any target $s$, we define $\mathit{shift}(X, s)$ as $\cup_{\chi \in X}[\mathit{shift}(\chi, s)]$.

We view the bottom-level auction as taking a pair $(X, s)$ as input, where $X$ is a quiescent ECC and $s$ is a target, and updating this pair over a sequence of rounds. For any agent $u$ in $X$, the nonnegative integer $\alpha(u)$ represents the number of additional *raise* invocations desired by agent $u$. In a general round of the auction with input $(X_0, s_0)$, a single agent $u$ in $\mathit{enabled}(X_0)$ having $s_0(u) > 0$ invokes *raise*, and the output of the round, denoted by $\mathit{raise}(X_0, u, s_0)$ is given by $(\mathit{raise}(X_0, u), \mathit{shift}(s_0, u, -1))$. The auction terminates when no enabled agent has pending *raise* invocations.

We define $\mathit{bottom}(X, s)$ as the output of the bottom-level auction when given the pair $(X, s)$ as input. By Lemma 5.4.3 and Lemma 5.5.9, it follows that $\mathit{bottom}(X, s)$ is uniquely defined.

In Section 6.1.1, we establish various properties of $\mathit{bottom}(X, s)$. These properties are crucial for describing and analyzing the auctions of Chapters 7 and 8.

The facts below follow from the definition of the function *raise* and the commutativity of *raise* invocations established in Lemma 5.5.9.

**Fact 6.1.1.** *For any quiescent ECC $X$, any target $s$, and any agent $u$ in* $\text{enabled}(X)$*, we have*

$$\text{bottom}(\text{raise}(X, u), s) = \text{bottom}(X, \text{shift}(s, u, 1)).$$

**Fact 6.1.2.** *For any quiescent ECC $X_0$ of the form* $\text{add}(X, u, \beta)$ *and any target $s$, we have* $\text{bottom}(X_0, s) = \text{bottom}(\text{add}(X', u, \beta), s')$ *where* $(X', s') = \text{bottom}(X, s)$.

**Fact 6.1.3.** *For any quiescent ECC $X$, any agent $u$ in* $\text{white}(X)$*, and any target $s$, if* $(X_0, s_0) = \text{bottom}(X, s)$*, then*

$$\text{bottom}(X, \text{shift}(s, u, 1)) = \text{bottom}(X_0, \text{shift}(s_0, u, 1)).$$

### 6.1.1 Properties

The goal of this section is to establish Lemma 6.1.10. Lemma 6.1.10 is useful in establishing truthfulness of first phase of the top-level auction of Chapter 7.

[Definitions] For any quiescent ECC $X$ and any target $s$, we define $\mathit{matched}(X, s)$ as the set of agents in $\mathit{matched}(X')$, where $(X', s') = \mathit{bottom}(X, s)$.

For any quiescent ECC $X$, any target $s$, and any item $v$ in $items(X)$, we define $agents(X, s, v)$ as $agents(X', v)$, where $(X', s') = bottom(X, s)$.

**Lemma 6.1.1.** *For any quiescent ECC $X$, any quiescent ECC $X'$ of the form* $\mathrm{subst}(X, u, u')$*, and any target $s$ such that $\alpha(u) = \alpha(u')$, if $u$ belongs to* $\mathrm{matched}(X) \cap \mathrm{white}(X)$*, then* $\mathrm{gap}(X_0, u) = \mathrm{gap}(X_1, u')$*, where* $(X_0, s_0) = \mathrm{bottom}(X, s)$ *and* $(X_1, s_1) = \mathrm{bottom}(X', s)$.

*Proof.* By Lemma 5.5.9, the *raise* invocations of the bottom-level auction instances with inputs $X$ and $subst(X, u, u')$ can be reordered such that at each round, either the same agent invokes *raise* in both executions, or agents $u$ and $u'$ invoke *raise* in their corresponding executions. By the definitions of the functions $raise'$ and $raise''$, the executions treat agents $u$ and $u'$ identically until both agents attain a utility of zero. By Lemma 5.4.2, we know that $u$ and $u'$ remain white in every round of their corresponding executions, and by Fact 5.3.2, we know that the potentials are nondecreasing over the rounds of both executions. Thus, agents $u$ and $u'$ continue to have zero utility for the remainder of the executions, and we have $gap(X_0, u) = gap(X_1, u')$. $\square$

**Lemma 6.1.2.** *Let $X$ be a quiescent ECC and let $X'$ be a quiescent ECC of the form* $\mathrm{subst}(X, u, u')$ *such that for any agent $u''$ in* $\mathrm{agents}(X)$*, we have $u'' < u$ if and only if $u'' < u'$. Let $s$ and $s'$ be targets such that $\alpha(u) = s'(u')$ and $\alpha(u'') = s'(u'')$ for any agent $u''$ in* $\mathrm{agents}(X) - u$*. If* $(X_0, s_0) = \mathrm{bottom}(X, s)$ *and* $(X_1, s_1) = \mathrm{bottom}(X', s')$*, then* $\mathrm{gap}(X_0, u) + s_0(u) = \mathrm{gap}(X_1, u') + s_1(u')$.

*Proof.* By Lemma 5.5.9, the *raise* invocations of the bottom-level auction instances with inputs $X$ and $\mathit{subst}(X, u, u')$ can be reordered such that at each round, either the same agent invokes *raise* in both executions, or agents $u$ and $u'$ invoke *raise* in their corresponding executions. Since agents $u$ and $u'$ have the same relative ordering with respect to the agents in $\mathit{agents}(X) - u$, it is easy to see that if $X_0$ and $X_0'$ are the output ECCs corresponding to the same round in both executions, then we have $X_0' = \mathit{subst}(X_0, u, u')$. □

**Lemma 6.1.3.** *For any quiescent ECC $X'$ of the form* $\mathrm{add}(X, u, v, z)$ *and any target $s$, there exists a unique integer $z^*$ and a unique agent $u^*$ in* $\mathrm{agents}(X) + \epsilon$ *such that $u$ belongs to* $\mathrm{matched}(X', s)$ *if and only if $(z + s(u), u) > (z^*, u^*)$. Moreover, if $u$ belongs to* $\mathrm{matched}(X', s)$*, then* $\mathrm{potential}(X'', v) = z^*$ *where* $(X'', s'') = \mathrm{bottom}(X', s)$.

*Proof.* Let $S$ be the ordered sequence of all pairs of the form $(z', u')$ where $z'$ is an integer and $u'$ is an agent that does not belong to $\mathit{agents}(X) \cup \epsilon$. Consider any pair $(z_0, u_0)$ in $S$ such that $z_0 + s_0(u) < \mathit{potential}(X_0, v)$, where $s_0 = \mathit{subst}(s, u_0, \alpha(u))$ and $(X_0, s_0') = \mathit{bottom}(\mathit{add}(X, u_0, v, z_0), s_0)$. By repeated application of Fact 5.3.2, we know that $\mathit{potential}(X_0) \geq \mathit{potential}(X)$ and by repeated application of Lemma 5.4.2, we have $u_0$ belongs to $\mathit{white}(X_0)$. Thus, $u_0$ does not belong to $\mathit{matched}(X_0)$. Further, since $u_0$ belongs to $\mathit{white}(X_0)$, it follows that $\mathit{potential}(X_0, v) \geq z_0$. Since prices cannot grow indefinitely, there must be a first pair $(z_1, u_1) > (z_0, u_0)$ in $S$ such that $u_1$ belongs to $\mathit{matched}(X_1, s_1)$ where $X_1 = \mathit{add}(X, u_1, v, z_1)$ and $s_1 = \mathit{subst}(s, u_1, \alpha(u))$.

Consider the pair $(z_1, u_2)$ where $u_2$ is the maximum agent such that $u_2 < u_1$. By Lemma 6.1.2, if $u_2$ does not belong to $\mathit{agents}(X) \cup \epsilon$, then $u_2$ and $u_1$ have the same relative ordering with respect to the remaining agents in $X$ and thus, $u_2$ belongs to $\mathit{matched}(\mathit{subst}(X_1, u_1, u_2), \mathit{subst}(s, u_2, \alpha(u)))$. However, we know that $(z_1, u_1)$ is the first pair in $S$ such that $u_1$ belongs to $\mathit{matched}(X_1, s_1)$. Thus, it follows that $u_2$ belongs to $\mathit{agents}(X) \cup \epsilon$. Consider any pair $(z_3, u_3) > (z_1, u_1)$ in $S$; by the definition of the bottom-level auction, we find that $u_3$ belongs to $\mathit{matched}(\mathit{subst}(X_1, u_1, u_3), \mathit{subst}(s, u_3, \alpha(u)))$. Thus $u^* = u_2$ and $z^* = z_1$.

We now show that if $u$ belongs to $\mathit{matched}(X', s)$, then $\mathit{potential}(X'', v) = z^*$ where $(X'', s'') = \mathit{bottom}(X', s)$. Suppose that $\mathit{potential}(X'', v) < z^*$. Then consider the case where $u < u^*$ and $z+\alpha(u) = z^*$. Since $(z+\alpha(u), u) < (z^*, u^*)$, we have $u$ belongs to $\mathit{unmatched}(X'')$ and since $z+\alpha(u) > \mathit{potential}(X'', v)$, we have $u$ belongs to $\mathit{matched}(X'')$; a contradiction. Suppose that $\mathit{potential}(X'', v) > z^*$. Then consider the case where $u > u^*$ and $z + \alpha(u) = z^*$. Since $(z + \alpha(u) + z, u) > (z^*, u^*)$, we have $u$ belongs to $\mathit{matched}(X'')$ and since $z + \alpha(u) < \mathit{potential}(X'', v)$, we have $u$ belongs to $\mathit{unmatched}(X'')$; a contradiction. It follows that $\mathit{potential}(X'', v) = z^*$. □

[Definitions] For any quiescent ECC $X$, any target $s$, and any item $v$ in $\mathit{items}(X)$, we define $\mathit{threshold}^*(X, s, v)$ as the unique pair $(z^*, u^*)$ of Lemma 6.1.3, and we define $\mathit{threshold}^*(X, s)$ as the function that maps each item $v$ in $\mathit{items}(X)$ to $\mathit{threshold}^*(X, s, v)$. In addition, we define $\mathit{threshold}(X, s, v)$ as the integer $z^*$ and we define $\mathit{threshold}(X, s)$ as the function that maps each item $v$ in *items*$(X)$ to *threshold*$(X, s, v)$.

**Lemma 6.1.4.** *For any quiescent ECC $X$, any target $s$, and any agent $u$ in* enabled$(X)$*, we have*

$$\text{threshold}^*(X, s) \leq \text{threshold}^*(\text{raise}(X, u), s).$$

*Proof.* Assume $\textit{threshold}^*(\textit{raise}(X, u), s, v_0) < \textit{threshold}^*(X, s, v_0)$ for some item $v_0$ in *items*$(X)$. Let $X_0$ be an ECC of the form $\textit{add}(X, u_0, v_0, \textit{min}(v_0))$ and let $s_0$ be a target such that

1. $s_0(u') = \alpha(u')$ for any agent $u'$ in *agents*$(X)$, and
2. $\textit{threshold}^*(\textit{raise}(X, u), s, v_0) < s_0(u_0) + \textit{min}(v_0) < \textit{threshold}^*(X, s, v_0)$

Note that $X_0$ is quiescent. We have $\textit{threshold}^*(X, s) = \textit{threshold}^*(X, s_0)$, and $\textit{threshold}^*(\textit{raise}(X, u), s) = \textit{threshold}^*(\textit{raise}(X, u), s_0)$; thus,

$$\textit{threshold}^*(\textit{raise}(X, u), s_0, v_0) < s_0(u_0) + \textit{min}(v_0) < \textit{threshold}^*(X, s_0, v_0)$$

Let $(X_1, s_1) = \textit{bottom}(X_0, s_0)$. Since $s_0(u_0) + \textit{min}(v_0)$ is less than $\textit{threshold}^*(X, s_0, v_0)$, by Lemma 6.1.3 we find that $u_0$ belongs to *unmatched*$(X_1)$. Since $X_0$ is quiescent and $u$ belongs to *unmatched*$(X_0)$, we have $u$ belongs to *white*$(X_0)$, and by repeated application of Lemma 5.4.2, we find that $u_0$ belongs to *white*$(X_1)$. We conclude that $s_1(u_0) = 0$.

By Fact 6.1.1, $\textit{bottom}(\textit{raise}(X_0, u), s_0) = \textit{bottom}(X_0, \textit{shift}(s_0, u, 1))$, and by Fact 6.1.3, $\textit{bottom}(X_0, \textit{shift}(s_0, u, 1)) = \textit{bottom}(X_1, \textit{shift}(s_1, u, 1))$. Since

$\mathit{threshold}^*(\mathit{raise}(X,u),s_0,v_0) < s_0(u_0)+\mathit{min}(v_0)$, by Lemma 6.1.3, we find that $u_0$ belongs to $\mathit{matched}(\mathit{raise}(X_0,u),s_0)$, and since we established above that $s_1(u_0)=0$, we have $u_0$ does not belong to $\mathit{matched}(X_1,\mathit{shift}(s_1,u,1))$. Since $\mathit{bottom}(\mathit{raise}(X_0,u),s_0) = \mathit{bottom}(X_1,\mathit{shift}(s_1,u,1))$, this yields a contradiction. Thus, we have $\mathit{threshold}^*(X,\mathit{shift}(s,u,1)) \leq \mathit{threshold}^*(\mathit{raise}(X,u),s)$. □

**Lemma 6.1.5.** *For any quiescent ECC $X_0$ of the form* $\mathrm{add}(X,u,\beta)$ *and any target $s$, if $u$ does not belong to* $\mathrm{matched}(X_0,s)$*, then* $\mathrm{threshold}^*(X_0,s) = \mathrm{threshold}^*(X,s)$.

*Proof.* Let $(X',s') = \mathit{bottom}(X,s)$ and let $(X_0',s_0') = \mathit{bottom}(X_0,s)$. By Fact 6.1.2, we have $\mathit{bottom}(X_0,s) = \mathit{bottom}(\mathit{add}(X',u,\beta),s')$. By repeated application of Lemma 6.1.4, it follows that $\mathit{threshold}^*(X_0',s_0') \geq \mathit{threshold}^*(X',s')$. Suppose $\mathit{threshold}^*(X',s',v_1) < \mathit{threshold}^*(X_0',s_0',v_1)$ for some $v_1$ in $\mathit{items}(X)$. Let $X_1 = \mathit{add}(X_0,u_1,v_1,\mathit{min}(v_1))$ for some agent $u_1$, and let $s_1$ be a target such that

1. $\mathit{threshold}^*(X',s',v_1) < (s_1(u_1)+\mathit{min}(v_1),u_1) < \mathit{threshold}^*(X_0',s_0',v_1)$, and
2. $s_1(u') = \alpha(u')$ for any $u'$ in $\mathit{agents}(X_0)$. Note that $\mathit{threshold}^*(X_0,s_0) = \mathit{threshold}^*(X_0,s_1)$

Similarly, $\mathit{threshold}^*(X_1,s_0) = \mathit{threshold}^*(X_1,s_1)$.

Let $X_2 = \mathit{add}(X, u_1, v_1, \mathit{min}(v_1))$; then it follows from above that $X_1 = \mathit{add}(X_2, u, \beta)$. Let $(X_2', s_2') = \mathit{bottom}(X_2, s_1)$; by Fact 6.1.2, we find that $\mathit{bottom}(X_1, s_1) = \mathit{bottom}(\mathit{add}(X_2', u, \beta), s_2')$. By Lemma 6.1.3, since $\mathit{min}(v_0) + s_1(u_1) > \mathit{threshold}^*(X', s', v_1)$ and $\mathit{threshold}^*(X_0', s_0') \geq \mathit{threshold}^*(X', s')$, $u_0$ is in $\mathit{matched}(X_2', s_2')$. By Lemma 6.1.4, $\mathit{threshold}^*(X_2', s_2') \geq \mathit{threshold}^*(X, s_1)$, and since $u$ is not in $\mathit{matched}(X_0, s_1)$, agent $u$ does not belong to $\mathit{matched}(X_1, s_1)$; thus $u_1$ belongs to $\mathit{matched}(X_1, s_1)$.

Since $u$ belongs to $\mathit{enabled}(X_1)$, by Fact 6.1.2, it follows that

$$\mathit{bottom}(\mathit{add}(X_0', u_1, v_1, \mathit{min}(v_1)), s_1') = \mathit{bottom}(X_1, s_1)$$

By Lemma 6.1.3, since $\mathit{min}(v_0) + s_0(u_0) < \mathit{threshold}^*(X_0', s_0', v_1)$, agent $u_0$ does not belong to $\mathit{matched}(X_0', s_1')$; thus $u$ and $u_1$ do not belong to $\mathit{matched}(X_1, s_1)$, a contradiction. $\square$

[Definitions] For any quiescent ECC $X$, any target $s$, and any item $v$ in $\mathit{items}(X)$, we define $\mathit{price}(X, s, v)$ as $\mathit{potential}(X', v)$ where $(X', s') = \mathit{bottom}(X, s)$, and we define $\mathit{price}(X, s)$ as the function that maps every item $v$ in $\mathit{items}(X)$ to $\mathit{price}(X, s, v)$.

[Definitions] For any quiescent ECC $X$, any target $s$, and any item $v$ in $\mathit{items}(X)$, we define $\mathit{price}^*(X, s, v)$ as $(\mathit{price}(X, s, v), u_0)$, where $u_0$ is the maximum agent in $\mathit{agents}(X, s, v)$. In addition, we define $\mathit{price}^*(X, s)$ as the function that maps each item $v$ in $\mathit{items}(X)$ to $\mathit{price}^*(X, s, v)$.

**Lemma 6.1.6.** *For any quiescent ECC $X$ and any target $s$, we have*

$$\text{price}^*(X, s) \leq \text{threshold}^*(X, s)$$

*Proof.* Assume that there exists an item $v$ in $\mathit{items}(X)$ such that $\mathit{price}^*(X, s, v) > \mathit{threshold}^*(X, s, v)$. Let $(X', s') = \mathit{bottom}(X, s)$. Let $X_0 = \mathit{add}(X', u, v, \mathit{min}(v))$ for some agent $u$, and let $s_0$ be a target such that

1. $\mathit{threshold}^*(X, s, v) < s_0(u) + \mathit{min}(v) < \mathit{price}^*(X, s, v)$, and

2. $s_0(u') = s'(u')$ for any agent $u'$ in $\mathit{agents}(X)$.

Note that $X_0$ is quiescent. It is easy to see that $\mathit{threshold}^*(X, s) = \mathit{threshold}^*(X', s') = \mathit{threshold}^*(X', s_0)$; thus $\mathit{threshold}^*(X', s_0, v) < s_0(u) + \mathit{min}(v)$.

We have $\mathit{bottom}(X_0, s_0) = \mathit{bottom}(\mathit{add}(X', u, v, \mathit{min}(v)), \mathit{subst}(s', u, s_0(u)))$. Thus, by repeated application of Fact 5.3.2, we have

$$\mathit{price}^*(X_0, s_0) \geq \mathit{price}^*(X', s') \geq \mathit{price}^*(X, s)$$

Thus, $\mathit{threshold}^*(X', s_0, v) < s_0(u) + \mathit{min}(v) < \mathit{price}^*(X_0, s_0, v)$.

Let $(X'_0, s'_0) = \mathit{bottom}(X_0, s_0)$. By Lemma 6.1.3, since $s_0(u) + \mathit{min}(v) > \mathit{threshold}^*(X', s_0, v)$ we find that $u$ belongs to $\mathit{matched}(X'_0)$. Since $u$ belongs to $\mathit{unmatched}(X_0)$, we find that $u$ belongs to $\mathit{white}(X_0)$; thus, by repeated application of Lemma 5.4.2, we have $u$ belongs to $\mathit{white}(X'_0)$. However, since $s_0(u) + \mathit{min}(v) < \mathit{price}^*(X, s, v)$, it follows that $u$ belongs to $\mathit{nonwhite}(X'_0)$, thus yielding a contradiction. Thus, $\mathit{price}^*(X, s) \leq \mathit{threshold}^*(X, s)$. $\square$

**Lemma 6.1.7.** *Let $X_0$ be a quiescent ECC. Let $u_0$ be an agent in* $\mathrm{unmatched}(X_0)$ *and let $X_1$ be a quiescent ECC of the form* $\mathrm{subst}(X_0, u_0, u_1)$*, where $u_1 < u_0$. Then for any target $s$ such that $\alpha(u_0) = \alpha(u_1)$, we have* $\mathrm{gap}(X'_0, u_0) = \mathrm{gap}(X'_1, u_1)$*, where* $(X'_0, s'_0) = \mathrm{bottom}(X_0, s)$ *and* $(X'_1, s'_1) = \mathrm{bottom}(X_1, s)$.

*Proof.* Let $\beta = \mathit{bid}(\mathit{bid\text{-}graph}(X_0), u)$ and let $X$ be the ECC such that $X_0 = \mathit{add}(X, u, \beta)$. Observe that $X$ and $X_1$ are quiescent. Let $(X', s') = \mathit{bottom}(X, s)$. By Fact 6.1.2, we have $\mathit{bottom}(X_0, s) = \mathit{bottom}(\mathit{add}(X', u_0, \beta), s')$ and we have $\mathit{bottom}(X_1, s) = \mathit{bottom}(\mathit{subst}(X_0, u_0, u_1), s')$. We refer to the instance of the bottom-level auction with inputs $X_0$ and $s$ as execution $A$ and we refer to the instance of the bottom-level auction with inputs $X_1$ and $s$ as execution $B$. By Lemma 5.5.9, *raise* invocations of executions $A$ and $B$ can be reordered such that agents $u_0$ and $u_1$ exhaust their *raise* invocations before any other agent invokes the function *raise*. If $u_0$ and $u_1$ are unmatched when they exhaust their *raise* invocations, then by the description of the bottom-level auction, agents $u_0$ and $u_1$ have zero utility in executions $A$ and $B$ respectively, and they continue to have zero utility for the rest of the corresponding executions; thus $\mathit{gap}(X'_0, u_0) = \mathit{gap}(X'_1, u_1) = 0$.

For the remainder of this proof, we may assume that consider the following cases at least one of agents $u_0$ and $u_1$ is matched by a *raise* invocation in either execution $A$ or execution $B$. Let $k$ be the first round in which either $u_0$ or $u_1$ is matched and let $X_k$ and $X'_k$ be the output ECCs of round $k$ of executions $A$ and $B$. By repeated application of Lemma 5.4.5, we have $\mathit{gap}(X_k, u_0) = \mathit{gap}(X'_k, u_1) = 0$, and either $X_k = \mathit{subst}(X'_k, u_1, u_0)$, or $\mathit{raise}(X_k, u_0) = \mathit{raise}(X'_k, u_1)$.

First we consider the case where $X_k = \mathit{subst}(X'_k, u_1, u_0)$. In this case, $u_0$ belongs to $\mathit{matched}(X_k) \cap \mathit{white}(X_k)$ and $u_1$ belongs to $\mathit{matched}(X'_k) \cap \mathit{white}(X'_k)$; thus, by Lemma 6.1.1 we have $\mathit{gap}(X'_0, u_0) = \mathit{gap}(X'_1, u_1)$.

Next we consider the case where $X_k \neq \mathit{subst}(X'_k, u_1, u_0)$. If agents $u_0$ and $u_1$ have exhausted their *raise* invocations, then by the description of the bottom-level auction, they continue to have zero utility for the rest of the auction; if $u_0$ and $u_1$ have one or more pending *raise* invocations, then by Lemma 5.4.5, $\mathit{raise}(X_k, u_0) = \mathit{subst}(\mathit{raise}(X'_k, u_0), u_1, u_0)$, and by Lemma 6.1.1, we have $\mathit{gap}(X'_0, u_0) = \mathit{gap}(X'_1, u_1)$. □

**Lemma 6.1.8.** *Let $X$ be a quiescent ECC of the form* $\mathrm{add}(X_0, u, \beta)$ *and for each item $v$ in* $\mathrm{items}(X)$*, let* $X_v = \mathrm{add}(X_0, u, v, z)$ *where* $z = \beta(v)$*. Then for any target $s$, agent $u$ belongs to* $\mathrm{matched}(X, s)$ *if and only if $u$ belongs to* $\mathrm{matched}(X_v, s)$ *for some item $v$ in* $\mathrm{items}(X)$*.*

*Proof.* We refer to the bottom-level auction instance with inputs $(X, s)$ as execution $A$, and for each item $v$, we refer to the bottom-level auction instance with input $(X_v, s)$ as execution $A_v$. We represent the output of round $i$ of execution $A$ by $(X_i, s_i)$, and for any $v$ in $V$, we represent the output of round $i$ of execution $A_v$ by $(X_{v,i}, s_{v,i})$. Note that agent $u$ is unmatched and therefore enabled in all rounds of all executions under consideration. By Lemma 5.5.9, we choose to defer the *raise* invocations of agent $u$ in each execution to a round $j$ in which $u$ is the only enabled agent. Further, we choose to allow the same agent to invoke *raise* in each round of every execution.

We now allow agent $u$ to exhaust its *raise* invocations in rounds $j$ to $k$ of all executions, where $k = j + \alpha(u)$. We consider the following two cases.

- Case (1) : $(\beta(v) + \alpha(u), u) < \textit{threshold}^*(X, \alpha(u), v)$ for every item $v$ in $\textit{items}(X)$.

  By Lemma 6.1.3, since $(\beta(v)+\alpha(u), u) < \textit{threshold}^*(X, \alpha(u), v)$ for every item $v$ in $\textit{items}(X)$, we find that $u$ does not belong to $\textit{matched}(X_v, s)$ and thus $u$ belongs to $\textit{unmatched}(X_{v,k})$ for every item $v$ in $\textit{items}(X)$. Assume that $u$ belongs to $\textit{matched}(X, s)$; thus, $u$ belongs to $\textit{matched}(X_k)$. Let $s'$ be a target such that $s'(u') = s_k(u')$ for any $u'$ in $\textit{agents}(X_k)$ and for any agent of the form $u_v$ where $v$ is an item in $\textit{items}(X)$, we have $(\beta(v) + \alpha(u), u) < (s'(u_v) + \textit{min}(v), u_v) < \textit{threshold}^*(X, s, v)$. By Lemma 6.1.4, we have $\textit{threshold}^*(X, s) \leq \textit{threshold}^*(X_k, s')$; thus, we have $(\beta(v) + \alpha(u), u) < (s'(u_v) + \textit{min}(v), u_v) < \textit{threshold}^*(X_k, s', v)$ for any item $v$ in $\textit{items}(X)$.

  Let $X'$ be an ECC that is constructed from $X_k$ as follows: initialize $X' = X_k$, and for each item $v$ in $\textit{items}(X)$, set $X' = \textit{add}(X', u_v, v, \textit{min}(v))$. Consider the execution $A'$ of the bottom-level auction with input $(X', s')$, and for any round $i$ of execution $A'$, let $(X'_i, s'_i)$ represent the output of round $i$ of execution $A'$. We now use Lemma 5.5.9, to allow all agents in $\cup_{v \in \textit{items}(X)} u_v$ to exhaust their *raise* invocations. If $m$ is the last round of the *raise* invocations by agents in $\cup_{v \in \textit{items}(X)} u_v$, then by Lemmas 6.1.3, since $(s'(u_v)+\textit{min}(v), u_v) < \textit{threshold}^*(X_k, s', v)$ for every item $v$, we find that agent $u_v$ belongs to $\textit{unmatched}(X'_m)$ for every $v$ in $\textit{items}(X)$, and by Lemma 6.1.5, we have $\textit{threshold}^*(X'_m, s'_m) = \textit{threshold}^*(X_k, s_k)$. Since every agent $u_v$ belongs to $\textit{unmatched}(X'_m)$ and $X'_m$ is quiescent, we have *potential*$(X'_m, v) \geq (s'(u_v) + \mathit{min}(v), u_v)$ for every item $v$ in *items*$(X)$; thus by Fact 5.3.2, we have *price*$^*(X', s', v) \geq (s'(u_v) + \mathit{min}(v), u_v)$ for every item $v$. Since $(\beta(v) + \alpha(u), u) < (s'(u_v) + \mathit{min}(v), u_v)$ for every item $v$, we have *price*$^*(X', s', v) \geq (\beta(v) + \alpha(u), u)$ for every item $v$.

However, by repeated use of Lemma 5.4.2, agent $u$ is white at the end of execution $A'$, and by our assumption that $u$ belongs to *matched*$(X, s)$, we have *price*$^*(X', s', v) < (\beta(v) + \alpha(u), u)$ for some item $v$; this yields a contradiction. Thus, we have $u$ does not belong to *matched*$(X, s)$.

- Case (2) : $(\beta(v) + \alpha(u), u) >$ *threshold*$^*(X, s, v)$ for some item $v$ in *items*$(X)$.

  By Lemma 6.1.3, since $(\beta(v) + \alpha(u), u) >$ *threshold*$^*(X, s, v)$ for some item $v$ in *items*$(X)$, we find that $u$ belongs to *matched*$(X_v, s)$ for some item $v$ in *items*$(X)$. Assume that $u$ does not belong to *matched*$(X, s)$. Consider the execution $A'$ defined as in Case 1 above. By Lemma 6.1.5, we have *threshold*$^*(X_k, s_k) =$ *threshold*$^*(X'_m, s'_m)$. By Lemma 6.1.6, we have *price*$^*(X'_m, s'_m) \leq$ *threshold*$^*(X_k, s_k)$. Thus, there exists some item $v$ in *items*$(X)$ such that $u$ belongs to *unmatched*$(X'_m)$ and $(\beta(v) + \alpha(u), u) >$ *price*$^*(X'_m, s'_m, v)$; this violates the quiescent property of $X'_m$. Thus, $u$ belongs to *matched*$(X, s)$.

We conclude that agent $u$ belongs to *matched*$(X, s)$ if and only if $u$ belongs to *matched*$(X_v, s)$ for some item $v$ in *items*$(X)$, as required. □

**Lemma 6.1.9.** *Let $X$ be a quiescent ECC of the form* $\mathrm{add}(X_0, u, \beta)$*, let $s$ be a target, and for each item $v$ in* $\mathrm{items}(X)$*, let* $X_v = \mathrm{add}(X_0, u, v, z)$*, where* $z = \beta(v)$*. Then, we have* $\mathrm{gap}(X', u) + s'(u) = \max_{v \in \mathrm{items}(X)}\{\mathrm{gap}(X'_v, u) + s_{v'}(u)\}$*, where* $(X', s') = \mathrm{bottom}(X, s)$ *and* $(X'_v, s'_v) = \mathrm{bottom}(X_v, s)$ *for each item $v$ in* $\mathrm{items}(X)$*.*

*Proof.* By Lemma 6.1.7, if $(X^*, s^*) = \mathit{bottom}(\mathit{add}(X_0, u', \beta), s)$ for any agent $u'$, then $\mathit{gap}(X', u) = \mathit{gap}(X^*, u')$. Thus, without loss of generality, we can assume that $u > u'$ for any agent $u'$ in $\mathit{agents}(X)$. By Lemma 6.1.8, $u$ belongs to $\mathit{matched}(X')$ if and only if $u$ belongs to $\mathit{matched}(X'_v)$ for some $v$ in $\mathit{items}(X_0)$. Thus, if $u$ belongs to $\mathit{unmatched}(X')$, we have $\mathit{gap}(X', u) = \mathit{gap}(X'_v, u) = 0$ for all $v$ in $\mathit{items}(X)$.

We now focus on the case where $u$ belongs to $\mathit{matched}(X')$. Let $z$ be the largest integer such that $u$ belongs to $\mathit{matched}(\mathit{shift}(X', u, -z))$. By Fact 6.1.3, we have $(X', s') = (X'', \mathit{shift}(s'', u, z))$ where $(X'', s'') = \mathit{bottom}(\mathit{shift}(X, u, -z), s)$; thus $\mathit{gap}(X', u) + s'(u) = \mathit{gap}(X'', u) + s''(u) + z$. By Lemma 6.1.8, agent $u$ belongs to $\mathit{matched}(X''_v)$ for some item $v$ in $\mathit{items}(X)$, where $(X''_v, s''_v) = \mathit{bottom}(\mathit{shift}(X_v, u, -z), s)$.

By Fact 6.1.3, we have $(X'_v, s'_v) = \mathit{bottom}(X''_v, \mathit{shift}(s'_v, u, z))$; thus we have $\mathit{gap}(X'_v, u) + s'_v = \mathit{gap}(X''_v, u) + s''_v(u) + z$. Since $u$ belongs to $\mathit{white}(X) \cap \mathit{white}(X_v)$, we have $s''(u) = s''_v(u) = 0$. To complete the proof, it remains to be shown that $\mathit{gap}(X''_v, u) = \mathit{gap}(X''_v, u)$.

Let the bottom-level auction instance with input $(\mathit{shift}(X, u, -z), s)$ as execution $A$, and for each item $v$, let the bottom-level auction instance with input $(\mathit{shift}(X_v, u, -z), s)$ as execution $A_v$. We represent the output of round $i$ of execution $A$ by $(X_i, s_i)$, and for any $v$ in $V$, we represent the output of round $i$ of execution $A_v$ by $(X_{v,i}, s_{v,i})$. Since $u$ belongs to $\mathit{unmatched}(X)$, it follows that $u$ belongs to $\mathit{enabled}(X) \cap \mathit{enabled}(X_v)$. By Lemma 5.5.9, we choose to allow agent $u$ to first exhaust its *raise* invocations in all executions. Since $u > u'$ for any agent $u'$ in $\mathit{agents}(X)$, it follows that for each round $j$ in which $u$ invokes *raise*, either $u$ belongs to $\mathit{unmatched}(X_j) \cap \mathit{unmatched}(X_{v,j})$ or there exists a first round $j$ such that $u$ belongs to $\mathit{matched}(X'_j)$ and $u$ belongs to $\mathit{matched}(X'_{v,j})$ for some item $v$ in $\mathit{items}(X)$. Since $z$ was chosen to be the largest integer such that $u$ belongs to $\mathit{matched}(\mathit{shift}(X', u, -z))$, we have $\mathit{gap}(X_j, u) = \mathit{gap}(X_{v,j}, u) = 0$, and thus $\mathit{gap}(X'', u) = \mathit{gap}(X''_v, u) = 0$. $\Box$

**Lemma 6.1.10.** *Let $X'$ be a quiescent ECC of the form* $\mathrm{add}(X, u, \beta)$ *and let* $(X'', s'') = \mathrm{bottom}(X', s)$ *for some target $s$. Let $\Delta$ denote the maximum, over all items $v$ in* $\mathrm{items}(X)$*, of* $\beta(v) + \alpha(u) - \mathrm{threshold}(X, s, v)$*, and let $V$ denote the set of all items $v$ in* $\mathrm{items}(X)$ *such that* $\beta(v) + \alpha(u) - \mathrm{threshold}(X, s, v) = \Delta$*. Let $u_0$ denote the minimum, over all items $v$ in $V$ of the second component of the pair given by* $\mathrm{threshold}^*(X, s, v)$*. Then the following conditions hold:*

- *If the pair $(\Delta, u) < (0, u_0)$, then agent $u$ belongs to* $\mathrm{unmatched}(X'')$*, and* $\mathrm{threshold}^*(X', s) = \mathrm{threshold}^*(X, s)$*.*
- *If the pair $(\Delta, u) > (0, u_0)$, then agent $u$ belongs to* $\mathrm{matched}(X'')$ *and,* (1) *for every configuration $\chi$ in $X''$, there exists an item $v$ in $V$ such*

*that* $\text{match}(\chi, v) = u$, *and* (2) $\text{potential}(X'', v) = \text{threshold}(X, s, v)$ *for any item* $v$ *in* $V$.

*Proof.* First, we show that $u$ belongs to $\mathit{matched}(X'')$ if and only if $(\Delta, u) > (0, u_0)$.

Let $v_0$ be any item in $\mathit{items}(X)$; thus, we find that $\beta(v_0) + \alpha(u) - \mathit{threshold}(X, s, v_0) = \Delta$. If $(\Delta, u) < (0, u_0)$, then by adding $\mathit{threshold}(X, s, v_0)$ to the first component of both pairs, we find that

$$(\beta(v_0) + \alpha(u), u) < (\mathit{threshold}(X, s, v_0), u_0)$$

Similarly, if $(\Delta, u) > (0, u_0)$, we have $(\beta(v_0)+\alpha(u), u) > (\mathit{threshold}(X, s, v_0), u_0)$. By Lemma 6.1.3, it follows that agent $u$ belongs to $\mathit{matched}(\mathit{add}(X, u, v_0, \beta(v_0)), s)$ if and only if $(\beta(v_0) + \alpha(u), u) > (\mathit{threshold}(X, s, v_0), u_0)$. Additionally, by Lemma 6.1.8, we find that agent $u$ belongs to $\mathit{matched}(X'') = \mathit{matched}(X', s)$ if and only if there exists some item $v'$ in $\mathit{items}(X)$ such that agent $u$ belongs to $\mathit{matched}(\mathit{add}(X, u, v', \beta(v')), s)$; thus, we find that agent $u$ belongs to $\mathit{matched}(X'')$ if and only if $(\Delta, u) > (0, u_0)$.

Next we show that if agent $u$ does not belongs to $\mathit{matched}(X'')$, then $\mathit{threshold}^*(X', s) = \mathit{threshold}^*(X, s)$. The result follows directly from Lemma 6.1.5.

Finally, we show that if agent $u$ belongs to $\mathit{matched}(X'')$, then (1) and (2) stated above hold. By Lemma 6.1.9, if agent $u$ belongs to $\mathit{matched}(X'')$, then we have $\mathit{gap}(X'', u) = \mathit{max}_{v \in \mathit{items}(X)}\{\mathit{gap}(X'_v, u)\}$, where $X'_v$ is equal to $\mathit{bottom}(\mathit{add}(X, u, v, \beta(v)), s)$, and by Lemma 6.1.3, it follows that $\Delta =$

$max_{v \in items(X)}\{gap(X'_v, u)\}$; thus $gap(X'', u) = \Delta$. Let $v_0$ be any item in $V$. By Lemma 6.1.6, we have $potential(X'', v_0) \leq threshold(X, s, v_0)$, and since $gap(X'', u) = \Delta$, we have $potential(X'', v_0) \geq threshold(X, s, v_0)$; thus $potential(X'', v_0) = threshold(X, s, v_0)$ and condition (1) holds. Now consider any item $v$ not in $V$. By definition, we have

$$\beta(v) + \alpha(u) - threshold^*(X, s, v) < \Delta. \tag{3}$$

By Lemma 6.1.6, we have

$$potential(X'', v) \leq threshold^*(X, s, v). \tag{4}$$

By subtracting (4) from (3), we have $\beta(v) + \alpha(u) - potential(X'', v) < \Delta$; since $u$ belongs to $white(X'')$, agent $u$ attains its highest utility by being matched to some item in $V$ in every configuration of $X''$ and condition (2) holds. □

## 6.2 A Determinized *Raise* Operation

We have been dealing with ECCs in the discussion of the bottom-level auction in the current and previous chapters. The inputs and outputs of our top-level auction are configurations; accordingly, we define a suitable determinization of the bottom-level auction.

For any quiescent configuration $\chi$ and any agent $u$ in $enabled(\chi)$, we would like to define $raise(\chi, u)$ as a specific configuration in $raise([\chi], u)$ such that for any agent $u_0$ in $enabled(\chi)$, if $u$ does not belong to $matched(\chi) \cap unmatched(raise(\chi, u_0))$, then $raise(raise(\chi, u_0), u) = raise(raise(\chi, u), u_0)$.

In order to do so, we determinize the choice of the augmenting path in function *augment* defined in Section 5.2. Specifically, we pick a lexicographically first (with respect to item identifiers) shortest path.

## 6.3 Description

The input to the mid-level auction is a pair $(\chi, s)$, where $\chi$ is a quiescent configuration and $s$ is a target.

We view the bottom-level auction instance as taking the pair $(\chi, s)$ as input, and updating this pair over a sequence of rounds. A general round of the bottom-level auction with input $(\chi_0, s_0)$ is defined as follows: if $\mathit{enabled}(\chi_0) = \emptyset$, then the auction terminates; if the minimum agent in $\mathit{matched}(\chi_0) \cap \mathit{enabled}(\chi_0) = \epsilon$ then the minimum agent in $\mathit{enabled}(\chi_0)$ invokes *raise*; otherwise, the minimum agent in $\mathit{matched}(\chi_0) \cap \mathit{enabled}(\chi_0)$ invokes *raise*.

The output of the mid-level auction is defined to be $\mathit{bottom}(\chi, s)$, the output of the bottom-level auction when given the pair $(\chi, s)$ as input.

Below we establish some key lemmas relating to the mid-level auction.

**Lemma 6.3.1.** *For any configuration $\chi'$ of the form* $\text{raise}(\chi, u')$*, and any agent $u$ in* $\text{nonwhite}(\chi)$*, either* (1) *$u$ belongs to* $\text{unmatched}(\chi')$*, or* (2) *$u$ belongs to* $\text{nonwhite}(\chi')$*, and there exists an item $v$ in* $\text{items}(\chi)$ *such that* $\text{potential}(\chi, v) = \text{potential}(\chi', v)$ *and* $\text{match}(\chi, v) = \text{match}(\chi', v) = u$.

*Proof.* Let $X' = \mathit{raise}([\chi], u')$. By definition, $\chi'$ belongs to $X'$. The result follows from Lemma 5.4.1. □

**Lemma 6.3.2.** *For any quiescent configuration* $\chi$ *and any agent* $u$ *in* $\text{enabled}(\chi)$, *if* $\chi' = \text{raise}(\chi, u)$, *then*

$$\text{gray}(\chi) \subseteq \text{nonblack}(\chi') \wedge \text{white}(\chi) \subseteq \text{white}(\chi').$$

*Proof.* Let $X' = \mathit{raise}([\chi], u)$. By definition, $\chi'$ belongs to $X'$. The result follows from Lemma 5.4.2. □

**Lemma 6.3.3.** *For any quiescent configuration* $\chi$ *and any agent* $u$ *in* $\text{enabled}(\chi)$, *we have* $\text{enabled}(\chi) - u \subseteq \text{enabled}(\text{raise}(\chi, u))$.

*Proof.* Let $X' = \mathit{raise}([\chi], u)$. By definition, $\mathit{raise}(\chi, u)$ belongs to $X'$. The result follows from Lemma 5.4.3. □

**Lemma 6.3.4.** *For any quiescent configuration* $\chi$, *any quiescent configuration* $\chi'$ *of the form* $\text{subst}(\chi, u, u')$, *and any target* $s$ *such that* $\alpha(u) = \alpha(u')$, *if* $u$ *belongs to* $\text{matched}(\chi) \cap \text{white}(\chi)$, *then* $\text{gap}(\chi_0, u) = \text{gap}(\chi_1, u')$, *where* $(\chi_0, s_0) = \text{bottom}(\chi, s)$ *and* $(\chi_1, s_1) = \text{bottom}(\chi', s)$.

*Proof.* Let $X = [\chi]$ and let $X' = [\chi']$. Let $(X_0, s_0) = \mathit{bottom}(X, s)$ and let $(X_1, s_1) = \mathit{bottom}(X', s)$. By definition, $\chi_0$ belongs to $X_0$ and $\chi_1$ belongs to $X_1$. The result follows from Lemma 6.1.1. □

**Lemma 6.3.5.** *Let* $\chi$ *be a quiescent configuration and let* $\chi'$ *be a quiescent configuration of the form* $\text{subst}(\chi, u, u')$ *such that for any agent* $u''$ *in* $\text{agents}(\chi)$, *we have* $u'' < u$ *if and only if* $u'' < u'$. *Let* $s$ *and* $s'$ *be targets such that* $\alpha(u) = s'(u')$ *and* $\alpha(u'') = s'(u'')$ *for any agent* $u''$ *in* $\text{agents}(\chi) - u$. *If*

$(\chi_0, s_0) = \text{bottom}(\chi, s)$ *and* $(\chi_1, s_1) = \text{bottom}(\chi', s')$*, then* $\text{gap}(\chi_0, u) + s_0(u) = \text{gap}(\chi_1, u') + s_1(u')$.

*Proof.* Let $X = [\chi]$ and let $X' = [\chi']$. Let $(X_0, s_0) = \mathit{bottom}(X, s)$ and let $(X_1, s_1) = \mathit{bottom}(X', s')$. By definition, $\chi_0$ belongs to $X_0$ and $\chi_1$ belongs to $X_1$. The result follows from Lemma 6.1.2. □

**Lemma 6.3.6.** *Let* $\chi'$ *be a quiescent configuration of the form* $\text{add}(\chi, u, \beta)$ *and let* $(\chi'', s'') = \text{bottom}(\chi', s)$ *for some target* $s$*. Let* $\Delta$ *denote the maximum, over all items* $v$ *in* $\text{items}(\chi)$*, of* $\beta(v) + \alpha(u) - \text{threshold}(\chi, s, v)$*, and let* $V$ *denote the set of all items* $v$ *in* $\text{items}(\chi)$ *such that* $\beta(v) + \alpha(u) - \text{threshold}(\chi, s, v) = \Delta$*. Let* $u_0$ *denote the minimum, over all items* $v$ *in* $V$ *of the second component of the pair given by* $\text{threshold}^*(\chi, s, v)$*. Then the following conditions hold:*

- *If the pair* $(\Delta, u) < (0, u_0)$*, then agent* $u$ *belongs to* $\text{unmatched}(\chi'')$*, and* $\text{threshold}^*(\chi', s) = \text{threshold}^*(\chi, s)$.
- *If the pair* $(\Delta, u) > (0, u_0)$*, then agent* $u$ *belongs to* $\text{matched}(\chi'')$ *and,* (1) *for every configuration* $\chi$ *in* $\chi''$*, there exists an item* $v$ *in* $V$ *such that* $\text{match}(\chi, v) = u$*, and* (2) $\text{potential}(\chi'', v) = \text{threshold}(\chi, s, v)$ *for any item* $v$ *in* $V$.

*Proof.* Let $X = [\chi]$ and let $X' = [\chi']$. Let $(X'', s'') = \mathit{bottom}(X', s)$. By definition, $\chi''$ belongs to $X''$. The result follows from Lemma 6.1.10. □

**Lemma 6.3.7.** *For any quiescent configuration $\chi$ and any agents $u_0$ and $u_1$ in* enabled($\chi$)*, we have*

$$\text{raise}(\text{raise}(\chi, u_0), u_1) = \text{raise}(\text{raise}(\chi, u_1), u_0).$$

*Proof.* By the definition of the function *raise* that takes an ECC as an argument, either $u_0$ does not belong to $\mathit{matched}(\chi) \cap \mathit{unmatched}(\mathit{raise}([\chi], u_1))$, or $u_1$ does not belong to $\mathit{matched}(\chi) \cap \mathit{unmatched}(\mathit{raise}([\chi], u_0))$. The result follows from the definition of the function *raise* that takes a configuration as argument. □

# Chapter 7

# Top-Level Auction

The top-level auction is a sealed-bid unit-demand auction consisting of two phases. The first phase corresponds to running an instance of the mid-level auction and the second phase corresponds to solving an instance of the house allocation problem [37]. The sealed-bid unit-demand auction proposed in this dissertation is a variant of the top-level auction that admits a fast implementation.

We provide a formal description of the first and second phases of the top-level auction in the following sections. Here we briefly mention some of the high-level ideas underlying the design of the first phase. To ensure that the price of an item $v$ does not decrease, at the outset of the first phase, we tentatively impose the following obligation on the agent $u$ who is the target of item $v$'s put: Agent $u$ will remain allocated to $v$ at the strike price of $v$. Next, we drop the bids of all agents sufficiently until equilibrium properties 1, 2, and 3 of Section 4.3 are satisfied. The first phase then proceeds to update the tentative allocation and pricing in an iterative manner by invoking the mid-level auction. In Section 8.7, we discuss a fast implementation of this iterative procedure. In our fast implementation, each iteration either permanently releases an initially tentatively allocated agent from its obligation, or eliminates an unallocated agent whose unit-demand bid is too low to ever be allocated. The latter property ensures termination of the first phase.

The rest of this chapter is organized as follows. In Sections 7.1 and 7.2, we describe the first and second phases respectively of the top-level auction. In Section 7.3, we establish some basic properties of the top-level auction. In Section 7.4, we show that the top-level auction is truthful.

## 7.1 First Phase

For any configuration $\chi$, we define $\mathit{top}'_0(\chi)$ as follows. Let $\chi_0$ be a quiescent configuration and let $s_0$ be a target such that $\chi = \mathit{shift}(\chi_0, s_0)$, $\mathit{white}(\chi_0) \cap \mathit{matched}(\chi_0) = \emptyset$, and for any agent $u$ in $\mathit{unmatched}(\chi)$, we have $\mathit{items}(\chi_0, u) = \emptyset$. We define $\mathit{top}'_0(\chi)$ as the configuration $\mathit{shift}(\chi', s')$, where $(\chi', s') = \mathit{bottom}(\chi_0, s_0)$. The uniqueness of $\mathit{top}'_0(\chi)$ is established by Lemma 7.3.1.

## 7.2 Second Phase

The second phase of the top-level auction affects only the allocation and uses a single application of either the TTC algorithm [37] or the TC$^{\prec}$ algorithm of Jaramillo and Manjunath [19] to exchange items within a certain subset of the allocated agents.

For any configuration $\chi = (G, M, \Phi)$, we define an instance of the house allocation problem on $\chi$ as follows. Each agent in $black(\chi)$ represents a house owner and the item matched to $u$ in $M$ represents the house owned by $u$. Each agent $u$ in $black(\chi)$ is associated with a preference ordering over the items as follows, where $\beta = bid(\chi, u)$: for any pair of items $v$ and $v'$, if $\beta(v) - \Phi(v) > \beta(v') - \Phi(v')$, then agent $u$ prefers item $v$ over item $v'$; ties, if any, are broken using item identifiers.

For any configuration $\chi$, we define $top''(\chi)$ as the configuration obtained by using the TTC algorithm to resolve the house allocation problem defined on $\chi$. Alternatively, the second phase of the top-level auction can be implemented using the polynomial time TC$^{\prec}$ algorithm. The TC$^{\prec}$ algorithm has a slower running time than the TTC algorithm but yields an outcome with stronger efficiency-related properties than the TTC algorithm. (see Section 8.5 for details.)

For any instance of the top-level auction with configuration $\chi$ as input, the second phase of the top-level auction takes the configuration $\chi' = top'_0(\chi)$ as input and produces the configuration $top''(\chi')$ as output. For any instance of the top-level auction with configuration $\chi$ as input, the output of the top-level auction is given by $top''(top'_0(\chi))$.

Recall that equilibrium properties 1, 2, and 3 of Section 4.3 are satisfied by the first phase of the top-level auction. In the second phase, item prices and the allocation of non-black agents remain unchanged. Thus, it is easy to see that equilibrium properties 1, 2, and 3 are retained in the second phase. The second phase involves resolving an instance of the house allocation problem on the subset of black agents; thus, by definition, equilibrium properties $4(a)$ and $4(b)$ are satisfied. Finally, it follows from known results on the $\mathrm{TC}^{\prec}$ (TTC) algorithm that the solution computed in the second phase is in the (weak) core. This establishes equilibrium property 5 of Section 4.3.

**Fact 7.2.1.** *For any configuration $\chi'$ of the form $\mathrm{top}''(\chi)$ where $\chi = (G, M, \Phi)$ and $\chi' = (G, M', \Phi')$, we have $\Phi' = \Phi$, $\mathrm{unmatched}(\chi') = \mathrm{unmatched}(\chi)$, and $\mathrm{white}(\chi) \subseteq \mathrm{white}(\chi')$.*

**Fact 7.2.2.** *For any configuration $\chi = (G, M, \Phi)$, if $\mathrm{top}(\chi) = (G, M', \Phi')$, then $\Phi' \geq \Phi$.*

## 7.3 Properties

The following lemmas establish basic properties of the top-level auction.

**Lemma 7.3.1.** *Let $\chi_0$ and $\chi_1$ be configurations and let $s_0$ and $s_1$ be targets such that (1) configurations $\chi_0$ and $\chi_1$ are quiescent, and $\chi = \mathrm{shift}(\chi_0, s_0) = \mathrm{shift}(\chi_1, s_1)$, (2) $\mathrm{white}(\chi_0) \cap \mathrm{matched}(\chi_0) = \mathrm{white}(\chi_1) \cap \mathrm{matched}(\chi_1) = \emptyset$, and (3) for any agent $u$ in $\mathrm{unmatched}(\chi)$, we have $\mathrm{items}(\chi_0, u) = \mathrm{items}(\chi_1, u) = \emptyset$. Then, we have $\mathrm{bottom}(\chi_0, s_0) = \mathrm{bottom}(\chi_1, s_1)$.*

*Proof.* Let $s^*$ be the target such that for any agent $u$ in $agents(\chi)$, we have $s^*(u) = min(s_0(u), s_1(u))$, and let $\chi = shift(\chi^*, s^*)$; thus $s_0(u) \geq s^*(u)$. Let $S$ be the set of agents $u$ in $agents(\chi)$ such that $s_0(u) > s^*(u)$. By the definitions of $s_0$ and $s^*$, for any agent $u$ in $S$, we find that $u$ belongs to $enabled(\chi_0)$ and $raise(\chi_0, u) = shift(\chi_0, u_0, 1)$; by repeated use of this fact and Lemma 6.3.7, agents in $S$ can commute their *raise* invocations forward until each agent $u$ in $S$ has $s^*(u)$ pending *raise* invocations and the resulting configuration is $\chi^*$; thus, $\mathit{bottom}(\chi_0, s_0) = \mathit{bottom}(\chi^*, s^*)$. By a similar argument, we have $\mathit{bottom}(\chi_1, s_1) = \mathit{bottom}(\chi^*, s^*)$. Thus, $\mathit{bottom}(\chi_0, s_0) = \mathit{bottom}(\chi_1, s_1)$. $\square$

**Lemma 7.3.2.** *The second phase of the top-level auction is truthful.*

*Proof.* Consider any configuration $\chi'$ of the form $\mathit{top}''(\chi)$. The second phase of the top-level auction, which is implemented using an application of the TTC algorithm or the TC$^{\prec}$, updates only the matching of black agents. By known results on the truthfulness of the TTC and TC$^{\prec}$ algorithms, the second phase of the top-level auction is truthful for agents in $\mathit{black}(\chi)$. By Fact 7.2.1, we have $\mathit{potential}(\mathit{top}''(\chi)) = \mathit{potential}(\chi)$; thus no agent $u$ in $\mathit{nonblack}(\chi)$ can achieve a utility higher than $\mathit{gap}(\chi, u)$ by submitting a false bid. Thus, the second phase of the top-level auction is truthful. $\square$

**Lemma 7.3.3.** *For any configuration $\chi_0$ of the form* $\mathrm{subst}(\chi, u, \beta)$*, if* $\beta \neq \mathrm{bid}(\text{bid-graph}(\chi), u)$*, then either* $\mathrm{top}'_0(\chi_0) = \mathrm{subst}(\mathrm{top}'_0(\chi), u, \beta)$ *or $u$ belongs to* $\mathrm{white}(\mathrm{top}'_0(\chi)) \cap \mathrm{white}(\mathrm{top}'_0(\chi_0))$.

*Proof.* It follows from the description of the bottom-level auction, that either $u$ is unmatched in the same round of both bottom-level auction instances and hence $u$ belongs to $\mathit{white}(\mathit{top}'_0(\chi)) \cap \mathit{white}(\mathit{top}'_0(\chi_0))$, or $u$ remains matched throughout to the same item in both auction instances and $\mathit{top}'_0(\chi_0) = \mathit{subst}(\mathit{top}'_0(\chi), u, \beta)$. $\square$

## 7.4 Truthfulness

A sealed-bid auction is said to be truthful if it is a weakly dominant strategy for every agent in the auction to bid truthfully. Formally, we say the first phase of the top-level auction is truthful if it satisfies the following condition: for any configuration $\chi$ and any agent $u$ in $\mathit{agents}(\chi)$, if $\chi' = \mathit{subst}(\chi, u, \beta)$ for some bid $\beta$ in $\mathit{bids}(\mathit{bid\text{-}graph}(\chi))$, then

$$\mathit{gap}(\mathit{top}'_0(\chi), u) \geq \mathit{gap}(\chi'', u)$$

where $\chi'' = \mathit{subst}(\mathit{top}'_0(\chi'), u, \mathit{bid}(\chi, u))$.

In what follows, we establish Lemma 7.4.3 on the truthfulness of the top-level auction. We establish Lemma 7.4.1 based on the claim of Lemma 6.3.6 of Section 6.1.1. Lemma 7.4.2 follows from Lemma 7.4.1. The proof of Lemma 7.4.3 follows from Lemma 7.4.2 and known results on the truthfulness of the TTC and TC$^{\prec}$ algorithms.

We use the claims of this section to establish Lemma 8.4.10 on the truthfulness of our proposed sealed-bid unit-demand auction in Section 8.4.

**Lemma 7.4.1.** *For any configuration $\chi$, any agent $u$ in* $\mathrm{agents}(\chi)$ *where* $\mathrm{bid}(\mathrm{bid\text{-}graph}(\chi), u) = \beta'$, *and any configuration $\chi'$ of the form* $\mathrm{subst}(\chi, u, \beta)$ *where $\beta$ is a bid in* $\mathrm{bids}(\mathrm{bid\text{-}graph}(\chi))$*, we have*

$$\mathrm{gap}(\mathrm{top}'_0(\chi), u) \geq \mathrm{gap}(\mathrm{subst}(\mathrm{top}'_0(\chi'), u, \beta'), u).$$

*Proof.* Let $\chi = \mathit{shift}(\chi_0, s_0)$, where $\chi_0$ is a quiescent configuration and $s_0$ is a target such that $\mathit{white}(\chi_0) \cap \mathit{matched}(\chi_0) = \emptyset$, and for any agent $u$ in

$\mathit{unmatched}(\chi)$, we have $\mathit{items}(\chi_0, u) = \emptyset$. By definition, $\mathit{top}'_0(\chi) = \mathit{shift}(\chi'_0, s'_0)$, where $(\chi'_0, s'_0) = \mathit{bottom}(\chi_0, s_0)$. Note that $\mathit{threshold}^*(\chi, s_0) = \mathit{threshold}^*(\chi, s_0)$. Similarly, let $\chi' = \mathit{shift}(\chi_1, s_1)$, where $\chi_1$ is a quiescent configuration and $s_1$ is a target such that $\mathit{white}(\chi_1) \cap \mathit{matched}(\chi_1) = \emptyset$, and $\mathit{items}(\chi_1, u) = \emptyset$ for any agent $u$ in $\mathit{unmatched}(\chi)$. By definition, $\mathit{top}'_0(\chi') = \mathit{shift}(\chi'_1, s'_1)$, where $(\chi'_1, s'_1) = \mathit{bottom}(\chi_1, s_1)$.

Let $\beta_T = \mathit{bid}(\mathit{bid\text{-}graph}(\chi), u)$ and let $\chi'' = \mathit{subst}(\chi'_1, u, \beta_T)$. Assume that $\mathit{gap}(\mathit{top}'_0(\chi), u) < \mathit{gap}(\mathit{subst}(\mathit{top}'_0(\chi'), u, \beta_T), u)$; thus $\mathit{gap}(\chi'_0, u) + s'_0(u) < \mathit{gap}(\chi'', u) + s'_1(u)$.

We first consider the case where $u$ belongs to $\mathit{unmatched}(\chi)$. By repeated application of Lemma 6.3.2, we find that $u$ belongs to $\mathit{white}(\chi'_0) \cap \mathit{white}(\chi'_1)$ and $s'_0(u) = s'_1(u) = 0$; thus, by our assumption, $\mathit{gap}(\chi'_0, u) < \mathit{gap}(\chi'', u)$. Since $\mathit{gap}(\chi'_0, u) < \mathit{gap}(\chi'', u)$ and $u$ belongs to $\mathit{white}(\chi'_0)$, we have $\mathit{gap}(\chi'', u) \geq 1$; thus $u$ belongs to $\mathit{matched}(\chi'_1)$. By Lemma 6.3.7, we choose to defer the *raise* invocations of $u$ until a round in which $u$ is the only remaining enabled agent with pending *raise* invocations. Let $(\chi_a, s_a)$ be the input of the first round in which $u$ invokes the function *raise* in the bottom-level auction instance with input $(\chi_0, s_0)$, and let $\chi'_a$ be the configuration such that $\chi_a = \mathit{add}(\chi'_a, u, \beta_T)$. Let $(\chi_b, s_b)$ be the input of the first round in which $u$ invokes the function *raise* in the bottom-level auction instance with input $(\chi_1, s_1)$, and let $\chi_b{}'$ be the configuration such that $\chi_b = \mathit{add}(\chi'_b, u, \beta)$. Let $S'$ be the set of items $v$ in $\mathit{items}(\chi)$ for which $\beta(v) - \mathit{threshold}^*(\chi'_b, s_b, v)$ is maximized. By Lemma 6.3.6, we have $\mathit{potential}(\chi'_1, v) = \mathit{threshold}^*(\chi'_b, s_b, v)$; thus, we have

$gap(\chi'_1, u) = \beta(v) - threshold^*(\chi'_b, s_b, v)$ and we have $gap(\chi''_1, u) = \beta_T(v) - threshold^*(\chi'_b, s_b, v)$. By Lemma 6.3.6, $gap(\chi'_0, u) = max_{v \in items(\chi)}(\beta_T(v) - threshold^*(\chi'_a, s_a, v))$; since $threshold^*(\chi'_b, s_b) = threshold^*(\chi'_a, s_a)$, we have $gap(\chi'_0, u) \geq gap(\chi'', u)$; a contradiction. Thus, it follows that

$$gap(subst(top'_0(\chi'), u, \beta_T), u) \leq gap(top'_0(\chi), u)$$

Next we consider the case where agent $u$ belongs to $matched(\chi)$. Since $matched(\chi_0) \cap white(\chi_0) = matched(\chi_1) \cap white(\chi_1) = \emptyset$, the definition of the function *shift* implies that there exists an item $v$ in $items(\chi)$ such that $match(\chi_0, v) = match(\chi_1, v) = u$. Since agent $u$ belongs to $nonwhite(\chi_0) \cap nonwhite(\chi_1)$, by the description of the bottom-level auction, either we have $match(\chi'_0, v) = match(\chi'_1, v) = u$, or $u$ is unmatched in some round of the bottom-level auction instances with inputs $(\chi_0, s_0)$ and $(\chi_1, s_1)$. In the case where $match(\chi'_0, v) = match(\chi'_1, v) = u$, it is easy to see that $gap(top'_0(\chi), u) \geq gap(subst(top'_0(\chi'), u, \beta_T), u)$. In the case where $u$ is unmatched in some round of the bottom-level auction instances with inputs $(\chi_0, s_0)$ and $(\chi_1, s_1)$, the analysis is similar to the previous case in which $u$ belongs to $unmatched(\chi)$. Thus, we have

$$gap(top'_0(\chi), u) \geq gap(subst(top'_0(\chi'), u, bid(bid\text{-}graph(\chi), u)), u)$$

□

**Lemma 7.4.2.** *The first phase of the top-level auction is truthful.*

*Proof.* Follows from Lemma 7.4.1 and the definition of truthfulness. □

**Lemma 7.4.3.** *The top-level auction is truthful.*

*Proof.* For any configuration $\chi_0$, let $f(\chi_0)$ denote $\mathit{top}''(\mathit{top}'_0(\chi_0))$. Consider any instance of the top-level auction with configuration $\chi$ as input and let $u$ be an agent in $\mathit{agents}(\chi)$. Let $\beta = \mathit{bid}(\mathit{bid\text{-}graph}(\chi), u)$ and let $\beta_T \neq \beta$ be the truthful bid of $u$. Let $\chi_T = \mathit{subst}(\chi, u, \beta_T)$. We are required to show that $\mathit{gap}(\mathit{subst}(f(\chi), u, \beta_T), u) \leq \mathit{gap}(f(\chi_T), u)$. By Lemma 7.3.3, it follows that either agent $u$ belongs to $\mathit{white}(\mathit{top}'_0(\chi)) \cap \mathit{white}(\mathit{top}'_0(\chi_T))$ or $\mathit{top}'_0(\chi_T) = \mathit{subst}(\mathit{top}'_0(\chi), u, \beta_T)$.

First, we consider the case where agent $u$ belongs to $\mathit{white}(\mathit{top}'_0(\chi)) \cap \mathit{white}(\mathit{top}'_0(\chi_T))$. By Fact 7.2.1, we have $\mathit{potential}(f(\chi)) = \mathit{potential}(\mathit{top}'_0(\chi))$ and $u$ belongs to $\mathit{white}(f(\chi))$; thus we have $\mathit{gap}(f(\chi), u) = \mathit{gap}(\mathit{top}'_0(\chi), u)$ and $\mathit{gap}(f(\chi_T), u) = \mathit{gap}(\mathit{top}'_0(\chi_T), u)$. By Lemma 7.4.1, we have

$$\mathit{gap}(\mathit{subst}(\mathit{top}'_0(\chi), u, \beta_T), u) \leq \mathit{gap}(\mathit{top}'_0(\chi_T), u)$$

Thus, we have $\mathit{gap}(\mathit{subst}(f(\chi), u, \beta_T), u) \leq \mathit{gap}(\chi_T, u)$.

Next, we consider the case where $\mathit{top}'_0(\chi_T) = \mathit{subst}(\mathit{top}'_0(\chi), u, \beta_T)$. By Lemma 7.3.2, the second phase of the top-level auction is truthful; thus, we have $\mathit{gap}(\mathit{subst}(f(\chi), u, \beta_T), u) \leq \mathit{gap}(f(\chi_T), u)$. Thus, any auction that takes an configuration $\chi$ as input and produces the configuration $f(\chi) = \mathit{top}''(\mathit{top}'_0(\chi))$ as output is truthful. $\square$

# Chapter 8

# A Sealed-Bid Unit-Demand Auction with Put Options

In this chapter, we describe our proposed sealed-bid unit-demand auction with put options.

The top-level auction described in Chapter 7 does not immediately incorporate bid revision requests of tentatively allocated agents at the beginning of the auction. Instead, the bid of each tentatively allocated agent is incorporated by a suitable number of *raise* invocations in the underlying mid-level auction. Our proposed sealed-bid unit-demand auction described in this chapter is a variant of the top-level auction that immediately incorporates bid revision requests of tentatively allocated agents whenever possible; consequently our proposed sealed-bid auction admits a fast implementation.

The rest of this chapter is organized as follows. In Section 8.1, we introduce some preliminary definitions. In Section 8.2, we describe our proposed sealed-bid unit-demand auction. We establish truthfulness of this auction in Section 8.4 by using results on the truthfulness of the top-level auction from Section 7.4. In Section 8.5, we establish efficiency-related properties of our sealed-bid auction. In Section 8.6, we show that our sealed-bid auction maintains bid privacy for tentatively allocated agents. Finally, in Section 8.7, we describe a polynomial-time implementation in which our sealed-bid auction is resolved by performing a number of single-source shortest paths computations, with the number of such computations bounded by the number of agents in the auction.

## 8.1 Preliminaries

For any configuration $\chi$, we define $\mathit{targets}(\chi)$ as the set of all targets $s$ such that there exists a quiescent configuration $\chi_0$ satisfying the following conditions:

1. $\mathit{shift}(\chi_0, s) = \chi$
2. $\mathit{white}(\chi_0) \cap \mathit{matched}(\chi_0) = \mathit{white}(\chi) \cap \mathit{matched}(\chi)$
3. for any agent $u$ in $\mathit{unmatched}(\chi)$ we have $\mathit{items}(\chi_0, u) = \emptyset$

For any configuration $\chi$, we define $\mathit{target}(\chi)$ as the unique pointwise minimum target in $\mathit{targets}(\chi)$.

## 8.2 Description

Consider any instance of our sealed-bid auction with a configuration $\chi$ as input and let $\chi = \mathit{shift}(\chi_0, \mathit{target}(\chi))$. The output of the first phase of our sealed-bid auction, denoted $\mathit{top}'(\chi)$, is given by $\mathit{shift}(\chi', s')$, where

$(\chi', s') = \mathit{bottom}(\chi_0, \mathit{target}(\chi))$. The output of the second phase of our sealed-bid auction is given by $\mathit{top}''(\mathit{top}'(\chi))$. The output of our sealed-bid auction, denoted $\mathit{top}(\chi)$, is the output of its second phase.

## 8.3 Properties

The following lemmas establish some basic properties of our proposed sealed-bid auction.

**Lemma 8.3.1.** *For any configuration $\chi$ and any targets $s_0$ and $s_1$ in* $\mathrm{targets}(\chi)$ *such that* $\chi = \mathrm{shift}(\chi_0, s_0) = \mathrm{shift}(\chi_1, s_1)$*, we have*

$$\mathrm{bottom}(\chi_0, s_0) = \mathrm{bottom}(\chi_1, s_1)$$

*Proof.* Let $\chi = \mathit{shift}(\chi^*, s^*)$, where $s^* = \mathit{target}(\chi)$. Since $s^*$ is the pointwise minimum target in $\mathit{targets}(\chi)$, we have $s_0(u) \geq s^*(u)$ for any agent $u$ in $\mathit{agents}(\chi)$. Let $S$ be the set of agents $u$ in $\mathit{agents}(\chi)$ such that $s_0(u) > s^*(u)$. By the definitions of $\mathit{targets}(\chi)$ and $s^*$, for any agent $u$ in $S$, we find that $u$ belongs to $\mathit{enabled}(\chi_0)$ and $\mathit{raise}(\chi_0, u) = \mathit{shift}(\chi_0, u_0, 1)$; by repeated use of this fact and Lemma 6.3.7, agents in $S$ can commute their *raise* invocations forward until each agent $u$ in $S$ has $s^*(u)$ pending *raise* invocations and the resulting configuration is $\chi^*$; thus, $\mathit{bottom}(\chi_0, s_0) = \mathit{bottom}(\chi^*, s^*)$. By a similar argument, we have $\mathit{bottom}(\chi_1, s_1) = \mathit{bottom}(\chi^*, s^*)$. Thus, $\mathit{bottom}(\chi_0, s_0) = \mathit{bottom}(\chi_1, s_1)$. □

**Lemma 8.3.2.** *Let $\chi$ be any configuration and let $\chi^*$ be the configuration* *such that* $\chi = \text{shift}(\chi^*, \text{target}(\chi))$. *If* $(\chi_0, s_0) = \text{bottom}(\chi^*, \text{target}(\chi))$*, then* $\text{unmatched}(\text{top}'(\chi)) \subseteq \text{white}(\text{top}'(\chi))$ *and* $\text{nonwhite}(\text{top}'(\chi)) \subseteq \text{nonwhite}(\chi_0)$.

*Proof.* Let $s^* = \mathit{target}(\chi)$. By definition, we have $\mathit{top}'(\chi) = \mathit{shift}(\chi_0, s_0)$. Since $\chi_0$ is quiescent, we have $\mathit{unmatched}(\chi_0) \subseteq \mathit{white}(\chi_0)$ and for any agent $u$ in $\mathit{unmatched}(\chi_0)$, we find that $\mathit{agents}(\chi_0, u) \cap \mathit{nonwhite}(\chi_0) = \emptyset$. Moreover, these facts imply that by definition of the bottom-level auction, we have $s_0(u) = 0$ for any agent $u$ in $\mathit{white}(\chi_1)$; we conclude that $u$ belongs to $\mathit{white}(\mathit{shift}(\chi_0, s_0))$, where $\mathit{top}'(\chi) = \mathit{shift}(\chi_0, s_0)$.

It remains to show that $\mathit{nonwhite}(\mathit{top}'(\chi)) \subseteq \mathit{nonwhite}(\chi_0)$. Consider any agent $u$ in $\mathit{nonwhite}(\mathit{top}'(\chi))$; since $s_0(u) \geq 0$, we conclude that $u$ belongs to $\mathit{nonwhite}(\chi_0)$. □

**Lemma 8.3.3.** *For any configuration* $\chi$ *and any agent* $u$ *in* $\text{nonwhite}(\text{top}'(\chi))$*, the following conditions hold:* (1) $u$ *belongs to* $\text{nonwhite}(\chi)$*,* (2) *there exists an item* $v$ *in* $\text{items}(\chi)$ *such that* $\text{potential}(\chi, v) = \text{potential}(\text{top}'(\chi), v)$*, and* (3) $\text{match}(\chi, v) = \text{match}(\text{top}'(\chi), v) = u$.

*Proof.* Let $\chi = \mathit{shift}(\chi^*, s^*)$, where $s^* = \mathit{target}(\chi)$. Let $(\chi_0, s_0) = \mathit{bottom}(\chi^*, s^*)$. By definition, we have $\mathit{top}'(\chi) = \mathit{shift}(\chi_0, s_0)$. By Lemma 8.3.2, $u$ belongs to $\mathit{nonwhite}(\chi_0)$; since $\chi_0$ is quiescent, we find that $u$ belongs to $\mathit{matched}(\chi_0)$; further, since $\mathit{top}'(\chi) = \mathit{shift}(\chi_0, s_0)$, there exists an item $v$ in $\mathit{items}(\chi)$ such that $\mathit{potential}(\mathit{top}'(\chi), v) = \mathit{potential}(\chi_0, v)$ and $\mathit{match}(\chi, v) = \mathit{match}(\mathit{top}'(\chi), v)$. By repeated application of Lemma 6.3.1, we find that $u$ belongs to $\mathit{nonwhite}(\chi^*)$, $\mathit{potential}(\chi^*, v) = \mathit{potential}(\chi_0, v)$ and $\mathit{match}(\chi^*, v) = \mathit{match}(\chi_0, v) = u$.

By the description of our sealed-bid auction it follows that $u$ belongs to $\mathit{nonwhite}(\chi)$, $\mathit{potential}(\chi, v) = \mathit{potential}(\chi^*, v)$, and $\mathit{match}(\chi, v) = u$. These facts imply that $u$ belongs to $\mathit{nonwhite}(\chi)$ and there exists an item $v$ in $\mathit{items}(\chi)$ such that $\mathit{potential}(\chi, v) = \mathit{potential}(\mathit{top}'(\chi), v)$, and $\mathit{match}(\chi, v) = \mathit{match}(\mathit{top}'(\chi), v) = u$. $\square$

**Lemma 8.3.4.** *For any configuration $\chi$, we have*

$$\text{nonwhite}(\text{top}(\chi)) \subseteq \text{nonwhite}(\chi) \wedge (\text{unmatched}(\chi) \cup \text{white}(\chi)) \subseteq \text{white}(\text{top}(\chi))$$

*Proof.* Let $\chi = \mathit{shift}(\chi^*, s^*)$ where $s^* = \mathit{target}(\chi)$. Let $(\chi_0, s_0) = \mathit{bottom}(\chi^*, s^*)$. We have $\mathit{unmatched}(\chi) = \mathit{unmatched}(\chi^*) \cap \mathit{white}(\chi^*)$ and we have $\mathit{white}(\chi) \cap \mathit{matched}(\chi) = \mathit{white}(\chi^*) \cap \mathit{matched}(\chi^*)$; thus $\mathit{unmatched}(\chi) \cup \mathit{white}(\chi) \subseteq \mathit{white}(\chi^*)$. By repeated application of Lemma 6.3.2, we have $\mathit{white}(\chi^*) \subseteq \mathit{white}(\chi_0)$, and since $\mathit{top}'(\chi) = \mathit{shift}(\chi_0, s_0)$, we have $\mathit{white}(\chi^*) \subseteq \mathit{white}(\mathit{top}'(\chi))$. Finally, by Fact 7.2.1, we have $\mathit{white}(\mathit{top}'(\chi)) \subseteq \mathit{white}(\mathit{top}(\chi))$. Thus, we have $\mathit{unmatched}(\chi) \cup \mathit{white}(\chi) \subseteq \mathit{white}(\mathit{top}(\chi))$. By Fact 7.2.1, we have $\mathit{white}(\mathit{top}'(\chi)) \subseteq \mathit{white}(\mathit{top}(\chi))$; thus $\mathit{nonwhite}(\mathit{top}(\chi)) \subseteq \mathit{nonwhite}(\mathit{top}'(\chi))$. $\square$

**Lemma 8.3.5.** *For any configuration $\chi_0$ of the form* $\text{subst}(\chi, u, \beta)$*, if $\beta \neq$* $\text{bid}(\text{bid-graph}(\chi), u)$*, then either* $\text{top}'(\chi_0) = \text{subst}(\text{top}'(\chi), u, \beta)$ *or $u$ belongs to* $\text{white}(\text{top}'(\chi)) \cap \text{white}(\text{top}'(\chi_0))$.

*Proof.* Let $\chi = \mathit{shift}(\chi^*, \mathit{target}(\chi))$ and let $\chi_0 = \mathit{shift}({\chi_0}^*, \mathit{target}(\chi_0))$. By definition, we have $\mathit{top}'(\chi) = \mathit{shift}(\chi', s')$ where $(\chi', s') = \mathit{bottom}(\chi^*, \mathit{target}(\chi))$

and we have $\mathit{top}'(\chi_0) = \mathit{shift}(\chi_0', s_0')$ where $(\chi_0', s_0') = \mathit{bottom}(\chi_0{}^*, \mathit{target}(\chi_0))$. We consider the following cases. First we consider the case where $u$ belongs to $\mathit{white}(\chi^*) \cap \mathit{white}(\chi_0{}^*)$. By repeated application of Lemma 6.3.2, we find that $u$ belongs to $\mathit{white}(\chi')$, and since $s'(u) \geq 0$, we find that $u$ belongs to $\mathit{white}(\mathit{top}'(\chi)) \cap \mathit{white}(\mathit{top}'(\chi_0))$.

Next we consider the case where agent $u$ belongs to $\mathit{nonwhite}(\chi^*) \cap \mathit{nonwhite}(\chi_0{}^*)$. It follows from the description of the bottom-level auction, that either $u$ is unmatched in the same round of both bottom-level auction instances and hence $u$ belongs to $\mathit{white}(\mathit{top}'(\chi)) \cap \mathit{white}(\mathit{top}'(\chi_0))$, or $u$ remains matched throughout to the same item in both auction instances and $\mathit{top}'(\chi_0) = \mathit{subst}(\mathit{top}'(\chi), u, \beta)$.

Finally we look at the case where $u$ either belongs to $\mathit{nonwhite}(\chi^*)$ or belongs to $\mathit{nonwhite}(\chi_0^*)$. Without loss of generality, assume that $u$ belongs to $\mathit{nonwhite}(\chi^*)$. It follows from the description of the bottom-level auction, that either $u$ is unmatched in some round of the auction instance with input $(\chi^*, \mathit{target}(\chi))$ and hence either $u$ belongs to $\mathit{white}(\mathit{top}'(\chi)) \cap \mathit{white}(\mathit{top}'(\chi_0))$, or agent $u$ remains matched throughout in the auction instance with input $(\chi^*, \mathit{target}(\chi))$ and hence $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\chi, u)$. It follows that $\mathit{top}'(\chi_0) = \mathit{subst}(\mathit{top}'(\chi), u, \beta)$. $\square$

## 8.4 Truthfulness

The goal of this section is to establish Lemma 8.4.10 on the truthfulness of our sealed-bid auction described in Section 8.2. We first establish that for any configuration $\chi$ and any white agent $u$ in $top'(\chi)$, agent $u$ has the same utility in $top'(\chi)$ as it does in $top'_0(\chi)$ (see Lemma 8.4.7). Lemmas 8.4.8 and 8.4.9 follow easily from Lemma 8.4.7. We use Lemmas 8.4.9 and known results on the truthfulness of the TTC algorithm and the TC$^{\prec}$ algorithm to establish Lemma 8.4.10.

**Lemma 8.4.1.** *For any configuration $\chi'$ of the form* $\text{subst}(\chi, u, u')$*, if $u$ is an agent in* $\text{matched}(\chi) \cap \text{white}(\chi)$*, then* $\text{gap}(\text{top}'(\chi), u) = \text{gap}(\text{top}'(\chi'), u')$*.*

*Proof.* Let $target(\chi) = s$ and let $target(\chi') = s'$. We first show that $s = s'$. Since $u$ belongs to $matched(\chi) \cap white(\chi)$, we have $\alpha(u) = 0$, and since $s$ is the pointwise minimum target in $targets(\chi)$ and $u'$ does not belong to $agents(\chi)$, we have $\alpha(u') = 0$. Similarly, we have $s'(u') = s'(u) = 0$. Since $s$ and $s'$ are the pointwise minimum targets in $targets(\chi)$ and $targets(\chi')$ respectively, we have $\alpha(u'') = s'(u'')$ for any agent $u''$ in $agents(\chi) - u$. It follows that $s = s'$.

Let $top'(\chi) = shift(\chi'_0, s'_0)$ where $(\chi'_0, s'_0) = bottom(\chi_0, s)$. Let $top'(\chi') = shift(\chi'_1, s'_1)$ where $(\chi'_1, s'_1) = bottom(\chi_1, s)$. By Lemma 6.3.4, $gap(\chi_0, u) = gap(\chi_1, u')$. Further, since $\alpha(u) = \alpha(u') = 0$, we have $s'_0(u) = s'_1(u') = 0$. Thus, $gap(top'(\chi), u) = gap(top'(\chi'), u')$. □

**Lemma 8.4.2.** *Let $\chi$ be a configuration and let $\chi_A$ be the quiescent configuration such that* $\chi = \text{shift}(\chi_A, \text{target}(\chi))$*. Let $u$ be an agent in* $\text{matched}(\chi) \cap \text{white}(\chi)$*, and let $z$ be an integer such that $u$ belongs to* $\text{gray}(\text{shift}(\chi_A, u, -z))$*. If $u$ is the minimum agent in* $\text{agents}(\chi)$*, then* $\text{gap}(\text{top}'(\chi), u) = \text{gap}(\chi_B, u) + s_B(u)$*, where* $(\chi_B, s_B) = \text{bottom}(\text{shift}(\chi'_A, u, -z), \text{shift}(\text{target}(\chi), u, z))$*.*

*Proof.* Let $(\chi'_B, s'_B) = \mathit{bottom}(\chi_A, \mathit{target}(\chi))$. We refer to the executions of the bottom-level auction with inputs $(\mathit{shift}(\chi'_A, u, -z), \mathit{shift}(\mathit{target}(\chi), u, z))$ and $(\chi_A, \mathit{target}(\chi))$ as executions $R$ and $R'$ respectively. Let $(\chi_i, s_i)$ and $(\chi'_i, s'_i)$ be the outputs of round $i$ of executions $R$ and $R'$ respectively. By Lemma 6.3.7, the *raise* invocations of enabled agents commute. Thus, by repeated application of Lemma 6.3.7, executions $R$ and $R'$ can be reordered such that for any round $i$, if $S$ is a nonempty set of agents in $\mathit{enabled}(\chi_i) \cap \mathit{enabled}(\chi'_i)$ such that $s_i(u') = s'_i(u') > 0$, then some agent $u'$ in $S$ invokes *raise* in round $i+1$ of executions $R$ and $R'$. For any round $i$, we define the predicate $P(i)$ to hold if $(\chi'_i, s'_i) = (\mathit{shift}(\chi_i, u, -z), \mathit{shift}(s_i, u, z))$.

First we consider the case where $P(i)$ holds for every round of executions $R$ and $R'$. In this case, we have $(\chi'_B, s'_B) = (\mathit{shift}(\chi_B, u, -z), \mathit{shift}(s_B, u, z))$. Thus, we have $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\chi'_B, u) + s'_B = \mathit{gap}(\chi_B, u) - z + s_B(u) + z$. Since $u$ belongs to $\mathit{white}(\mathit{top}(\chi))$, we have $s'_B(u) = 0$. Thus, $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\chi'_B, u) + s'_B$.

Next we consider the case where there exists a first round $k$ such that $P(k)$ does not hold. In this case, it is easy to see that either $u$ belongs to $\mathit{unmatched}(\chi_k)$ or $u$ belongs to $\mathit{unmatched}(\chi'_k)$; further, since $u$ belongs to $\mathit{nonwhite}(\chi'_k)$, we find that $u$ belongs to $\mathit{unmatched}(\chi_k)$. Let $u'$ be the agent in $\mathit{matched}(\chi'_{k-1}) \cap \mathit{unmatched}(\chi'_k)$. We now allow $u$ to exhaust all its *raise* invocations in rounds $(k+1) \cdots (k+z)$ of execution $B$; thus, we have $s_B(u) = 0$. Since $u$ belongs to $\mathit{white}(\mathit{top}'(\chi))$, we have $\mathit{agents}(\chi'_j, u) \neq \emptyset$ for some round $j$ of execution $B$ where $(k+1) \leq j \leq (k+z)$, and thus, $u' = \mathit{victim}(\chi'_j, u)$.

It is straightforward to see that for any round $j > k$ of execution $R$, we have $\chi_j = \chi_{j+z}$; thus, $\chi'_B = \chi_B$. Since $\alpha(u) = 0$, we have $s'_B(u) = 0$, and we established above that $s_B(u) = 0$. Thus, $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\chi_B, u) = \mathit{gap}(\chi_B, u) + s_B(u_0)$. $\square$

[Definition] For any configuration $\chi$, any agent $u$ in $\mathit{matched}(\chi) \cap \mathit{white}(\chi)$, and any agent $u'$ such that $u < u'$ and there exists no agent $u''$ in $\mathit{agents}(\chi)$ such that $u' < u'' < u$, we define $\mathit{split}(\chi, u, u')$ as the configuration $\mathit{add}(\mathit{shift}(\chi, u, -z), u', \beta)$ where $z$ is the integer such that $\beta = \mathit{bid}(\mathit{bid\text{-}graph}(\chi), u)$ and $\mathit{max\text{-}gap}(\mathit{shift}(\chi, u, -z), u) = -1$.

**Lemma 8.4.3.** *Let $\chi'$ be a configuration of the form* $\mathrm{split}(\chi, u, u')$ *and let* $\chi = \mathrm{shift}(\chi_A, \mathrm{target}(\chi))$. *For any integer $z$ such that $u$ belongs to* $\mathrm{gray}(\mathrm{shift}(\chi_A, u, -z))$, *we have*

$$\mathrm{gap}(\mathrm{top}'(\chi'), u') = \mathrm{gap}(\chi_B, u) + s_B(u)$$

*where* $(\chi_B, s_B) = \mathrm{bottom}(\mathrm{shift}(\chi_A, u, -z), \mathrm{shift}(\mathrm{target}(\chi), u, z))$.

*Proof.* Let $\chi'' = \mathit{shift}(\chi_A, u, -z^*)$, where $z^*$ is the integer such that $\mathit{max\text{-}gap}(\chi'', u)$ is equal to $-1$. Let $s = \mathit{target}(\chi)$. By the definition of the function *split*, we find that $u$ belongs to $\mathit{matched}(\chi) \cap \mathit{white}(\chi)$; thus, $\alpha(u) = 0$. Note that $u$ belongs to $\mathit{white}(\mathit{shift}(\chi'', u, 1))$; thus, $z^* \le z$. By repeated application of Fact 6.1.1, it follows that $(\chi_B, s_B) = \mathit{bottom}(\chi'', \mathit{shift}(\mathit{target}(\chi), u, z^*))$. Let $\chi'_A$ be the quiescent configuration such that $\chi' = \mathit{shift}(\chi'_A, \mathit{target}(\chi'))$ and let $(\chi'_B, s'_B) = \mathit{bottom}(\chi'_A, \mathit{target}(\chi'))$.

We refer to the execution of the bottom-level auction instance with inputs $(\chi'', \mathit{shift}(\mathit{target}(\chi), u, z^*))$ as execution $A$ and we refer to the bottom-level auction instance with inputs $(\chi'_A, \mathit{target}(\chi'))$ as execution $B$. By Lemma 6.3.7, we can assume that the same agent invokes *raise* in both executions whenever possible. From the description of the bottom-level auction, we find that either $u$ becomes unmatched in the same round of both executions, or $u$ remains matched in both executions until $u'$ is the only remaining enabled agent with pending *raise* invocations in execution $B$.

We first consider the case where $u$ is unmatched in the same round of both executions. In this case, we immediately process the *raise* invocations of agent $u$ in execution $A$ and agent $u'$ in execution $B$. If $\chi_B$ and $\chi'_B$ are the resultant configurations of executions $A$ and $B$ after agents $u$ and $u'$ have exhausted their *raise* invocations, then by Lemma 6.3.5, we have $\mathit{gap}(\chi'_B, u') + s'_B(u') = \mathit{gap}(\chi_B, u) + s_B(u)$, and the proof is complete.

Next we consider the case where $u$ remains matched to some item $v$ in both executions, all enabled agents have exhausted their *raise* invocations in execution $A$, and $u'$ is the only remaining enabled agent with pending *raise* invocations in execution $B$. We now allow agent $u'$ to exhaust its *raise* invocations. While $u$ remains allocated for the rest of execution $B$, the potential of $v$ remains unchanged; thus $u'$ attains its highest utility of $\mathit{gap}(\chi_B, u) + s_B(u)$. If $u$ is unmatched by some *raise* invocation, then the auction terminates with the potential of item $v$ unchanged, and with $u'$ attaining its highest utility of $\mathit{gap}(\chi_B, u) + s_B(u)$. Thus, we have $\mathit{gap}(\mathit{top}'(\chi'), u') = \mathit{gap}(\chi_B, u) + s_B(u)$ $\square$

**Lemma 8.4.4.** *Let $\chi_0$ be an configuration of the form* $\text{subst}(\chi, u, u_0)$ *where $u$ is an agent in* $\text{white}(\chi) \cap \text{matched}(\chi)$*, and $u_0$ is an agent such that $u_0 > u$ and there exists exactly one agent $u''$ in* $\text{agents}(\chi)$ *such that $u < u'' < u_0$. Then* $\text{gap}(\text{top}'(\chi'), u') = \text{gap}(\text{top}'(\chi'_0), u'_0)$ *where $\chi'$ is any configuration of the form* $\text{split}(\chi, u, u')$ *and $\chi'_0$ is any configuration of the form* $\text{split}(\chi_0, u_0, u'_0)$*.*

*Proof.* Let $\chi_A$ be the quiescent configuration such that $\chi' = \mathit{shift}(\chi_A, \mathit{target}(\chi'))$ and let $\chi_B$ be the quiescent configuration such that $\chi'_0 = \mathit{shift}(\chi_B, \mathit{target}(\chi'_0))$. We refer to the executions of the bottom-level auction instances with inputs $(\chi_A, \mathit{target}(\chi'))$ and $(\chi_B, \mathit{target}(\chi'_0))$ as executions $A$ and $B$ respectively. By Lemma 6.3.7, the *raise* invocations by enabled agents commute. Thus, we choose to defer the *raise* invocations of agents in the set $S = \{u', u'_0, u''\}$ in executions $A$ and $B$ until $S$ is the only set of enabled agents. Further, we commute *raise* invocations in both executions such that whenever possible, the same agent invokes *raise* in each round.

For any nonnegative integer $i$, let $\chi_i$ and $\chi'_i$ be the output configurations of round $i$ of executions $A$ and $B$ respectively. We define the predicate $\mathit{sync}(A, B, i)$ to hold if $\chi'_i = \mathit{subst}(\mathit{subst}(\chi_i, u, u_0), u', u'_0)$. We define the predicate $\mathit{coupled}(A, B, i)$ to hold if there exists exactly one maximal set of items $V_i$ in $\mathit{items}(\chi)$ and agents $u_a$ in $\mathit{matched}(\chi_i) \cap \mathit{unmatched}(\chi'_i)$ and $u_b$ in $\mathit{unmatched}(\chi_i) \cap \mathit{matched}(\chi'_i)$ such that for any agent $u_c$ in $\mathit{unmatched}(\chi_i) \cap \mathit{unmatched}(\chi'_i)$ such that $\mathit{items}(\chi_i, u^*) \cap V_i \neq \emptyset$, we have $\mathit{victim}(\chi_i, u_c, 1) = u_a$ and $\mathit{victim}(\chi'_i, u_c, 1) = u_b$.

Consider the first round $j$ such that $\mathit{coupled}(A, B, j)$ holds; then, by the definition of the bottom-level auction, it is easy to see that either: (1) execution $A$ evicts $u'$ in round $j$ and execution $B$ evicts $u''$ in round $j$, or (2) execution $A$ evicts $u$ in round $j$ and execution $B$ evicts $u''$ in round $j$. In case (1), since $u'$ is evicted by execution $A$ in round $j$, and $\mathit{sync}(A, B, j-1)$ holds, agents $u'$ and $u'_0$ have zero utility in round $j$, and by the definition of the bottom-level auction, $u'$ and $u'_0$ continue to have zero utility for the rest of the auction; this completes the proof for case (1).

We now consider case (2). Consider each round $i > j$ of executions $A$ and $B$ where some agent $u_1$ in $\mathit{agents}(\chi) - \{u''\}$ invokes the function *raise*. If $\mathit{items}(\chi_i, u_1) \cap V_i \neq \emptyset$, then it follows that $\mathit{coupled}(A, B, i+1)$ holds. Consider the first round $k > j$ in which some agent $u_1$ invokes *raise* and $\mathit{victim}(\chi_k, u_1, 1) = u''$, thus we have $\mathit{victim}(\chi'_k, u_1, 1) = u_0$. and $\mathit{sync}(A, B, k+1)$ holds; further, it is easy to see that $\mathit{sync}(A, B, i)$ holds for every round $i > k$ in which some agent in $\mathit{agents}(\chi) \setminus S$ invokes the function *raise*.

We now look at executions $A$ and $B$ in a round $k$ when $u''$ is the only enabled agent with pending *raise* invocations. We consider two cases. We first consider the case where $\mathit{sync}(A, B, k-1)$ holds; in this case, $u''$ is the only enabled agent with pending *raise* invocations in both executions $A$ and $B$; thus for any $i \geq k$, if $\mathit{coupled}(A, B, i)$ holds, then condition (1) holds where execution $A$ evicts $u'$ and execution $B$ evicts $u''$ and the proof follows from the analysis of case (1) discussed above. Next, we consider the case where $\mathit{coupled}(A, B, k-1)$ holds; in this case execution $A$ has terminated, and $u''$

is the only enabled agent with pending *raise* invocations in execution $B$, and agents $u''$ and $u'$ are matched in executions $A$ and $B$ respectively. While $u''$ and $u'$ remains allocated for the rest of executions $A$ and $B$, the potentials of items on $P$ remain unchanged; thus $u'$ and $u'_0$ both attain zero utility. If $u'$ is unmatched by some *raise* invocation of execution $B$, then execution $B$ terminates and thus, agents $u'$ and $u'_0$ attain zero utility. □

**Lemma 8.4.5.** *Let* $\chi_0$ *be an configuration of the form* $\text{subst}(\chi, u, u_0)$ *where* $u$ *is an agent in* $\text{white}(\chi) \cap \text{matched}(\chi)$*. Then* $\text{gap}(\text{top}'(\chi'), u') = \text{gap}(\text{top}'(\chi'_0), u'_0)$ *where* $\chi'$ *is any configuration of the form* $\text{split}(\chi, u, u')$ *and* $\chi'_0$ *is any configuration of the form* $\text{split}(\chi_0, u_0, u'_0)$*.*

*Proof.* Without loss of generality, we can assume that $u < u_0$. If there is no agent $u_1$ in *agents*$(\chi)$ such that $u < u_1 < u_0$, then the result follows by repeated application of Lemma 6.3.5. If there is an agent $u_1$ in *agents*$(\chi)$ such that $u < u_1 < u_0$, then the result follows by induction using Lemma 8.4.4. □

**Lemma 8.4.6.** *For any configuration* $\chi$ *and any agent* $u$ *in* $\text{matched}(\chi) \cap \text{white}(\chi)$*, if configuration* $\chi'$ *is of the form* $\text{subst}(\chi, u, u')$*, then* $\text{gap}(\text{top}'(\chi), u) = \text{gap}(\text{top}'(\chi'), u')$*.*

*Proof.* Let $\chi_0$ be an configuration of the form $\mathit{split}(\chi, u, u_0)$ and let $\chi_1$ be an configuration of the form $\mathit{split}(\chi', u', u_1)$. By Lemma 8.4.3, $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\mathit{top}'(\chi_0), u_0)$ and $\mathit{gap}(\mathit{top}'(\chi'), u') = \mathit{gap}(\mathit{top}'(\chi_1), u_1)$. Further, since $\chi' = \mathit{subst}(\chi, u, u')$, by Lemma 8.4.5, we have $\mathit{gap}(\mathit{top}'(\chi_0), u_0) = \mathit{gap}(\mathit{top}'(\chi_1), u_1)$. Thus, we have $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\mathit{top}'(\chi'), u')$. □

**Lemma 8.4.7.** *For any configuration $\chi$ such that $\chi = \text{shift}(\chi_0, \text{target}(\chi))$, any agent $u$ in $\text{matched}(\chi) \cap \text{white}(\chi)$, and any integer $z$ such that $u$ belongs to $\text{gray}(\text{shift}(\chi_0, u, -z))$, we have $\text{gap}(\text{top}'(\chi), u) = \text{gap}(\chi_1, u) + s_1(u)$ where $(\chi_1, s_1) = \text{bottom}(\text{shift}(\chi_0, u, -z), \text{shift}(\text{target}(\chi), u, z))$.*

*Proof.* Let $u'$ be an agent such that $u' < u''$ for every agent $u''$ in $\mathit{agents}(\chi)$. Let $\chi_1'$ be a configuration and let $s_1'$ be a target such that

$$(\chi_1', s_1') = \mathit{bottom}(\mathit{shift}(\mathit{subst}(\chi_0, u, u'), u', -z), \mathit{shift}(\mathit{target}(\chi), u', z))$$

By Lemma 8.4.2, we have $\mathit{gap}(\chi_1', u') + s_1'(u') = \mathit{gap}(\mathit{top}'(\mathit{subst}(\chi, u, u')), u')$.

By Lemma 8.4.6, we have $\mathit{gap}(\chi_1, u) + s_1(u) = \mathit{gap}(\chi_1', u') + s_1'(u')$. By Lemma 8.4.1, we have $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\mathit{top}'(\mathit{subst}(\chi, u, u')), u')$. Thus, we have $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\chi_1, u) + s_1(u)$. □

**Lemma 8.4.8.** *For any configuration $\chi$, any agent $u$ in $\text{agents}(\chi)$, and any configuration $\chi'$ of the form $\text{subst}(\chi, u, \beta)$ where $\beta$ is a bid in $\text{bids}(\text{bid-graph}(\chi))$, we have*

$$\text{gap}(\text{top}'(\chi), u) \geq \text{gap}(\text{subst}(\text{top}'(\chi'), u, \text{bid}(\text{bid-graph}(\chi), u)), u).$$

*Proof.* The analysis for agents in $\mathit{unmatched}(\chi)$ is identical to the analysis for unmatched agents in the proof of Lemma 7.4.1, and the analysis for agents in $\mathit{matched}(\chi) \cap \mathit{nonwhite}(\chi)$ is identical to the analysis for matched nonwhite agents in the proof of Lemma 7.4.1.

We now consider any agent $u$ in $\mathit{matched}(\chi) \cap \mathit{white}(\chi)$. Let $\chi = \mathit{shift}(\chi_0, \mathit{target}(\chi))$ and let $z$ be any integer such that agent $u$ belongs to

$\mathit{gray}(\mathit{shift}(\chi_0, u, -z))$. By Lemma 8.4.7, we have $\mathit{gap}(\mathit{top}'(\chi), u) = \mathit{gap}(\chi_1, u) + s_1(u)$ where $(\chi_1, s_1) = \mathit{bottom}(\mathit{shift}(\chi_0, u, -z), \mathit{shift}(\mathit{target}(\chi), u, z))$. Thus, $u$ obtains the same utility as it would have obtained if its bid had been shifted down sufficiently to make $u$ gray; it follows that we can restrict attention to bottom-level auction instances that have no white matched agents in their input configurations. Let $\chi = \mathit{shift}(\chi_1, s_1)$, where $\chi_1$ is the configuration obtained by shifting down the bid of every white agent in $\chi_0$ such that $\mathit{white}(\chi_1) \cap \mathit{matched}(\chi_1) = \emptyset$. If $(\chi_1', s_1') = \mathit{bottom}(\chi_1, s_1)$, then by definition, we have $\mathit{top}_0'(\chi) = \mathit{shift}(\chi_1', s_1')$. The proof now follows from Lemma 7.4.1. □

**Lemma 8.4.9.** *The first phase of our proposed sealed-bid unit-demand auction is truthful.*

*Proof.* Follows from Lemma 8.4.8 and the definition of truthfulness. □

We use Lemma 8.4.9 and Lemma 7.3.2 on the truthfulness of the second phase to establish Lemma 8.4.10.

**Lemma 8.4.10.** *The proposed sealed-bid unit-demand auction is truthful.*

*Proof.* By Lemma 8.4.9 and Lemma 7.3.2, the first and second phases of our sealed-bid auction are individually truthful. We now show that the sealed-bid auction which combines the two phases is truthful. Consider any instance of our sealed-bid auction with configuration $\chi$ as input and let $u$ be an agent in $\mathit{agents}(\chi)$. Let $\beta = \mathit{bid}(\mathit{bid\text{-}graph}(\chi), u)$ and let $\beta_T \neq \beta$ be the truthful bid of $u$. Let $\chi_T = \mathit{subst}(\chi, u, \beta_T)$. We wish to show that

$\mathit{gap}(\mathit{subst}(\mathit{top}(\chi), u, \beta_T), u) \leq \mathit{gap}(\mathit{top}(\chi_T), u)$. By Lemma 8.3.5, either $u$ belongs to $\mathit{white}(\mathit{top}'(\chi)) \cap \mathit{white}(\mathit{top}'(\chi_T))$ or $\mathit{top}'(\chi_T) = \mathit{subst}(\mathit{top}'(\chi), u, \beta_T)$.

First, we consider the case where agent $u$ belongs to $\mathit{white}(\mathit{top}'(\chi)) \cap \mathit{white}(\mathit{top}'(\chi_T))$. By Fact 7.2.1, we have $\mathit{potential}(\mathit{top}(\chi)) = \mathit{potential}(\mathit{top}'(\chi))$ and $u$ belongs to $\mathit{white}(\mathit{top}(\chi))$; thus $\mathit{gap}(\mathit{top}(\chi), u) = \mathit{gap}(\mathit{top}'(\chi), u)$ and $\mathit{gap}(\mathit{top}(\chi_T), u) = \mathit{gap}(\mathit{top}'(\chi_T), u)$. Further, it follows from Lemma 8.4.8 that $\mathit{gap}(\mathit{top}'(\chi), u) \leq \mathit{gap}(\mathit{subst}(\mathit{top}'(\chi_T), u, \beta_T), u)$. Thus, we conclude that $\mathit{gap}(\mathit{top}(\chi), u) \leq \mathit{gap}(\mathit{subst}(\mathit{top}(\chi_T), u, \beta_T), u)$.

Next, we consider the case where $\mathit{top}'(\chi_T) = \mathit{subst}(\mathit{top}'(\chi), u, \beta_T)$. By Lemma 7.3.2 on the truthfulness of the second phase of our sealed-bid auction, we have $\mathit{gap}(\mathit{top}(\chi), u) \leq \mathit{gap}(\mathit{subst}(\mathit{top}(\chi_T), u, \beta_T), u)$. □

## 8.5 Efficiency

Recall that put options impose lower bound constraints on item prices. As a result, we cannot in general achieve efficiency in our auction setting. Lemmas 8.5.3 and 8.5.4 establish a relaxed form of efficiency for our sealed-bid auction — the outcome is efficient if the target of every put that is exercised satisfies voluntary participation and envy-freedom.

[Definitions] For any configuration $\chi$ such that $\mathit{unmatched}(\chi) \subseteq \mathit{white}(\chi)$ and any agent $u$ in $\mathit{nonwhite}(\chi)$, we define $\mathit{admissible}(\chi, u)$ as the set of all bids $\beta$ in $\mathit{bids}(\mathit{bid\text{-}graph}(\chi))$ such that $u$ belongs to $\mathit{white}(\mathit{subst}(\chi, u, \beta))$. For any configuration $\chi$ such that $\mathit{unmatched}(\chi) \subseteq \mathit{white}(\chi)$, we define $\mathit{admissible}(\chi)$

as the set of all possible configurations that can be obtained from $\chi$ by replacing the bid of every agent $u$ in $\mathit{black}(\chi)$ by a bid in $\mathit{admissible}(\chi, u)$.

**Lemma 8.5.1.** *For any configuration $\chi$ and any agent $u$ in* $\text{matched}(\text{top}(\chi)) \cap \text{nonwhite}(\text{top}(\chi))$*, if $u$ belongs to* $\text{white}(\text{subst}(\chi, u, \beta))$ *where $\beta$ is a bid in* $\text{bids}(\text{bid-graph}(\chi))$*, then $u$ belongs to* $\text{white}(\text{subst}(\text{top}(\chi), u, \beta))$.

*Proof.* By Lemma 8.3.3, we know that there exists an item $v$ in $\mathit{items}(\chi)$ such that $\mathit{potential}(\mathit{top}(\chi), v) = \mathit{potential}(\chi, v)$ and $\mathit{match}(\chi, v) = \mathit{match}(\mathit{top}(\chi), v) = u$. Let $\beta$ be any bid in $\mathit{admissible}(\chi, u)$. By definition, $\beta(v) - \mathit{potential}(\chi, v) \geq \beta(v') - \mathit{potential}(\chi, v')$ for any item $v'$ in $\mathit{items}(\chi)$. By Lemma 7.2.2, it follows that $\mathit{potential}(\mathit{top}(\chi), v') \geq \mathit{potential}(\chi, v)$. Thus, $\beta(v) - \mathit{potential}(\mathit{top}(\chi), v) \geq \beta(v') - \mathit{potential}(\mathit{top}(\chi), v')$ for any item $v'$ in $\mathit{items}(\chi)$. Thus, $\beta$ belongs to $\mathit{admissible}(\mathit{top}(\chi), u)$. $\square$

**Lemma 8.5.2.** *If $\chi$ is an configuration such that* $\text{unmatched}(\chi) \subseteq \text{white}(\chi)$*, then any configuration in* $\text{admissible}(\chi)$ *is white.*

*Proof.* Let $\chi'$ be any configuration in $\mathit{admissible}(\chi)$. By the definition of $\mathit{admissible}(\chi)$, for every agent $u$ in $\mathit{nonwhite}(\chi)$, we have $\mathit{bid}(\chi', u)$ belongs to $\mathit{admissible}(\chi, u)$; thus we have $\mathit{nonwhite}(\chi) \subseteq \mathit{white}(\chi')$. Further, for any agent $u$ in $\mathit{white}(\chi)$, we have $\mathit{bid}(\chi, u) = \mathit{bid}(\chi', u)$, and thus $u$ belongs to $\mathit{white}(\chi')$. Thus, configuration $\chi'$ is white. $\square$

**Lemma 8.5.3.** *For any configuration $\chi$ and any agent $u$ in* $\text{agents}(\chi)$*, if $u$ belongs to* $\text{nonwhite}(\text{top}(\chi))$*, then $u$ belongs to* $\text{nonwhite}(\chi)$ *and* $\text{admissible}(\chi, u) \subseteq$

admissible(top($\chi$), $u$).

*Proof.* By Lemma 8.3.2, we have $\mathit{nonwhite}(\mathit{top}(\chi)) \subseteq \mathit{nonwhite}(\chi)$. Thus, $u$ belongs to $\mathit{nonwhite}(\chi)$. Let $\beta$ be any bid in $\mathit{admissible}(\chi, u)$; then by definition, $u$ belongs to $\mathit{white}(\mathit{subst}(\chi, u, \beta))$. By Lemma 8.5.1, $u$ belongs to $\mathit{white}(\mathit{subst}(\mathit{top}(\chi), u, \beta))$. Thus, $\beta$ belongs to $\mathit{admissible}(\mathit{top}(\chi), u)$. $\square$

**Lemma 8.5.4.** *For any configuration $\chi$,* unmatched(top($\chi$)) $\subseteq$ white(top($\chi$)), *and every configuration in* admissible(top($\chi$)) *is efficient.*

*Proof.* By definition, $\mathit{top}(\chi) = \mathit{top}''(\mathit{top}'(\chi))$. By Fact 7.2.1, we have

$$\mathit{unmatched}(\mathit{top}'(\chi)) = \mathit{unmatched}(\mathit{top}(\chi)) \wedge \mathit{potential}(\mathit{top}(\chi)) = \mathit{potential}(\mathit{top}'(\chi))$$

and by Lemma 8.3.2, we have $\mathit{unmatched}(\mathit{top}'(\chi)) \subseteq \mathit{white}(\mathit{top}'(\chi))$; thus $\mathit{unmatched}(\mathit{top}(\chi)) \subseteq \mathit{white}(\mathit{top}(\chi))$.

Let $\chi' = (G, M, \Phi)$ be any configuration in $\mathit{admissible}(\mathit{top}(\chi))$. By Lemma 8.5.2, $\chi'$ is white, and by Lemmas 3.6.1, $\chi'$ is Walrasian; thus, it follows from Lemma 3.5.2 that $M$ is an MWMCM of $G$. Thus, $\chi'$ is efficient. $\square$

[Definitions] We define a sealed-bid auction to be Pareto-efficient if it is truthful — so that no agent has an incentive to lie — and it satisfies the strong version of Condition 5. Lemmas 8.5.5 and 8.5.6 establish the strong and weak versions of equilibrium condition 5 of Section 4.3.

**Lemma 8.5.5.** *The proposed sealed-bid unit-demand auction is Pareto-efficient when the second phase of the auction is implemented using the $TC^{\ltimes}$ algorithm.*

*Proof.* Consider any configuration $\chi$ and let $\chi' = \mathit{top}(\chi)$. Suppose by way of contradiction that there is a nonempty set of agents $U_0$ who can trade their allocated items amongst themselves such that every agent in $U_0$ experiences an increase in utility. By definition, for any agent $u$ in $\mathit{white}(\chi')$ and any item $v$ in $\mathit{items}(\chi)$, we have $\mathit{gap}(\chi', u) \geq \beta(v) - \mathit{potential}(\chi', v)$, where $\beta = \mathit{bid}(\mathit{bid\text{-}graph}(\chi), u)$. Thus, $\mathit{white}(\chi') \cap U_0 = \emptyset$. It follows that $U_0 \subseteq \mathit{nonwhite}(\chi')$; this contradicts the Pareto-efficient property of the $\mathrm{TC}^{\prec}$ algorithm. Thus, $U_0 = \emptyset$ and our sealed-bid auction is Pareto-efficient. $\square$

**Lemma 8.5.6.** *The proposed sealed-bid auction produces an outcome in the weak core when the second phase of the auction is implemented using the TTC algorithm.*

*Proof.* Consider any configuration $\chi$ and let $\chi' = \mathit{top}(\chi)$. Suppose by way of contradiction that there is a nonempty set of agents $U_0$ who can trade their allocated items amongst themselves such that every agent in $U_0$ experiences an increase in utility. By definition, for any agent $u$ in $\mathit{white}(\chi')$ and any item $v$ in $\mathit{items}(\chi)$, we have $\mathit{gap}(\chi', u) \geq \beta(v) - \Phi'(v)$, where $\beta = \mathit{bid}(\mathit{bid\text{-}graph}(X), u)$ and $\chi' = (G, M', \Phi')$. Thus, $\mathit{white}(\chi') \cap U_0 = \emptyset$. It follows that $U_0 \subseteq \mathit{nonwhite}(\chi')$; this is a contradiction to the well established property that in the absence of strict preferences, the TTC algorithm produces an outcome in the weak core. Thus, $U_0 = \emptyset$. $\square$

## 8.6 Privacy Preservation

A motivating application of the sealed-bid unit-demand auction proposed in this dissertation is the design of a dynamic unit-demand auction in which each round is implemented using the proposed sealed-bid auction. If the seller of an item in the dynamic auction has access to the maximum price that an agent who is tentatively allocated to the item is willing to pay for the item, then the seller can extract this price by submitting a "shill" offer just below the agent's offer. Many dynamic auctions including the popular eBay auction suffer from shill bidding [38]. Thus, a goal of our proposed dynamic auction is to ensure bid privacy for tentatively allocated agents. Below we establish Lemmas 8.6.1 and 8.6.2 which are useful in showing a certain privacy preserving property of the dynamic auction — no seller can artificially raise the price of an item by more than one unit without risking forfeiture of sale. Ideally, we would want to prevent a seller from raising the price of an item even by a single unit; however, our adoption of tie-breaking to handle degeneracy limits our auction to giving up one unit in being shill proof.

Consider an agent $u$ with bid $\beta$, and let $u$ be white and allocated in an outcome of our sealed-bid auction. Lemma 8.6.1 establishes that the outcome of our sealed-bid auction remains unchanged for any bid $\beta'$ of $u$ that exceeds $\beta$ in all its components. Lemma 8.6.2 establishes that if $u$ has a positive utility, then the outcome of our sealed-bid auction remains unchanged if $u$ drops all components of its bid by one unit. It follows from Lemmas 8.6.1 and 8.6.2 that the seller of an item cannot deduce if the offer of an agent who wins the item exceeds the price of the item by more than one unit.

**Lemma 8.6.1.** *For any configuration* $\chi$ *and any agent* $u$ *in* $\text{matched}(\text{top}(\chi)) \cap \text{white}(\text{top}(\chi))$*, we have* $\text{top}(\text{shift}(\chi, u, 1)) = \text{shift}(\text{top}(\chi), u, 1)$.

*Proof.* Let $\chi' = \mathit{shift}(\chi, u, 1)$. Let $\chi = \mathit{shift}(\chi_A, \mathit{target}(\chi))$ and let $\chi' = \mathit{shift}(\chi'_A, \mathit{target}(\chi'))$. We refer to the execution of the bottom-level auction with inputs $\chi_A$ and $\mathit{target}(\chi)$ as execution $R$ and we refer to the execution of the bottom-level auction with inputs $\chi'_A$ and $\mathit{target}(\chi')$ as execution $S$. Let $(\chi_i, s_i)$ and $(\chi'_i, s'_i)$ be the outputs of round $i$ of executions $R$ and $S$ respectively. Let $S$ be the sequence of agents where the $i$th element of sequence $S$, denoted $S(i)$, is the agent that invoked *raise* in round $i$ of execution $R$. Similarly, we define $S'$ to be the sequence of agents that invoked *raise* in execution $S$. Let $j$ be the round in which $u$ makes its last *raise* invocation in execution $R$.

We first claim that $u$ has the same color in configurations $\chi$ and $\chi'$ and that $u$ is not gray in either configuration. If $u$ belongs to $\mathit{gray}(\chi)$, then by the definition of our sealed-bid auction, $u$ belongs to $\mathit{gray}(\chi_0)$ and $\alpha(u) = 0$, where $s = \mathit{target}(\chi)$; thus, $u$ either belongs to $\mathit{gray}(\mathit{top}(\chi))$ or $u$ belongs to $\mathit{unmatched}(\mathit{top}(\chi))$, which is a contradiction. Thus, $u$ does not belong to $\mathit{gray}(\chi)$. Since $\chi' = \mathit{shift}(\chi, u, 1)$, we find that $u$ does not belong to $\mathit{gray}(\chi')$; it is thus straightforward to argue that $u$ has the same color in configurations $\chi$ and $\chi'$.

We now show that (1) for all $i \leq j$, we have $S(i) = S'(i)$ and $\chi_i = \chi'_i$, and (2) $S'_{j+1} = u$.

By definition of our sealed-bid auction, $\chi'_0 = \mathit{shift}(\chi_0, u, 1)$. Further, we established above that agent $u$ is non-gray has the same color in configurations $\chi_0$ and $\chi'_0$. Thus, it follows from the definition of the determinized bottom-level auction that for all $i \leq j$, we have $S(i) = S'(i)$ and $\chi_i = \chi'_i$. Since $u$ belongs to $\mathit{matched}(\mathit{top}(\chi))$ and $u$ invoked its last *raise* in round $j$, we find that $u$ belongs to $\mathit{matched}(\chi_j) \cap \mathit{white}(\chi_j)$. Further, since $\chi_j = \chi_{j'}$, we find that $u$ belongs to $\mathit{matched}(\chi'_j) \cap \mathit{white}(\chi'_j)$ and $\mathit{enabled}(\chi_j) = \mathit{enabled}(\chi'_j)$. Since $S(j) = u$, and by the definition of the function *raise*, no matched agents were enabled in round $j$ of both executions, we find that $u$ belongs to $\mathit{enabled}(\chi_{j+1})$; it follows from these facts that $S'(j+1) = u$.

Next we show that $S(i) = S'(i+1)$ for any $i > j$. We established above that $u$ belongs to $\mathit{matched}(\chi'_j)$ and that $u$ makes its last *raise* invocation of execution $S$ in round $j+1$; thus $s'_{j+1}(u) = 0$. By the definition of the function *raise*, we have $\chi'_{j+1} = \mathit{shift}(\chi'_j, u, 1)$ and $\mathit{enabled}(\chi'_j) = \mathit{enabled}(\chi'_{j+1})$. Since $\chi_j = \chi'_j$, we have $\chi'_{j+1} = \mathit{shift}(\chi_j, u, 1)$ and $\mathit{enabled}(\chi'_{j+1}) = \mathit{enabled}(\chi_j)$; further, we established that $s'_{j+1}(u) = 0$; thus, $S(i) = S'(i+1)$ for $i \geq j$.

We now show that if $S'(j+2) = u'$, then $\mathit{raise}(\mathit{raise}(\chi'_j, u), u') = \mathit{raise}(\mathit{raise}(\chi'_j, u'), u)$. From the preceding claim, we have $S(j+1) = u'$. Since $u$ belongs to $\mathit{matched}(\mathit{top}(\chi)) \cap \mathit{white}(\mathit{top}(\chi))$ and $u$ invoked its last *raise* in round $j$ of execution $R$, it follows that $u \neq \mathit{victim}(\chi_{j+1}, u')$. Since $\chi'_{j+1} = \mathit{shift}(\chi_j, u, 1)$ and $S'(j+2) = u'$, it follows that $u \neq \mathit{victim}(\chi'_{j+1}, u')$; thus, by the definition of the determinized function *raise*, we have $\mathit{raise}(\mathit{raise}(\chi'_j, u), u') = \mathit{raise}(\mathit{raise}(\chi'_j, u'), u)$.

By repeated application of the preceding argument, the last *raise* invocation of $u$ in execution $S$ can be commuted to the last round $k$ of execution $S$. Thus, it follows that $(\chi'_{k-1}, s'_{k-1}) = (\chi_B, \mathit{shift}(s_B, u, 1))$, where $(\chi_B, s_B) = \mathit{bottom}(\chi_A, \mathit{target}(\chi))$.

By the definition of the function *raise*, we have $\mathit{bottom}(\chi_B, \mathit{target}(\chi')) = (\mathit{shift}(\chi_B, u, 1), s_B)$. By the description of the second phase of our sealed-bid auction, it follows that $\mathit{top}(\chi') = \mathit{shift}(\mathit{top}(\chi), u, 1)$. □

**Lemma 8.6.2.** *For any configuration $\chi$ and any agent $u$ in* $\mathrm{agents}(\chi)$*, if* $\mathrm{gap}(\mathrm{top}(\chi), u) > 1$*, then we have* $\mathrm{gap}(\mathrm{top}(\chi_0), u) \geq 1$*, where* $\chi_0 = \mathrm{shift}(\chi, u, -1)$.

*Proof.* Let $\chi = (G, M, \Phi)$. It follows that $\chi_0 = (G_0, M, \Phi)$, where $G_0 = \mathit{shift}(G, u, -1)$. Let $\beta = \mathit{bid}(G_0, u)$. Let $\mathit{top}(\chi_0) = (G_0, M'_0, \Phi'_0)$ and let $\mathit{top}(\chi) = (G, M', \Phi')$. We consider the following cases.

Suppose $u$ belongs to $\mathit{white}(\mathit{top}(\chi_0)) \cap \mathit{matched}(\mathit{top}(\chi_0))$. By Lemma 8.6.1, it follows that $\mathit{top}(\chi) = \mathit{shift}(\mathit{top}(\chi_0), u, 1)$. Thus, $\mathit{gap}(\mathit{top}(\chi_0), u) \geq 1$.

Suppose $u$ belongs to $\mathit{white}(\mathit{top}(\chi_0)) \cap \mathit{unmatched}(\mathit{top}(\chi_0))$. It follows that $\mathit{gap}(\mathit{top}(\chi_0), u) = 0$. Thus, $\beta(v) \leq \Phi'_0(v)$ for every item $v$ in $\mathit{items}(\chi_0)$. However, since $\mathit{gap}(\mathit{top}(\chi), u) > 1$, there exists an item $v$ in $\mathit{items}(\chi_0)$ such that $\mathit{match}(\mathit{top}(\chi), v) = u$ and $\beta(v) - \Phi'(v) \geq 1$. Thus, $\mathit{gap}(\mathit{top}(\chi_0), u) < \mathit{gap}(\chi'', u)$, where $\chi'' = \mathit{subst}(\mathit{top}(\chi), u, \beta)$. This contradicts Lemma 8.4.8.

Suppose $u$ belongs to $\mathit{nonwhite}(\mathit{top}(\chi_0))$. By Lemma 8.3.4, $u$ belongs to $\mathit{nonwhite}(\chi_0)$, and by Lemma 8.3.5, we have $\mathit{top}'(\chi_0) = \mathit{subst}(\mathit{top}'(\chi), u, \beta)$; thus, it follows from the definition of the second phase of our sealed-bid auction that $\mathit{top}(\chi_0) = \mathit{subst}(\mathit{top}(\chi), u, \beta)$. It is now straightforward to see that since $\mathit{gap}(\mathit{top}(\chi), u) > 1$, we have $\mathit{gap}(\mathit{top}(\chi_0), u) \geq 1$. $\square$

## 8.7 Scalability

In this section we briefly sketch a polynomial-time of our proposed sealed-bid auction. We say that a bid component is "active" if it is at least equal to the price (viewing the bid components and prices as pairs, as in the discussion on white configurations in Section 3.6) of the corresponding item. We only need to maintain information concerning the active bid components. We first define an initial tentative pricing and allocation at the start of the auction: each item is allocated to the seller of its put and has a price equal to the strike price of its put. The agents that are not tentatively allocated do not have any active bid components, and so we do not need to maintain any information concerning such agents. We do not maintain an explicit color value (black, gray, or white) for each tentatively allocated agent. Instead, when we need to determine the color of an agent, we do so by examining its active bid components along with the current prices of the associated items.

We now iteratively process bids of unallocated agents. At the start of an iteration, our auction state specifies the current pricing and allocation, the target bid of each tentatively allocated agent, and a set of unallocated agents for which the associated bids have yet to be processed. We pick an arbitrary unallocated agent $u$ from the latter set, and in the style of the well-known Hungarian algorithm for weighted bipartite matching [24], or the closely related successive shortest paths algorithm [2, Chapter 9], we proceed to update the tentative pricing and allocation to account for the bid of $u$. The high-level strategy is to grow a Hungarian tree (which involves increasing certain prices, while maintaining the allocation) rooted at $u$ until one of the following two conditions occurs: (1) one or more nonwhite tentatively allocated agents enter the tree; (2) the utility of $u$ or one or more of the white tentatively allocated agents drops to zero.

If (2) occurs before (1), then we update the allocation via an augmentation that unallocates (and discards) the minimum zero-utility agent, and allocates $u$. (If agent $u$ is itself the minimum zero-utility agent, then no augmentation is performed, and the allocation remains unchanged.) Using a standard primal-dual approach [25], it is possible to update the pricing and allocation in time proportional to the time required to solve a single-source shortest paths (SSSP) problem [8, 11] on the active subgraph of the current bid-graph. For a directed graph with $n$ vertices and $m$ edges, Thorup presents an $O(m + n \log \log n)$ algorithm for the SSSP problem [39]. Thus the time complexity of the update is close to linear in the number of active bid components.

If (1) occurs before (2), then we update the allocation via an augmentation that unallocates the minimum nonwhite tentatively allocated agent, call it $u'$, and allocates $u$. The time complexity for performing this update is the same as in the case of the preceding paragraph. The difference is that here we cannot necessarily discard agent $u'$. In particular, if agent $u'$ was black before the update, then it may still have one or more active bid components; if so, we add agent $u'$ to the set of unallocated agents for which the associated bids have yet to be processed. While the size of the latter set does not decrease (because we removed $u$ and added $u'$), we are able to prove that the number of black tentatively allocated agents has decreased by at least one. Consequently, in any execution of the main auction, the total number of SSSP computations performed is at most the total number of agents in the auction.

Recall that our proposed sealed-bid auction consists of two phases. The foregoing discussion has focused on the implementation of the first phase. In the second phase, any black tentatively allocated agents are given the opportunity to exchange items with one another. As discussed in Section 7.2, either the TTC algorithm or the $\mathrm{TC}^{\prec}$ algorithm is used to update the allocation, and the item prices are left unchanged. The second phase of our sealed-bid auction can be implemented in linear time in the size of the active bid-graph using the TTC algorithm [37], and in polynomial time using the $\mathrm{TC}^{\prec}$ algorithm [19].

# Chapter 9

# A Dynamic Unit-Demand Auction Supporting Arbitrary Bid Revision

In this section, we present our proposed dynamic unit-demand auction supporting arbitrary bid revision. The dynamic auction proceeds in rounds and a single application of the sealed-bid unit-demand auction with put options described in Chapter 8 is used to update the tentative allocation and pricing in each round. The output of the last round determines the final allocation and pricing. Below we give an informal description of the input to each application of the sealed-bud auction.

At the beginning of the first round, the tentative pricing is given by the starting prices of the items. Each item $v$ is tentatively allocated to a "dummy agent" for item $v$ whose bid is a single offer on $v$ equal to the reserve price of $v$. There may be other (non-dummy) agents present in the first round, each of whom has an associated unit-demand bid, which may be arbitrary.

At the beginning of any non-first round, the tentative allocation and pricing is given by the solution to the application of the sealed-bid auction associated with the previous round. The set of agents appearing in the round is equal to the union of the following two sets: (1) agents that were tentatively allocated at the end of the previous round; (2) (non-dummy) agents that were not tentatively allocated at the end of the previous round, and are submitting a new unit-demand bid in the current round. For each agent $u$ in set (1), the associated unit-demand bid in the current round is determined as follows: if $u$ submits a revised bid in this round, then this revised bid is taken to be the bid of $u$; otherwise, the bid of $u$ is taken to be the same as in the previous round. We do not allow a dummy agent to revise its bid, since the bid of a dummy agent is merely intended to model the fixed reserve price of the seller.

The rest of this chapter in organized as follows. In Section 9.1, we provide a formal description of the dynamic auction. In Section 9.2, we introduce some auxiliary definitions that are required in Section 9.3. In Section 9.3, we discuss properties of the dynamic auction related to truthfulness, efficiency, and shill-resistance. In Section 9.4, we discuss an implementation of the dynamic auction with a fast amortized time bound for processing each bidding operation.

## 9.1 Description

The input to the first round of the dynamic auction is a configuration $\chi$ satisfying the following conditions: (1) for any item $v$ is $\mathit{items}(\chi)$, the integer $\mathit{min}(v)$ is equal to the seller-specified starting price of item $v$; (2) there exists exactly $|\mathit{items}(\chi)|$ agents in $\mathit{agents}(\chi)$ that are designated as dummy agents, and for any dummy agent $u$ and any non-dummy agent $u'$ in the set $\mathit{agents}(\chi)$, we have $u < u'$; (3) for any item $v$ in $\mathit{items}(\chi)$, there is a dummy agent $u$ in *agents*$(\chi)$ such that $w(u, v)$ is equal to the seller-specified reserve price of $v$ (which is required to be at least the starting price of $v$), *match*$(\chi, v) = u$, and $w(u, v') = \mathit{min}(v') - 1$ for any item $v'$ in *items*$(\chi) - v$.

Each round of the dynamic auction is resolved using the sealed-bid unit-demand auction with put options described in Chapter 8. We now describe the input of a general non-first round of the auction. Let $\chi = (G, M, \Phi)$ where $G = (U, V, w)$ be the output of round $i - 1$ of the auction. The input to round $i$ is a configuration $\chi'$ of the form $(G', M, \Phi)$ where $G' = (U', V, w')$ satisfying the following conditions: (1) $U$ does not include an agent $u$ in *unmatched*$(\chi)$ if $u$ is either a dummy agent or if $u$ did not submit a bid in round $i$, (2) For each item $v$ in $V$ and for each agent $u$ that is either a dummy agent or is in *matched*$(\chi)$ and did not submit a bid in round $i$, $w'(u, v) = w(u, v)$.

## 9.2 Auxiliary Definitions

A dynamic unit-demand auction $D$ is a sequence of sealed-bid unit-demand auctions where each round of the dynamic auction is resolved using the corresponding sealed-bid auction in the sequence.

For any execution of a dynamic unit-demand auction, we have an associated history of bids that specifies the bids received in each round. We define a bid-history $H$ as a sequence of sets of bids where each set includes a unit-demand bid for each agent in the auction. For any bid-history $H$, we define *length*$(H)$ as the length of the sequence $H$. For any bid-history $H$ and any nonnegative integer $i \leq$ *length*$(H)$, we define *prefix*$(H, i)$ as the prefix of $H$ of length $i$. For any bid-history $H$, we define $\mathit{prefix}(H)$ as $\mathit{prefix}(H, \mathit{length}(H) - 1)$. For any bid-history $H$ and any agent $u$, we define $\mathit{bid}(H, u)$ as the bid of agent $u$ in the last set of bids of sequence $H$. For any bid-history $H$, any agent $u$, and any bid $\beta$, we define $\mathit{subst}(H, u, \beta)$ as the history $H'$ obtained by substituting $\mathit{bid}(H, u)$ with $\beta$.

For any dynamic unit-demand auction $D$ and any bid-history $H$, we define $\mathit{config}(D, H)$ as the output configuration obtained by running auction $D$ on bid-history $H$. It follows that for any dynamic unit-demand auction $D$ and any bid-history $H$, the input and output configurations of each round of auction $D$ can be deduced for the sequence of bids in $H$.

For any configuration $\chi = (G, M, \Phi)$ where $G = (U, V, w)$, and any agent $u$ in $U$, we define $\mathit{envy\text{-}free}(\chi, u)$ to hold if $\mathit{gap}(\chi, u) \geq 0$ and $\mathit{gap}(\chi, u) \geq w(u, v)$ for any item $v$ in $V$. For any configuration $\chi = (G, M, \Phi)$ and any agent $u$ such that $\neg\mathit{envy\text{-}free}(\chi, u)$, we define $\mathit{admissible}(\chi, u)$ as the set of all bids $\beta$ in $\mathit{bids}(G)$ such that $\mathit{envy\text{-}free}(\mathit{subst}(\chi, u, \beta), u)$.

For any bid-history $H$ and any agent $u$, we say $\mathit{submit}(H, u)$ holds if $\mathit{bid}(H, u) \neq \mathit{bid}(\mathit{prefix}(H), u)$.

## 9.3 Properties

Recall that a dynamic auction is essentially a sequence of sealed-bid auctions. We say a dynamic unit-demand auction satisfies property 1 if each round of the dynamic auction satisfies property 1 of Section 4.3. We define what it means for a dynamic unit-demand auction to satisfy properties 2, 3, 4, 5, and 6 similarly. In this section, we establish properties of any dynamic unit-demand auction that satisfies certain subsets of properties 1 through 6. Theorems 9.3.1 and 9.3.2 establish efficiency-related properties of the dynamic auction and are discussed in Section 9.3.2. Theorem 9.3.3 establishes a certain shill-resistant property of the dynamic auction and is discussed in Section 9.3.3.

### 9.3.1 Truthfulness

As we have previously noted in Section 8.4, the sealed-bid unit-demand auction with put options is truthful. Since each round of the dynamic auction is resolved using this sealed-bid auction, it follows that each round of the dynamic auction is truthful.

### 9.3.2 Efficiency

Each round of the dynamic auction implements the solution concept of Section 4.3. Thus, it follows from property 6 that each round of the dynamic auction produces an outcome that is either (strong version) Pareto-efficient, or (weak version) contained in the weak core.

In Theorems 9.3.1 and 9.3.2, we establish efficiency-related properties that hold over multiple rounds of the dynamic auction. We now informally motivate the claims of Theorems 9.3.1 and 9.3.2.

Theorem 9.3.1 establishes that if an agent $u$ is envy-free in a round, then agent $u$ remains envy-free in each subsequent round in which $u$'s preference does not change and $u$ does not submit a bid.

Consider an agent $u$ who is tentatively allocated to an item $v$. Assume that agent $u$ submits a bid revision request in round $i$ of the auction, thereby expressing a desire to be allocated to some item $v'$ different from $v$. After round $i$, agent $u$ may not be envy-free; informally, this means that the revised bid of $u$ is not fully respected by the auction. Theorem 9.3.2 establishes that in each round subsequent to round $i$ in which $u$'s preference does not change, $u$ does not submit a bid revision request, and $u$ remains allocated to the same item, the dynamic auction makes progress towards respecting the revised bid submitted by $u$ in round $i$. Specifically, with each successive round, the revised bid of $u$ can only find better and better approximations in the growing set of admissible bids.

**Lemma 9.3.1.** *For any dynamic unit-demand auction $D$ that satisfies properties 1 and 3, any agent $u$ in auction $D$, and any bid-history $H$, if $u$ belongs to* $\text{unmatched}(\text{config}(D, \text{prefix}(H)))$*, then we have* $\text{envy-free}(\text{config}(D, H), u)$*.*

*Proof.* Let configuration $\chi = \mathit{config}(D, \mathit{prefix}(H))$ and let configuration $\chi' = \mathit{config}(D, H)$. By property 1 of auction $D$, configuration $\chi'$ is semi-Walrasian. If $u$ belongs to $\mathit{unmatched}(\chi')$, then by the definition of semi-Walrasian configurations, we have $\mathit{envy\text{-}free}(\chi', u)$. We now consider the case where $u$ belongs to $\mathit{matched}(\chi')$. Let $v$ be the item such that $\mathit{match}(\chi', v) = u$. Suppose $\neg\mathit{envy\text{-}free}(\chi', u)$; then by the definition of semi-Walrasian configurations and the definition of items that are priced above market, item $v$ is priced above market. By property 4(a) of auction $D$, if $v$ is priced above market, then agent $u$ belongs to *matched*$(\chi)$, a contradiction. Thus, *envy-free*$(\chi', u)$. $\square$

**Lemma 9.3.2.** *For any dynamic unit-demand auction $D$ that satisfies properties 1, 2, 3, and 4, any bid-history $H$, and any agent $u$, if* $\neg$submit$(H, u)$ *and* envy-free(config$(D$, prefix$(H)), u)$*, then* envy-free(config$(D, H), u)$.

*Proof.* Let configuration $\chi =$ *config*$(D,$ *prefix*$(H))$ and let configuration $\chi' =$ *config*$(D, H)$. If agent $u$ belongs to *unmatched*$(\chi)$, then the result follows from Lemma 9.3.1. We now focus on the case where $u$ belongs to *matched*$(\chi)$. By property 1 of auction $D$, configurations $\chi$ and $\chi'$ are semi-Walrasian. Suppose $\neg$*envy-free*$(\chi', u)$; then by the definition of semi-Walrasian configurations, $u$ belongs to *matched*$(\chi')$. Let $v$ and $v'$ be the items in auction $D$ such that *match*$(\chi, v) = u =$ *match*$(\chi', v')$. By the definition of semi-Walrasian configurations and the definition of items that are priced above market, item $v'$ is priced above market. By property 4(a), $v$ is also priced above market. By property 4(b), *gap*$(\chi', u) \geq$ *gap*$(\chi, u)$ and by property 3, *potential*$(\chi, v') =$ *potential*$(\chi', v')$ and *potential*$(\chi, v) =$ *potential*$(\chi', v)$. Since *envy-free*$(\chi, u)$, it follows that *gap*$(\chi', u) =$ *gap*$(\chi, u)$. Finally, by properties 2 and 3, *potential*$(\chi, v'') \geq$ *potential*$(\chi', v'')$ for any item $v''$ in *items*$(\chi) \setminus \{v, v'\}$. It follows that *envy-free*$(\chi', u)$, a contradiction. Thus, *envy-free*$(\chi', u)$. $\square$

**Lemma 9.3.3.** *For any dynamic unit-demand auction $D$ that satisfies properties 1, 2, and 3, any bid-history $H$, and any agent $u$, if* $\neg$envy-free(config$(D, H), u)$

*and if $u$ is matched to the same item $v$ in configurations* $\text{config}(D, \text{prefix}(H))$ *and* $\text{config}(D, H)$*, then*

$$\text{admissible}(\text{config}(D, \text{prefix}(H)), u) \subseteq \text{admissible}(\text{config}(D, H), u)$$

*Proof.* Let configuration $\chi = \textit{config}(D, \textit{prefix}(H))$ and let configuration $\chi' = \textit{config}(D, H)$. By property 1 of auction $D$, configurations $\chi$ and $\chi'$ are semi-Walrasian. Let $\beta$ be any bid in $\textit{admissible}(\chi, u)$. By definition, $\beta(v) - \textit{potential}(\chi, v) \geq \beta(v') - \textit{potential}(\chi, v')$ for any item $v'$ in $\textit{items}(\chi)$. Since $\neg\textit{envy-free}(\chi', u)$ and $\chi'$ is semi-Walrasian, by the definition of items that are priced above market, item $v$ is priced above market. By properties 2 and 3, $\textit{potential}(\chi, v) = \textit{potential}(\chi', v)$ and $\textit{potential}(\chi, v') \leq \textit{potential}(\chi', v')$ for any item $v'$ in $\textit{items}(\chi) - v$. It follows that, $\beta(v) - \textit{potential}(\chi', v) \geq \beta(v') - \textit{potential}(\chi', v')$ for any item $v'$ in $\textit{items}(\chi')$. Thus, $\beta$ is in $\textit{admissible}(\chi', u)$. □

Theorems 9.3.1 and 9.3.2 follow directly by induction on Lemmas 9.3.2 and 9.3.3 respectively.

**Theorem 9.3.1.** *For any dynamic unit-demand auction $D$ that satisfies properties 1, 2, 3, and 4, any bid-history $H$, any prefix $H'$ of bid-history $H$, and any agent $u$ such that* $\text{envy-free}(\text{config}(D, H'), u)$*, if* $\neg\text{submit}(\text{prefix}(H, j), u)$ *for* $\text{length}(H') < j \leq \text{length}(H)$*, then* $\text{envy-free}(\text{config}(D, H), u)$*.*

**Theorem 9.3.2.** *Let $D$ be a dynamic unit-demand auction that satisfies properties 1, 2, and 3, let $H$ be a bid-history, and let $H'$ be a prefix of $H$. Let $u$ be an agent in $D$ such that* $\neg\text{envy-free}(\text{config}(D, H), u)$ *and let $v$ be the item such that*

$\text{match}(\text{config}(D, H), v) = u$. *For each* $j$ *where* $\text{length}(H') < j \leq \text{length}(H)$, *if* $\text{match}(\text{config}(D, \text{prefix}(H, j)), v) = u$ *and* $\neg\text{submit}(\text{prefix}(H, j), u)$, *then*

$$\text{admissible}(\text{config}(D, H'), u) \subseteq \text{admissible}(\text{config}(D, H), u).$$

### 9.3.3 Shill-resistance

If the seller of an item in a dynamic auction has access to the maximum price that an agent who is tentatively allocated to the item is willing to pay for the item, then the seller can extract this price by submitting a shill offer just below the agent's offer. Thus, a goal of the dynamic auction is to ensure bid privacy for tentatively allocated agents. Below we formalize what it means for an agent to be shilled by $\Delta$ units for some nonnegative integer $\Delta$. In Theorem 9.3.3, we establish that no agent in the proposed dynamic auction can be shilled by more than one unit. A consequence of this shill-resistant property is that no seller can artificially raise the price of an item by more than one unit without risking forfeiture of sale. The running time of our auction is independent of the monetary units used; thus each unit can be considered to be as low as one cent, making our auction highly resistant to shilling.

We establish bid privacy of an agent $u$ in the auction with respect to the grand coalition of all agents in the auction except agent $u$. For any dynamic unit-demand auction $D$ and any agent $u$, we define *coalition*$(D, u)$ as the set of all agents in $D$ except agent $u$. The agents in *coalition*$(D, u)$ are assumed to learn the following in each round of auction $D$:

1. the bids of all agents except agent $u$,
2. whether agent $u$ submitted a bid in the round,
3. the shape (relative differences between offers) of agent $u$'s bid, and
4. the pricing and allocation at the end of the round.

Assumption 1 ensures that our dynamic auction preserves the privacy of agent $u$ even when all of the other agents conspire against $u$. Assumptions 2 and 3 ensure that our notion of privacy does not merely exploit the fact that agent $u$ is allowed to submit a bid revision in every round. Assumption 4 is natural since the dynamic auction publishes the tentative outcome in every round.

[Definitions] For any dynamic unit-demand auction $D$, any bid-history $H$, and any agent $u$ in auction $D$, we define $\mathit{possible}(D, H, u)$ as the set of all bids $\beta$ such that $\mathit{config}(D, \mathit{subst}(H, u, \beta)) = \mathit{subst}(\mathit{config}(D, H), u, \beta)$. The set $\mathit{possible}(D, H, u)$ corresponds to the set of possible bids of agent $u$ at the end of the auction that can be deduced by the agents in $\mathit{coalition}(D, u)$. For any dynamic unit-demand auction $D$, any bid-history $H$, any agent $u$ in $D$, and any bid $\beta$ in $\mathit{possible}(D, H, u)$, we define $\mathit{possible}(D, H, u, \beta)$ as the set of all integers $z$ such that $\mathit{shift}(\beta, z)$ belongs to $\mathit{possible}(D, H, u)$.

[Definitions] For any dynamic unit-demand auction $D$, any bid-history $H$, and any agent $u$, we define $\mathit{risk}(D, H, u)$ to hold if

1. agent $u$ belongs to $\mathit{matched}(\mathit{config}(D, \mathit{prefix}(H)))$, and

2. there exists a bid $\beta$ in $\textit{possible}(D, \textit{prefix}(H), u)$ such that agent $u$ belongs to $\textit{unmatched}(\textit{config}(D, \textit{subst}(H, u, \beta)))$.

For any dynamic unit-demand auction $D$, any bid-history $H$, and any agent $u$, we say $u$ is shilled out of $\Delta$ units if $\Delta$ is the maximum integer such that there exists integers $i$ and $j > i$ that satisfy the following conditions:

1. Agent $u$ belongs to $\textit{matched}(\textit{config}(D, \textit{prefix}(H, j)))$
2. $\neg \textit{submit}(\textit{prefix}(H, k), u)$ for each integer $k$ where $i < k \leq j$
3. $\neg \textit{risk}(D, \textit{prefix}(H, k), u)$ for each integer $k$ where $i < k \leq j$
4. $\textit{gap}(\textit{config}(D, \textit{prefix}(H, i)), u) \geq \textit{gap}(\textit{config}(D, \textit{prefix}(H, j)), u) + \Delta$.

**Lemma 9.3.4.** *For any dynamic unit-demand auction $D$ that satisfies property 1, 2, 3, and 4, any bid-history $H$, any agent $u$ in* matched(config($D, H$))*, and any bid $\beta$ in* possible($D, H, u$)*, either* envy-free(config($D, H$)$, u$)*, or* shift($\beta, z$) *belongs to* possible($D, H, u$) *for any integer $z$.*

*Proof.* We use induction on the round number $i$ of the auction. For the base case, we consider $i = 1$, the first round of the auction. By definition, $u$ is unmatched in the input configuration of the first round. By Lemma 9.3.1, it follows that $\textit{envy-free}(\textit{config}(D, \textit{prefix}(H, 1)), u)$.

For the induction step, we consider the case where $i > 1$. By the induction hypothesis, the lemma holds for all $1 \leq j < i$. Let $\chi = \textit{config}(D, \textit{prefix}(H, i-1))$ and let $\chi' = \textit{config}(D, \textit{prefix}(H, i))$. If $u$ belongs to $\textit{unmatched}(\chi)$, then by Lemma 9.3.1, we have *envy-free*$(\chi', u)$. We now consider the case where $u$ belongs to *matched*$(\chi)$. Let $v$ be the item such that *match*$(\chi, v) = u$. We consider the following subcases:

- *envy-free*$(\chi, u)$

  If $\neg$*submit*(*prefix*$(H, i-1), u)$, then by Lemma 9.3.2, we have *envy-free*$(\chi', u)$. We now consider the case where *submit*(*prefix*$(H, i-1), u)$.

  First, we consider the case where $v$ does not belong to *items*$(\chi', u')$ for any agent $u'$ in *unmatched*$(\chi')$, then by using property 1 of auction $D$ and the definition of items that are priced above market, item $v$ is priced above market. By property 3 of auction $D$, *potential*$(\chi, v) =$ *potential*$(\chi', v)$. Since *potential*$(\chi, v) =$ *potential*$(\chi', v)$ and the bid $\beta$ submitted by $u$ could be arbitrary, it follows that for any integer $z$, the bid *shift*$(\beta, z)$ is contained in the set *possible*$(D,$ *prefix*$(H, i), u)$.

  Next, we consider the case where item $v$ belongs to *items*$(\chi', u')$ for some agent $u'$ in *unmatched*$(\chi')$. In this case, by property 1 of auction $D$, configuration $\chi'$ is semi-Walrasian, and by the semi-Walrasian property, we have *envy-free*$(\chi', u)$.

- For any integer $z$, the bid *shift*$(\beta, z)$ is an element of *possible*$(D,$ *prefix*$(H, i-1), u)$.

  First, we consider the case where $v$ does not belong to *items*$(\chi', u')$ for any agent $u'$ in *unmatched*$(\chi')$. In this case, by property 1 of auction

$D$, configuration $\chi'$ is semi-Walrasian, and by definition of items that are priced above market, item $v$ is priced above market. By property 3 of auction $D$, we have $\mathit{potential}(\chi, v) = \mathit{potential}(\chi', v)$. Since $\mathit{potential}(\chi, v) = \mathit{potential}(\chi', v)$ and for any integer $z$, $\mathit{shift}(\beta, z)$ is contained in the set $\mathit{possible}(D, \mathit{prefix}(H, i-1), u)$, it follows that for any integer $z$, $\mathit{shift}(\beta, z)$ is in $\mathit{possible}(D, \mathit{prefix}(H, i), u)$.

Next, we consider the case where $v$ is in $\mathit{items}(\chi', u')$ for some agent $u'$ in $\mathit{unmatched}(\chi')$. In this case, by property 1 of auction $D$, $\chi'$ is semi-Walrasian, and by the semi-Walrasian property of configuration $\chi'$, we have $\mathit{envy\text{-}free}(\chi', u)$.

**Lemma 9.3.5.** *For any dynamic unit-demand auction $D$ that satisfies properties 1 and 6, any bid-history $H$, any agent $u$ in* $\mathrm{matched}(\mathrm{config}(D, H))$*, and any bid $\beta$ in* $\mathrm{possible}(D, H, u)$*, if* $\mathrm{envy\text{-}free}(\mathrm{config}(D, H), u)$*, then there exists a smallest integer $z_0$ such that for any integer $z \geq z_0$,* $\mathrm{shift}(\beta, z)$ *belongs to* $\mathrm{possible}(D, H, u)$*, and either (a) any nonnegative integer is a possible value of* $\mathrm{gap}(\mathrm{config}(D, H), u)$*, or (b) any positive integer is a possible value of* $\mathrm{gap}(\mathrm{config}(D, H), u)$*.*

*Proof.* Let configuration $\chi = \mathit{config}(D, H)$ and let $v$ be the item in auction $D$ such that $\mathit{match}(\chi, v) = u$. Since $\mathit{envy\text{-}free}(\chi, u)$, we have $\mathit{gap}(\chi, u) \geq 0$. It follows that there exists a maximum integer $z'$ such that, $\mathit{gap}(\mathit{shift}(\chi, u, z), u) < 0$ for any integer $z < z'$. By property 1 of auction $D$, $\chi$ is semi-Walrasian; since $\mathit{envy\text{-}free}(\chi, u)$, by definition, $v$ is priced at market. By property 6 of auction

$D$, for any integer $z > z'$, the bid $\mathit{shift}(\beta, z)$ belongs to $\mathit{possible}(D, H, u)$. Let $H' = \mathit{subst}(H, u, \mathit{shift}(\beta, z'))$. By definition, we have $\mathit{gap}(\mathit{shift}(\chi, u, z'), u) = 0$; thus, if $\mathit{envy\text{-}free}(\mathit{config}(D, H'), u)$, then $z_0 = z'$ and any nonnegative integer is a possible value of $\mathit{gap}(\chi, u)$; otherwise, $z_0 = z' + 1$ and any positive integer is a possible value of $\mathit{gap}(\chi, u)$. □

**Lemma 9.3.6.** *For any dynamic unit-demand auction $D$ that satisfies properties 1, 2, 3, 4, and 6, any bid-history $H$, and any agent $u$ in* $\text{matched}(\text{config}(D, H))$, *if* $\neg\text{submit}(H, u)$ *and* $\neg\text{risk}(D, H, u)$, *then for any bid $\beta$ in* $\text{possible}(D, H, u)$,

$$\text{possible}(D, H, u, \beta) = \text{possible}(D, \text{prefix}(H), u, \beta)$$

*Proof.* Let configuration $\chi = \mathit{config}(D, \mathit{prefix}(H))$ and let configuration $\chi' = \mathit{config}(D, H)$. Let $v$ be the item in auction $D$ such that $\mathit{match}(\chi', v) = u$. Since $\beta$ belongs to $\mathit{possible}(D, H, u)$ and $\neg\mathit{submit}(H, u)$, it follows from properties 2 and 3 of auction $D$, that $\beta$ is contained in the set $\mathit{possible}(D, \mathit{prefix}(H), u)$. By Lemma 9.3.4, we know that either $\mathit{envy\text{-}free}(\mathit{config}(D, \mathit{prefix}(H)), u)$, or $\mathit{shift}(\beta, z)$ belongs to $\mathit{possible}(D, \mathit{prefix}(H), u)$ for any integer $z$. We consider the following cases:

- $\mathit{envy\text{-}free}(\chi, u)$

  By Lemma 9.3.2, we have $\mathit{envy\text{-}free}(\chi', u)$. It follows from Lemma 9.3.5 that the possible values for $\mathit{gap}(\chi', u)$ deduced by $\mathit{coalition}(D, u)$ either include $(a)$ all nonnegative integers or, $(b)$ all integers greater than 0. Since $\mathit{envy\text{-}free}(\chi, u)$, by Lemma 9.3.5, there exists a smallest integer $k$ in in $\mathit{possible}(D, \mathit{prefix}(H), u, \beta)$. Since $\neg \mathit{risk}(D, H, u)$ and $\neg \mathit{submit}(H, u)$, it follows that $k$ belongs to $\mathit{possible}(D, H, u, \beta)$. By Lemma 9.3.5, we know that every integer $z > k$ is in $\mathit{possible}(D, H, u, \beta)$. Thus, we have $\mathit{possible}(D, \mathit{prefix}(H), u) \subseteq \mathit{possible}(D, H, u)$. Since $\neg \mathit{submit}(H, u)$, by properties 2 and 3 of auction $D$, we have

$$\mathit{possible}(D, H, u) \subseteq \mathit{possible}(D, \mathit{prefix}(H), u)$$

Thus, $\mathit{possible}(D, H, u) = \mathit{possible}(D, \mathit{prefix}(H), u)$.

- $\mathit{shift}(\beta, z)$ belongs to $\mathit{possible}(D, \mathit{prefix}(H), u)$ for any integer $z$

  By property 1 of auction $D$, configuration $\chi$ is semi-Walrasian. Since $\neg \mathit{risk}(D, H, u)$, item $v$ does not belong to $\mathit{items}(\chi, u')$ for any agent $u'$ in $\mathit{unmatched}(\chi)$. By the semi-Walrasian property of configuration $\chi$, item $v$ is priced above market, and by property 4(a) of auction $D$, $u$ belongs to $\mathit{matched}(\chi)$. Let $v'$ be the item such that $\mathit{match}(\chi, v') = u$. By properties 2 and 3 of auction $D$, $\mathit{potential}(\chi, v) = \mathit{potential}(\chi', v)$, $\mathit{potential}(\chi, v') = \mathit{potential}(\chi', v')$, and $\mathit{potential}(\chi, v'') \geq \mathit{potential}(\chi', v'')$ for any item $v''$ in $\mathit{items}(\chi) \setminus \{v, v'\}$. Thus, $\mathit{possible}(D, \mathit{prefix}(H), u) \subseteq \mathit{possible}(D, H, u)$. Since $\neg \mathit{submit}(H, u)$, by properties 2 and 3 of auction $D$, we have $\mathit{possible}(D, H, u) \subseteq \mathit{possible}(D, \mathit{prefix}(H), u)$. Thus, $\mathit{possible}(D, H, u) = \mathit{possible}(D, \mathit{prefix}(H), u)$.

**Theorem 9.3.3.** *For any dynamic unit-demand auction $D$ that satisfies properties 1, 2, 3, 4, and 6, and any bid-history $H$, no agent in auction $D$ can be shilled by more than one unit.*

*Proof.* Suppose there exists an agent $u$ who is shilled by $\Delta > 1$ in an execution of auction $D$ with bid-history $H$. Then, by definition, there exists integer $i$ and $j$, where $j > i$m such that the following conditions hold:

1. $u$ belongs to $\mathit{matched}(\mathit{config}(D, \mathit{prefix}(H, j)))$,
2. $\neg \mathit{submit}(\mathit{prefix}(H, k), u)$ holds for each integer $k$ where $i < k \leq j$,
3. $\neg \mathit{risk}(D, \mathit{prefix}(H, k), u)$ holds for each integer $k$ where $i < k \leq j$, and
4. $\mathit{gap}(\mathit{config}(D, \mathit{prefix}(H, i)), u) \geq \mathit{gap}(\mathit{config}(D, \mathit{prefix}(H, j)), u) + \Delta$.

Let $\chi = \mathit{config}(D, \mathit{prefix}(H, i))$. If $\neg \textit{envy-free}(\chi, u)$ then $\beta$ can be arbitrary, and any integer is a possible value of $\mathit{gap}(\chi, u)$. If $\textit{envy-free}(\chi, u)$, then by Lemma 9.3.5, the possible values of $\mathit{gap}(\chi, u)$ either includes $(a)$ all nonnegative integers, or $(b)$ all integers greater than 0. When zero is a possible value for $\mathit{gap}(\chi, u)$, it is easy to see that $u$ cannot be shilled even by a single unit without $\mathit{coalition}(D, u)$ risking forfeiture of sale. We now consider the case where $\textit{envy-free}(\chi, u)$ holds and any positive integer is a possible value of $\mathit{gap}(\chi, u)$. By Lemma 9.3.6, we know that $\mathit{possible}(D, \mathit{prefix}(H, k), u, \beta) = \mathit{possible}(D, \mathit{prefix}(H, k+1), u, \beta)$ for $i \leq k < j$. Thus, if $u$ is shilled by one unit in a round $k$ where $i < k \leq j$, then zero is a possible value of $\mathit{gap}(\mathit{config}(D, \mathit{prefix}(H, l)), u)$ for $k < l \leq j$ and $u$ cannot be shilled further, a contradiction. Thus, $u$ cannot be shilled by more than a unit. □

## 9.4 Scalability

In this section, we briefly sketch the details of a fast implementation of the dynamic auction. In each round of the dynamic auction, new bid data (i.e., bid revision requests from tentatively allocated agents, and bids from unallocated agents) is received and the tentative allocation and pricing is updated using an instance of the sealed-bid unit-demand auction with put options. Recall from Section 8.7 that in the first round of the sealed-bid auction, the bid of each unallocated agent can be processed in time proportional to the time required to solve a single-source shortest paths (SSSP) problem on the active subgraph of the associated bid-graph.

At the start of a round, one or more tentatively allocated agents are not envy-free. If a tentatively allocated agent $u$ who is not envy-free becomes unallocated in the round, then $u$ is added to the set of unallocated agents whose bids are yet to be processed. For each such agent $u$, the number of tentatively allocated agents who are not envy-free decreases by at least one as the only way a tentatively allocated agent can cease to be envy-free is by revising its bid. Furthermore, when we use an SSSP computation to process the bid of the now-unallocated agent $u$, we can charge the cost of this SSSP computation to the most recent bid revision of $u$. Consequently, the total number of SSSP computations performed across all rounds is at most the total number of bidding operations (i.e., bid revisions or new bids) over all rounds.

For our fast implementation, we choose the TTC [37] algorithm to implement the second phase of the sealed-bid auction used in each round. From Section 8.7, the TTC algorithm can be implemented in time linear in the size of the active subgraph.

In summary, it is possible to implement the dynamic auction in such a way that the amortized cost of each bidding operation is close to linear in the size of the active subgraph of the bid-graph, which is at most quadratic in the number of items. Moreover, in many practical auction settings, the average number of active bid components of a tentatively allocated agent is likely to be small, say at most a constant. In such settings, the number of active bid components is linear in the number of items, and hence the amortized cost of each bidding operation is close to linear in the number of items.

# Chapter 10

# Sniping Fees

In this section we describe how to modify our dynamic unit-demand auction to encourage early bidding, while preserving all of the theoretical properties established earlier in the preceding chapters. In Section 10.1 we review a standard technique for incorporating agent-specific adjustments into the selling prices of the items. In Section 10.2, we generalize this technique to allow for price adjustments that may increase from one round to the next. Such dynamic price adjustments are used to discourage "sniping", i.e., waiting until close to the end of the auction to submit a bid. Sniping diminishes agents' ability to focus value discovery efforts on the most relevant items, thereby increasing participation costs and degrading efficiency.

## 10.1 Static Price Adjustments

At the conclusion of a typical single-item online auction, the item is shipped to the winning agent. The winning agent pays the auction price plus shipping costs. The cost of shipping is agent-specific, in general, because it may vary depending on the shipping address. The cost of shipping is typically made available to the agents during the auction (e.g., via a shipping calculator).

Viewed more abstractly, the seller publishes a static function $adjustment(u)$ as part of the auction listing, and if agent $u$ wins the auction, then agent $u$ pays the auction price plus $adjustment(u)$.

Such an abstraction is also useful for selling an item with multiple "variants". For example, consider a computer that can be sold with or without a monitor. The auction listing for the computer might specify an additional charge for the optional monitor. Such variant-related charges might be agent-specific (e.g., due to the cost of shipping the monitor), and in general might be positive or negative. The auction listing of the seller provides the necessary information (e.g., shipping calculator, fixed price adjustments for different variants) to allow each agent $u$ to determine the relevant price adjustment $adjustment(u)$ to be paid in the event that agent $u$ wins the auction. We view the adjustment as a function of the agent only, as opposed to the agent and variant, because the agent selects the relevant variant based on the published cost adjustments. In this sense, the problem of supporting multiple variants of an item is reduced to the single-variant case.

The framework discussed above generalizes immediately to the unit-demand setting, where we have a static price adjustment function $adjustment(u, v)$ that specifies the amount to be added to the auction price to determine the total price paid by agent $u$ for item $v$. It is natural to ask whether the theoretical properties established for our auction continue to hold in the presence of such an adjustment function. Apart from the price adjustment performed at the end of the auction, the computations performed by our auction depend only on the non-adjusted bids. Consequently, it is straightforward to argue that all of the theoretical properties established for our auction continue to hold with respect to the non-adjusted bids/prices. (Regarding our claim that each individual round of our auction is truthful, we point out that a non-adjusted bid of agent $u$ is truthful if the corresponding adjusted bid is equal to the truthful preferences of agent $u$.)

## 10.2 Dynamic Price Adjustments

The static price adjustment framework discussed in Section 10.1 reflects standard practice in single-item auctions, and generalizes straightforwardly to the unit-demand setting. We now introduce a variation of this framework in which there is a separate price adjustment function $\mathit{adjustment}_i$ for each round $i$ of the auction. We require that the choice of the function $\mathit{adjustment}_i$ is determined by the public component of the bidding history up to the start of round $i$, and that for any agent $u$, item $v$, and rounds $i$ and $j$ such that $i < j$, we have $\mathit{adjustment}_i(u, v) \leq \mathit{adjustment}_j(u, v)$.

When an agent $u$ wins an item $v$, agent $u$ pays the auction price plus $\mathit{adjustment}_i(u, v)$, where $i$ is the index of the earliest round such that for all rounds with index at least $i$, the output configuration $\chi = (G, M, \Phi)$ satisfies $w(u, v) - \Phi(v) \geq \min(0, \mathit{gap}(\chi, u))$. Roughly speaking, the latter condition checks whether agent $u$'s unit-demand bid still has the possibility of winning item $v$ in a later round, even if it is left unchanged.

Reasoning as in the case of a static price adjustment function discussed in Section 10.1, it is straightforward to argue that our dynamic price adjustment scheme continues to enjoy all of the theoretical properties established in Chapter 9. In this regard, we remark that our requirement that function $adjustment_i$ is determined by the public component of the bidding history up to the start of round $i$ ensures that the shill-resistance of the auction is preserved. (In the absence of this requirement, the choice of the function $adjustment_i$ could reveal private information related to the bids of the tentatively allocated agents.) The scalability of our auction is unaffected since it is easy to compute the price adjustments to be applied at the end of the auction. Indeed, we can compute tentative price adjustments at the end of each round without increasing the asymptotic complexity of processing a round.

The requirement that $adjustment_i(u, v)$ is nondecreasing in $i$ is motivated by our desire to encourage early bidding in the auction. Suppose agent $u$ wins item $v$, and our price adjustment rule prescribes that agent $u$ pays the auction price plus $adjustment_i(u, v)$. We view the nonnegative value obtained by subtracting $adjustment_1(u, v)$ from $adjustment_i(u, v)$ as the "sniping fee" incurred by agent $u$ for not bidding earlier. (Remark: The term "sniping" is often used narrowly to refer to submitting a bid in the last few seconds of an auction. Here we use the term more broadly, since our sniping fee structure can be multi-tiered to discriminate between bids submitted with arbitrarily varying amounts of time remaining in the auction.)

We now describe a simple but practically important use case of the dynamic price adjustment framework introduced above. Consider a unit-demand auction in which the listing of each item $v$ specifies — at the outset of the auction — the value of $adjustment_i(u, v)$ for all agents $u$ and rounds $i$. For $i = 1$, these values can be used to model shipping costs and item variants as discussed in Section 10.1. The sniping fee applicable to bids submitted in the first round is zero. For any agent $u$ and round $i > 1$, the quantity $adjustment_i(u, v) - adjustment_{i-1}(u, v)$ models the nonnegative change in sniping fee from round $i - 1$ to round $i$. A potential drawback of this scheme is that the sniping fees for an item $v$ accumulate even while the item remains tentatively allocated to the dummy agent for item $v$ (because the reserve price has not been reached). It may be preferable for the seller of item $v$ to specify how sniping fees are to grow once the reserve price has been met. Such a more complex sniping fee schedule — which has a nontrivial dependence on the bidding history — still falls well within the general dynamic price adjustment framework introduced above.

We now discuss considerations associated with the design of a suitable sniping fee schedule for a given item in the auction. For the sake of concreteness, we pursue this discussion in the context of a specific popular online auction format: A continuous auction with a fixed one-week duration. (In a continuous auction, the tentative pricing and allocation is updated immediately once a bidding operation is received. Equivalently, we can imagine that each round corresponds to a fixed, infinitesimally small, interval of time.) Similar considerations arise in the design of sniping fee schedules for other auction formats.

Under one simple sniping fee schedule for a one-week continuous auction, the sniping fee might increase linearly from zero — at the time when the reserve is met — to a seller-specified maximum value at the end of the auction. However, such a schedule is unlikely to be appropriate. Notice, for example, that the sniping fee would remain virtually constant over the last hour of the auction. From the perspective of allowing competing agents to engage in additional value discovery in response to one's bid, there is a significant difference between bidding with ten seconds left in the auction, with one minute left, with five minutes left, or with an hour left. This observation leads us to conclude that the sniping fee should increase more rapidly as the time remaining in the auction diminishes. For example, it might be reasonable to make the sniping fee proportional to the logarithm of the ratio of the auction duration to the time remaining. Doing so results in an additive increase in the sniping fee whenever the time remaining decreases geometrically, and a geometric decrease in the time remaining has a qualitative impact on the ability of agents to engage in value discovery.

# Chapter 11

# Concluding Remarks

The two main contributions of this dissertation are (1) a sealed-bid unit-demand auction with put options, and (2) a dynamic unit-demand auction supporting arbitrary bid revision. Our proposed dynamic unit-demand auction is essentially a generalization of the eBay auction for the unit-demand setting and has practical potential. In this section, we offer some recommendations and extensions for our dynamic auction.

In the foregoing presentation of our dynamic auction, we have assumed that the set of items for sale in the auction is static. It is straightforward to modify the auction to allow new items to be introduced in each round. Further, we have assumed that all items in the auction have the same expiry time. It is possible to relax this assumption. For example, we can specify a separate expiration time for each item in the auction, and allow unit-demand bidding across items that expire within the same interval of time (e.g., the same day).

The binary string identifiers associated with agents and items in our dynamic auction provide fixed total orders over the sets of agents and items. The properties of our auction continue to hold even if these total orders are changed from one round of the auction to the next. We recommend using a single total order over the items for all rounds of the dynamic auction; for example, this total order could be derived by sorting a fixed set of item identifiers. We recommend a slightly more complex scheme for determining the total order over the agents to be used in each round. First, we recommend that all dummy agents be ordered lower than all non-dummy agents in every round; this ensures that an item can be sold to a non-dummy agent at the starting price. Within the set of dummy agents, we recommend using the same (arbitrary) fixed total order in all rounds.

Within the set of non-dummy agents, we recommend using a dynamic timestamp-based ordering, where the timestamp of an agent is determined as follows. In the first round, all agents are assigned the same timestamp. In any non-first round $i$, recall that the agents may be partitioned into sets (1) agents that were tentatively allocated at the end of the previous round; and (2) (non-dummy) agents that were not tentatively allocated at the end of the previous round, and are submitting a new unit-demand bid in the current round. Timestamp $i$ is assigned to all of the agents in set (2). Each agent $u$ in set (1) is assigned the minimum timestamp $j$ less than $i$ such that $u$ is allocated in the solution associated with all applications of the sealed-bid unit-demand auction with put options in rounds $j$ through $i-1$. Having determined these timestamps, we recommend ordering any pair of agents $u$ and $u'$ participating in round $i$ as follows: if $u$ and $u'$ have distinct timestamps, then the agent with the higher timestamp is considered lower; if $u$ and $u'$ have equal timestamps, then the order of the agents is determined by an arbitrary fixed total order. The motivation for the proposed dynamic timestamp-based scheme is that it breaks ties in favor of agents who have been allocated longer.

The single-item auction mechanism employed by eBay is essentially a dynamic second-price auction. We have shown how to generalize this popular auction format to the unit-demand case, while supporting arbitrary bid revision by tentatively allocated agents. Our auction maintains strong properties related to efficiency, truthfulness, privacy preservation, and scalability. We have implemented our auction in Java and verified that it is capable of processing large numbers of bidding operations per second. Such speed is important in practice, since it is desirable for a dynamic auction to compute and publish updates to the pricing and allocation in real time.

# Bibliography

[1] G. Aggarwal, S. Muthukrishnan, D. Pál, and M. Pál. General auction mechanism for search advertising. In *Proceedings of the 18th World Wide Web Conference*, volume 18, 2009.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows.* Prentice Hall, New York, NY, 1993.

[3] L. Ausubel. An efficient ascending-bid auction for multiple objects. *American Economic Review*, 94(5):1452–1475, 2004.

[4] L. Ausubel. An efficient dynamic auction for heterogeneous commodities. *American Economic Review*, 96(3):602–629, 2006.

[5] L. M. Ausubel, P. Cramton, and P. Milgrom. *The Clock-Proxy Auction: A Practical Combinatorial Auction Design.* MIT Press, Cambridge, Massachussetts, 2005.

[6] S. Bickchandani and J. Ostroy. Ascending price Vickrey auctions. *Games and Economic Behavior*, 55(2):215–241, 2006.

[7] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C Stein. *Introduction to Algorithms*. MIT Press, Cambridge, Massachussetts, 1990.

[9] S. de Vries, J. Schummer, and R. Vohra. On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, 132(1):95–118, 2007.

[10] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[11] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[12] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, Massachussetts, 1993.

[13] S. Fujishige and A. Tamura. A two-sided discrete-concave market with possibly bounded side payments: An approach by discrete concave analysis. *Mathematics of Operations Research*, 32(1):136–155, 2007.

[14] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[15] J. Green and J. J. Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 44:427–438, 1977.

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

[17] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.

[18] F. Gul and E. Stacchetti. The English auction with differentiated commodities. *Journal of Economic Theory*, 92(1):66–95, 2000.

[19] P. Jaramillo and V. Manjunath. The difference indifference makes in strategy-proof alloocatin of objects. *Under Review*, 2009.

[20] T. Koopmans and M. Beckmann. Assignment problem and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[21] V. Krishna and M. Perry. Efficient mechanism design. Technical report, Pennsylvania State University, 1998.

[22] C. Krishnappa and C. G. Plaxton. A dynamic unit-demand auction supporting arbitrary bid revision. In *Proceedings of the 13th International Conference on Electronic Commerce*, August 2011.

[23] C. Krishnappa and C. G. Plaxton. A sealed-bid unit-demand auction with put options. In *Proceedings of the 22nd International Conference on Game Theory*, July 2011.

[24] H. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 3:253–258, 1955.

[25] H. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *Journal of Political Economy*, 91(3):461–479, 1983.

[26] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic Theory.* Oxford University Press, 1995.

[27] P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

[28] D. Mishra and D. C. Parkes. Ascending price Vickrey auctions for general valuations. *Journal of Economic Theory*, 132(1):335–366, 2007.

[29] D. Mishra and D. C. Parkes. Multi-item Vickrey-Dutch auctions. *Games and Economic Behavior*, 66(1):326–347, 2009.

[30] R. Myerson. Incentive compatibility and the bargaining problem. *Econometrica*, 47:61–73, 1979.

[31] R. Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.

[32] J. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 36, pages 48–49, 1950.

[33] J. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, New Jersey, 1947.

[34] D. C. Parkes. Iterative Combinatorial Auctions. In P. Cramton, Y. Shoham, and R Steinberg, editors, *Combinatorial Auctions*, chapter 2. MIT Press, Cambridge, Massachussetts, 2006.

[35] H. Rothkopf, M, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1999.

[36] J. Schummer and R. V. Vohra. Mechanism design Without Money. In N. Nisan, T. Roughgarden, É. Tardos, and V Vazirani, editors, *Algorithmic Game Theory*, chapter 10. Cambridge University Press, Cambridge, Massachussetts, 2007.

[37] L. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974.

[38] K. Steiglitz. *Snipers, Shills & Sharks: eBay and Human Behavior*. Princeton University Press, Princeton, New Jersey, 2007.

[39] M. Thorup. Integer priority queues with decrease key in constant time and the single source shortest paths problem. *Journal of Computer and System Sciences*, 69(3):330–353, 2004.

[40] H. Varian. Economic mechanism design for computerized agents. In *Proceedings of the 1st USENIX Conference on Electronic Commerce*, 1995.

[41] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[42] L. Walras. *Elements of Pure Economics*. Allen and Unwin, 1954.

[43] S. Williams. A characterization of efficient, Bayesian incentive-compatible mechanisms. *Economic Theory*, 14:155–180, 1999.

[44] P. Wurman and M. Wellman. A parameterization of the auction design space. *Games and Economic Behavior*, 35:304–338, 2001.

# Vita

Chinmayi Krishnappa was born in Bangalore, India, the daughter of C. K. Kumudini and Dr. P. Krishnappa. She received her Bachelor's degree in Computer Science from Visveswaraiah Technological University in India. After a short stint in the software industry, she enrolled at the University of Texas at Austin and earned a PhD in Computer Science in 2011. She was the recipient of the MCD (Microelectronics and Computer Development) fellowship and the Dean's Excellence Award at the University of Texas at Austin.

Permanent address: No. 19, H.M.T Layout
R.T Nagar Main Road
Bangalore 560 032, India

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.