

A Dynamic Unit-Demand Auction Supporting Bid Revision*

Chinmayi Krishnappa
chinmayi@cs.utexas.edu

C. Greg Plaxton
plaxton@cs.utexas.edu

July 31, 2010

Abstract

We specify rules for a dynamic unit-demand auction that supports arbitrary bid revision. Each round of our dynamic auction is implemented via a single application of a novel sealed-bid unit-demand auction that 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. In order to specify the input-output behavior of our sealed-bid auction, we develop a framework that is based in part on a natural iterative bargaining process, and that guarantees certain strong properties related to efficiency and privacy preservation. Within this framework, we succeed in refining the design of our sealed-bid auction to ensure that it is truthful. We present a fast algorithm for implementing the proposed dynamic 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.

*Department of Computer Science, University of Texas at Austin, Austin, TX 78712. Supported by NSF Grant CCF-0635203 and THECB NHARP Grant 003658-0235-2007.

1 Introduction

Consider an auction in which many different items are for sale. Assume that a bidding agent assigns a separate value to each item, and is interested in acquiring at most one item. Such an agent is said to have unit-demand preferences. In a unit-demand auction, the bid of an agent takes the same form as a unit-demand preference function: The agent specifies an offer for each item, 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 remaining 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 indeed 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.

Unit-demand auctions are well understood in the standard sealed-bid framework alluded to above. In this context, we can apply the well-known Vickrey-Clarke-Groves (VCG) mechanism to obtain a truthful auction that produces an efficient allocation and envy-free pricing [17]. In the discussion that follows, we refer to this standard sealed-bid unit-demand auction as Auction A. The allocation returned by Auction A corresponds to a maximum-weight matching of a suitable edge-weighted bipartite graph. Such a matching may be computed efficiently using the Hungarian method [9], or the closely related successive shortest paths algorithm [1, Chapter 9]. The dual variables maintained by either of these algorithms directly provide the desired item prices, which are uniquely determined.

We propose a novel set of rules for running a unit-demand auction. Our auction is dynamic, meaning that it proceeds in 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 solution (allocation and pricing). The tentative solution is made public at the end of each round. The final solution is given by the allocation and pricing at the end of the last round. We show that our update rule, which corresponds to the mid-level auction discussed below, satisfies a number of desirable mathematical properties.

The primary distinguishing feature of our dynamic unit-demand auction is its support for arbitrary bid revision by tentatively allocated agents. Agent valuations are not necessarily fixed, even in single-item auctions, but are estimates that evolve over time. An agent may update his preset valuation for an item based on several factors that include bidding history and research on the item. For instance, Myerson [12] designs a single-item auction in which agents revise their valuations based on preference and quality uncertainty factors. Unit-demand bids are more expressive than the single-item bids of traditional auctions, and bid formulation is correspondingly more complex. Accordingly, 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. Such an artificial reduction in the number of bids undercuts the main value proposition of unit-demand auctions, i.e., improved efficiency.

It is worth pointing out that in the absence of bid revision by tentatively allocated agents, we can proceed by simply applying the same approach as in Auction A to process the set of new bids received in each round. In the discussion that follows, we refer to this dynamic variant of Auction A as Auction B. We now highlight several of the key theoretical properties that Auction B inherits from Auction A. Each round of Auction B is truthful. The allocation computed in each round of Auction B is efficient with respect to the set of bids thus far submitted. Auction B also preserves the privacy of the bids associated with the tentatively allocated agents, in the following sense: If agent u is tentatively allocated to item v at price p , then no coalition of agents and sellers that does not include u can determine the amount by which the offer of agent u for item

v exceeds p . The latter privacy preservation property is crucial to the integrity of Auction B, because it means that such a coalition cannot force agent u to pay a higher price without risking forfeiture of sale. Auction B is also scalable in the sense that it admits a fast implementation. Each new bid can be processed using a single Hungarian augmentation. 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.) The high-level objective of the present paper is to support arbitrary bid revision by tentatively allocated agents, while preserving — to the extent possible — the theoretical properties of Auction B.

Auction B is inherently ascending-price because enlarging the set of unit-demand bids cannot cause any price to fall. However, we are interested in the setting where tentatively allocated agents are allowed to revise their bids, and in this setting, we need to decide whether we are willing to allow prices to fall. Allowing prices to fall is incompatible with achieving privacy preservation in the sense achieved by Auction B. For this reason, we opt to enforce the ascending-price property. Given the broadly applicable nature of the VCG paradigm, one might expect that a VCG mechanism can be used in each round to determine the appropriate price increase for each item. For example, one might consider applying Auction A in each round to the modified instance obtained by shifting the price of each item to zero, and applying the same shift to all bids on that item. Unfortunately, the VCG framework is not directly applicable to our problem; the fundamental reason is that while the bids assign values to all feasible outcomes, they also serve to restrict the set of feasible outcomes. Accordingly, we proceed to design and analyze our auction from first principles instead of relying on the VCG framework. At the same time, we use the VCG-based Auction A as a building block to specify the input-output behavior of our auction, and our analysis exploits known properties of Auction A.

Within the technical body of this paper, we present three unit-demand auctions that we refer to as the top-level, mid-level, and bottom-level auctions. The top-level auction is our proposed dynamic unit-demand auction. Each round of the top-level auction corresponds to a single application of the mid-level auction. Like Auction A, the mid-level auction is a sealed-bid unit-demand auction; however, as discussed in greater detail below, the input to the mid-level auction specifies a tentative allocation and allows for arbitrary bid revision, resulting in additional complexity. The mid-level auction consists of two phases. The first phase, which affects both the tentative allocation and pricing, is defined in terms of the bottom-level auction. The bottom-level auction is dynamic. We associate with each mid-level agent u a bottom-level proxy agent u' . Given the sealed unit-demand bid of u , the associated proxy u' employs a fixed strategy to bid on behalf of u in each round of the bottom-level auction. The second phase of the mid-level auction affects only the tentative allocation, and uses a single application of the well-known Top Trading Cycles (TTC) algorithm [14] to exchange items within a certain subset of the tentatively allocated agents.

The design of the first phase of our mid-level auction constitutes a key technical component of our work. A formal description of this component may be found in Section 5. 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 to which v is tentatively allocated: Agent u will remain allocated to v at the tentative price of v . The first phase then proceeds to update the tentative allocation and pricing in an iterative manner. Apart from updating the allocation and pricing, 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. But should we expect such an iterative procedure to be truthful? By lying, an agent can alter the set of initially tentatively allocated agents that are released from their obligations, and can thereby alter the set of bids that are taken into account in determining the final allocation and pricing. Accordingly, one might expect lying to be advantageous in at least some situations. This intuition was borne out in our initial — and unsuccessful — attempt to design the mid-

level auction, in which we implemented each iteration of the first phase by solving a suitable instance of Auction A. Fortunately, and somewhat surprisingly, we are able to obtain a truthful mid-level auction by carefully restricting the nondeterminacy associated with these instances of Auction A. Specifically, in scenarios where there are multiple allocations of maximum weight, and Auction A selects an arbitrary such allocation, we employ a natural tie-breaking convention to restrict the choice of allocation. Under this tie-breaking convention, the allocation is still not uniquely determined, but the set of allocated agents is uniquely determined. Moreover, the tie-breaking convention is easy to enforce, allowing for a fast implementation.

In the foregoing discussion, we have placed special emphasis on the truthfulness property of the mid-level auction. In the technical body of the paper, we establish various relevant properties of the bottom-level, mid-level, and top-level auctions, related to efficiency, truthfulness, privacy preservation, and scalability. To give the reader a sense for the strength of our results, recall that our high-level objective is to match — to the extent possible — the properties achieved by Auction B, while supporting (unlike Auction B) bid revision. Much of the technical development of the paper is geared towards establishing that, like Auction B, each round of our top-level auction is truthful (see Lemma 5.22). We also prove (see Lemma 6.5) that our top-level auction achieves the same strong privacy preservation property as Auction B. With respect to scalability, our fast implementation of the top-level auction (see Section 6.5) processes each bidding operation (i.e., new bid or bid revision) using an amortized constant number of augmentations, thereby matching the asymptotic complexity associated with Auction B, which uses a single augmentation to process each new bid. With regard to efficiency, recall that Auction B produces an efficient allocation in each round. In the absence of bid revision by tentatively allocated agents, our top-level auction mimics the behavior of Auction B, and hence achieves the same efficiency guarantee. When tentatively allocated 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 maintain a relaxed form of efficiency (see Lemmas 6.1 and 6.2). Informally, our efficiency-related guarantees imply that 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. Moreover, we show that the auction can only make progress towards achieving efficiency with respect to the most recent revision of a given bid. We believe that our efficiency-related guarantees are essentially the strongest that can be achieved without sacrificing other properties.

Our proposed auction is relatively straightforward to specify. Indeed, the informal description given earlier in this introduction is not far from complete. However, the analysis of our auction is highly nontrivial. For example, it is a major challenge to prove that each round of the auction is truthful. A second major challenge is to justify the correctness of our fast implementation. To overcome these challenges, our formal development introduces a good deal of technical machinery, resulting in a lengthy presentation. To reduce the length of this submission, we have chosen to present some basic lemma statements without proofs. Given the formal framework established, the proofs that have been omitted are straightforward and are not too difficult to prove.

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 provides a foundation for the technical presentation to follow. Sections 4, 5, and 6 present the bottom-level, mid-level, and top-level auctions, respectively. Section 7 offers some concluding remarks.

2 Related Work

Demange et al. [6] 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 overdemand set of items, a minimal overdemand 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. Assuming that each agent behaves in accordance with a fixed unit-demand bid, the input-output behavior of the DGS-exact auction is exactly the same as that of Auction A described in Section 1. 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 [11] 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 δ , 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 δ ; 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; the bid revision framework is restrictive, since it does not allow for trading of items between tentatively allocated agents.

Gul and Stacchetti [8] 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 [13]. Various generalizations of unit-demand have been considered in the literature, including recent work on dynamic auctions for homogeneous [2, 4] and heterogeneous [3, 5, 10] commodities.

In the special case where there is only one item in the auction, our top-level auction is effectively a dynamic Vickrey auction. This kind of auction, referred to as the “California auction” by Steiglitz [15], closely approximates the auction format of eBay, and hence is of significant practical importance. It is well known, and straightforward to prove, that the California auction is not truthful. Since our top-level auction generalizes the California auction, it is also not truthful. Nevertheless, the privacy preservation property that we establish for the top-level auction, together with the truthfulness of any single round, suggests that truthful bidding is a reasonable strategy in practice.

3 Preliminaries

As pointed out in the introduction, while the specification of our auction is straightforward, the analysis is a significant technical challenge. In order to work through this challenge, and to facilitate a rigorous and formal analysis of our auction, we introduce below a number of basic types and their auxiliary functions.

We refer to the bidders in our auctions as *agents*. Formally, we treat an agent as a binary string. We define the maximum over an empty set of agents as the empty agent ϵ . An item v in our auction is a pair $(id(v), min(v))$, where $id(v)$ is a binary string identifier and $min(v)$ is an integer lower bound on the price

of v . We allow the price of an item in our auction to be negative in order to support procurement-type auctions. 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.

3.1 Bid-Graphs

A *bid-graph* encodes the input to a unit-demand auction and encapsulates a set of items and a set of agents having unit-demand bids on the items. In general, an agent's unit-demand bid may not include an offer for every item in the bid-graph. In the full version of this paper, we represent the absence of an offer by an offer of $-\infty$. Handling an offer of $-\infty$ requires special treatment in our formalism and lengthens the presentation. Due to space constraints in this paper, we find it convenient to 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.

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 $id(v) \neq id(v')$.

Let $G = (U, V, w)$ be a bid-graph. We introduce the following definitions. We define $bids(G)$ as the set of all possible bids on the set V of items. For any agent u in U , we define $bid(G, u)$ as the bid β 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 β in $bids(G)$, we define $add(G, u, \beta)$ as the bid-graph $G' = (U + u, V, w')$ where $bid(G', u) = \beta$ and $bid(G', u') = bid(G, u')$ 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 $add(G, u, v, z)$ as $add(G, u, \beta)$, where β is the bid in $bids(G)$ such that $\beta(v) = z$ and $\beta(v') = \min(v') - 1$ for any item v' in $V - v$. For any any agent u in U , and any integer z , we define $shift(G, u, z)$ 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 β in $bids(G)$, we define $subst(G, u, \beta)$ as the bid-graph $G' = (U, V, w')$ where $bid(G', u) = \beta$ and $bid(G', u') = bid(G, u')$ for any agent u' in $U - u$.

3.2 Configurations

The outcome of a unit-demand auction consists of a pricing of the items and an allocation of items to agents such that every agent is allocated at most one item. A *configuration* encodes the input to an auction along with an associated outcome.

Formally, a configuration χ is a triple (G, M, Φ) , where $G = (U, V, w)$ is a bid-graph, M is a maximum cardinality matching (MCM) of G , and Φ is a potential function that maps each item v in V to an integer $\Phi(v)$ such that $\Phi(v) \geq \min(v)$. Let $\chi = (G, M, \Phi)$ be a configuration where bid-graph $G = (U, V, w)$. We introduce the following definitions. The function $agents(\chi)$ is the set U and the function $items(\chi)$ is the set V . We say χ is *efficient* if M is a maximum weight MCM (MWMCM) of G . We define $matched(\chi)$ as the subset of agents in U that are matched in M , and we define $unmatched(\chi)$ as the set of agents in $U \setminus matched(\chi)$. For any item v in V , we define $match(\chi, v)$ as the agent u in U such that the edge (u, v) belongs to M . For any item v in V , we define $amount(\chi, v)$ as $w(match(\chi, v), v)$, and we define $amount(\chi)$ as the function that maps each item v in V to $amount(\chi, v)$.

For any bid β in $bids(G)$, we define $pseudo-utility(\chi, \beta)$ as the maximum over all items v in V , of $\beta(v) - \Phi(v)$. We define $pseudo-demand(\chi, \beta)$ as the set of all items v in V such that $\beta(v) - \Phi(v)$ is equal to $pseudo-utility(\chi, \beta)$. For any agent u in U , we define $utility(\chi, u)$ as $w(u, v) - \Phi(v)$ if edge (u, v) belongs to M , and as 0 otherwise. We define $positive(\chi)$ as the set of agents u in U such that $utility(\chi, u) > 0$. We define the set $nonpositive(\chi)$ similarly. For any bid β in $bids(G)$, we define

$demand(\chi, \beta)$ as $pseudo-demand(\chi, \beta)$ if $pseudo-utility(\chi, \beta) \geq 0$, and as the empty set \emptyset otherwise. For any item v in V , we define $bids(\chi, v)$ as the set of all bids β in $bids(G)$ such that v belongs to $demand(\chi, \beta)$.

For any agent u in U and any bid β in $bids(G)$, we denote the configuration $(subst(G, u, \beta), M, \Phi)$ by $subst(\chi, u, \beta)$. For any nonempty agent u not in U , and any bid β in $bids(G)$, we define $add(\chi, u, \beta)$ as the configuration $(add(G, u, \beta), M, \Phi)$. 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 $(add(G, u, v, z), M, \Phi)$. 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 χ by replacing all occurrences of agent u with agent u' .

3.3 Agent Colors

Every agent in a configuration is colored white, gray, or black according to certain rules. We classify configurations based on the colors of their agents.

We define the following for any configuration $\chi = (G, M, \Phi)$, where $G = (U, V, w)$. We define $digraph(\chi)$ as the directed graph $(U \cup V, A)$, where A is the set of arcs that includes an arc (v, u) for every edge (u, v) in M such that v belongs to $demand(\chi, bid(G, u))$, and an arc (u, v) for every edge (u, v) in G such that v belongs to $demand(\chi, bid(G, u))$ and (u, v) does not belong to M . For any agent u in χ , 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 $digraph(\chi)$, and we define $agents(\chi, u)$ as the set of agents matched to the items in $items(\chi, u)$. For any item v in V , we define $agents(\chi, v)$ as the set of all agents u in $unmatched(\chi)$ such that v belongs to $items(\chi, u)$.

The color of any agent u in U is determined as follows, where β denotes $bid(G, u)$. 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 $pseudo-demand(\chi, \beta)$, then agent u is black. If v belongs to $demand(\chi, \beta)$, 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 $pseudo-utility(\chi, \beta) > 0$, then agent u is black. If $pseudo-utility(\chi, \beta) = 0$, 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 $utility(\chi, match(\chi, v)) = 0$, then agent u is gray. Otherwise, agent u is white.

We define $white(\chi)$ as the set of white agents in χ . The sets $gray(\chi)$, $black(\chi)$, $nonblack(\chi)$, and $nonwhite(\chi)$ are defined similarly. 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 $\beta = bid(G, u)$.

3.4 Walrasian configurations

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, Φ) . The following is a list of definitions and lemmas related to Walrasian configurations. The proofs of these lemmas are straightforward and follow from standard results on Walrasian equilibria [7].

Lemma 3.1 *For any bid-graph G' of the form $add(G, u, \beta)$, if bid-graph G is Walrasian then the bid-graph G' is Walrasian.*

Lemma 3.2 *If $\chi = (G, M, \Phi)$ is a Walrasian configuration, then M is an MWMCM of G .*

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

Lemma 3.3 *For any Walrasian bid-graph G , any MWMCM M of G , and any potential function Φ in $potentials(G)$, the configuration (G, M, Φ) is Walrasian.*

Lemma 3.4 For any Walrasian bid-graph G , the functions in $\text{potentials}(G)$ form a lattice with meet and join operations given by pointwise minimum and maximum, respectively.

For any Walrasian bid-graph G , we define $\text{max-potential}(G)$ and $\text{min-potential}(G)$ as the maximum and minimum functions in $\text{potentials}(G)$; the existence of these functions is guaranteed by Lemma 3.4.

Lemma 3.5 For any bid-graph G' of the form $\text{add}(G, u, v, z)$ 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 $\text{matched}(\chi)$.
- If $z < z_0$ and configuration $\chi = (G', M, \Phi)$ is Walrasian, then agent u belongs to $\text{unmatched}(\chi)$.
- If $z = z_0$, then there exist Walrasian configurations $\chi = (G', M, \Phi)$ and $\chi' = (G', M', \Phi)$ such that agent u belongs to $\text{matched}(\chi) \cap \text{unmatched}(\chi')$.

For any Walrasian bid-graph $G = (U, V, w)$ and any item v in V , we define $\text{threshold}(G, v)$ as the unique integer z_0 of Lemma 3.5, and we define $\text{threshold}(G)$ as the function that maps each item v in V to $\text{threshold}(G, v)$.

Lemma 3.6 For any Walrasian bid-graph G , we have $\text{threshold}(G) = \text{max-potential}(G)$.

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

Lemma 3.7 For any bid-graph G' of the form $\text{add}(G, u, \beta)$ where bid-graph $G = (U, V, w)$ is Walrasian, if $\beta(v) \leq \text{price}(G, v)$ for every item v in V , then $\text{price}(G') = \text{price}(G)$.

Lemma 3.8 For any Walrasian configuration $\chi = (G, M, \Phi)$, we have $\text{price}(G) \leq \text{threshold}(G) \leq \text{amount}(\chi)$.

Lemma 3.9 Let G' be a bid-graph of the form $\text{add}(G, u, \beta)$ where bid-graph $G = (U, V, w)$ is Walrasian. Let Δ 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 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 $\text{threshold}(G') = \text{threshold}(G)$.

3.5 White configurations

A configuration χ is *white* if $\text{agents}(\chi) = \text{white}(\chi)$. 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 3.4 for Walrasian configurations.

Lemma 3.10 For any Walrasian bid-graph G , there exists a white configuration of the form (G, M, Φ) , and for any white configuration of the form (G, M, Φ) , the bid-graph G is Walrasian.

Lemma 3.11 For any Walrasian bid-graph G and any potential function Φ in $\text{potentials}(G)$, there exists a white configuration of the form (G, M, Φ) .

Lemma 3.12 For any Walrasian bid-graph G and any pair of white configurations $\chi = (G, M, \Phi)$ and $\chi' = (G, M', \Phi')$, we have $\text{matched}(\chi) = \text{matched}(\chi')$.

By Lemmas 3.10 and 3.12, 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, Φ) . We denote this unique set of matched agents by $\text{matched}(G)$.

Lemma 3.13 For any white configuration (G, M, Φ) , and for any potential function Φ' in $\text{potentials}(G)$, the configuration (G, M, Φ') is white.

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

Lemma 3.14 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 $(z, u) > (\text{threshold}(G, v), u_0)$.

For any Walrasian bid-graph $G = (U, V, w)$ and any item v in V , we define $\text{threshold}^*(G, v)$ as the unique pair $(\text{threshold}(G, v), u_0)$ of Lemma 3.14.

Lemma 3.15 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 $\text{agents}(\chi, v) = \text{agents}(\chi', v)$.

For any Walrasian bid-graph $G = (U, V, w)$, any potential function Φ in $\text{potentials}(G)$, and any item v in V , we define $\text{agents}(G, \Phi, v)$ as the unique set $\text{agents}(\chi, v)$ of Lemma 3.15, where $\chi = (G, M, \Phi)$ is a white configuration whose existence is guaranteed by Lemma 3.11. For any Walrasian bid-graph $G = (U, V, w)$, and any item v in V , we define $\text{agents}(G, v)$ as $\text{agents}(G, \text{price}(G), v)$.

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

Lemma 3.16 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 $\text{price}^*(G') = \text{price}^*(G)$.

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

Lemma 3.17 For any white configuration $\chi = (G, M, \Phi)$, we have $\text{price}^*(G) \leq \text{threshold}^*(G) \leq \text{amount}^*(\chi)$.

Lemma 3.18 Let G' be a bid-graph of the form $\text{add}(G, u, \beta)$ where bid-graph $G = (U, V, w)$ is Walrasian. Let Δ 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 $\text{threshold}^*(G') = \text{threshold}^*(G)$.

3.6 Quiescent configurations

A configuration $\chi = (G, M, \Phi)$ is *quiescent* if $\text{unmatched}(\chi) \subseteq \text{white}(\chi)$, and for any agent u in $\text{black}(\chi)$ where $\beta = \text{bid}(G, u)$, we have $\beta(v) < \Phi(v)$ for all items v in $\text{items}(\chi)$. For any configuration $\chi =$

(G, M, Φ) where $G = (U, V, w)$, and any agent u in U , we say χ is u -quiescent if either (1) u belongs to $unmatched(\chi) \cap gray(\chi)$ and (G', M, Φ) is quiescent, where $G' = (U - u, V, w)$, or (2) u belongs to $matched(\chi)$ and $shift(\chi, u, 1)$ is quiescent.

3.7 ECCs

For any pair of configurations $\chi = (G, M, \Phi)$ and $\chi' = (G, M', \Phi)$, we write $\chi \sim \chi'$ if $matched(\chi) = matched(\chi')$, $nonwhite(\chi) = nonwhite(\chi')$, and for any item v in $items(\chi)$ such that $match(\chi, v)$ is non-white, we have $match(\chi, v) = match(\chi', v)$. 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 χ . By definition, for any ECC X , there exists a unique bid-graph G_0 and a unique potential function Φ_0 such that every configuration in X is of the form (G_0, M, Φ_0) . We define $bid-graph(X)$ and $potential(X)$ as G_0 and Φ_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 χ in X is quiescent. We define u -quiescent ECCs similarly.

For any ECC X , we define $agents(X)$ as the unique set of agents whose existence is guaranteed by Lemma 3.20. We define the following similarly: $items(X)$, $matched(X)$, $unmatched(X)$, $gray(X)$, $white(X)$, $black(X)$, $nonwhite(X)$, $nonblack(X)$, $enabled(X)$, $positive(X)$, $nonpositive(X)$, $utility(X, u)$, $pseudo-utility(X, u)$, $demand(X, u)$, $pseudo-demand(X, u)$, $bids(X, v)$, $items(X, u)$, $agents(X, u)$, and $agents(X, v)$.

For any configurations $\chi = (bid-graph(X), M, potential(X))$ and $\chi' = (bid-graph(X), M', potential(X))$ in X , we say χ is lexicographically smaller than χ' if $S < S'$, where S and S' are the lexicographically ordered lists of edges in M and M' respectively. For any ECC X , any agent u in $agents(X)$, and any integer z , we define $shift(X, u, z)$ as $[shift(\chi, z, u)]$ where χ is the configuration in ECC with the lexicographically smallest matching. For any ECC X and any agent u in $agents(X)$, we define $subst(X, u, \beta)$ as the ECC $\cup_{\chi \in X} [subst(\chi, u, \beta)]$ given by Lemma 3.22. We define the following similarly: $subst(X, u, u')$, $add(X, u, \beta)$, and $add(X, u, v, z)$.

Lemma 3.19 *For any quiescent configuration χ , the ECC $[\chi]$ is quiescent.*

Lemma 3.20 *For any ECC X , and any pair of configurations χ and χ' in X , we have $agents(\chi) = agents(\chi')$.*

Lemma 3.21 *For any ECC X , any agent u in $agents(X)$, and any pair of configurations χ and χ' in X , we have $items(\chi, u) = items(\chi', u)$.*

Lemma 3.22 *For any ECC X , any agent u in $agents(X)$, and any bid β in $bids(bid-graph(X))$, the set of configurations given by $\cup_{\chi \in X} [subst(\chi, u, \beta)]$ is an ECC.*

4 Bottom-Level Auction

For the purposes of our analysis, we find it useful to break the presentation of our auction into three layers, progressively building up to our final top-level auction in Section 6. Each layer describes an auction that uses the auction of the previous layer as a building block. In this section, we describe the bottom-level auction which corresponds to the lowest layer of our presentation.

4.1 Description

The bottom-level 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 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.

We now 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 χ' is constructed as follows: Let χ be an arbitrary configuration in X and let P be an arbitrary simple directed path from u to $victim(\chi, u, z)$ 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 4.1, it follows that $augment(X, u, z)$ is well defined.

For any ECC X and any agent u in $enabled(X)$ such that either (1) X is quiescent, or (2) X is u -quiescent and u belongs to $matched(X)$, we define $inc(X, u)$ as the ECC X' where $bid-graph(X') = shift(bid-graph(X), u, 1)$, and $potential(X')$ is defined as follows: if agent u belongs to $matched(X)$, then $potential(X') = potential(X)$; otherwise $potential(X', v) = potential(X, v) + 1$ for any item v in $items(X, u)$ and $potential(X', v) = potential(X, v)$ for any item v in $items(X) \setminus items(X, u)$.

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

For any quiescent ECC X and any agent u in $unmatched(X)$, we define $victim(X, u)$ as follows: if $matched(X) \cap unmatched(raise(X, u)) = \{u'\}$, then $victim(X, u) = u'$; otherwise, $victim(X, u) = \emptyset$. Recall that by Fact 4.4, $matched(X) \cap unmatched(X)$ has a cardinality of at most 1.

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

Fact 4.1 For any quiescent ECC X and any agent u in $enabled(X) \cap matched(X)$, we have $raise(X, u) = shift(X, u, 1)$.

Fact 4.2 For any quiescent ECC X and any agent u in $enabled(X)$, we have $potential(raise(X, u)) \geq potential(X)$.

Fact 4.3 For any quiescent ECC X and any agent u in $enabled(X)$ such that $bid(bid-graph(X), u) < potential(X)$, we have $potential(raise(X, u)) = potential(X)$.

Fact 4.4 For any ECC X' of the form $raise(X, u)$, we have $|S| \leq 1$, where $S = matched(X) \setminus matched(X')$.

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

Lemma 4.1 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, the ECC $augment(X, u, z)$ is independent of the choice of the configuration χ and the path P used in the definition of $augment(X, u, z)$.

Lemma 4.2 Any set of configurations of the form $inc(X, u)$ is an ECC.

Lemma 4.3 For any quiescent ECC X and any agent u in $enabled(X)$, the predicate $P_1(X, u)$ holds.

Lemma 4.4 For any ECC X' of the form $inc(X, u)$, the predicate $P_1(X', u)$ holds.

Lemma 4.5 For any quiescent ECC X and any agent u in $enabled(X)$, either $raise'(X, u)$ is quiescent, or $raise'(X, u)$ is u -quiescent and u belongs to $matched(raise'(X, u))$.

Lemma 4.6 Any ECC of the form $raise(X, u)$ is quiescent.

4.2 Basic Properties

Below we discuss some basic properties of the bottom-level auction that are useful in both proving lemmas of Section 4.5 and establishing properties of the mid-level auction of Section 5.

Lemma 4.7 For any ECC X' of the form $raise(X, u')$ and any agent u in $nonwhite(X)$, either (1) u belongs to $unmatched(X')$, or (2) u belongs to $nonwhite(X')$, and there exists an item v in $items(X)$ such that $potential(X, v) = potential(X', v)$ and $match(\chi, v) = u$ for any configuration χ in $X \cup X'$.

Proof. Since u belongs to $nonwhite(X)$, there exists an item v in $items(X)$ such that for any configuration χ 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 χ 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 $raise''$, $potential(X', v') = potential(X, v') + 1$ 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 $X'' = inc(raise'(X, u'), u')$. 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 χ in $X \cup X'$. \square

Lemma 4.8 For any quiescent ECC X and any agent u in $enabled(X)$, if $X' = raise(X, u)$, then

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

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 $agents(X) - u$. By Lemma 4.6, X' is quiescent, and by the definition of a quiescent ECC, $unmatched(X') \subseteq white(X')$. Thus, if u_0 belongs to $unmatched(X')$, then u_0 belongs to $white(X')$ and hence u_0 belongs to $enabled(X')$. Now suppose that u_0 belongs to $matched(X')$. We consider the following two cases.

First we consider the case where u_0 belongs to $gray(X) \cap matched(X')$. By Fact 4.2, $potential(X') \geq potential(X)$ and by Lemma 4.7, there exists an item v_0 in $items(X)$ such that $potential(X, v_0) = potential(X', v_0)$ and for any configuration χ in $X \cup X'$, we have $match(\chi, v_0) = u_0$. It follows that u_0 belongs to $gray(X')$.

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

Lemma 4.9 For any quiescent ECC X and any agent u in $enabled(X)$, we have $enabled(X) - u \subseteq enabled(raise(X, u))$.

Proof. Let $X' = raise(X, u)$. By Lemma 4.6, X' is quiescent. Consider any agent u_0 in $enabled(X) - u$. Suppose u_0 belongs to $white(X)$; then by Lemma 4.8, u_0 belongs to $white(X')$, and hence u_0 belongs to $enabled(X')$.

Suppose u_0 belongs to $nonwhite(X)$. Since u_0 belongs to $enabled(X)$, we have $\beta(v) < potential(X, v)$ for every item v in $items(X)$, where $\beta = bid(X, u_0)$. By Fact 4.2, $potential(X') \geq potential(X)$ and by Lemma 4.7, either u_0 belongs to $unmatched(X')$ or there exists an item v_0 in $items(X)$ such that for any configuration χ in $X \cup X'$, we have $match(\chi, v_0) = u_0$. Thus, u_0 belongs to $enabled(X')$. \square

Lemma 4.10 *For any quiescent ECC X and any agent u in $enabled(X)$, if there exists an item v in $items(X)$ such that $potential(X, v) = potential(raise(X, u), v)$, then $bids(X, v) \subseteq bids(raise(X, u), v)$.*

Proof. Let β be any bid 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 4.2, 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, β belongs to $bids(raise(X, u), v)$. \square

Lemma 4.11 *For any quiescent ECC X_0 and any quiescent ECC X_1 of the form $subst(X_0, u_0, u_1)$ where u_0 belongs to $unmatched(X_0)$ and $u_1 < u_0$, we have $utility(raise(X_0, u_0), u_0) = utility(raise(X_1, u_1), u_1) = 0$. Furthermore, either (1) $raise(X_1, u_1) = subst(raise(X_0, u_0), u_0, u_1)$, or (2) $raise(raise(X_1, u_1), u_1) = subst(raise(raise(X_0, u_0), u_0), u_0, u_1)$.*

Proof. Let $\beta = bid(bid-graph(X_0), u_0)$. 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 $X'_1 = subst(X'_0, u_0, u_1)$. If u_0 belongs to $matched(X'_0)$, then it is easy to see that $X''_1 = subst(X''_0, u_0, u_1)$ 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 $X''_1 = subst(X''_0, u_0, u_1)$, or u_0 belongs to $matched(X''_0)$ and u_1 belongs to $unmatched(X''_1)$. Thus, there exists an item v in $items(inc(X'_1, u_1), u_1)$ such that $match(inc(X'_1, u_1), v)$ belongs to $zero(inc(X'_1, u_1))$ 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 $raise'(X''_1, u_1) = subst(X''_0, u_0, u_1)$. Thus, $raise(X''_1, u_0) = subst(raise(X''_0, u_0), u_0, u_1)$. \square

4.3 Commutativity of *raise* invocations

A key property of the bottom-level auction is the commutativity of *raise* invocations. This property is formalized in Lemma 4.20 and is used extensively in the following sections of the paper.

Lemma 4.12 *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 find that v belongs to $items(raise(X, u_1), u_0)$ if and only if $potential(raise(X, u_1), v) = potential(X, v)$.*

Proof. Since $agents(X, u_0) \cap nonpositive(X) = \emptyset$, it follows that $victim(X, u_1, 1)$ does not belong to $agents(X, u_0)$, thus $agents(X, u_0) = agents(raise'(X, u_1), u_0)$ and $items(X, u_0) = items(raise'(X, u_1), u_0)$. Let $\chi = (G, M, \Phi)$ be any configuration in X and let $\chi' = (G', M', \Phi')$ be any configuration in $raise(X, u_1)$. By Lemma 3.21, $items(\chi, u_0) = items(X, u_0)$ and $items(\chi', u_0) = items(raise(X, u_1), u_0)$. By definition, v belongs to $items(\chi, u_0)$ if and only if there exists a directed path from u_0 to v in $digraph(\chi)$, where every edge of the form (u', v') in $digraph(\chi)$ is such that v' belongs to $demand(\chi, bid(bid-graph(X), u'))$.

It is easy to see that if $potential(raise(X, u_1), v) > potential(X, v)$, then there is no directed path from u_0 to v in $digraph(\chi)$. We now consider the case where $potential(raise(X, u_1), v) = potential(X, v)$. 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 $potential(raise(X, u_1), v) = potential(X, v)$. 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 $digraph(\chi')$, and thus, v belongs to $items(raise(X, u_1), u_0)$. \square

Lemma 4.13 For any quiescent ECC X and any agents u_0 in $unmatched(X)$ and u_1 in $enabled(X)$, if $agents(X, u_0) \cap nonpositive(X) = \emptyset$, then $agents(X_1, u_0) \subseteq agents(X, u_0)$, and $agents(X_1, u_0) \cap nonpositive(X_1) = \emptyset$, where $X_1 = raise(X, u_1)$.

Proof. If u_1 belongs to $matched(X)$, then by Fact 4.1, 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 4.12, 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 $agents(X_1, u_0) \cap nonpositive(X_1) = \emptyset$. \square

Lemma 4.14 Let X_0 and X_1 be quiescent ECCs such that $bid-graph(X_0) = bid-graph(X_1)$, $potential(X_0) = potential(X_1)$ 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 χ 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 $agents(X_1, u_1)$, 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 χ 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 4.15 For any quiescent ECC X and any agents u_0 and u_1 in $unmatched(X)$, if $victim(X, u_0) = victim(X, u_1, 1)$, then u_0 belongs to $agents(raise(X, u_0), u_1)$.

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 4.12, 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 4.16 For any quiescent ECC X and any agents u_0 and u_1 in $unmatched(X)$ such that $u_1 = victim(X, u_1, 1)$, if $victim(X, u_0) = victim(X'_1, u_1, 0)$ where $X'_1 = inc(raise'(X, u_1), u_1)$, then agent u_0 belongs to $agents(X'_{01}, u_1)$ where $X'_{01} = inc(raise'(raise(X, u_0), u_1), u_1)$.

Proof. Let $X_0 = raise(X, u_0)$ and let $victim(X, u_0) = victim(X'_1, u_1, 0) = u$. Note that the case where $victim(X, u_0, 1) = u$ is symmetric to the case handled by Lemma 4.15; thus the proof of this case follows from Lemma 4.15.

We now focus on the case where $victim(X, u_0, 1) = u_0$. By the definition of the function $raise''$, we have $potential(X_0, v) = potential(X, v) + 1$ for any item v in $items(X, u_0)$ and $potential(X, v) = potential(X, v)$ for any item v in $items(X) \setminus items(X, u_0)$. Since $victim(X, u_0) = u$, we have u_0 belongs to $agents(X_0, u')$ for any agent u' such that $demand(X_0, u) \cap items(X_0, u') \neq \emptyset$. Since $victim(X, u_1, 1) = u_1$, we have $nonpositive(X) \cap agents(X, u_1) = \emptyset$; thus by Lemmas 4.12 and 4.13, we have $potential(X'_{01}, v) = potential(X, v) + 1$ for any item v in $items(X, u_0) \cup items(X, u_1)$. Since $victim(X'_1, u_1, 0) = u$, we have $V \cap items(X'_{01}, u_1) \neq \emptyset$. It follows that u_0 belongs to $agents(X'_{01}, u_1)$. \square

Lemma 4.17 *Let X be a quiescent ECC and let u_0 and u_1 be agents in $\text{unmatched}(X)$. Let $X_0 = \text{raise}(X, u_0)$ and let $X_1 = \text{raise}(X, u_1)$. If $\text{victim}(X, u_0) \neq \text{victim}(X, u_1)$, then $\text{victim}(X_0, u_1) = \text{victim}(X, u_1)$ and $\text{potential}(\text{raise}(X_0, u_1), v) = \text{potential}(X_1, v)$ for any item v in $\text{items}(X)$ such that $\text{potential}(X_1, v) = \text{potential}(X, v) + 1$.*

Proof. First we consider the case where $\text{victim}(X, u_1, 1) \neq u_1$. In this case, we have $\text{victim}(X, u_1) = \text{victim}(X, u_1, 1)$ and $\text{potential}(X_1) = \text{potential}(X)$; thus $\text{victim}(X, u_1)$ belongs to $\text{nonpositive}(X)$. The statement of the lemma assumes that $\text{victim}(X, u_0) \neq \text{victim}(X, u_1)$; thus, by the definition of the function raise , we find that $\text{victim}(X, u_1)$ belongs to $\text{agents}(X_0, u_1) \cap \text{nonpositive}(X_0)$. If $\text{victim}(X_0, u_1) = \text{victim}(X, u_1)$, then the proof is complete. Suppose that $\text{victim}(X_0, u_1) \neq \text{victim}(X, u_1)$. Then there is an agent u' in $\text{agents}(X, u_1)$ such that $u' = \text{victim}(X, u_0)$, and hence u_0 belongs to $\text{agents}(X_0, u_1)$. Since $u' = \text{victim}(X_0, u_0)$, u' belongs to $\text{agents}(X, u_1)$, and $u' \neq \text{victim}(X, u_1, 1)$, the definition of the function victim implies that $\text{victim}(X_0, u_1, 1) = \text{victim}(X, u_1, 1)$.

Next we consider the case where $\text{victim}(X, u_1, 1) = u_1$; thus, by the definition of the function raise , we have $\text{nonpositive}(X) \cap \text{agents}(X, u_1) = \emptyset$ and $\text{potential}(X_1, v) = \text{potential}(X, v) + 1$ for any item v in $\text{items}(X, u_1)$. Thus, $\text{victim}(X, u_1) = \text{victim}(X'_1, u_1, 0)$, where $X'_1 = \text{inc}(\text{raise}'(X, u_1), u_1)$. By Lemma 4.12, we have $\text{items}(X_0, u_1) = \text{items}(X, u_1) \setminus \text{items}(X, u_0)$ and $\text{potential}(X_0, v) = \text{potential}(X, v) + 1$ for any item v in $\text{items}(X, u_0)$. By Lemmas 4.12 and 4.13, we have $\text{nonpositive}(X_0) \cap \text{agents}(X_0, u_1) = \emptyset$; thus, we have $\text{potential}(\text{raise}(X_0, u_1), v) = \text{potential}(X, v) + 1$ for any item v in $\text{items}(X_0, u_1)$. Since $\text{items}(X_0, u_1) = \text{items}(X, u_1) \setminus \text{items}(X, u_0)$, we have $\text{potential}(\text{raise}(X_1, u_1), v) = \text{potential}(X, v) + 1$ for any item v in $\text{items}(X, u_0) \cup \text{items}(X, u_1)$. Let $X'_{01} = \text{inc}(\text{raise}'(X_0, u_1), u_1)$. If $\text{victim}(X'_{01}, u_1, 0) = \text{victim}(X'_1, u_1, 0)$, then the proof is complete. Suppose that $\text{victim}(X'_{01}, u_1, 0) \neq \text{victim}(X'_1, u_1, 0)$; then there is an agent u' in $\text{agents}(X'_1, u_1)$ such that $u' = \text{victim}(X, u_0)$, and hence u_0 belongs to $\text{agents}(X'_{01}, u_1)$. Since $u' = \text{victim}(X_0, u_0)$, u' belongs to $\text{agents}(X'_1, u_1)$, and $u' \neq \text{victim}(X'_1, u_1, 0)$. The definition of the function victim implies that $\text{victim}(X'_{01}, u_1, 0) = \text{victim}(X'_1, u_1, 0)$.

Thus, $\text{victim}(X_0, u_1) = \text{victim}(X, u_1)$ and $\text{potential}(\text{raise}(X_0, u_1), v) = \text{potential}(X_1, v)$ for any item v in $\text{items}(X)$ such that $\text{potential}(X_1, v) = \text{potential}(X, v) + 1$. □

Lemma 4.18 *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 $X_0 = \text{raise}(X, u_0)$ and let $X_1 = \text{raise}(X, u_1)$.

We first consider the case where $|\{u_0, u_1\} \cap \text{matched}(X)| = 2$; thus, $\{u_0, u_1\} \subseteq \text{matched}(X)$. By Fact 4.1, we have $X_{01} = \text{shift}(\text{shift}(X, u_0, 1), u_1, 1)$, and $X_{10} = \text{shift}(\text{shift}(X, u_1, 1), u_0, 1)$; thus, $X_{01} = X_{10}$.

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

- Case 1: $\text{victim}(X, u_0) \neq u_1$.
 - Case 1.1: $\text{victim}(X, u_0, 1) \neq u_0$.

We have $victim(X, u_0) = victim(X, u_0, 1)$. In this case, u_0 belongs to $matched(raise'(X, u_0))$; thus, by the definition of the function $raise$, we have $potential(X_0) = potential(X)$ and $matched(X_0) = matched(X) + u_0 - victim(X, u_0)$. By Fact 4.1, we have $X_{01} = shift(X_0, u_1, 1)$; thus $potential(X_{01}) = potential(X)$ and $matched(X_{01}) = matched(X) + u_0 - victim(X, u_0)$. Since $victim(X, u_0) \neq u_1$, there exists an agent u' in $nonwhite(X) \cap agents(X, u_0)$ such that $victim(X, u_1, 1) = u'$. By Fact 4.1, we have $X_1 = shift(X, u_1, 1)$ and thus, $nonwhite(X) \cap agents(X, u_0) - u_1 = nonwhite(X_1) \cap agents(X_1, u_0) - u_1$; it follows that $victim(X_1, u_0) = victim(X, u_0)$. Thus, $matched(X_{10}) = matched(X) + u_0 - victim(X, u_0)$. Since $X_1 = shift(X, u, 1)$ and u_0 belongs to $matched(raise'(X, u_0))$, we have $potential(X_{10}) = potential(X)$. Thus, $matched(X_{01}) = matched(X_{10})$ and $potential(X_{01}) = potential(X_{10})$.

– Case 1.2: $victim(X, u_0, 1) = u_0$.

In this case, $victim(X, u_0) = victim(X'_0, u_0, 0)$ where $X'_0 = inc(raise'(X, u_0), u_0)$. Since $victim(X, u_0, 1) = u_0$, we have $nonwhite(X) \cap agents(X, u_0) = \emptyset$; thus, $potential(X_0, v) = potential(X, v) + 1$ for any item v in $items(X, u_0)$. By Fact 4.1, we have $X_{01} = shift(X_0, u_1, 1)$; thus $potential(X_{01}) = potential(X_0)$ and $matched(X_{01}) = matched(X) + u_0 - victim(X, u_0)$. We established above that $X_1 = shift(X, u_1, 1)$; thus $potential(X_1) = potential(X)$ and $matched(X_1) = matched(X)$. Since $agents(X, u_0) \cap nonpositive(X) = \emptyset$ and $potential(X_1) = potential(X_0)$, by Lemmas 4.12 and 4.13 can be used to argue that we have $items(X_1, u_0) = items(X, u_0)$ and $agents(X_1, u_0) \cap nonpositive(X_1) = \emptyset$; thus we have $potential(X_{10}, v) = potential(X, v) + 1$ for any item v in $items(X, u_0)$. Let $X'_1 = inc(raise'(X_1, u_0), u_0)$; it is easy to see that $X'_1 = shift(X'_0, u_1, 1)$. Since $u_1 \neq victim(X'_0, u_0)$ and $X'_1 = shift(X'_0, u_1, 1)$, we have $u_1 \neq victim(X'_1, u_0)$; thus, $victim(X'_1, u_0) = victim(X'_0, u_0) = victim(X, u_0)$, and $matched(X_{10}) = matched(X) + u_0 - victim(X, u_0)$.

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

• Case 2: $victim(X, u_0) = u_1$.

– Case 2.1: $victim(X, u_0, 1) \neq u_0$.

In this case, $victim(X, u_0) = victim(X, u_0, 1) = u_1$; thus, $potential(X_0) = potential(X)$ and $matched(X_0) = matched(X) + u_0 - u_1$. By the definition of the function $raise$, we have $utility(X, u_1) = 0$ and $utility(X_0, u_0) = 1$. We consider two subcases.

First we consider the subcase where $agents(X_0, u_1) \cap nonpositive(X_0) \neq \emptyset$. In this case, we have $potential(X_{01}) = potential(X_0)$ and by Lemma 4.13, we have $agents(X_0, u_1) \cap nonpositive(X_0) = nonpositive(X) \cap (agents(X, u_1) \cup agents(X, u_0))$. Since $utility(X, u_1) = 0$ and $X_1 = shift(X, u_1, 1)$, we have $utility(X_1, u_1) = 1$; by the definition of the function $raise'$, we find that u_1 does not belong to $nonpositive(X_0)$ and $victim(X_0, u_1) \neq u_1$. Thus, $potential(X_{01}) = potential(X)$ and $matched(X_{01}) = matched(X) + u_0 + u_1 - victim(X_0, u_1)$. Since u_1 belongs to $agents(X_1, u_0)$ and $X_1 = shift(X, u_1, 1)$, we have $potential(X_1) = potential(X_0)$ and $nonpositive(X_1) \cap agents(X_1, u_0) = nonpositive(X) \cap (agents(X, u_0) \cup agents(X, u_1)) - u_1$; thus $victim(X_1, u_0) = victim(X_0, u_1)$. Therefore, we have $potential(X_{10}) = potential(X)$ and $matched(X_{10}) = matched(X) + u_0 + u_1 - victim(X_0, u_1)$.

Next we consider the subcase where $agents(X_0, u_1) \cap nonpositive(X_0) = \emptyset$. In this case, we have $potential(X_{01}, v) = potential(X, v) + 1$ for any item v in $items(X_0, u_1)$ and $victim(X_0, u_1) = victim(X'_0, u_1, 0)$ where $X'_0 = inc(raise'(X, u_1), u_1)$. Since u_0 belongs to $agents(X_0, u_1)$ and $utility(X_0, u_0) = 1$, we have u_0 belongs to $nonpositive(X'_0) \cap agents(X'_0, u_1)$; thus u_1 belongs to $matched(X_{01})$. Thus, we have $matched(X_{01}) = matched(X) + u_0 + u_1 -$

$victim(X'_0, u_1, 0)$ and $potential(X_{01}, v) = potential(X, v) + 1$ for any item in $items(X, u_1)$. Since $potential(X_0) = potential(X)$ and $victim(X, u_0, 1) = u_1$, and since $agents(X_0, u_1) \cap nonpositive(X_0) = \emptyset$, we have $agents(X, u_0) \cap nonpositive(X_0) = \{u_1\}$. Since $X_1 = shift(X, u, 1)$, we have $agents(X_1, u_0) \cap nonpositive(X_1) = \emptyset$; thus $potential(X_{10}, v) = potential(X, v) + 1$ for any item v in $items(X_1, u_0)$, where $items(X_1, u_0) = items(X_0, u_1)$. Let $X'_1 = inc(raise'(X_1, u_1), u_1)$; thus we have $X'_1 = shift(X'_0, u_1, 1)$, $agents(X'_1, u_0) \cap nonpositive(X'_1) = agents(X'_0, u_1) \cap nonpositive(X'_0) - u_1$, and we have $victim(X'_1, u_0, 0) = victim(X'_0, u_1, 0)$. Therefore, we have $potential(X_{10}) = potential(X)$ and $matched(X_{10}) = matched(X) + u_0 + u_1 - victim(X'_0, u_1, 0)$.

– Case 2.2: $victim(X, u_0, 1) = u_0$.

In this case, $victim(X, u_0) = victim(X'_0, u_0, 0) = u_1$ where $X'_0 = inc(raise'(X, u_0), u_0)$. Since $victim(X, u_0, 1) = u_0$, we have $nonwhite(X) \cap agents(X, u_0) = \emptyset$; thus, $potential(X_0, v) = potential(X, v) + 1$ for any item v in $items(X, u_0)$, and by the definition of the function $raise''$, we have $utility(X_0, u_0) = 0$. Since u_0 belongs to $agents(X_0, u_1)$ and $utility(X_0, u_0) = 0$, we have u_1 belongs to $matched(raise'(X_0, u_1))$; thus $potential(X_{01}) = potential(X_0)$ and $matched(X_{01}) = matched(X) + u_0 + u_1 - victim(X_0, u_0, 1)$.

We established above that $X_1 = shift(X, u_1, 1)$; thus $potential(X_1) = potential(X)$ and $matched(X_1) = matched(X)$. Since $agents(X, u_0) \cap nonpositive(X) = \emptyset$ and $potential(X_1) = potential(X)$. Lemmas 4.12 and 4.13 imply that $items(X_1, u_0) = items(X, u_0)$ and $agents(X_1, u_0) \cap nonpositive(X_1) = \emptyset$; thus $potential(X'_1, v) = potential(X, v) + 1$ for any item v in $items(X, u_0)$, where $X'_1 = inc(raise'(X_1, u_0), u_0)$. Since $utility(X_0, u_1) = 0$, $potential(X'_1) = potential(X_0)$ and $X_1 = shift(X, u_1, 1)$, we have $utility(X'_1, u_1) = 1$; thus $u_1 \neq victim(X'_1, u_0, 0)$. By the definition of the function $raise$, we have $X'_1 = shift(X_0, u_1, 1)$; thus, $victim(X'_1, u_0) = victim(X_0, u_0, 0)$. It is now easy to see that $matched(X_{10}) = matched(X) + u_0 + u_1 - victim(X, u_0, 0)$ and $potential(X_{10}) = potential(X_0)$.

□

Lemma 4.19 *For any quiescent ECC X and any agents u_0 and u_1 in $unmatched(X)$, if $victim(X, u_0) = victim(X, u_1) = u$, then $potential(X_{01}) = potential(X_{10})$ and $matched(X_{01}) = matched(X_{10})$ where $X_{01} = raise(raise(X, u_0), u_1)$ and $X_{10} = raise(raise(X, u_1), u_0)$.*

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

• Case 1: $victim(X, u_1, 1) \neq u_1$

– Case 1.1 $victim(X, u_0, 1) \neq u_0$. We begin by establishing the following sequence of claims.

1. $potential(X_0) = potential(X)$. Follows from the fact that $victim(X, u_0, 1) \neq u_0$ and the definition of the function $raise''$.
2. u_0 belongs to $agents(X_0, u_1)$. Follows from the fact that $victim(X, u_0) = victim(X, u_1, 1) = u$ and Lemma 4.15.
3. $potential(X_1) = potential(X)$. Follows from the fact that $victim(X, u_1, 1) \neq u_1$ and the definition of the function $raise''$.
4. u_1 belongs to $agents(X_1, u_0)$. Follows from the fact that $victim(X, u_1) = victim(X, u_0, 1) = u$ and Lemma 4.15.

We now consider two subcases.

(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 we have $victim(X_1, u_0, 1) \neq u_0$. Since $victim(X, u_0) = victim(X, u_1) = u$, we have $matched(X_0) \setminus matched(X_1) = \{u_0\}$ 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 χ in $X_0 \cup X_1$. Thus, by Lemma 4.14, 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 we have $agents(X_0, u_1) = agents(X_1, u_0)$ and by the definition of the function *raise''*, we have $potential(X'_{01}) = potential(X'_{10})$. Since $victim(X, u_0) = victim(X, u_1) = u$, we have $matched(X'_{01}) \setminus matched(X'_{10}) = \{u_0\}$ 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 χ in $X'_{01} \cup X'_{10}$. Thus, by Lemma 4.14, 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. $potential(X_0, v) = potential(X, v) + 1$ for any item v in $items(X, u_0)$ and $potential(X_0, v) = potential(X, v)$ for any item v in $items(X) \setminus items(X, u_0)$. Follows from the fact that $victim(X, u_0, 1) = u_0$ and the definition of the function *raise''*.
2. $utility(X_0, u_0) = 0$. We have $victim(X, u_0, 1) = u_0$ and $victim(X, u_0) = u$; thus, u_0 is matched by a *raise''* invocation and $utility(X_0, u_0) = 0$.
3. u_0 belongs to $agents(X_0, u_1)$. Since $victim(X, u_0) = victim(X, u_1, 1)$, by Lemma 4.15, we have u_0 belongs to $agents(X_0, u_1)$.
4. $potential(X'_{01}) = potential(X_0)$. By 2 and 3, we have u_0 belongs to $nonpositive(X_0) \cap agents(X_0, u_1)$; thus by the definition of the function *raise'*, we have $potential(X'_{01}) = potential(X_0)$.
5. $potential(X_1) = potential(X)$. Follows from the fact that $victim(X, u_1, 1) = u$ and the definition of the function *raise'*.
6. $potential(X'_{10}) = potential(X_0)$. Since $victim(X, u_0, 1) = u$, we have $nonpositive(X) \cap agents(X, u_0) = \emptyset$, thus by 5 and Lemma 4.12, we have $items(X_1, u_0) = items(X, u_0)$, and by Lemma 4.13, we have $nonpositive(X_1) \cap agents(X_1, u_0) = \emptyset$; thus by the definition of the function *raise''* and 1, we have $potential(X'_{10}) = potential(X_0)$.
7. u_1 belongs to $agents(X'_{10}, u_0)$. Since $victim(X, u_0, 1) = u_0$, we have $victim(X, u_0) = victim(X'_0, u_0, 0) = u$ where $X'_0 = inc(raise'(X, u_0), u_0)$. We have $victim(X, u_1, 1) = u$. Thus, by Lemma 4.16, 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 have u_0 belongs to $agents(X_0, u_1)$ and u_1 belongs to $agents(X_{10}, u_0)$. It is 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 4.16 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 $items(X) \setminus items(X, u_0)$. Follows from the fact that $victim(X, u_0, 1) = u_0$ and the definition of the function *raise''*.
 2. $potential(X'_{01}, v) = potential(X, v) + 1$ for any item v in $items(X, u_0) \cup items(X, u_1)$ and $potential(X'_{01}, v) = potential(X, v)$ for any item v in $items(X) \setminus items(X, u_0) \cup items(X, u_1)$. Since $victim(X, u_1, 1) = u_1$, we have $nonpositive(X) \cap agents(X, u_1) = \emptyset$; by 1 and Lemma 4.12, we have $items(X_0, u_1) = items(X, u_1) \setminus items(X, u_0)$, and by Lemma 4.13, we have $nonpositive(X_0) \cap agents(X_0, u_1) = \emptyset$; thus, claim 2 follows by the definition of the function *raise''* and claim 1.
 3. u_0 belongs to $agents(X'_{01}, u_1)$. By claim 2, we have $potential(X'_{01}) > potential(X_0)$; thus $victim(X_0, u_1, 1) = u_1$ and $victim(X_0, u_1) = victim(X'_{01}, u_1, 0)$; and by Lemma 4.16, we find that u_0 belongs to $agents(X'_{01}, u_1)$.
 4. $potential(X_1, v) = potential(X, v) + 1$ for any item v in $items(X, u_1)$ and $potential(X_0, v) = potential(X, v)$ for any item v in $items(X) \setminus items(X, u_1)$. Follows from the fact that $victim(X, u_0, 1) = u_0$ and the definition of the function *raise''*.
 5. $potential(X'_{10}, v) = potential(X, v) + 1$ for any item v in $items(X, u_1) \cup items(X, u_0)$ and $potential(X'_{10}, v) = potential(X, v)$ for any item v in $items(X) \setminus items(X, u_1) \cup items(X, u_0)$. The analysis is similar to claim 2.
 6. u_1 belongs to $agents(X'_{10}, u_0)$. By claim 5, we have $potential(X'_{10}) > potential(X_1)$; thus $victim(X_1, u_0, 1) = u_0$; and $victim(X_1, u_0) = victim(X'_{10}, u_0, 0)$; and by Lemma 4.16, we find that u_1 belongs to $agents(X'_{10}, u_0)$.
 7. $victim(X'_{10}, u_0, 0) = victim(X'_{01}, u_1, 0)$. By claims 4 and 5, we have $potential(X'_{01}) = potential(X'_{10})$ and by claims 3 and 6 we have u_0 belongs to $agents(X'_{01}, u_1)$ and u_1 belongs to $agents(X'_{10}, u_0)$. Since $victim(X, u_0) = victim(X, u_1)$, we have $matched(X'_{01}) \setminus matched(X'_{10}) = \{u_1\}$ and $matched(X'_{10}) \setminus matched(X'_{01}) = \{u_0\}$, 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 $X'_{01} \cup X'_{10}$. It follows from Lemma 4.16 that $victim(X'_{10}, u_0, 0) = victim(X'_{01}, u_1, 0)$.

By claims 2 and 5, 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 7, we have $victim(X'_{10}, u_0, 0) = victim(X'_{01}, u_1, 0)$; thus, $matched(X_{01}) = matched(X_{10})$.

□

Lemma 4.20 *For any quiescent ECC X and any agents u_0 and u_1 in $\text{enabled}(X)$, we have*

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

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

We first prove the following claim: $\text{potential}(X_{01}) = \text{potential}(X_{10})$ and $\text{matched}(X_{01}) = \text{matched}(X_{10})$. By Lemma 4.18, the claim holds when $\text{matched}(X) \cap \{u_0, u_1\} \neq \emptyset$. It remains to show that the claim holds when $\{u_0, u_1\} \subseteq \text{unmatched}(X)$. By Lemma 4.19, the claim holds when $\{u_0, u_1\} \subseteq \text{unmatched}(X)$ and $\text{victim}(X, u_0) = \text{victim}(X, u_1)$. By Lemma 4.17, the claim holds when $\{u_0, u_1\} \subseteq \text{unmatched}(X)$ and $\text{victim}(X, u_0) \neq \text{victim}(X, u_1)$.

It now remains to be shown that if $\text{potential}(X_{01}) = \text{potential}(X_{10})$ and $\text{matched}(X_{01}) = \text{matched}(X_{10})$, then $X_{01} = X_{10}$. Consider any agent u in $\text{nonwhite}(X)$; thus, there exists an item v in $\text{items}(X)$ such that $\text{match}(\chi, v) = u$ for every configuration χ in X . Note that if u belongs to $\text{unmatched}(\text{raise}(X, u_0))$, then by the definition of the function raise , it follows that u belongs to $\text{unmatched}(X_{01})$. Using this fact and by repeated application of Lemma 4.7, it follows that either u belongs to $\text{unmatched}(X_{01})$ or $\text{match}(\chi, v) = u$ for every configuration χ in $X \cup X_{01}$. By an identical argument, we find that either u belongs to $\text{unmatched}(X_{10})$ or $\text{match}(\chi, v) = u$ for every configuration in $X \cup X_{10}$. However, since we established above that $\text{matched}(X_{01}) = \text{matched}(X_{10})$, it follows that $\text{match}(\chi, v) = u$ for every configuration χ in $X \cup X_{01} \cup X_{10}$, and hence, $X_{01} = X_{10}$. □

4.4 A restricted class of bidding strategies

We now analyze the bottom-level auction when all agents in the auction bid according to a certain restricted class of strategies.

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

We view the bottom-level auction as taking a pair (X, α) as input, where X is a quiescent ECC and α 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, α_0) , a single agent u in $\text{enabled}(X_0)$ having $\alpha_0(u) > 0$ invokes raise , and the output of the round, denoted by $\text{raise}(X_0, u, \alpha_0)$ is given by $(\text{raise}(X_0, u), \text{shift}(\alpha_0, u, -1))$. The auction terminates when no enabled agent has pending raise invocations.

We define $\text{bottom}(X, \alpha)$ as the output of the bottom-level auction when given the pair (X, α) as input. By Lemma 4.9 and Lemma 4.20, it follows that $\text{bottom}(X, \alpha)$ is uniquely defined. In Section 4.5, we establish various properties of $\text{bottom}(X, \alpha)$. These properties are crucial for describing and analyzing the mid-level auction of Section 5.

The facts below follow from the definition of the function raise and the commutativity of raise invocations established in Lemma 4.20.

Fact 4.5 *For any quiescent ECC X , any target α , and any agent u in $\text{enabled}(X)$, we have*

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

Fact 4.6 *For any quiescent ECC X_0 of the form $\text{add}(X, u, \beta)$ and any target α , we have $\text{bottom}(X_0, \alpha) = \text{bottom}(\text{add}(X', u, \beta), \alpha')$ where $(X', \alpha') = \text{bottom}(X, \alpha)$.*

Fact 4.7 For any quiescent ECC X , any agent u in $\text{white}(X)$, and any target α , if $(X_0, \alpha_0) = \text{bottom}(X, \alpha)$, then

$$\text{bottom}(X, \text{shift}(\alpha, u, 1)) = \text{bottom}(X_0, \text{shift}(\alpha_0, u, 1)).$$

4.5 Truthfulness-related properties

The goal of this section is to establish Lemma 4.30. We use Lemma 4.30 to establish Lemma 5.10 (see Section 5.3.1) on the truthfulness of the slow implementation of the mid-level auction.

For any quiescent ECC X and any target α , we define $\text{matched}(X, \alpha)$ as the set of agents in $\text{matched}(X')$, where $(X', \alpha') = \text{bottom}(X, \alpha)$. For any quiescent ECC X , any target α , and any item v in $\text{items}(X)$, we define $\text{agents}(X, \alpha, v)$ as $\text{agents}(X', v)$, where $(X', \alpha') = \text{bottom}(X, \alpha)$.

Lemma 4.21 For any quiescent ECC X , any quiescent ECC X' of the form $\text{subst}(X, u, u')$, and any target α such that $\alpha(u) = \alpha(u')$, if u belongs to $\text{matched}(X) \cap \text{white}(X)$, then $\text{utility}(X_0, u) = \text{utility}(X_1, u')$, where $(X_0, \alpha_0) = \text{bottom}(X, \alpha)$ and $(X_1, \alpha_1) = \text{bottom}(X', \alpha)$.

Proof. By Lemma 4.20, the *raise* invocations of the bottom-level auction instances with inputs X and $\text{subst}(X, u, u')$ 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 4.8, we know that u and u' remain white in every round of their corresponding executions, and by Fact 4.2, 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 $\text{utility}(X_0, u) = \text{utility}(X_1, u')$. \square

Lemma 4.22 Let X be a quiescent ECC and let X' be a quiescent ECC of the form $\text{subst}(X, u, u')$ such that for any agent u'' in $\text{agents}(X)$, we have $u'' < u$ if and only if $u'' < u'$. Let α and α' be targets such that $\alpha(u) = \alpha'(u')$ and $\alpha(u'') = \alpha'(u'')$ for any agent u'' in $\text{agents}(X) - u$. If $(X_0, \alpha_0) = \text{bottom}(X, \alpha)$ and $(X_1, \alpha_1) = \text{bottom}(X', \alpha')$, then $\text{utility}(X_0, u) + \alpha_0(u) = \text{utility}(X_1, u') + \alpha_1(u')$.

Proof. By Lemma 4.20, the *raise* invocations of the bottom-level auction instances with inputs X and $\text{subst}(X, u, u')$ 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 $\text{agents}(X) - u$, 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 $X'_0 = \text{subst}(X_0, u, u')$. \square

Lemma 4.23 For any quiescent ECC X' of the form $\text{add}(X, u, v, z)$ and any target α , there exists a unique integer z^* and a unique agent u^* in $\text{agents}(X) + \epsilon$ such that u belongs to $\text{matched}(X', \alpha)$ if and only if $(z + \alpha(u), u) > (z^*, u^*)$. Moreover, if u belongs to $\text{matched}(X', \alpha)$, then $\text{potential}(X'', v) = z^*$ where $(X'', \alpha'') = \text{bottom}(X', \alpha)$.

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 $\text{agents}(X) \cup \epsilon$. Consider any pair (z_0, u_0) in S such that $z_0 + \alpha_0(u) < \text{potential}(X_0, v)$, where $\alpha_0 = \text{subst}(\alpha, u_0, \alpha(u))$ and $(X_0, \alpha'_0) = \text{bottom}(\text{add}(X, u_0, v, z_0), \alpha_0)$. By repeated application of Fact 4.2, we know that $\text{potential}(X_0) \geq \text{potential}(X)$ and by repeated application of Lemma 4.8, we have u_0 belongs to $\text{white}(X_0)$. Thus, u_0 does not belong to $\text{matched}(X_0)$. Further, since u_0 belongs to $\text{white}(X_0)$, it follows that $\text{potential}(X_0, v) \geq z_0$. 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 $\text{matched}(X_1, \alpha_1)$ where $X_1 = \text{add}(X, u_1, v, z_1)$ and $\alpha_1 = \text{subst}(\alpha, u_1, \alpha(u))$. Consider the pair (z_1, u_2) where u_2 is the maximum agent such that $u_2 < u_1$. By Lemma 4.22, if u_2 does not belong to $\text{agents}(X) \cup \epsilon$, 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 $\text{matched}(\text{subst}(X_1, u_1, u_2), \text{subst}(\alpha, u_2, \alpha(u)))$. However, we know that (z_1, u_1) is the first pair in S such that u_1 belongs to $\text{matched}(X_1, \alpha_1)$. Thus, it follows that u_2 belongs to $\text{agents}(X) \cup \epsilon$. 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 $\text{matched}(\text{subst}(X_1, u_1, u_3), \text{subst}(\alpha, u_3, \alpha(u)))$. Thus $u^* = u_2$ and $z^* = z_1$.

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

For any quiescent ECC X , any target α , and any item v in $\text{items}(X)$, we define $\text{threshold}^*(X, \alpha, v)$ as the unique pair (z^*, u^*) of Lemma 4.23, and we define $\text{threshold}^*(X, \alpha)$ as the function that maps each item v in $\text{items}(X)$ to $\text{threshold}^*(X, \alpha, v)$. In addition, we define $\text{threshold}(X, \alpha, v)$ as the integer z^* and we define $\text{threshold}(X, \alpha)$ as the function that maps each item v in $\text{items}(X)$ to $\text{threshold}(X, \alpha, v)$.

Lemma 4.24 *For any quiescent ECC X , any target α , and any agent u in $\text{enabled}(X)$, we have*

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

Proof. Assume $\text{threshold}^*(\text{raise}(X, u), \alpha, v_0) < \text{threshold}^*(X, \alpha, v_0)$ for some item v_0 in $\text{items}(X)$. Let X_0 be an ECC of the form $\text{add}(X, u_0, v_0, \min(v_0))$ and let α_0 be a target such that (1) $\alpha_0(u') = \alpha(u')$ for any agent u' in $\text{agents}(X)$, and (2) $\text{threshold}^*(\text{raise}(X, u), \alpha, v_0) < \alpha_0(u_0) + \min(v_0) < \text{threshold}^*(X, \alpha, v_0)$. Note that X_0 is quiescent. Further, we have $\text{threshold}^*(X, \alpha) = \text{threshold}^*(X, \alpha_0)$, and $\text{threshold}^*(\text{raise}(X, u), \alpha) = \text{threshold}^*(\text{raise}(X, u), \alpha_0)$; thus, $\text{threshold}^*(\text{raise}(X, u), \alpha_0, v_0) < \alpha_0(u_0) + \min(v_0) < \text{threshold}^*(X, \alpha_0, v_0)$.

Let $(X_1, \alpha_1) = \text{bottom}(X_0, \alpha_0)$; since $\alpha_0(u_0) + \min(v_0) < \text{threshold}^*(X, \alpha_0, v_0)$, by Lemma 4.23 we find that u_0 belongs to $\text{unmatched}(X_1)$. Since X_0 is quiescent and u belongs to $\text{unmatched}(X_0)$, we have u belongs to $\text{white}(X_0)$, and by repeated application of Lemma 4.8, we find that u_0 belongs to $\text{white}(X_1)$. We conclude that $\alpha_1(u_0) = 0$.

By Fact 4.5, we have $\text{bottom}(\text{raise}(X_0, u), \alpha_0) = \text{bottom}(X_0, \text{shift}(\alpha_0, u, 1))$, and by Fact 4.7, we have $\text{bottom}(X_0, \text{shift}(\alpha_0, u, 1)) = \text{bottom}(X_1, \text{shift}(\alpha_1, u, 1))$.

Since $\text{threshold}^*(\text{raise}(X, u), \alpha_0, v_0) < \alpha_0(u_0) + \min(v_0)$, by Lemma 4.23, we find that u_0 belongs to $\text{matched}(\text{raise}(X_0, u), \alpha_0)$, and since we established above that $\alpha_1(u_0) = 0$, we have u_0 does not belong to $\text{matched}(X_1, \text{shift}(\alpha_1, u, 1))$. Since $\text{bottom}(\text{raise}(X_0, u), \alpha_0) = \text{bottom}(X_1, \text{shift}(\alpha_1, u, 1))$, this yields a contradiction. Thus, we have $\text{threshold}^*(X, \text{shift}(\alpha, u, 1)) \leq \text{threshold}^*(\text{raise}(X, u), \alpha)$. \square

Lemma 4.25 *For any quiescent ECC X_0 of the form $\text{add}(X, u, \beta)$ and any target α , if u does not belong to $\text{matched}(X_0, \alpha)$, then $\text{threshold}^*(X_0, \alpha) = \text{threshold}^*(X, \alpha)$.*

Proof. Let $(X', \alpha') = \text{bottom}(X, \alpha)$ and let $(X'_0, \alpha'_0) = \text{bottom}(X_0, \alpha)$. By Fact 4.6, we find that $\text{bottom}(X_0, \alpha) = \text{bottom}(\text{add}(X', u, \beta), \alpha')$. By repeated application of Lemma 4.24, it follows that $\text{threshold}^*(X'_0, \alpha'_0) \geq \text{threshold}^*(X', \alpha')$. Suppose $\text{threshold}^*(X', \alpha', v_1) < \text{threshold}^*(X'_0, \alpha'_0, v_1)$ for some item v_1 in $\text{items}(X)$. Let $X_1 = \text{add}(X_0, u_1, v_1, \min(v_1))$ for some agent u_1 , and let α_1 be a target such that (1) $\text{threshold}^*(X', \alpha', v_1) < (\alpha_1(u_1) + \min(v_1), u_1) < \text{threshold}^*(X'_0, \alpha'_0, v_1)$, and (2) $\alpha_1(u') = \alpha(u')$ for any agent u' in $\text{agents}(X_0)$. Note that $\text{threshold}^*(X_0, \alpha_0) = \text{threshold}^*(X_0, \alpha_1)$. Similarly, $\text{threshold}^*(X_1, \alpha_0) = \text{threshold}^*(X_1, \alpha_1)$.

Let $X_2 = \text{add}(X, u_1, v_1, \min(v_1))$; then $X_1 = \text{add}(X_2, u, \beta)$. Let $(X'_2, \alpha'_2) = \text{bottom}(X_2, \alpha_1)$; by Fact 4.6, we find that $\text{bottom}(X_1, \alpha_1) = \text{bottom}(\text{add}(X'_2, u, \beta), \alpha'_2)$. By Lemma 4.23, since $\min(v_0) + \alpha_1(u_1) > \text{threshold}^*(X', \alpha', v_1)$ and $\text{threshold}^*(X'_0, \alpha'_0) \geq \text{threshold}^*(X', \alpha')$, agent u_0 belongs to

$matched(X'_2, \alpha'_2)$. By Lemma 4.24, we have $threshold^*(X'_2, \alpha'_2) \geq threshold^*(X, \alpha_1)$, and since u does not belong to $matched(X_0, \alpha_1)$, we have u does not belong to $matched(X_1, \alpha_1)$; thus u_1 belongs to $matched(X_1, \alpha_1)$.

Since u belongs to $enabled(X_1)$, by Fact 4.6, it follows that $bottom(add(X'_0, u_1, v_1, min(v_1)), \alpha'_1) = bottom(X_1, \alpha_1)$. By Lemma 4.23, since $min(v_0) + \alpha_0(u_0) < threshold^*(X'_0, \alpha'_0, v_1)$, agent u_0 does not belong to $matched(X'_0, \alpha'_1)$; thus u and u_1 do not belong to $matched(X_1, \alpha_1)$, a contradiction. \square

For any quiescent ECC X , any target α , and any item v in $items(X)$, we define $price(X, \alpha, v)$ as $potential(X', v)$ where $(X', \alpha') = bottom(X, \alpha)$, and we define $price(X, \alpha)$ as the function that maps every item v in $items(X)$ to $price(X, \alpha, v)$.

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

Lemma 4.26 *For any quiescent ECC X and any target α , we have $price^*(X, \alpha) \leq threshold^*(X, \alpha)$.*

Proof. Assume that there exists an item v in $items(X)$ such that $price^*(X, \alpha, v) > threshold^*(X, \alpha, v)$. Let $(X', \alpha') = bottom(X, \alpha)$. Let $X_0 = add(X', u, v, min(v))$ for some agent u , and let α_0 be a target such that (1) $threshold^*(X, \alpha, v) < \alpha_0(u) + min(v) < price^*(X, \alpha, v)$, and (2) $\alpha_0(u') = \alpha'(u')$ for any agent u' in $agents(X)$. Note that X_0 is quiescent. It is easy to see that $threshold^*(X, \alpha) = threshold^*(X', \alpha') = threshold^*(X', \alpha_0)$; thus $threshold^*(X', \alpha_0, v) < \alpha_0(u) + min(v)$. We have $bottom(X_0, \alpha_0) = bottom(add(X', u, v, min(v)), subst(\alpha', u, \alpha_0(u)))$; thus, by repeated application of Fact 4.2, we have $price^*(X_0, \alpha_0) \geq price^*(X', \alpha') \geq price^*(X, \alpha)$. Thus, we have $threshold^*(X', \alpha_0, v) < \alpha_0(u) + min(v) < price^*(X_0, \alpha_0, v)$.

Let $(X'_0, \alpha'_0) = bottom(X_0, \alpha_0)$. By Lemma 4.23, since $\alpha_0(u) + min(v) > threshold^*(X', \alpha_0, v)$ we find that u belongs to $matched(X'_0)$. Since u belongs to $unmatched(X_0)$, we find that u belongs to $white(X_0)$; thus, by repeated application of Lemma 4.8, we have u belongs to $white(X'_0)$. However, since $\alpha_0(u) + min(v) < price^*(X, \alpha, v)$, it follows that u belongs to $nonwhite(X'_0)$, thus yielding a contradiction. Thus, $price^*(X, \alpha) \leq threshold^*(X, \alpha)$. \square

Lemma 4.27 *Let X_0 be a quiescent ECC and let u_0 be an agent in $unmatched(X_0)$. Let X_1 be a quiescent ECC of the form $subst(X_0, u_0, u_1)$, where $u_1 < u_0$. Then for any target α such that $\alpha(u_0) = \alpha(u_1)$, we have $utility(X'_0, u_0) = utility(X'_1, u_1)$, where $(X'_0, \alpha'_0) = bottom(X_0, \alpha)$ and $(X'_1, \alpha'_1) = bottom(X_1, \alpha)$.*

Proof. Let $\beta = bid(bid-graph(X_0), u)$ and let X be the ECC such that $X_0 = add(X, u, \beta)$. Observe that X and X_1 are quiescent. Let $(X', \alpha') = bottom(X, \alpha)$. By Fact 4.6, we have $bottom(X_0, \alpha) = bottom(add(X', u_0, \beta), \alpha')$ and $bottom(X_1, \alpha) = bottom(subst(X_0, u_0, u_1), \alpha')$. We refer to the instance of the bottom-level auction with inputs X_0 and α as execution A and we refer to the instance of the bottom-level auction with inputs X_1 and α as execution B . By Lemma 4.20, $raise$ 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 $utility(X'_0, u_0) = utility(X'_1, u_1) = 0$.

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 4.11, we have $utility(X_k, u_0) = utility(X'_k, u_1) = 0$, and either $X_k = subst(X'_k, u_1, u_0)$, or $raise(X_k, u_0) = raise(X'_k, u_1)$.

First we consider the case where $X_k = \text{subst}(X'_k, u_1, u_0)$. In this case, u_0 belongs to $\text{matched}(X_k) \cap \text{white}(X_k)$ and u_1 belongs to $\text{matched}(X'_k) \cap \text{white}(X'_k)$; thus, by Lemma 4.21 we have $\text{utility}(X'_0, u_0) = \text{utility}(X'_1, u_1)$.

Next we consider the case where $X_k \neq \text{subst}(X'_k, u_1, u_0)$. 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 4.11, $\text{raise}(X_k, u_0) = \text{subst}(\text{raise}(X'_k, u_0), u_1, u_0)$, and by Lemma 4.21, we have $\text{utility}(X'_0, u_0) = \text{utility}(X'_1, u_1)$. \square

Lemma 4.28 *Let X be a quiescent ECC of the form $\text{add}(X_0, u, \beta)$ and for each item v in $\text{items}(X)$, let $X_v = \text{add}(X_0, u, v, z)$ where $z = \beta(v)$. Then for any target α , agent u belongs to $\text{matched}(X, \alpha)$ if and only if u belongs to $\text{matched}(X_v, \alpha)$ for some item v in $\text{items}(X)$.*

Proof. We refer to the bottom-level auction instance with inputs (X, α) as execution A , and for each item v , we refer to the bottom-level auction instance with input (X_v, α) as execution A_v . We represent the output of round i of execution A by (X_i, α_i) , and for any v in V , we represent the output of round i of execution A_v by $(X_{v,i}, \alpha_{v,i})$. Note that agent u is unmatched and therefore enabled in all rounds of all executions under consideration. By Lemma 4.20, 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) : $(\beta(v) + \alpha(u), u) < \text{threshold}^*(X, \alpha(u), v)$ for every item v in $\text{items}(X)$.

By Lemma 4.23, since $(\beta(v) + \alpha(u), u) < \text{threshold}^*(X, \alpha(u), v)$ for every item v in $\text{items}(X)$, we find that u does not belong to $\text{matched}(X_v, \alpha)$ and thus u belongs to $\text{unmatched}(X_{v,k})$ for every item v in $\text{items}(X)$. Assume that u belongs to $\text{matched}(X, \alpha)$; thus, u belongs to $\text{matched}(X_k)$. Let α' be a target such that $\alpha'(u') = \alpha_k(u')$ for any u' in $\text{agents}(X_k)$ and for any agent of the form u_v where v is an item in $\text{items}(X)$, we have $(\beta(v) + \alpha(u), u) < (\alpha'(u_v) + \min(v), u_v) < \text{threshold}^*(X, \alpha, v)$. By Lemma 4.24, we have $\text{threshold}^*(X, \alpha) \leq \text{threshold}^*(X_k, \alpha')$; thus, we have $(\beta(v) + \alpha(u), u) < (\alpha'(u_v) + \min(v), u_v) < \text{threshold}^*(X_k, \alpha', v)$ for any item v in $\text{items}(X)$. Let X' be an ECC that is constructed from X_k as follows: initialize $X' = X_k$, and for each item v in $\text{items}(X)$, set $X' = \text{add}(X', u_v, v, \min(v))$. Consider the execution A' of the bottom-level auction with input (X', α') , and for any round i of execution A' , let (X'_i, α'_i) represent the output of round i of execution A' . We now use Lemma 4.20, to allow all agents in $\cup_{v \in \text{items}(X)} u_v$ to exhaust their *raise* invocations. If m is the last round of the *raise* invocations by agents in $\cup_{v \in \text{items}(X)} u_v$, then by Lemmas 4.23, since $(\alpha'(u_v) + \min(v), u_v) < \text{threshold}^*(X_k, \alpha', v)$ for every item v , we find that agent u_v belongs to $\text{unmatched}(X'_m)$ for every v in $\text{items}(X)$, and by Lemma 4.25, we have $\text{threshold}^*(X'_m, \alpha'_m) = \text{threshold}^*(X_k, \alpha_k)$. Since every agent u_v belongs to $\text{unmatched}(X'_m)$ and X'_m is quiescent, we have $\text{potential}(X'_m, v) \geq (\alpha'(u_v) + \min(v), u_v)$ for every item v in $\text{items}(X)$; thus by Fact 4.2, we have $\text{price}^*(X', \alpha', v) \geq (\alpha'(u_v) + \min(v), u_v)$ for every item v . Since $(\beta(v) + \alpha(u), u) < (\alpha'(u_v) + \min(v), u_v)$ for every item v , we have $\text{price}^*(X', \alpha', v) \geq (\beta(v) + \alpha(u), u)$ for every item v . However, by repeated use of Lemma 4.8, agent u is white at the end of execution A' , and by our assumption that u belongs to $\text{matched}(X, \alpha)$, we have $\text{price}^*(X', \alpha', v) < (\beta(v) + \alpha(u), u)$ for some item v ; this yields a contradiction. Thus, we have u does not belong to $\text{matched}(X, \alpha)$.

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

By Lemma 4.23, since $(\beta(v) + \alpha(u), u) > \text{threshold}^*(X, \alpha, v)$ for some item v in $\text{items}(X)$, we find that u belongs to $\text{matched}(X_v, \alpha)$ for some item v in $\text{items}(X)$. Assume that u does not be-

long to $matched(X, \alpha)$. Consider the execution A' defined as in Case 1 above. By Lemma 4.25, we have $threshold^*(X_k, \alpha_k) = threshold^*(X'_m, \alpha'_m)$. By Lemma 4.26, we have $price^*(X'_m, \alpha'_m) \leq threshold^*(X_k, \alpha_k)$. Thus, there exists some item v in $items(X)$ such that u belongs to $unmatched(X'_m)$ and $(\beta(v) + \alpha(u), u) > price^*(X'_m, \alpha'_m, v)$; this violates the quiescent property of X'_m . Thus, u belongs to $matched(X, \alpha)$.

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

Lemma 4.29 *Let X be a quiescent ECC of the form $add(X_0, u, \beta)$, let α be a target, and for each item v in $items(X)$, let $X_v = add(X_0, u, v, z)$, where $z = \beta(v)$. Then, we have $utility(X', u) + \alpha'(u) = \max_{v \in items(X)} \{utility(X'_v, u) + \alpha'_{v'}(u)\}$, where $(X', \alpha') = bottom(X, \alpha)$ and $(X'_v, \alpha'_v) = bottom(X_v, \alpha)$ for each item v in $items(X)$.*

Proof. By Lemma 4.27, if $(X^*, \alpha^*) = bottom(add(X_0, u', \beta), \alpha)$ for any agent u' , then $utility(X', u) = utility(X^*, u')$. Thus, without loss of generality, we can assume that $u > u'$ for any agent u' in $agents(X)$. By Lemma 4.28, u belongs to $matched(X')$ if and only if u belongs to $matched(X'_v)$ for some v in $items(X_0)$. Thus, if u belongs to $unmatched(X')$, we have $utility(X', u) = utility(X'_v, u) = 0$ for all v in $items(X)$.

We now focus on the case where u belongs to $matched(X')$. Let z be the largest integer such that u belongs to $matched(shift(X', u, -z))$. By Fact 4.7, we have $(X', \alpha') = (X'', shift(\alpha'', u, z))$ where $(X'', \alpha'') = bottom(shift(X, u, -z), \alpha)$; thus $utility(X', u) + \alpha'(u) = utility(X'', u) + \alpha''(u) + z$. By Lemma 4.28, it follows that u belongs to $matched(X''_v)$ for some item v in $items(X)$, where $(X''_v, \alpha''_v) = bottom(shift(X_v, u, -z), \alpha)$. By Fact 4.7, we have $(X'_v, \alpha'_v) = bottom(X''_v, shift(\alpha''_v, u, z))$; thus we have $utility(X'_v, u) + \alpha'_v = utility(X''_v, u) + \alpha''_v(u) + z$. Since u belongs to $white(X) \cap white(X_v)$, we have $\alpha''(u) = \alpha''_v(u) = 0$. To complete the proof, it remains to be shown that $utility(X''_v, u) = utility(X''_v, u)$.

We refer to the bottom-level auction instance with inputs $(shift(X, u, -z), \alpha)$ as execution A , and for each item v , we refer to the bottom-level auction instance with input $(shift(X_v, u, -z), \alpha)$ as execution A_v . We represent the output of round i of execution A by (X_i, α_i) , and for any v in V , we represent the output of round i of execution A_v by $(X_{v,i}, \alpha_{v,i})$. Since u belongs to $unmatched(X)$, it follows that u belongs to $enabled(X) \cap enabled(X_v)$. By Lemma 4.20, we choose to allow agent u to first exhaust its *raise* invocations in all executions. Since $u > u'$ for any agent u' in $agents(X)$, it follows that for each round j in which u invokes *raise*, either u belongs to $unmatched(X_j) \cap unmatched(X_{v,j})$ or there exists a first round j such that u belongs to $matched(X'_j)$ and u belongs to $matched(X'_{v,j})$ for some item v in $items(X)$. Since z was chosen to be the largest integer such that u belongs to $matched(shift(X', u, -z))$, we have $utility(X_j, u) = utility(X_{v,j}, u) = 0$, and thus $utility(X'', u) = utility(X''_v, u) = 0$. \square

Lemma 4.30 *Let X' be a quiescent ECC of the form $add(X, u, \beta)$ and let $(X'', \alpha'') = bottom(X', \alpha)$ for some target α . Let Δ denote the maximum, over all items v in $items(X)$, of $\beta(v) + \alpha(u) - threshold(X, \alpha, v)$, and let V denote the set of all items v in $items(X)$ such that $\beta(v) + \alpha(u) - threshold(X, \alpha, v) = \Delta$. Let u_0 denote the minimum, over all items v in V of the second component of the pair given by $threshold^*(X, \alpha, v)$. Then the following conditions hold:*

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

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

Let v_0 be any item in $items(X)$; thus, we find that $\beta(v_0) + \alpha(u) - threshold(X, \alpha, v_0) = \Delta$. If $(\Delta, u) < (0, u_0)$, then by adding $threshold(X, \alpha, v_0)$ to the first component of both pairs, we find that $(\beta(v_0) + \alpha(u), u) < (threshold(X, \alpha, v_0), u_0)$. Similarly, if $(\Delta, u) > (0, u_0)$, we find that $(\beta(v_0) + \alpha(u), u) > (threshold(X, \alpha, v_0), u_0)$. By Lemma 4.23, it follows that agent u belongs to $matched(add(X, u, v_0, \beta(v_0)), \alpha)$ if and only if $(\beta(v_0) + \alpha(u), u) > (threshold(X, \alpha, v_0), u_0)$, and by Lemma 4.28, we find that agent u belongs to $matched(X'') = matched(X', \alpha)$ if and only if there exists some item v' in $items(X)$ such that agent u belongs to $matched(add(X, u, v', \beta(v')), \alpha)$; thus, we find that agent u belongs to $matched(X'')$ if and only if $(\Delta, u) > (0, u_0)$.

Next we show that if u does not belong to $matched(X'')$, then $threshold^*(X', \alpha) = threshold^*(X, \alpha)$. The result follows directly from Lemma 4.25.

Finally, we show that if u belongs to $matched(X'')$, then (1) and (2) stated above hold. By Lemma 4.29, if u belongs to $matched(X'')$, then $utility(X'', u) = \max_{v \in items(X)} \{utility(X'_v, u)\}$, where X'_v is equal to $bottom(add(X, u, v, \beta(v)), \alpha)$, and by Lemma 4.23, we have $\max_{v \in items(X)} \{utility(X'_v, u)\} = \Delta$; thus $utility(X'', u) = \Delta$. Let v_0 be any item in V . By Lemma 4.26, we have $potential(X'', v_0) \leq threshold(X, \alpha, v_0)$, and since $utility(X'', u) = \Delta$, we have $potential(X'', v_0) \geq threshold(X, \alpha, v_0)$; thus $potential(X'', v_0) = threshold(X, \alpha, v_0)$ and condition (1) holds. Now consider any item v not in V . By definition, we have

$$\beta(v) + \alpha(u) - threshold^*(X, \alpha, v) < \Delta. \quad (3)$$

By Lemma 4.26, we have

$$potential(X'', v) \leq threshold^*(X, \alpha, v). \quad (4)$$

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. \square

5 Mid-Level Auction

In this section, we describe the mid-level auction, a sealed-bid auction that uses the bottom-level auction of Section 4 as a building block. We show that the mid-level auction satisfies strong properties related to truthfulness, efficiency, and privacy preservation.

5.1 Description

The mid-level auction proceeds in two phases, where the first phase corresponds to running an instance of the bottom-level auction and the second phase corresponds to solving an instance of the house allocation problem [14].

First phase. In this section, we describe the first phase of the mid-level auction. For any ECC X , we define $targets(X)$ as the set of all targets such that for any target α in $targets(X)$, there exists a quiescent ECC X_0 satisfying the following conditions: (1) $shift(X_0, \alpha) = X$, (2) $white(X) \cap matched(X) = white(X_0) \cap matched(X_0)$, and (3) for any agent u in $unmatched(X)$ we have $items(X_0, u) = \emptyset$. For any ECC X , we define $target(X)$ as the unique pointwise minimum target in $targets(X)$. For any instance of the mid-level auction with an ECC X as input where $X = shift(X_0, target(X))$, the output of the first phase of the mid-level auction, denoted by $mid'(X)$, is given by $shift(X_0, \alpha_0)$, where $(X_0, \alpha_0) = bottom(X_0, target(X))$.

Second phase. For any configuration $\chi = (G, M, \Phi)$, we define an instance of the house allocation problem on χ 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 χ , we define $mid''(\chi)$ as the configuration obtained by using the TTC algorithm [14] to solve the house allocation problem defined on χ . For any ECC X , we define $mid''(X)$ as the unique ECC $[mid''(\chi)]$ where χ is any configuration in X . The uniqueness of $mid''(\chi)$ is established by Lemma 5.4. For any instance of the mid-level auction with input X , the second phase of the mid-level auction takes the ECC $X' = mid'(X)$ as input and produces the ECC $mid''(X')$ as output. For any ECC X , we define $mid(X)$ as $mid''(mid'(X))$.

The facts below follow from the description of the mid-level auction.

Fact 5.1 *For any ECC X' of the form $mid''(X)$, there is no nonempty set of agents $U_0 \subseteq nonwhite(X')$ that can trade their allocated items amongst themselves in a way such that every agent in U_0 experiences an increase in utility.*

Fact 5.2 *For any ECC X' of the form $mid''(X)$, we have $potential(X') = potential(X)$, $unmatched(X') = unmatched(X)$, and $white(X) \subseteq white(X')$.*

Fact 5.3 *For any ECC X , we have $potential(mid(X)) \geq potential(X)$.*

5.2 Properties

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

Lemma 5.1 *For any ECC X and any targets α_0 and α_1 in $targets(X)$ such that $X = shift(X_0, \alpha_0) = shift(X_1, \alpha_1)$, we have $bottom(X_0, \alpha_0) = bottom(X_1, \alpha_1)$.*

Proof. Let $X = shift(X^*, \alpha^*)$, where $\alpha^* = target(X)$. Since α^* is the pointwise minimum target in $targets(X)$, we have $\alpha_0(u) \geq \alpha^*(u)$ for any agent u in $agents(X)$. Let S be the set of agents u in $agents(X)$ such that $\alpha_0(u) > \alpha^*(u)$. By the definitions of $targets(X)$ and α^* , for any agent u in S , we find that u belongs to $enabled(X_0)$ and $raise(X_0, u) = shift(X_0, u_0, 1)$; by repeated use of this fact and Lemma 4.20, agents in S can commute their $raise$ invocations forward until each agent u in S has $\alpha^*(u)$ pending $raise$ invocations and the resulting ECC is X^* ; thus, $bottom(X_0, \alpha_0) = bottom(X^*, \alpha^*)$. By a similar argument, we have $bottom(X_1, \alpha_1) = bottom(X^*, \alpha^*)$. Thus, $bottom(X_0, \alpha_0) = bottom(X_1, \alpha_1)$. \square

Lemma 5.2 *Let X be any ECC such that $X = shift(X^*, target(X))$ and let $(X_0, \alpha_0) = bottom(X^*, target(X))$. Then, $unmatched(mid'(X)) \subseteq white(mid'(X))$ and $nonwhite(mid'(X)) \subseteq nonwhite(X_0)$.*

Proof. Let $\alpha^* = target(X)$. By definition, we have $mid'(X) = shift(X_0, \alpha_0)$. Since X_0 is quiescent, we have $unmatched(X_0) \subseteq white(X_0)$ and for any agent u in $unmatched(X_0)$, we find that $agents(X_0, u) \cap nonwhite(X_0) = \emptyset$. Moreover, these facts imply that by definition of the bottom-level auction, we have $\alpha_0(u) = 0$ for any agent u in $white(X_1)$; we conclude that u belongs to $white(shift(X_0, \alpha_0))$, where $mid'(X) = shift(X_0, \alpha_0)$.

It remains to show that $nonwhite(mid'(X)) \subseteq nonwhite(X_0)$. Consider any agent u in $nonwhite(mid'(X))$; since $\alpha_0(u) \geq 0$, we conclude that u belongs to $nonwhite(X_0)$. \square

Lemma 5.3 *For any ECC X and any agent u in $nonwhite(mid'(X))$, the following conditions hold: (1) u belongs to $nonwhite(X)$, (2) there exists an item v in $items(X)$ such that $potential(X, v) = potential(mid'(X), v)$, and (3) $match(\chi, v) = u$ for any configuration χ in $X \cup mid'(X)$.*

Proof. Let $X = shift(X^*, \alpha^*)$, where $\alpha^* = target(Ecc)$, and let $(X_0, \alpha_0) = bottom(X^*, \alpha^*)$. By definition, we have $mid'(X) = shift(X_0, \alpha_0)$. By Lemma 5.2, u belongs to $nonwhite(X_0)$; since X_0 is

quiescent, we find that u belongs to $matched(X_0)$; further, since $mid'(X) = shift(X_0, \alpha_0)$, there exists an item v in $items(X)$ such that $potential(mid'(X), v) = potential(X_0, v)$ and $match(\chi, v) = u$ for any configuration χ in $X_0 \cup mid'(X)$. By repeated application of Lemma 4.7, we find that u belongs to $nonwhite(X^*)$, $potential(X^*, v) = potential(X_0, v)$ and $match(\chi, v) = u$ for any configuration χ in $X^* \cup X_0$. By the description of the mid-level auction it follows that u belongs to $nonwhite(X)$, $potential(X, v) = potential(X^*, v)$, and $match(\chi, v) = u$ for any configuration χ in X . These facts imply that u belongs to $nonwhite(X)$ and there exists an item v in $items(X)$ such that $potential(X, v) = potential(mid'(X), v)$, and $match(\chi, v) = u$ for any configuration χ in $X \cup mid'(X)$. \square

Lemma 5.4 *For any ECC X and any configurations χ and χ' in X , we have $mid''(\chi) = mid''(\chi')$.*

Proof. Let $\chi = (G, M, \Phi)$ where $G = (U, V, w)$, and let $\chi' = (G, M', \Phi)$. By the definition of an ECC, we have $black(\chi) = black(\chi')$, and for any agent u in $black(\chi)$, there exists an item v in V such that $match(\chi, v) = match(\chi', v) = u$. The second phase of the mid-level auction keeps item potentials unchanged and only updates the matching of black agents based on a preference ordering over the items; since black agents have the same matched items in χ and χ' , it is easy to see from the description of the second phase of the bottom-level auction that $mid''(\chi) = mid''(\chi')$. \square

Lemma 5.5 *For any ECC X , we have $unmatched(X) \cup white(X) \subseteq white(mid(X))$ and $nonwhite(mid(X)) \subseteq nonwhite(X)$.*

Proof. Let $X = shift(X^*, \alpha^*)$ where $\alpha^* = target(X)$, and let $(X_0, \alpha_0) = bottom(X^*, \alpha^*)$. We have $unmatched(X) = unmatched(X^*) \cap white(X^*)$ and $white(X) \cap matched(X) = white(X^*) \cap matched(X^*)$; thus $unmatched(X) \cup white(X) \subseteq white(X^*)$. By repeated application of Lemma 4.8, we have $white(X^*) \subseteq white(X_0)$, and since $mid'(X) = shift(X_0, \alpha_0)$, we have $white(X^*) \subseteq white(mid'(X))$. Finally, by Fact 5.2, we have $white(mid'(X)) \subseteq white(mid(X))$. Thus, we have $unmatched(X) \cup white(X) \subseteq white(mid(X))$. By Fact 5.2, we have $white(mid'(X)) \subseteq white(mid(X))$; thus $nonwhite(mid(X)) \subseteq nonwhite(mid'(X))$. \square

Lemma 5.6 *Let X_0 and X_1 be ECCs and let α_0 and α_1 be targets such that (1) ECCs X_0 and X_1 are quiescent, and $X = shift(X_0, \alpha_0) = shift(X_1, \alpha_1)$, (2) $white(X_0) \cap matched(X_0) = white(X_1) \cap matched(X_1) = \emptyset$, and (3) for any agent u in $unmatched(X)$, we have $items(X_0, u) = items(X_1, u) = \emptyset$. Then, we have $bottom(X_0, \alpha_0) = bottom(X_1, \alpha_1)$.*

Proof. Let α^* be the target such that for any agent u in $agents(X)$, we have $\alpha^*(u) = \min(\alpha_0(u), \alpha_1(u))$, and let $X = shift(X^*, \alpha^*)$; thus $\alpha_0(u) \geq \alpha^*(u)$. Let S be the set of agents u in $agents(X)$ such that $\alpha_0(u) > \alpha^*(u)$. By the definitions of α_0 and α^* , for any agent u in S , we find that u belongs to $enabled(X_0)$ and $raise(X_0, u) = shift(X_0, u_0, 1)$; by repeated use of this fact and Lemma 4.20, agents in S can commute their $raise$ invocations forward until each agent u in S has $\alpha^*(u)$ pending $raise$ invocations and the resulting ECC is X^* ; thus, $bottom(X_0, \alpha_0) = bottom(X^*, \alpha^*)$. By a similar argument, we have $bottom(X_1, \alpha_1) = bottom(X^*, \alpha^*)$. Thus, $bottom(X_0, \alpha_0) = bottom(X_1, \alpha_1)$. \square

Lemma 5.7 *The second phase of the mid-level auction is truthful.*

Proof. Consider any ECC X' of the form $mid''(X)$. The second phase of the mid-level auction, which is implemented using an application of the TTC algorithm, updates only the matching of black agents. By known results on the truthfulness of the TTC algorithm, the second phase of the mid-level auction is truthful for agents in $black(X)$. By Fact 5.2, we have $potential(mid''(X)) = potential(X)$; thus no agent u in $nonblack(X)$ can achieve a utility higher than $utility(X, u)$ by submitting a false bid. Thus, the second phase of the mid-level auction is truthful. \square

Lemma 5.8 For any ECC X_0 of the form $\text{subst}(X, u, \beta)$, if $\beta \neq \text{bid}(\text{bid-graph}(X), u)$, then either $\text{mid}'(X_0) = \text{subst}(\text{mid}'(X), u, \beta)$ or u belongs to $\text{white}(\text{mid}'(X)) \cap \text{white}(\text{mid}'(X_0))$.

Proof. Let $X = \text{shift}(X^*, \text{target}(X))$ and let $X_0 = \text{shift}(X_0^*, \text{target}(X_0))$. By definition, $\text{mid}'(X) = \text{shift}(X', \alpha')$ where $(X', \alpha') = \text{bottom}(X^*, \text{target}(X))$ and $\text{mid}'(X_0) = \text{shift}(X'_0, \alpha'_0)$ where $(X'_0, \alpha'_0) = \text{bottom}(X_0^*, \text{target}(X_0))$. We consider the following cases.

First we consider the case where u belongs to $\text{white}(X^*) \cap \text{white}(X_0^*)$. By repeated application of Lemma 4.8, we find that u belongs to $\text{white}(X')$, and since $\alpha'(u) \geq 0$, we find that u belongs to $\text{white}(\text{mid}'(X)) \cap \text{white}(\text{mid}'(X_0))$.

Next we consider the case where u belongs to $\text{nonwhite}(X^*) \cap \text{nonwhite}(X_0^*)$. 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 $\text{white}(\text{mid}'(X)) \cap \text{white}(\text{mid}'(X_0))$, or u remains matched throughout to the same item in both auction instances and $\text{mid}'(X_0) = \text{subst}(\text{mid}'(X), u, \beta)$.

Finally we look at the case where u either belongs to $\text{nonwhite}(X^*)$ or belongs to $\text{nonwhite}(X_0^*)$. Without loss of generality, assume that u belongs to $\text{nonwhite}(X^*)$. It follows from the description of the bottom-level auction, that either u is unmatched in some round of the auction instance with input $(X^*, \text{target}(X))$ and hence u belongs to $\text{white}(\text{mid}'(X)) \cap \text{white}(\text{mid}'(X_0))$, or u remains matched throughout in the auction instance with input $(X^*, \text{target}(X))$ and hence $\text{utility}(\text{mid}'(X), u) = \text{utility}(X, u)$. It follows that $\text{mid}'(X_0) = \text{subst}(\text{mid}'(X), u, \beta)$. \square

Lemma 5.9 For any ECC X_0 of the form $\text{subst}(X, u, \beta)$, if $\beta \neq \text{bid}(\text{bid-graph}(X), u)$, then either $\text{mid}'_0(X_0) = \text{subst}(\text{mid}'_0(X), u, \beta)$ or u belongs to $\text{white}(\text{mid}'_0(X)) \cap \text{white}(\text{mid}'_0(X_0))$.

Proof. The proof is identical to the proof of Lemma 5.8 for the case where u belongs to $\text{nonwhite}(X^*) \cap \text{nonwhite}(X_0^*)$. \square

5.3 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 mid-level auction is truthful if it satisfies the following condition: for any ECC X and any agent u in $\text{agents}(X)$, if $X' = \text{subst}(X, u, \beta)$ for some bid β in $\text{bids}(\text{bid-graph}(X))$, then $\text{utility}(\text{mid}(X), u) \geq \text{utility}(X'', u)$, where $X'' = \text{subst}(\text{mid}(X'), u, \text{bid}(X, u))$.

For any ECC X , we define $\text{mid}'_0(X)$ as follows. Let X_0 be a quiescent ECC and let α_0 be a target such that $X = \text{shift}(X_0, \alpha_0)$, $\text{white}(X_0) \cap \text{matched}(X_0) = \emptyset$, and for any agent u in $\text{unmatched}(X)$, we have $\text{items}(X_0, u) = \emptyset$. We define $\text{mid}'_0(X)$ as the ECC $\text{shift}(X', \alpha')$, where $(X', \alpha') = \text{bottom}(X_0, \alpha_0)$. The uniqueness of $\text{mid}'_0(X)$ is established by Lemma 5.6.

The auction that takes an ECC X as input and produces $\text{mid}'_0(X)$ as output does not immediately incorporate bid revision requests of tentatively allocated agents at the beginning of each round, and thus corresponds to a slow implementation of the first phase of the mid-level auction. As discussed in the introduction, proving truthfulness of the mid-level auction (which constitutes each round of the top-level auction) is a significant technical challenge. We find it useful to first establish Lemma 5.12 on the truthfulness of the slow implementation of the mid-level auction (see Section 5.3.1). We then use the claims of Section 5.3.1 to establish Lemma 5.22 below.

5.3.1 Slow implementation

The goal of this section is to establish Lemma 5.12 on the truthfulness of a slow implementation of the mid-level auction. We establish Lemma 5.10 based on the claim of Lemma 4.30 of Section 4.5. Lemma 5.11 follows from Lemma 5.10. The proof of Lemma 5.12 follows from Lemma 5.11 and known results on the truthfulness of the TTC algorithm.

Lemma 5.10 For any ECC X , any agent u in $agents(X)$, and any ECC X' of the form $subst(X, u, \beta)$ where β is a bid in $bids(\text{bid-graph}(X))$, we have

$$utility(\text{mid}'_0(X), u) \geq utility(\text{subst}(\text{mid}'_0(X'), u, \text{bid}(\text{bid-graph}(X), u)), u).$$

Proof. Let $X = \text{shift}(X_0, \alpha_0)$, where X_0 is a quiescent ECC and α_0 is a target such that $\text{white}(X_0) \cap \text{matched}(X_0) = \emptyset$, and for any agent u in $\text{unmatched}(X)$, we have $\text{items}(X_0, u) = \emptyset$. By definition, $\text{mid}'_0(X) = \text{shift}(X'_0, \alpha'_0)$, where $(X'_0, \alpha'_0) = \text{bottom}(X_0, \alpha_0)$. Note that $\text{threshold}^*(X, \alpha_0) = \text{threshold}^*(X_0, \alpha_0)$. Similarly, let $X' = \text{shift}(X_1, \alpha_1)$, where X_1 is a quiescent ECC and α_1 is a target such that $\text{white}(X_1) \cap \text{matched}(X_1) = \emptyset$, and $\text{items}(X_1, u) = \emptyset$ for any agent u in $\text{unmatched}(X)$. By definition, $\text{mid}'_0(X') = \text{shift}(X'_1, \alpha'_1)$, where $(X'_1, \alpha'_1) = \text{bottom}(X_1, \alpha_1)$.

Let $\beta_T = \text{bid}(\text{bid-graph}(X), u)$ and let $X'' = \text{subst}(X'_1, u, \beta_T)$. Assume that $utility(\text{mid}'_0(X), u) < utility(\text{subst}(\text{mid}'_0(X'), u, \beta_T), u)$; thus $utility(X'_0, u) + \alpha'_0(u) < utility(X'', u) + \alpha'_1(u)$.

We first consider the case where u belongs to $\text{unmatched}(X)$. By repeated application of Lemma 4.8, we find that u belongs to $\text{white}(X'_0) \cap \text{white}(X'_1)$ and $\alpha'_0(u) = \alpha'_1(u) = 0$; thus, by our assumption, $utility(X'_0, u) < utility(X'', u)$. Since $utility(X'_0, u) < utility(X'', u)$ and u belongs to $\text{white}(X'_0)$, we have $utility(X'', u) \geq 1$; thus u belongs to $\text{matched}(X'_1)$. By Lemma 4.20, 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 (X_a, α_a) be the input of the first round in which u invokes the function *raise* in the bottom-level auction instance with input (X_0, α_0) , and let X'_a be the ECC such that $X_a = \text{add}(X'_a, u, \beta_T)$. Let (X_b, α_b) be the input of the first round in which u invokes the function *raise* in the bottom-level auction instance with input (X_1, α_1) , and let X'_b be the ECC such that $X_b = \text{add}(X'_b, u, \beta)$. Let S' be the set of items v in $\text{items}(X)$ for which $\beta(v) - \text{threshold}^*(X'_b, \alpha_b, v)$ is maximized. By Lemma 4.30, we have $\text{potential}(X'_1, v) = \text{threshold}^*(X'_b, \alpha_b, v)$; thus, $utility(X'_1, u) = \beta(v) - \text{threshold}^*(X'_b, \alpha_b, v)$ and $utility(X'', u) = \beta_T(v) - \text{threshold}^*(X'_b, \alpha_b, v)$. By Lemma 4.30, we find that $utility(X'_0, u) = \max_{v \in \text{items}(X)} (\beta_T(v) - \text{threshold}^*(X'_a, \alpha_a, v))$; since $\text{threshold}^*(X'_b, \alpha_b) = \text{threshold}^*(X'_a, \alpha_a)$, we have $utility(X'_0, u) \geq utility(X'', u)$; a contradiction. Thus, it follows that $utility(\text{subst}(\text{mid}'_0(X'), u, \beta_T), u) \leq utility(\text{mid}'_0(X), u)$.

Next we consider the case where u belongs to $\text{matched}(X)$. Since $\text{matched}(X_0) \cap \text{white}(X_0) = \text{matched}(X_1) \cap \text{white}(X_1) = \emptyset$, the definition of the function *shift* implies that there exists an item v in $\text{items}(X)$ such that $\text{match}(\chi, v) = u$ for any configuration χ in $X_0 \cup X_1$. Since u belongs to $\text{nonwhite}(X_0) \cap \text{nonwhite}(X_1)$, by the description of the bottom-level auction, either $\text{match}(\chi, v) = u$ for any configuration χ in $X'_0 \cup X'_1$, or u is unmatched in some round of the bottom-level auction instances with inputs (X_0, α_0) and (X_1, α_1) . In the case where $\text{match}(\chi, v) = u$ for any configuration χ in $X'_0 \cup X'_1$, it is not difficult to see that $utility(\text{mid}'_0(X), u) \geq utility(\text{subst}(\text{mid}'_0(X'), u, \beta_T), u)$. In the case where u is unmatched in some round of the bottom-level auction instances with inputs (X_0, α_0) and (X_1, α_1) , the analysis is similar to the previous case in which u belongs to $\text{unmatched}(X)$.

Thus, we have $utility(\text{mid}'_0(X), u) \geq utility(\text{subst}(\text{mid}'_0(X'), u, \text{bid}(\text{bid-graph}(X), u)), u)$. \square

Lemma 5.11 Any auction that takes an ECC X as input and produces the ECC $\text{mid}'_0(X)$ as output is truthful.

Proof. Follows from Lemma 5.10 and the definition of truthfulness. \square

Lemma 5.12 Any auction that takes an ECC X as input and produces the ECC $\text{mid}''(\text{mid}'_0(X))$ as output is truthful.

Proof. For any ECC X_0 , let $f(X_0)$ denote $\text{mid}''(\text{mid}'_0(X_0))$. Consider any instance of the mid-level auction with ECC X as input and let u be an agent in $agents(X)$. Let $\beta = \text{bid}(\text{bid-graph}(X), u)$ and let $\beta_T \neq \beta$ be the truthful bid of u . Let $X_T = \text{subst}(X, u, \beta_T)$. We wish to show that $utility(\text{subst}(f(X), u, \beta_T), u) \leq$

$utility(f(X_T), u)$. By Lemma 5.9, either u belongs to $white(mid'_0(X)) \cap white(mid'_0(X_T))$ or $mid'_0(X_T) = subst(mid'_0(X), u, \beta_T)$.

First, we consider the case where u belongs to $white(mid'_0(X)) \cap white(mid'_0(X_T))$. By Fact 5.2, we have $potential(f(X)) = potential(mid'_0(X))$ and u belongs to $white(f(X))$; thus we have $utility(f(X), u) = utility(mid'_0(X), u)$ and $utility(f(X_T), u) = utility(mid'_0(X_T), u)$. It follows from Lemma 5.10 that $utility(subst(mid'_0(X), u, \beta_T), u) \leq utility(mid'_0(X_T), u)$; thus, we have $utility(subst(f(X), u, \beta_T), u) \leq utility(X_T, u)$.

Next, we consider the case where $mid'_0(X_T) = subst(mid'_0(X), u, \beta_T)$. By Lemma 5.7, the second phase of the mid-level auction is truthful; thus, we have $utility(subst(f(X), u, \beta_T), u) \leq utility(f(X_T), u)$. Thus, any auction that takes an ECC X as input and produces the ECC $f(X) = mid''(mid'_0(X))$ as output is truthful. \square

5.3.2 Fast implementation

The goal of this section is to establish Lemma 5.22 on the truthfulness of the the mid-level auction. We first establish that any white, matched agent achieves the same utility in the mid-level auction and it does in the slow implementation of the mid-level auction (see Lemma 5.19). Lemmas 5.20 and 5.21 follow easily from Lemma 5.19. We use Lemmas 5.21 and known results on the truthfulness of the TTC algorithm to establish Lemma 5.22.

Lemma 5.13 *For any ECC X' of the form $subst(X, u, u')$, if u is an agent in $matched(X) \cap white(X)$, then $utility(mid'(X), u) = utility(mid'(X'), u')$.*

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

Let $mid'(X) = shift(X'_0, \alpha'_0)$ where $(X'_0, \alpha'_0) = bottom(X_0, \alpha)$ and let $mid'(X') = shift(X'_1, \alpha'_1)$ where $(X'_1, \alpha'_1) = bottom(X_1, \alpha)$. By Lemma 4.21, we have $utility(X_0, u) = utility(X_1, u')$. Further, since $\alpha(u) = \alpha(u') = 0$, we have $\alpha'_0(u) = \alpha'_1(u') = 0$. Thus, $utility(mid'(X), u) = utility(mid'(X'), u')$. \square

Lemma 5.14 *Let X be an ECC and let X_A be the quiescent ECC such that $X = shift(X_A, target(X))$. Let u be an agent in $matched(X) \cap white(X)$, and let z be an integer such that u belongs to $gray(shift(X_A, u, -z))$. If u is the minimum agent in $agents(X)$, then $utility(mid'(X), u) = utility(X_B, u) + \alpha_B(u)$, where $(X_B, \alpha_B) = bottom(shift(X'_A, u, -z), shift(target(X), u, z))$.*

Proof. Let $(X'_B, \alpha'_B) = bottom(X_A, target(X))$. We refer to the executions of the bottom-level auction with inputs $(shift(X'_A, u, -z), shift(target(X), u, z))$ and $(X_A, target(X))$ as executions R and R' respectively. Let (X_i, α_i) and (X'_i, α'_i) be the outputs of round i of executions R and R' respectively. By Lemma 4.20, the *raise* invocations of enabled agents commute. Thus, by repeated application of Lemma 4.20, executions R and R' can be reordered such that for any round i , if S is a nonempty set of agents in $enabled(X_i) \cap enabled(X'_i)$ such that $\alpha_i(u') = \alpha'_i(u') > 0$, 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 $(X'_i, \alpha'_i) = (shift(X_i, u, -z), shift(\alpha_i, u, z))$.

First we consider the case where $P(i)$ holds for every round of executions R and R' . In this case, we have $(X'_B, \alpha'_B) = (shift(X_B, u, -z), shift(\alpha_B, u, z))$. Thus, we have $utility(mid'(X), u) = utility(X'_B, u) +$

$\alpha'_B = \text{utility}(X_B, u) - z + \alpha_B(u) + z$. Since u belongs to $\text{white}(\text{mid}(X))$, we have $\alpha'_B(u) = 0$. Thus, $\text{utility}(\text{mid}'(X), u) = \text{utility}(X'_B, u) + \alpha'_B$.

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 $\text{unmatched}(X_k)$ or u belongs to $\text{unmatched}(X'_k)$; further, since u belongs to $\text{nonwhite}(X'_k)$, we find that u belongs to $\text{unmatched}(X_k)$. Let u' be the agent in $\text{matched}(X'_{k-1}) \cap \text{unmatched}(X'_k)$. We now allow u to exhaust all its *raise* invocations in rounds $(k+1) \cdots (k+z)$ of execution B ; thus, we have $\alpha_B(u) = 0$. Since u belongs to $\text{white}(\text{mid}'(X))$, we have $\text{agents}(X'_j, u) \neq \emptyset$ for some round j of execution B where $(k+1) \leq j \leq (k+z)$, and thus, $u' = \text{victim}(X'_j, u)$. It is straightforward to see that for any round $j > k$ of execution R , we have $X_j = X_{j+z}$; thus, $X'_B = X_B$. Since $\alpha(u) = 0$, we have $\alpha'_B(u) = 0$, and we established above that $\alpha_B(u) = 0$. Thus, $\text{utility}(\text{mid}'(X), u) = \text{utility}(X_B, u) = \text{utility}(X_B, u) + \alpha_B(u_0)$. \square

For any ECC X , any agent u in $\text{matched}(X) \cap \text{white}(X)$, and any agent u' such that $u < u'$ and there exists no agent u'' in $\text{agents}(X)$ such that $u' < u'' < u$, we define $\text{split}(X, u, u')$ as the ECC $\text{add}(\text{shift}(X, u, -z), u', \beta)$ where z is the integer such that $\text{pseudo-utility}(\text{shift}(X, u, -z), u) = -1$, and $\beta = \text{bid}(\text{bid-graph}(X), u)$.

Lemma 5.15 *Let X' be an ECC of the form $\text{split}(X, u, u')$ and let $X = \text{shift}(X_A, \text{target}(X))$. For any integer z such that u belongs to $\text{gray}(\text{shift}(X_A, u, -z))$, we have $\text{utility}(\text{mid}'(X'), u') = \text{utility}(X_B, u) + \alpha_B(u)$ where $(X_B, \alpha_B) = \text{bottom}(\text{shift}(X_A, u, -z), \text{shift}(\text{target}(X), u, z))$.*

Proof. Let $X'' = \text{shift}(X_A, u, -z^*)$, where z^* is the integer such that $\text{pseudo-utility}(X'', u) = -1$. Let $\alpha = \text{target}(X)$. By the definition of the function split , we find that u belongs to $\text{matched}(X) \cap \text{white}(X)$; thus, $\alpha(u) = 0$. Note that u belongs to $\text{white}(\text{shift}(X'', u, 1))$; thus, $z^* \leq z$. By repeated application of Fact 4.5, it follows that $(X_B, \alpha_B) = \text{bottom}(X'', \text{shift}(\text{target}(X), u, z^*))$. Let X'_A be the quiescent ECC such that $X' = \text{shift}(X'_A, \text{target}(X'))$ and let $(X'_B, \alpha'_B) = \text{bottom}(X'_A, \text{target}(X'))$.

We refer to the bottom-level auction instance with inputs $(X'', \text{shift}(\text{target}(X), u, z^*))$ as execution A and we refer to the bottom-level auction instance with inputs $(X'_A, \text{target}(X'))$ as execution B . By Lemma 4.20, 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 X_B and X'_B are the resultant ECCs of executions A and B after agents u and u' have exhausted their *raise* invocations, then by Lemma 4.22, we have $\text{utility}(X'_B, u') + \alpha'_B(u') = \text{utility}(X_B, u) + \alpha_B(u)$, 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 $\text{utility}(X_B, u) + \alpha_B(u)$. 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 $\text{utility}(X_B, u) + \alpha_B(u)$. Thus, we have $\text{utility}(\text{mid}'(X'), u') = \text{utility}(X_B, u) + \alpha_B(u)$ \square

Lemma 5.16 *Let X_0 be an ECC of the form $\text{subst}(X, u, u_0)$ where u is an agent in $\text{white}(X) \cap \text{matched}(X)$, and u_0 is an agent such that $u_0 > u$ and there exists exactly one agent u'' in $\text{agents}(X)$ such that $u < u'' < u_0$. Then $\text{utility}(\text{mid}'(X'), u') = \text{utility}(\text{mid}'(X'_0), u'_0)$ where X' is any ECC of the form $\text{split}(X, u, u')$ and X'_0 is any ECC of the form $\text{split}(X_0, u_0, u'_0)$.*

Proof. Let X_A be the quiescent ECC such that $X' = \text{shift}(X_A, \text{target}(X'))$ and let X_B be the quiescent ECC such that $X'_0 = \text{shift}(X_B, \text{target}(X'_0))$. We refer to the executions of the bottom-level auctions with inputs $(X_A, \text{target}(X'))$ and $(X_B, \text{target}(X'_0))$ as executions A and B respectively. By Lemma 4.20, 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 X_i and X'_i be the output ECCs of round i of executions A and B respectively. We define the predicate $\text{sync}(A, B, i)$ to hold if $X'_i = \text{subst}(\text{subst}(X_i, u, u_0), u', u'_0)$. We define the predicate $\text{coupled}(A, B, i)$ to hold if there exists exactly one maximal set of items V_i in $\text{items}(X)$ and agents u_a in $\text{matched}(X_i) \cap \text{unmatched}(X'_i)$ and u_b in $\text{unmatched}(X_i) \cap \text{matched}(X'_i)$ such that for any agent u_c in $\text{unmatched}(X_i) \cap \text{unmatched}(X'_i)$ such that $\text{items}(X_i, u^*) \cap V_i \neq \emptyset$, we have $\text{victim}(X_i, u_c, 1) = u_a$ and $\text{victim}(X'_i, u_c, 1) = u_b$.

Consider the first round j such that $\text{coupled}(A, B, j)$ 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 $\text{sync}(A, B, j-1)$ 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 $\text{agents}(X) - \{u''\}$ invokes the function *raise*. If $\text{items}(X_i, u_1) \cap V_i \neq \emptyset$, then it follows that $\text{coupled}(A, B, i+1)$ holds. Consider the first round $k > j$ in which some agent u_1 invokes the function *raise* and $\text{victim}(X_k, u_1, 1) = u''$, thus we have $\text{victim}(X'_k, u_1, 1) = u_0$. and $\text{sync}(A, B, k+1)$ holds; further, it is easy to see that $\text{sync}(A, B, i)$ holds for every round $i > k$ in which some agent in $\text{agents}(X) \setminus S$ 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 $\text{sync}(A, B, k-1)$ 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 $\text{coupled}(A, B, i)$ 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 $\text{coupled}(A, B, k-1)$ 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. \square

Lemma 5.17 *Let X_0 be an ECC of the form $\text{subst}(X, u, u_0)$ where u is an agent in $\text{white}(X) \cap \text{matched}(X)$. Then $\text{utility}(\text{mid}'(X'), u') = \text{utility}(\text{mid}'(X'_0), u'_0)$ where X' is any ECC of the form $\text{split}(X, u, u')$ and X'_0 is any ECC of the form $\text{split}(X_0, u_0, u'_0)$.*

Proof. Without loss of generality, we can assume that $u < u_0$. If there is no agent u_1 in $\text{agents}(X)$ such that $u < u_1 < u_0$, then the result follows by repeated application of Lemma 4.22. If there is an agent u_1 in $\text{agents}(X)$ such that $u < u_1 < u_0$, then the result follows by induction using Lemma 5.16. \square

Lemma 5.18 *For any ECC X and any agent u in $\text{matched}(X) \cap \text{white}(X)$, if ECC X' is of the form $\text{subst}(X, u, u')$, then $\text{utility}(\text{mid}'(X), u) = \text{utility}(\text{mid}'(X'), u')$.*

Proof. Let X_0 be an ECC of the form $\text{split}(X, u, u_0)$ and let X_1 be an ECC of the form $\text{split}(X', u', u_1)$. By Lemma 5.15, we have $\text{utility}(\text{mid}'(X), u) = \text{utility}(\text{mid}'(X_0), u_0)$ and $\text{utility}(\text{mid}'(X'), u') =$

$utility(mid'(X_1), u_1)$. Since $X' = subst(X, u, u')$, by Lemma 5.17, we have $utility(mid'(X_0), u_0) = utility(mid'(X_1), u_1)$. Thus, we have $utility(mid'(X), u) = utility(mid'(X'), u')$. \square

Lemma 5.19 *For any ECC X such that $X = shift(X_0, target(X))$, any agent u in $matched(X) \cap white(X)$, and any integer z such that u belongs to $gray(shift(X_0, u, -z))$, we have $utility(mid'(X), u) = utility(X_1, u) + \alpha_1(u)$ where $(X_1, \alpha_1) = bottom(shift(X_0, u, -z), shift(target(X), u, z))$.*

Proof. Let u' be any agent such that $u' < u''$ for every agent u'' in $agents(X)$, and let $(X'_1, \alpha'_1) = bottom(shift(subst(X_0, u, u'), u', -z), shift(target(X), u', z))$. By Lemma 5.14, we have $utility(X'_1, u') + \alpha'_1(u') = utility(mid'(subst(X, u, u')), u')$.

By Lemma 5.18, we have $utility(X_1, u) + \alpha_1(u) = utility(X'_1, u') + \alpha'_1(u')$. By Lemma 5.13, we have $utility(mid'(X), u) = utility(mid'(subst(X, u, u')), u')$. Thus, we have $utility(mid'(X), u) = utility(X_1, u) + \alpha_1(u)$. \square

Lemma 5.20 *For any ECC X , any agent u in $agents(X)$, and any ECC X' of the form $subst(X, u, \beta)$ where β is a bid in $bids(bid-graph(X))$, we have*

$$utility(mid'(X), u) \geq utility(subst(mid'(X'), u, bid(bid-graph(X), u)), u).$$

Proof. The analysis for agents in $unmatched(X)$ is identical to the analysis for unmatched agents in the proof of Lemma 5.10, and the analysis for agents in $matched(X) \cap nonwhite(X)$ is identical to the analysis for matched nonwhite agents in the proof of Lemma 5.10.

We now consider any agent u in $matched(X) \cap white(X)$. Let $X = shift(X_0, target(X))$ and let z be any integer such that u belongs to $gray(shift(X_0, u, -z))$. By Lemma 5.19, we have $utility(mid'(X), u) = utility(X_1, u) + \alpha_1(u)$ where $(X_1, \alpha_1) = bottom(shift(X_0, u, -z), shift(target(X), u, z))$. 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 ECCs. Let $X = shift(X_1, \alpha_1)$, where X_1 is the ECC obtained by shifting down the bid of every white agent in X_0 such that $white(X_1) \cap matched(X_1) = \emptyset$. If $(X'_1, \alpha'_1) = bottom(X_1, \alpha_1)$, then by definition, we have $mid'_0(X) = shift(X'_1, \alpha'_1)$. The proof now follows from Lemma 5.10. \square

Lemma 5.21 *The first phase of the mid-level auction is truthful.*

Proof. Follows from Lemma 5.20 and the definition of truthfulness. \square

We use Lemma 5.21 and known results on the truthfulness of the TTC algorithm to establish Lemma 5.22.

Lemma 5.22 *The mid-level auction is truthful.*

Proof. By Lemma 5.21 and Lemma 5.7, the first and second phases of the mid-level auction are individually truthful. We now show that the mid-level auction which combines the two phases is truthful. Consider any instance of the mid-level auction with ECC X as input and let u be an agent in $agents(X)$. Let $\beta = bid(bid-graph(X), u)$ and let $\beta_T \neq \beta$ be the truthful bid of u . Let $X_T = subst(X, u, \beta_T)$. We wish to show that $utility(subst(mid(X), u, \beta_T), u) \leq utility(mid(X_T), u)$. By Lemma 5.8, either u belongs to $white(mid'(X)) \cap white(mid'(X_T))$ or $mid'(X_T) = subst(mid'(X), u, \beta_T)$.

First, we consider the case where u belongs to $white(mid'(X)) \cap white(mid'(X_T))$. By Fact 5.2, $potential(mid(X)) = potential(mid'(X))$ and u belongs to $white(mid(X))$; thus $utility(mid(X), u) = utility(mid'(X), u)$ and $utility(mid(X_T), u) = utility(mid'(X_T), u)$. It follows from Lemma 5.20 that $utility(mid'(X), u) \leq utility(subst(mid'(X_T), u, \beta_T), u)$. Thus, we conclude that $utility(mid(X), u) \leq utility(subst(mid(X_T), u, \beta_T), u)$.

Next, we consider the case where $mid'(X_T) = subst(mid'(X), u, \beta_T)$. By the truthfulness of the second phase of the mid-level auction, we have $utility(mid(X), u) \leq utility(subst(mid(X_T), u, \beta_T), u)$. \square

5.4 Efficiency

We now discuss efficiency-related properties of the mid-level auction. Lemmas 5.26, and 5.27 are the main technical lemmas related to the efficiency of the mid-level auction. We also show that the mid-level auction is weakly Pareto-optimal (see Lemma 5.23). These lemmas are used in Section 6 to establish efficiency-related properties of the top-level auction.

For any ECC X such that $unmatched(X) \subseteq white(X)$ and any agent u in $nonwhite(X)$, we define $admissible(X, u)$ as the set of all bids β in $bids(bid-graph(X))$ such that u belongs to $white(subst(X, u, \beta))$. For any configuration χ such that $unmatched(\chi) \subseteq white(\chi)$, we define $admissible(\chi)$ as the set of all possible configurations that can be obtained from χ by replacing the bid of every agent u in $black(\chi)$ by a bid in $admissible([\chi], u)$. For any ECC X such that $unmatched(X) \subseteq white(X)$, we define $admissible(X)$ as $\cup_{\chi \in X} admissible(\chi)$.

A sealed-bid auction is said to be weakly Pareto-optimal if it produces an output allocation such that no group of agents can trade their allocated items amongst themselves in such a way that all agents in the group experience an increase in utility.

Lemma 5.23 *The mid-level auction is weakly Pareto-optimal.*

Proof. Consider any ECC X and let $X' = mid(X)$. Suppose by way of contradiction that there is a nonempty set of agents $U_0 \subseteq nonwhite(X')$ 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 $white(X')$ and any item v in $items(X)$, we have $utility(X', u) \geq \beta(v) - potential(X', v)$, where $\beta = bid(bid-graph(X), u)$. Thus, $white(X') \cap U_0 = \emptyset$. It follows that $U_0 \subseteq nonwhite(X')$; a contradiction. Thus, $U_0 = \emptyset$ and the mid-level auction is weakly Pareto-optimal. \square

Lemma 5.24 *For any ECC X and any agent u in $matched(mid(X)) \cap nonwhite(mid(X))$, if u belongs to $white(subst(X, u, \beta))$ for some bid β in $bids(bid-graph(X))$, then u belongs to $white(subst(mid(X), u, \beta))$.*

Proof. By Lemma 5.3, there exists an item v in $items(X)$ such that $potential(mid(X), v) = potential(X, v)$ and $match(\chi, v) = u$ for any configuration χ in $X \cup mid(X)$. Let β be any bid in $admissible(X, u)$. By definition, $\beta(v) - potential(X, v) \geq \beta(v') - potential(X, v')$ for any item v' in $items(X)$. By Lemma 5.3, we have $potential(mid(X), v') \geq potential(X, v)$. Thus, we have $\beta(v) - potential(mid(X), v) \geq \beta(v') - potential(mid(X), v')$ for any item v' in $items(X)$. Thus, β belongs to $admissible(mid(X), u)$. \square

Lemma 5.25 *If X is an ECC such that $unmatched(X) \subseteq white(X)$, then any configuration in $admissible(X)$ is white.*

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

Lemma 5.26 *For any ECC X and any agent u in $agents(X)$, if u belongs to $nonwhite(mid(X))$, then u belongs to $nonwhite(X)$ and $admissible(X, u) \subseteq admissible(mid(X), u)$.*

Proof. By Lemma 5.2, we have $nonwhite(mid(X)) \subseteq nonwhite(X)$. Thus, u belongs to $nonwhite(X)$. Let β be any bid in $admissible(X, u)$; then by definition, u belongs to $white(subst(X, u, \beta))$. By Lemma 5.24, u belongs to $white(subst(mid(X), u, \beta))$. Thus, β belongs to $admissible(mid(X), u)$. \square

Lemma 5.27 *For any ECC X , we have $unmatched(mid(X)) \subseteq white(mid(X))$, and every configuration in $admissible(mid(X))$ is efficient.*

Proof. By definition, we have $\text{mid}(X) = \text{mid}''(\text{mid}'(X))$. By Fact 5.2, we have $\text{unmatched}(\text{mid}'(X)) = \text{unmatched}(\text{mid}(X))$ and $\text{potential}(\text{mid}(X)) = \text{potential}(\text{mid}'(X))$; further, by Lemma 5.2, we have $\text{unmatched}(\text{mid}'(X)) \subseteq \text{white}(\text{mid}'(X))$; thus $\text{unmatched}(\text{mid}(X)) \subseteq \text{white}(\text{mid}(X))$.

Let $\chi' = (G, M, \Phi)$ be any configuration in $\text{admissible}(\text{mid}(\chi))$. By Lemma 5.25, χ' is white, and by Lemmas 3.10, χ' is Walrasian; thus, it follows from Lemma 3.2 that M is an MWMCM of G . Thus, χ' is efficient. \square

5.5 Privacy Preservation

The goal of this section is to establish Lemmas 5.28 and 5.29, which we use in Section 6 to establish a certain privacy preservation property of the top-level auction.

For any quiescent configuration χ and any agent u in $\text{enabled}(\chi)$, we would like to define $\text{raise}(\chi, u)$ as a specific configuration in $\text{raise}([\chi], u)$ such that for any agent u_0 in $\text{enabled}(\chi)$, if u does not belong to $\text{matched}(\chi) \cap \text{unmatched}(\text{raise}(\chi, u_0))$, then $\text{raise}(\text{raise}(\chi, u_0), u) = \text{raise}(\text{raise}(\chi, u), u_0)$. In order to do so, we determinize the choice of the augmenting path in function augment defined in Section 4. Specifically, we pick a lexicographically first (with respect to item identifiers) shortest path.

We view the bottom-level auction as taking a pair (χ, α) as input, where χ is a quiescent configuration and α is a target, and updating this pair over a sequence of rounds. A general round of the auction with input (χ_0, α_0) is defined as follows: if $\text{enabled}(\chi_0) = \emptyset$, then the auction terminates; if the minimum agent in $\text{matched}(\chi_0) \cap \text{enabled}(\chi_0) = \epsilon$ then the minimum agent in $\text{enabled}(\chi_0)$ invokes raise ; otherwise, the minimum agent in $\text{matched}(\chi_0) \cap \text{enabled}(\chi_0)$ invokes raise . We define $\text{bottom}(\chi, \alpha)$ as the output of the bottom-level auction when given the pair (χ, α) as input.

We now describe a determinized implementation of the mid-level auction that takes a configuration as input and produce a configuration as output. The determinized mid-level auction described in this section is used as a building block for the top-level auction of Section 6.

For any configuration $\chi = \text{shift}(\chi_0, \text{target}([\chi]))$, we define $\text{mid}'(\chi)$ as $\text{shift}(\chi', \alpha')$ where $(\chi', \alpha') = \text{bottom}(\chi_0, \text{target}([\chi]))$. For any configuration χ , we define $\text{mid}(\chi)$ as $\text{mid}''(\text{mid}'(\chi))$.

Lemma 5.28 *For any configuration χ and any agent u in $\text{matched}(\text{mid}(\chi)) \cap \text{white}(\text{mid}(\chi))$, we have $\text{mid}(\text{shift}(\chi, u, 1)) = \text{shift}(\text{mid}(\chi), u, 1)$.*

Proof. Let $\chi' = \text{shift}(\chi, u, 1)$. Let $\chi = \text{shift}(\chi_A, \text{target}([\chi]))$ and let $\chi' = \text{shift}(\chi'_A, \text{target}([\chi']))$. We refer to the execution of the bottom-level auction with inputs χ_A and $\text{target}([\chi])$ as execution R and we refer to the execution of the bottom-level auction with inputs χ'_A and $\text{target}([\chi'])$ as execution S . Let (χ_i, α_i) and (χ'_i, α'_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 χ and χ' and that u is not gray in either configuration. If u belongs to $\text{gray}(\chi)$, then by the definition of the mid-level auction, u belongs to $\text{gray}(\chi_0)$ and $\alpha(u) = 0$, where $\alpha = \text{target}([\chi])$; thus, u either belongs to $\text{gray}(\text{mid}(\chi))$ or u belongs to $\text{unmatched}(\text{mid}(\chi))$, which is a contradiction. Thus, u does not belong to $\text{gray}(\chi)$. Since $\chi' = \text{shift}(\chi, u, 1)$, we find that u does not belong to $\text{gray}(\chi')$; it is thus straightforward to argue that u has the same color in configurations χ and χ' .

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 the mid-level auction, $\chi'_0 = \text{shift}(\chi_0, u, 1)$. Further, we established above that agent u is nongray has the same color in configurations χ_0 and χ'_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 $\text{matched}(\text{mid}(\chi))$ and u invoked its last raise in round j , we find that u belongs to $\text{matched}(\chi_j) \cap$

$white(\chi_j)$. Further, since $\chi_j = \chi_{j'}$, we find that u belongs to $matched(\chi'_j) \cap white(\chi'_j)$ and $enabled(\chi_j) = enabled(\chi'_j)$. 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 $enabled(\chi_{j+1})$; 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 $matched(\chi'_j)$ and that u makes its last $raise$ invocation of execution S in round $j+1$; thus $\alpha'_{j+1}(u) = 0$. By the definition of the function $raise$, we have $\chi'_{j+1} = shift(\chi'_j, u, 1)$ and $enabled(\chi'_j) = enabled(\chi'_{j+1})$. Since $\chi_j = \chi'_j$, we have $\chi'_{j+1} = shift(\chi_j, u, 1)$ and $enabled(\chi'_{j+1}) = enabled(\chi_j)$; further, we established that $\alpha'_{j+1}(u) = 0$; thus, $S(i) = S'(i+1)$ for $i \geq j$.

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

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 $(\chi'_{k-1}, \alpha'_{k-1}) = (\chi_B, shift(\alpha_B, u, 1))$, where $(\chi_B, \alpha_B) = bottom(\chi_A, target([\chi]))$.

By the definition of the function $raise$, we have $bottom(\chi_B, target([\chi'])) = (shift(\chi_B, u, 1), \alpha_B)$. By the description of the second phase of the mid-level auction, it follows that $mid(\chi') = shift(mid(\chi), u, 1)$. \square

Lemma 5.29 *For any configuration χ and any agent u in $agents(\chi)$, if $utility(mid(\chi), u) > 1$, then $utility(mid(\chi_0), u) \geq 1$, where $\chi_0 = shift(\chi, u, -1)$.*

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

Suppose u belongs to $white(mid(\chi_0)) \cap matched(mid(\chi_0))$. By Lemma 5.28, it follows that $mid(\chi) = shift(mid(\chi_0), u, 1)$. Thus, $utility(mid(\chi_0), u) \geq 1$.

Suppose u belongs to $white(mid(\chi_0)) \cap unmatched(mid(\chi_0))$. It follows that $utility(mid(\chi_0), u) = 0$. Thus, $\beta(v) \leq \Phi'_0(v)$ for every item v in $items(\chi_0)$. However, since $utility(mid(\chi), u) > 1$, there exists an item v in $items(\chi_0)$ such that $match(mid(\chi), v) = u$ and $\beta(v) - \Phi'(v) \geq 1$. Thus, $utility(mid(\chi_0), u) < utility(\chi'', u)$, where $\chi'' = subst(mid(\chi), u, \beta)$. This contradicts Lemma 5.20.

Suppose u belongs to $nonwhite(mid(\chi_0))$. By Lemma 5.5, u belongs to $nonwhite(\chi_0)$, and by Lemma 5.8, we have $mid'(\chi_0) = subst(mid'(\chi), u, \beta)$; thus, it follows from the definition of the second phase of the mid-level auction that $mid(\chi_0) = subst(mid(\chi), u, \beta)$. It is now straightforward to see that since $utility(mid(\chi), u) > 1$, we have $utility(mid(\chi_0), u) \geq 1$. \square

6 Top-Level Auction

In this section, we describe our top-level auction, the dynamic unit-demand auction proposed in this paper. The top-level auction proceeds in rounds. Agents submit bid data at the beginning of each round, where bid data includes bid revision requests from tentatively allocated agents and bids from unallocated agents. If a tentatively allocated agent does not submit a bid revision in a round, then the bid of that agent from the previous round is carried forward to the current round. Each round of the auction is processed by an invocation of the mid-level auction of Section 5.

6.1 Description

A general round of the top-level auction takes a configuration $\chi = (G, M, \Phi)$ as input and produces the configuration $mid(\chi)$ as output. The input to the first round of the auction is a configuration χ that satisfies the following conditions: (1) for any item v in $items(\chi)$, the integer $min(v)$ is equal to the seller-specified starting price of item v ; (2) there exists exactly $|items(\chi)|$ agents in $agents(\chi)$ that are designated as “dummy” agents, and for any dummy agent u and any non-dummy agent u' in $agents(\chi)$, we have $u < u'$; (3) for any item v in $items(\chi)$, 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 $w(u, v') = min(v') - 1$ for any item v' in $items(\chi) - v$.

For any round of the auction that produces a configuration $\chi = (G, M, \Phi)$ as output, the input configuration of the next round is a configuration of the form $\chi' = (G', M, \Phi)$ where $items(\chi) = items(\chi')$.

Below, we discuss some important properties of the top-level auction related to truthfulness, efficiency, privacy preservation, and scalability.

6.2 Truthfulness

Each round of the top-level auction corresponds to a single instance of the mid-level auction. Thus, it follows from Lemma 5.22, that each round of the top-level auction is truthful.

6.3 Efficiency

As mentioned in the introduction, our top-level auction achieves a relaxed form of efficiency. Lemmas 6.1 and 6.2 are the key efficiency-related lemmas of the top-level auction. Furthermore, it follows from Lemma 5.23 that every round of the top-level auction is weakly Pareto-optimal.

We now informally motivate the efficiency-related claims of Lemmas 6.1 and 6.2. 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 be black; informally, this means that the revised bid of u is not fully respected by the auction. By Lemmas 6.1 and 6.2, we establish that in each round subsequent to round i in which u remains allocated and does not submit a bid revision request, the top-level auction makes progress towards respecting the revised bid submitted by u in round i . Lemma 6.1 establishes that in each subsequent round, the top-level auction continues to respect the last admissible bid submitted by u . Lemma 6.2 establishes that in each subsequent round, the set of admissible bids of u can only grow. Thus, with each successive round, the revised bid of u can only find better and better approximations in the growing set of admissible bids.

Lemma 6.1 *Let χ and χ' be the output configurations of successive rounds of the top-level auction where $\chi = (G, M, \Phi)$. and let u be an agent in $matched(\chi)$. If u belongs to $white(\chi) \cap nonwhite(\chi')$, then $bid(G, u)$ belongs to $admissible(\chi', u)$.*

Proof. Let $G = (U, V, w)$. and let χ_0 be the configuration such that $\chi' = mid(\chi_0)$.

First we claim that $bid(G, u)$ belongs to $admissible(\chi_0, u)$. By the definition of the top-level auction, χ_0 is of the form (G', M, Φ) , where $G' = (U, V, w')$. Since χ and χ_0 have the same potential functions and the same matching, and since u belongs to $white(\chi)$, it follows that u belongs to $white(subst(\chi_0, u, bid(G, u)))$ and thus, $bid(G, u)$ belongs to $admissible(\chi_0, u)$.

Next we claim that $admissible(\chi_0, u) \subseteq admissible(\chi', u)$. By Lemma 5.26, since u belongs to $nonwhite(\chi')$, we find that u belongs to $nonwhite(\chi_0)$ and $admissible(\chi_0, u) \subseteq admissible(\chi', u)$.

It follows from the claims established above that $bid(G, u)$ belongs to $admissible(\chi', u)$. \square

Lemma 6.2 *Let χ and χ' be the output configurations of successive rounds of the top-level auction. If agent u belongs to $nonwhite(\chi) \cap nonwhite(\chi')$, then $admissible(\chi, u) \subseteq admissible(\chi', u)$.*

Proof. Let $G = (U, V, w)$. and let χ_0 be the configuration such that $\chi' = mid(\chi_0)$. Let β be any bid in $admissible(\chi, u)$; thus, u belongs to $white(subst(\chi, u, \beta))$.

First we claim that β belongs to $admissible(\chi_0, u)$. By the definition of the top-level auction, χ_0 is of the form (G', M, Φ) , where $G' = (U, V, w')$. Since χ and χ_0 have the same potential functions and the same matching, and since u belongs to $white(subst(\chi, u, \beta))$, it follows that u belongs to $white(subst(\chi_0, u, \beta))$; thus, $bid(G, u)$ belongs to $admissible(\chi_0, u)$.

Next we claim that $admissible(\chi_0, u) \subseteq admissible(\chi', u)$. By Lemma 5.26, since u belongs to $nonwhite(\chi')$, we find that u belongs to $nonwhite(\chi_0)$ and $admissible(\chi_0, u) \subseteq admissible(\chi', u)$.

It follows from the claims established above that $admissible(\chi, u) \subseteq admissible(\chi', u)$. \square

6.4 Privacy Preservation

If the seller of an item in our top-level 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 our top-level auction is to ensure bid privacy for tentatively allocated agents. Below we give a formal definition of a privacy preserving property; one consequence of this property is that no seller can artificially raise the price of an item by more than one unit without risking forfeiture of sale.

For the rest of this section, we discuss the privacy of an arbitrarily designated agent u^* in the top-level auction with respect to an observer of the auction. An observer of the auction is assumed to learn the following in each round of the auction: (1) the matching and allocation published in the round; (2) the bid of every agent in the round except the bid of agent u^* ; (3) whether agent u^* submitted a bid in the round.

Assumption (1) is natural since the top-level auction publishes the tentative allocation and pricing in every round. Assumption (2) ensures that our top-level auction preserves the privacy of agent u^* even when all of the other agents conspire against u^* . Assumption (3) ensures that our privacy preservation property does not merely exploit the fact that agent u^* is allowed to submit a bid revision in every round.

We denote the configuration at the end of round i of the top-level auction by χ_i where $\chi_i = (G_i, M_i, \Phi_i)$. We use Y_i to denote the set of configurations that are consistent with the information learned by an observer up to the end of round i . Note that any pair of configurations in Y_i differ only in the bid of agent u^* . Formally, the top-level auction is said to be *privacy preserving* if the following holds for any designated agent u^* : for any round i of the auction such that u^* belongs to $matched(\chi_i)$, if configuration χ belongs to Y_i , then for any positive integer k , the configuration $shift(\chi, u^*, k - utility(\chi, u^*))$ also belongs to Y_i .

Lemma 6.3 *For any positive integer k and any configuration χ in Y_k such that u^* belongs to $matched(\chi) \cap white(\chi)$, the configuration $shift(\chi, u^*, 1)$ belongs to Y_k . Further, if $utility(\chi, u^*) > 1$, then $shift(\chi, u^*, -1)$ belongs to Y_k .*

Proof. Let χ' be a configuration such that $mid(\chi') = \chi$. Let t be the smallest integer such that for every configuration χ_j where $t < j \leq k$ we have u^* belongs to $matched(\chi_j) \cap white(\chi_j)$ and $bid(G_k, u^*) = bid(G_j, u^*)$. Thus, it follows that either u^* belongs to $nonwhite(\chi_{t-1})$ or $bid(G_{t-1}, u^*) \neq bid(G_t, u^*)$. The proof proceeds by induction on k .

For the base step, we show that the claim holds for $k = t$. First, we show that $shift(\chi, u^*, 1)$ belongs to Y_t . By Lemma 5.28, since u^* belongs to $matched(\chi) \cap white(\chi)$, we have $mid(shift(\chi', u^*, 1)) = shift(\chi, u^*, 1)$. Further, since u^* belongs to $white(shift(\chi, u^*, 1))$, and χ and $shift(\chi, u^*, 1)$ have the same matching and potential function, it follows that $shift(\chi, u^*, 1)$ belongs to Y_t . Next, we show that if $utility(\chi, u^*) > 1$, then $shift(\chi, u^*, -1)$ belongs to Y_t . By Lemma 5.29, $utility(\chi'', u^*) \geq 1$, where

$\chi'' = \text{mid}(\text{shift}(\chi', u^*, -1))$. Thus, u^* belongs to $\text{matched}(\chi'') \cap \text{white}(\chi'')$, and by Lemma 5.28, we have $\chi = \text{shift}(\chi'', u^*, 1)$. It follows that $\chi'' = \text{shift}(\chi, u^*, -1)$. Further, since u^* belongs to $\text{white}(\chi'')$, and χ and χ'' have the same matching and potential function, it follows that χ'' belongs to Y_t .

For the induction step, we assume the claim holds for some round ℓ where $t \leq \ell < k$. We now show that the claim holds when $k = \ell + 1$. By the description of the top-level auction and the fact that $\text{bid}(G_\ell, u^*) = \text{bid}(G_{\ell+1}, u^*)$, there exists a configuration χ'' in Y_k such that $\text{bid}(G', u^*) = \text{bid}(G'', u^*)$. By the induction assumption, it follows that $\text{shift}(\chi'', u^*, 1)$ and $\text{shift}(\chi'', u^*, -1)$ belong to Y_ℓ . First, we show that $\text{shift}(\chi, u^*, 1)$ belongs to $Y_{\ell+1}$. By Lemma 5.28, it follows that $\text{mid}(\text{shift}(\chi', u^*, 1)) = \text{shift}(\chi, u^*, 1)$. Further, since u^* belongs to $\text{white}(\chi'')$, and χ and χ'' have the same matching and potential function, it follows that χ'' belongs to $Y_{\ell+1}$. Thus, $\text{shift}(\chi, u^*, 1)$ belongs to $Y_{\ell+1}$. Next, we show that $\text{shift}(\chi, u^*, -1)$ belongs to $Y_{\ell+1}$. By Lemma 5.29, if $\text{utility}(\chi, u^*) > 1$, then $\text{utility}(\chi'', u^*) \geq 1$, where $\chi'' = \text{mid}(\text{shift}(\chi', u^*, -1))$. Thus, u^* belongs to $\text{matched}(\chi'') \cap \text{white}(\chi'')$, and by Lemma 5.28, we have $\chi = \text{shift}(\chi'', u^*, 1)$ and hence $\chi'' = \text{shift}(\chi, u^*, -1)$. Further, since u^* belongs to $\text{white}(\chi'')$, and χ and χ'' have the same matching and potential function, we find that χ'' belongs to $Y_{\ell+1}$. \square

Lemma 6.4 *For any positive integers k and ℓ and any configuration χ in Y_k such that u^* belongs to $\text{matched}(\chi)$, there exists a configuration χ' in Y_k such that $\text{utility}(\chi', u^*) = \ell$.*

Proof. Let χ' be the configuration such that $\text{mid}(\chi') = \chi$. If u^* belongs to $\text{white}(\chi)$, then the result follows by repeated application of Lemma 6.3. If u^* belongs to $\text{nonwhite}(\chi)$, then by Lemma 5.3, u^* belongs to $\text{nonwhite}(\chi') \cap \text{matched}(\chi')$. It follows that u^* submitted a bid in round k and this bid could be arbitrary. Thus, there exists a configuration χ'' in Y_k of the form $\text{subst}(\chi, u^*, \beta)$ in Y_k such that $\text{utility}(\chi'', u^*) = \ell$. \square

Lemma 6.5 *The top-level auction is privacy preserving.*

Proof. Let k be any positive integer and let u^* be any agent in $\text{matched}(\chi_k)$. The result follows from Lemma 6.4. \square

6.5 Scalability

In this section we briefly sketch a fast implementation of the top-level auction. In each round of the top-level auction, new bid data (i.e., bid revision requests from tentatively allocated agents, and bids from unallocated agents) is received and processed.

We first describe the general state of the auction just before any new bid data is received. In such a state, we know the current pricing and allocation, and the “target” bid of each agent (i.e., the most recent bid submitted by the agent). 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 Section 3.5) of the corresponding item. We only need to maintain information concerning the active bid components. 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.

When new bid data is received at the start of a round, we first incorporate all of the bid revision requests by tentatively allocated agents. This is easy to do since it has no impact on the allocation or pricing; we simply update the relevant target bids. It remains to process the new bids introduced by unallocated agents. We process these bids iteratively. 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 [9], or the closely related

successive shortest paths algorithm [1, 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, 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 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 [16]. 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. Furthermore, we can show that the only way a tentatively allocated agent can become black is by revising its bid. Thus, when we later 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, in any execution of the top-level algorithm, 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.

Recall that each round of our top-level auction consists of two phases. The foregoing discussion has focused on the implementation and analysis 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 5.1, the TTC algorithm is used to update the allocation, and the item prices are left unchanged. It is straightforward to identify all of the black tentatively allocated agents, and set up the input for the TTC algorithm, in linear time in the number of active bid components. It is easy to implement the TTC algorithm so that it runs within the same linear time bound.

In our discussion of the first phase above, we have sketched an argument that the amortized cost of each bidding operation is proportional to the cost of an SSSP computation over the active subgraph of the bid-graph. (The size of the active sub-graph may vary throughout the execution of the top-level auction, but the number of nodes is proportional to the number of items, and the number of edges is at most quadratic in the number of items.) The lone gap in this argument is that it does not address any potential effect of the presence of the second phase. Fortunately, the second phase can only strengthen the amortization argument. The reason is that the second phase has no impact on the color of any non-black tentatively allocated agent, and therefore cannot increase the number of black tentatively allocated agents. (It is possible for the second phase to decrease the number of black tentatively allocated agents, which only helps the amortization argument.)

In summary, it is possible to implement the top-level 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.

7 Concluding Remarks

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 bidders. Our auction maintains strong theoretical 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.

In our current presentation of the 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).

References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, New York, NY, 1993.
- [2] L. Ausubel. An efficient ascending-bid auction for multiple objects. *American Economic Review*, 94(5):1452–1475, 2004.
- [3] L. Ausubel. An efficient dynamic auction for heterogeneous commodities. *American Economic Review*, 96(3):602–629, 2006.
- [4] S. Bickchandani and J. Ostroy. Ascending price Vickrey auctions. *Games and Economic Behavior*, 55(2):215–241, 2006.
- [5] S. de Vries, J. Schummer, and R. Vohra. On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, 132(1):95–118, 2007.
- [6] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.
- [7] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.
- [8] F. Gul and E. Stacchetti. The English auction with differentiated commodities. *Journal of Economic Theory*, 92(1):66–95, 2000.
- [9] H. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 3:253–258, 1955.
- [10] D. Mishra and D. C. Parkes. Ascending price Vickrey auctions for general valuations. *Journal of Economic Theory*, 132(1):335–366, 2007.
- [11] D. Mishra and D. C. Parkes. Multi-item Vickrey-Dutch auctions. *Games and Economic Behavior*, 66(1):326–347, 2009.
- [12] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

- [13] H. Rothkopf, M. A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1999.
- [14] L. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974.
- [15] K. Steiglitz. *Snipers, Shills & Sharks: eBay and Human Behavior*. Princeton University Press, Princeton, New Jersey, 2007.
- [16] 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.
- [17] L. Walras. *Elements of Pure Economics*. Allen and Unwin, 1954.