Modularity and Greed in Double Auctions☆

Paul Dütting[a], Inbal Talgam-Cohen[b], Tim Roughgarden[b]

[a] Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, UK.
[b] Department of Computer Science, Stanford University, 353 Serra Mall, Stanford, CA 94305, USA.

Abstract

Designing double auctions is a complex problem, especially when there are restrictions on the sets of buyers and sellers that may trade with one another. The goal of this paper is to develop a *modular* approach to the design of double auctions, by relating it to the exhaustively-studied problem of designing one-sided mechanisms with a single seller (or, alternatively, a single buyer).

We consider several desirable properties of a double auction: feasibility, dominant-strategy incentive compatibility, the still stronger incentive constraints offered by a deferred-acceptance implementation, exact and approximate welfare maximization, and budget balance. For each of these properties, we identify sufficient conditions on two one-sided algorithms—one for ranking the buyers, one for ranking the sellers—and on a method for their composition into trading pairs, which guarantee the desired property of the double auction.

Our framework also offers new insights into classic double auction designs, such as the VCG and McAfee auctions with unit-demand buyers and unit-supply sellers.

Keywords: Mechanism Design, Double Auctions, Trade Reduction Mechanism, Deferred-Acceptance Auctions

1. Introduction

Double auctions play an important role in mechanism design theory and practice. They are of theoretical importance because they solve the fundamental problem of how to organize trade between a set of buyers and a set of sellers, when both the buyers and the sellers act strategically. Important practical applications include the New York Stock Exchange (NYSE), where buyers and sellers trade shares, and the upcoming spectrum auction conducted by the US Federal Communication Commission (FCC), which aims at reallocating spectrum licenses from TV broadcasters to mobile communication providers [32].

Designing double auctions can be a complex task, with several competing objectives. These include, but are not limited to: feasibility, dominant-strategy incentive compatibility (DSIC), the still stronger incentive constraints offered by a deferred-acceptance implementation such as weak group-strategyproofness (WGSP) or implementability as a clock auction [31], exact or approximate welfare maximization, and budget balance (BB). (See Section 2 for definitions.)

In this paper we utilize the fact that the problem of designing *single*-sided mechanisms is well-studied, and develop the following modular approach to designing *double*-sided auctions for complex settings. We split the design task into three algorithmic modules: one for the buyers, one for the sellers and one for their combination. Designing the algorithm for the buyers or the sellers is based on the well-developed theory of designing single-sided mechanisms, in which problems such as feasibility checking have been exhaustively studied—see examples below. The third module incorporates a composition rule that combines buyers and sellers to determine the final allocation. Payments are

☆A preliminary version of this article appeared in *Proceedings of the 15th ACM Conference on Economics and Computation.*
Email addresses: p.d.duetting@lse.ac.uk (Paul Dütting), italgam@cs.stanford.edu (Inbal Talgam-Cohen), tim@cs.stanford.edu (Tim Roughgarden)

determined by computing *thresholds* from the three algorithms.[1] The goal of this paper is to develop the theory that explains when and how such a modular approach works.[2]

1.1. Motivating Examples

Suppose there are n buyers and m sellers. Each buyer i wants to acquire one unit of an identical good, and has a value v_i for it. Each seller j produces one unit of the good, and producing it incurs a cost of c_j.

Assume first there are no restrictions on which buyers and sellers can trade with one another (we refer to this below as the *unconstrained* problem). Is it possible, by composing two single-sided algorithms, to implement the Vickrey-Clarke-Groves (VCG) mechanism [45, 11, 22] that maximizes welfare and is DSIC? What about McAfee's trade reduction mechanism [30], which accepts all buyer-seller pairs from the welfare-maximizing solution except for the least valuable one, and is DSIC and BB?[3]

The answer to these questions is "yes". We can implement the VCG mechanism using simple *greedy* algorithms that sort the buyers by non-increasing value and the sellers by non-decreasing cost. We iteratively query these algorithms for the next buyer-seller pair and accept it if the buyer has a larger value v_i than the seller's cost c_j, and we apply threshold payments. For McAfee's trade reduction mechanism we use reverse versions of these algorithms, that sort the players by non-decreasing value and non-increasing cost. We iteratively query these algorithms for the next buyer-seller pair and reject it if none of the previously inspected buyer-seller pairs had non-negative gain from trade $v_i - c_j \geq 0$, and we again apply threshold payments.

Now, *what happens if we add feasibility constraints on which buyers and sellers can trade*? Such feasibility constraints are important in practical applications, and their potential richness mirrors the richness of real-life economic settings; see, for example, the recent work on the proposed FCC double auctions for spectrum [32, 35, 28, 29].

As a first example, consider the variation of the above problem in which the buyers belong to one of three categories (e.g., they are firms that are either small, medium, or large in market share). To ensure diversity among buyers, the policy maker requires that no more than k_i buyers from each category i shall be accepted (for additional *quota* examples see, e.g., [23]).

In this example it is still possible to implement the VCG and trade reduction mechanisms by composing two one-sided algorithms. The only change is to the algorithm used for the buyers. In its forward version we would go through the buyers in order of their value and accept the next buyer if and only if we haven't already accepted k_c buyers from that buyer's category c. In its backward version we would go through the buyers in reverse order and reject the next buyer unless there are k_c or fewer buyers from that category left.

As a second example, consider the variant of the original (unconstrained) problem in which sellers have one of two "sizes", s or S, where $S > s$. For instance, sellers could be firms that pollute the environment to different extents. Suppose there is a cap C on the combined size of the sellers that can be accepted (for additional *packing* examples see, e.g., [1]).

In this example it is less clear what to do. Even putting aside our goal of modular design, computing the welfare-maximizing solution is an NP-hard packing problem [e.g., 26], and so specifically the one-sided greedy by cost algorithm is no longer optimal. We thus shift our attention to approximately-maximizing solutions, but it is not clear which one-sided approximation methods—greedy according to cost, greedy according to cost divided by size, non-greedy algorithms, etc.—would offer good approximation guarantees in the double auction context, where the choices of buyers and sellers are entangled. Furthermore, it is not clear if the good properties of the double-sided VCG and McAfee mechanisms, such as DSIC or BB, would continue to hold.

---

[1] *Threshold payments* are defined in Section 2; informally they are based on the threshold bids of the players, which differentiate between acceptance and rejection by the mechanism.

[2] Using terminology from the study of algorithms in computer science, our approach can also be described at a high level as a "black-box reduction" from designing double auctions to the problem of designing single-sided mechanisms.

[3] For simplicity we focus on a version of McAfee's trade reduction mechanism in which the least valuable pair is always rejected. *Cf.* the full version which is defined as follows: Sort buyers by non-increasing value $v_1 \geq v_2 \geq \ldots$ and sellers by non-decreasing cost $c_1 \leq c_2 \leq \ldots$. Let k be the largest index such that $v_k \geq c_k$. Compute $t = (v_{k+1} + c_{k+1})/2$. If $t \in [c_k, v_k]$ let buyers/sellers $1, \ldots, k$ trade with each other. Otherwise exclude the buyer-seller pair with the k-th highest value and the k-th lowest cost from trade.

1.2. Approach and Results

We advocate a modular approach to the design of double auctions, which is applicable to complex feasibility constraints on both sides of the market. All of the resulting double auction mechanisms are deterministic.

The modular approach breaks the design task into two subtasks: (a) the design of two one-sided algorithms and (b) the design of a composition rule that pairs buyers and sellers. To identify what we want from the respective subtasks, we prove a number of *compositions theorems* of the following general form:

> If the one-sided algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ have properties X_1 and X_2 and the composition rule has property Y, then the resulting double auction has property Z.

A main theme of this work is thus to identify sufficient conditions on the two one-sided algorithms and on the method of composition that guarantee a desired property of the double auction.

We start with sufficient conditions that ensure that the double auction is DSIC, resp., has the stronger incentive properties shared by *deferred-acceptance* implementations generalizing the Gale-Shapley mechanism (see [32, 19] and Section 4). Interestingly, *monotonicity* of all involved components (where the standard definition appears in Section 2) is not sufficient for DSIC; we also need that the one-sided algorithms return the players in order of their "quality", i.e., their contribution to social welfare,[4] and for this reason greedy approaches play an essential role in our designs. In the above examples, the greedy by value and greedy by cost algorithms have this property, while a greedy algorithm based on cost divided by size may violate it. An important consequence of the sufficient conditions we obtain is that trade reduction mechanisms can be implemented within the deferred-acceptance framework, and therefore share the stronger incentive properties of mechanisms within this class. In particular, this approach shows for the first time that McAfee's trade reduction mechanism is WGSP.

We then identify conditions that ensure the double auction obtains a certain fraction of the optimal welfare. These conditions ask that the one-sided algorithms achieve a certain approximation ratio "at all times"—the intuition being that the final number of accepted players is extrinsic since it depends on the interplay with the other side of the market, and so the algorithm should be close to optimal for any possible number of accepted players rather than just for the final number of accepted players. We analyze such guarantees for a number of algorithms, including the greedy by value and greedy by cost algorithms used in the examples above.

We complement the above results with a lower bound on the welfare obtainable by any WGSP mechanism (whether based on composition or not). We conclude that in some cases, including the unconstrained setting and the setting with diversity constraints discussed above, the trade reduction mechanism is not only implementable via the modular composition approach, but it also minimizes the worst-case welfare loss subject to WGSP.

The last property we consider is BB. Here we show that the same conditions on the one-sided algorithms and the composition that enable implementation within the deferred-acceptance framework also lead to BB. We complement this result with a lower bound on the welfare achievable by any BB double auction. We again conclude that in several settings, the trade reduction mechanism minimizes the worst-case welfare loss subject to BB.

1.3. Applications

To demonstrate the usefulness of our modular approach, we use it to design novel double auctions for problems with non-trivial feasibility structure. We focus on three types of feasibility constraints. These serve to illustrate our design framework in several concrete settings, and are not an exhaustive list of the applications of our results. We describe them somewhat informally below, and formally in Section 2.3.

1. Matroids:[5] The set of feasible subsets of players is downward-closed (if a set S is feasible any subset $T \subseteq S$ is feasible), and satisfies an *exchange* axiom (if two sets S, T are feasible and T is larger than S, then there must be an element $u \in T \setminus S$ such that $S \cup \{u\}$ is feasible). The unconstrained problem discussed above as well as the problem with diversity quota constraints are special cases of this category.

[4] A buyer contributes more to social welfare as his value increases; a seller contributes more to social welfare as his cost decreases. In other words, a buyer's quality increases as his ability to extract value from the good increases, and a seller's quality increases as his ability to produce the good at lower cost increases.

[5] Matroid structure corresponds to an economic substitutability condition referred to as *players are substitutes* in the related literature [*cf.*, 46]. This condition requires that the welfare—the total value of all buyers minus the total cost of all sellers—is a submodular function of the set of buyers or sellers, which is the case for matroid feasibility constraints.

2. Knapsacks: Each player has a size and a set of players is feasible if their combined size does not exceed a given threshold. The variation of the unconstrained problem in which sellers have one of two distinct sizes is a special case of this constraint.
3. Matchings: We are given a graph such that each player corresponds to an edge in this graph, and a set of players is feasible if it corresponds to a matching in this graph. A concrete example of this constraint is a setting where the sellers on the market correspond to certain pairs of firms, who can cooperate to produce complementary goods, both of which are required to provide the service being sold on the market. The sellers can thus be thought of as edges of a bipartite graph, where on one side there are firms producing the first complementary good and on the other there are firms producing the other good [*cf.*, 38].

Intuitively, the first setting is precisely the setting in which greedy by quality is optimal. The second and third settings can be thought of as different relaxations of the matroid constraint, in which greedy by quality is not optimal but often performs well.

Our framework yields novel VCG- and trade reduction-style mechanisms for all three settings. The former are DSIC, whereas the latter are WGSP, implementable as a clock auction and BB. It also translates approximation guarantees for greedy algorithms into welfare guarantees for these double auctions. These guarantees show that the welfare degrades gracefully as we move away from settings in which greedy is optimal.

1.4. Further Related Work

The design principle of modularity is embraced in a diverse range of complex design tasks, from mechanical systems through software design to architecture [e.g., 4]. Splitting a complex design task into parts or modules, addressing each separately and then combining the modules into a complete system helps make the design and analysis tractable and robust. Economic mechanisms that operate in complex incentive landscapes while balancing multiple objectives are natural candidates for reaping the benefits of modularity. Two predecessors of our work that apply a modular approach to a mechanism design problem are [33, 13], but they consider different settings than ours (one-sided rather than two-sided), or different objectives (profit in the "competitive analysis" framework rather than welfare, strategyproofness and budget balance).

Most prior work on double auctions is motivated by the impossibility results of [24] and [34], which state that optimal welfare and BB cannot be achieved simultaneously subject to DSIC or even Bayes-Nash incentive compatibility (BIC). One line of work escapes this impossibility by relaxing the efficiency requirement. This direction can be divided into mechanisms that are BIC and mechanisms that are DSIC. An important example of the former is the buyer's bid double auction [42, 41, 43], which sets a single price to equate supply and demand. More recent work that falls into this category is [12, 18]. A prominent example of the latter is McAfee's trade reduction mechanism, which allows all but the least efficient trade. This mechanism has been generalized to more complex settings in [2, 8, 21, 3, 10]. More recent work that falls into this category is [27, 6] (where [27] actually applies *ex post* incentive compatibility, as appropriate for interdependent values). A second line of work that seeks to escape the impossibility results was recently initiated by [9], by analyzing the trade-off between incentives and efficiency while insisting on budget balance. Our work is different in that it adds to the double auction design problem the objectives of feasibility and WGSP, and takes an explicitly modular approach to achieve the objectives.

The WGSP property that we highlight was studied in detail in [25], although a complete characterization of WGSP mechanisms is not known. Deferred-acceptance algorithms on which part of our work is based are proposed in [32], and their performance is analyzed in [15]. Our work extends the deferred-acceptance framework from one-sided settings to two-sided settings.

The greedy approach has been extensively studied in the context of one-sided mechanism design, for both single- and multi-parameter settings; see, e.g., [7] and references within.

1.5. Paper Organization

Section 2 covers preliminaries of the settings to which our framework applies, and formally defines properties of double auctions that we are interested in, including incentive compatibility of different types (DSIC and WGSP), welfare-maximization, and BB; this section can be skipped by the expert reader. Section 3 describes our composition framework: First we define the one-sided algorithms, and then we turn to different methods of composing these one-sided algorithms.

The next three sections are roughly organized by the desired double auction property. Section 4 proves our DSIC and WGSP composition theorems. Section 5 gives our welfare composition theorem. Section 6 proves our BB composition theorem. Finally, Section 7 studies the interplay of welfare, incentives and BB.

2. Problem Statement

This section defines the double auction settings and the properties of double auction mechanisms that we are interested in. We also single out three settings that will serve as running examples.

2.1. Double Auction Settings with Feasibility Constraints

We study single-parameter double auction settings. These are two-sided markets, with n buyers on one side of the market and m sellers on the other. There is a single kind of item for sale. The buyers each want to acquire a single unit of this item, and the sellers each have a single unit to sell. A set of buyers and sellers is *feasible* if the set of buyers is feasible and the set of sellers is feasible, and there are at least as many sellers as there are buyers. Which sets of buyers are feasible is expressed as a set system $(N, \mathcal{I}_N)$, where N is the *ground set* of all n buyers, and $\mathcal{I}_N \subseteq 2^N$ is a non-empty collection of all the feasible buyer subsets. Similarly, feasible seller sets are given as a set system $(M, \mathcal{I}_M)$, where M is the ground set of all m sellers, and $\mathcal{I}_M \subseteq 2^M$ is a non-empty collection of all the feasible seller subsets. The set systems are *downward-closed*, meaning that for every nonempty feasible set, removing any element of the set results in another feasible set. We assume that the two feasibility set systems are represented in a *computationally tractable* way,[6] and are known to the mechanism designer.

Each buyer i has a value $v_i \in [v_i^0, v_i^1]$ where $v_i^1 < \bar{v}$, and each seller j has a cost $c_j \in [c_j^0, c_j^1]$ where $c_j^1 < \bar{c}$, and $\bar{v} = \bar{c}$ are the maximum possible value and cost. For simplicity and without loss of generality, unless stated otherwise, we assume that values and costs are unique and non-negative. The *type spaces* $[v_i^0, v_i^1], [c_j^0, c_j^1]$ are publicly known, and the *bid spaces* are equal to the type spaces. A player's *quality* is his value if he is a buyer, and *minus* his cost if he is a seller. We denote by $\vec{v}$ (resp. $\vec{c}$) the value (resp. cost) profile of all buyers (resp. sellers). The players' utilities are quasi-linear, i.e., buyer i's utility from acquiring a unit at price p_i is $v_i - p_i$, and seller j's utility from selling his unit for payment p_j is $p_j - c_j$. The *optimal welfare* is the maximum difference between the total value and total cost over feasible subsets of players. That is,

$$OPT(\vec{v}, \vec{c}) = \max_{B,S : B \in \mathcal{I}_N, S \in \mathcal{I}_M, |B| \le |S|} \left\{ \sum_{i \in B} v_i - \sum_{j \in S} c_j \right\}.$$

Note that since we consider downward-closed set systems, the optimal will always be attained by buyer set B and seller set S such that $|B| = |S|$.

2.2. Double Auction Mechanisms

We study direct and deterministic[7] double auction mechanisms, which consist of an *allocation rule* $x(\cdot, \cdot)$ and a *payment rule* $p(\cdot, \cdot)$. The allocation rule takes a pair of value and cost profiles $\vec{v}, \vec{c}$ as input, and outputs the set of players who are *accepted* for trade, also referred to as *allocated*. For every buyer i (resp. seller j), $x_i(\vec{v}, \vec{c})$ (resp., $x_j(\vec{v}, \vec{c})$) indicates whether he is allocated by the mechanism. The payment rule also takes a pair of value and cost profiles $\vec{v}, \vec{c}$ as input, and computes payments that the mechanism charges the buyers and pays to the sellers. We use $p_i(\vec{v}, \vec{c})$ to denote the payment buyer i is charged, and $p_j(\vec{v}, \vec{c})$ to denote the payment seller j is paid. A buyer who is not accepted is charged 0 and a seller who is not accepted is paid 0.

The *welfare of a mechanism* is the total value of buyers that it accepts minus the total cost of the sellers that it accepts. That is,

$$W(\vec{v}, \vec{c}) = \sum_{i \in N} x_i(\vec{v}, \vec{c}) \cdot v_i - \sum_{j \in M} x_j(\vec{v}, \vec{c}) \cdot c_j.$$

---

[6]More formally, consider for example the buyer set system. We assume that it is represented succinctly by a tractable algorithm, called a *feasibility oracle*, which for every set of buyers returns whether this set is feasible or not.

[7]A double auction mechanism is *deterministic* if for every input $(\vec{v}, \vec{c})$ and every execution of the mechanism on this input, the mechanism selects the same sets of buyers $B \subseteq N$ and sellers $S \subseteq M$ and sets the same payments $p = (p_i)_{i \in N \cup M}$.

Non-Strategic Properties. We study the following non-strategic properties of double auction mechanisms:

- *Feasibility*. A double auction mechanism is feasible if for every value and cost profiles $\vec{v}, \vec{c}$, the set of accepted buyers and sellers is feasible. Formally, if B is the set of accepted buyers and S is the set of accepted sellers, then $B \in \mathcal{I}_N$, $S \in \mathcal{I}_M$ and $|B| \leq |S|$.
- *Budget balance (BB)*. A double auction mechanism is budget balanced if for every value and cost profiles $\vec{v}, \vec{c}$, the difference between the sum of payments charged from the accepted buyers and the sum of payments paid to the accepted sellers is non-negative.
- *Efficiency*. A double auction mechanism is δ-approximately efficient if for every value and cost profiles $\vec{v}, \vec{c}$, its welfare $W(\vec{v}, \vec{c})$ is at least a $(1/\delta)$-fraction of the optimal welfare $OPT(\vec{v}, \vec{c})$. Clearly, for feasible mechanisms $\delta \geq 1$, and $\delta = 1$ precisely if the mechanism achieves optimal welfare.

Strategic Properties. We also study the following strategic properties of double auction mechanisms:

- *Individual rationality (IR)*. A double auction mechanism is IR if for every value and cost profiles $\vec{v}, \vec{c}$, every accepted buyer i is not charged more than his value v_i, and every accepted seller j is paid at least his cost c_j. Non-accepted players are charged/paid zero.
- *Dominant-strategy incentive compatible (DSIC)*. A double auction mechanism is DSIC if for every value and cost profiles $\vec{v}, \vec{c}$ and for every i, j, v'_i, c'_j, it holds that buyer i is (weakly) better off reporting his true value v_i than any other value v'_i, and seller j is (weakly) better off reporting his true cost c_j than any other cost c'_j. Formally,

$$x_i(\vec{v}, \vec{c}) \cdot v_i - p_i(\vec{v}, \vec{c}) \geq x_i((v'_i, v_{-i}), \vec{c}) \cdot v_i - p_i((v'_i, v_{-i}), \vec{c}),$$

  and similarly for seller j.
- *Weak group-strategyproofness (WGSP)*. A double auction mechanism is WGSP if for every value and cost profiles $\vec{v}, \vec{c}$, for every set of buyers and sellers $B \cup S$ and every alternative value and cost reports of these players v'_B, c'_S, there is at least one player in $B \cup S$ who is (weakly) better off when the players report truthfully as when they report v'_B, c'_S. Intuitively, such a player does not have a strict incentive to join the deviating group.[8]

The following characterization of DSIC and IR double auction mechanisms follows from standard arguments. A similarly simple characterization of WGSP and IR double auction mechanisms is not available.[9]

Definition 2.1. The allocation rule $x(\cdot, \cdot)$ is *monotone* if for all value and cost profiles $\vec{v}, \vec{c}$, every accepted buyer who raises his value while other values and costs remain fixed is still accepted, and every accepted seller who lowers his cost while other values and costs remain fixed is still accepted.

Definition 2.2. For a monotone allocation rule $x(\cdot, \cdot)$, the *threshold payment* of buyer i given profiles $\vec{v}_{-i}, \vec{c}$, and the threshold payment of seller j given profiles $\vec{v}, \vec{c}_{-j}$, are, respectively,

$$\sup_{v_i | \neg x_i(\vec{v}, \vec{c})} v_i, \inf_{c_j | \neg x_j(\vec{v}, \vec{c})} c_j.$$

Intuitively, the threshold payment of a player is the highest value (resp. lowest cost) he can report without being accepted. We can now state the characterization result:

Proposition 2.3. *A double auction mechanism is DSIC and IR if and only if the allocation rule is monotone and the payment rule applies threshold payments.*

Note that threshold payments are sufficient to guarantee IR, and since all the mechanisms we consider apply threshold payments, we do not discuss individual rationality further. Also, since we focus on DSIC mechanisms, we use the terms true value (resp. cost), reported value (resp. cost) and bid interchangeably.

[8] A stronger notion of group strategyproofness requires that no group of buyers and sellers can jointly deviate to make some member of the group strictly better off while all other members are no worse off. This stronger notion is violated by all common double auction formats. For example, if a seller's cost sets the price for a buyer, then the seller can claim to have a lower cost to lower the buyer's payment without affecting his own utility.

[9] See [25] for recent progress towards characterizing WGSP and BB mechanisms in the context of cost sharing mechanisms.

2.3. Running Examples

We now define formally the examples of feasibility constraints mentioned and motivated in Section 1.3. We denote by U the ground set of players (either the buyers or sellers in our context), and by $\mathcal{I}$ the collection of feasible subsets.

1. *Matroids*: We give here a formal definition—see below for intuitive examples from linear algebra and graph theory, and for an explanation of the connection between matroids and the greedy approach. A set system $(U, \mathcal{I})$ is a matroid if it satisfies the following three axioms: (1) $\emptyset \in \mathcal{I}$, (2) for all $S \subset T \subseteq U : T \in \mathcal{I}$ implies $S \in \mathcal{I}$ (downward-closed property), (3) if $S, T \in \mathcal{I}$ and $|T| > |S|$, then there exists $u \in T \setminus S$ such that $S \cup \{u\} \in \mathcal{I}$ (exchange property). The sets in $\mathcal{I}$ are called *independent* and all other sets are called *dependent*. A maximal independent set—that is, an independent set which becomes dependent upon adding any element of U—is called a *basis*, and a minimal dependent set—that is, a dependent set whose proper subsets are all independent—is called a *circuit*.
2. *Knapsacks*: In this case, the elements of the ground set U have publicly-known *sizes* $(s_1, \ldots, s_{|U|})$, and the family of feasible sets $\mathcal{I}$ includes every subset $S \subseteq U$ such that its total size $\sum_{i \in S} s_i$ is at most the capacity C of the knapsack. We denote the ratio between the size of the largest element and the size of the knapsack by $\lambda \leq 1$, and the ratio between the size of the smallest element and the size of the largest element by $\mu \leq 1$. It is assumed that $1/\mu$ is integral.
3. *Matchings*: A third class of feasibility restrictions are bipartite matching constraints. In this case the ground set U is the edge set of some bipartite graph $G = (V, U)$, and the family of feasible sets $\mathcal{I}$ are the subsets of the ground set that correspond to bipartite matchings in this graph.

Matroid Background. As noted in the introduaciton, matroid feasibility constraints correspond to the economic condition of player substitutability. We give two matroid examples and explain the connection between matroids and the greedy approach.

As a first example, consider vector spaces from linear algebra. Define a matroid by letting the ground set U be the set of vectors, and letting the independent sets in $\mathcal{I}$ be the linearly independent vector subsets. It is not hard to verify that the axioms hold, that the bases of the matroid coincide with the bases of the vector space, and that the circuits of the matroid coincide with the minimal dependent sets of vectors.

As a second example, consider undirected graphs from graph theory. Define a matroid by letting the ground set U be the set of graph edges, and letting the independent sets in $\mathcal{I}$ be all the forests in the graph. It is not hard to verify that the axioms hold, that the bases of the matroid coincide with the spanning forests of the graph, and that the circuits of the matroid coincide with the simple cycles of the graph.

There is a close relation between matroids and the greedy algorithm: Consider a matroid whose elements have non-negative weights. The greedy algorithm can be used to find a maximum-weight basis—for example, a maximum-weight spanning forest—by starting from the empty set and repeatedly adding a maximum-weight element among the elements whose addition would preserve the independence of the set. Moreover, matroids are precisely the set system for which such a greedy algorithm works for all weights [16].

While further details about matroids are beyond the scope of this paper, for a comprehensive survey of matroid theory see [36]; for more on matroids in the context of mechanism design see [5].

3. Composition Framework

In this section we describe our framework for designing double auctions via composition. We first describe the one-sided algorithms and then the different ways of composing them.

3.1. Ranking Algorithms

The one-sided algorithms we use for our compositions are called *ranking algorithms*. A ranking algorithm for buyers (resp. sellers) is a deterministic algorithm that receives as input a value profile $\vec{v}$ (resp. cost profile $\vec{c}$), and returns as output an ordered set of buyers (resp. sellers), which we refer to as a *stream*. Not all buyers (resp. sellers) necessarily appear in the stream, e.g., due to feasibility considerations. The *rank* of a buyer (resp. seller), denoted by $r_i(\vec{v})$ (resp. $r_j(\vec{c})$), is his position in the stream (e.g., 1 if he appears first), or ∞ if he does not appear in the stream.[10]

[10]This notion of a player's rank by a ranking algorithm is not to be confused with the notion of a matroid's rank; the latter is not referred to in this paper.

The closer a player's rank is to 1, the smaller or *lower* his rank. Accessing the next player in the stream is called *querying* the ranking algorithm. When querying the kth player, the query *history* is the identities and qualities (values or costs) of the $k-1$ previously-queried players. We say that a history h is a *prefix* of another history h' when the queries recorded in h are the first queries recorded in h'.

We now distinguish between two natural *directions* of ranking algorithms by their different feasibility guarantees. The rest of this section is stated for buyers but applies to sellers as well.

Definition 3.1. A ranking algorithm is *forward-feasible* for a given feasibility set system $(N, \mathcal{I}_N)$ if for every input $\vec{v}$, it returns a stream of buyers $s_1, \ldots, s_n$ such that the following holds: for every $1 \le i \le n$, the set of buyers $\{s_1, \ldots, s_i\}$ is $\in \mathcal{I}_N$.

That is, a ranking algorithm is forward-feasible if for every input, every prefix of the output stream (including the entire stream) is a feasible set according to the feasibility set system.

Definition 3.2. A ranking algorithm is *backward-feasible* for a given set system $(N, \mathcal{I}_N)$ if for every input $\vec{v}$, it returns a stream of buyers $s_1, \ldots, s_n$ and a rank $0 \le \ell < n$ such that the following holds: for every $\ell \le i \le n$, the set of buyers $N \setminus \{s_1, \ldots, s_i\}$ is $\in \mathcal{I}_N$ (if $\ell = 0$ then $N \in \mathcal{I}_N$).

That is, a ranking algorithm is backward-feasible if for every input, there exists a prefix of the output stream such that after discarding it or any larger prefix that contains it, the remaining buyer set is feasible according to the feasibility set system.

The semantic difference between forward-feasible and backward-feasible ranking algorithms is as follows: The former returns a stream of buyers who can be greedily accepted, resulting in a feasible set of accepted buyers; the latter returns a stream of buyers who can be greedily rejected, such that after sufficiently many rejections the remaining buyers can be accepted, again resulting in a feasible set of accepted buyers.

The following example demonstrates the above concepts.

Example 3.3 (Greedy Ranking by Value with Knapsack Constraints)**.** Suppose there are 4 buyers $1, 2, 3, 4$ with sizes $4, 3, 2, 1$ and values $3, 7, 1, 8$, and the feasibility constraint is that the total size of accepted buyers cannot exceed 4. A possible forward-feasible ranking algorithm could rank buyers by highest value first. So it would first return buyer 4 and then buyer 2 at which point the total size has reached 4. In other words, the stream of buyers would be $4, 2$ and the ranks would be $\infty, 2, \infty, 1$. A possible backward-feasible ranking algorithm could rank buyers by lowest value first. In this case the output stream would be $3, 1, 2, 4$ with buyer ranks $2, 3, 1, 4$. The rank ℓ of the largest-rank buyer that must be rejected for feasibility would be $\ell = 2$. Note that in both cases the set of feasible players is the same and achieves optimal welfare. It is not difficult to see that with different values the set of feasible players achieved by these algorithms may be suboptimal.

Running Examples. We discuss two algorithmic approaches to designing ranking algorithms for our running examples.

One general possibility to obtain a ranking algorithm, say for the buyers, is to compute the set of buyers that maximizes total value among all feasible buyer sets. A forward-feasible ranking algorithm could then output the buyers in this set in an arbitrary order. Similarly, a backward-feasible ranking algorithm could first output all the buyers not in this set, followed by all the buyers in the set. This is computationally tractable for matroids and bipartite matching, but for knapsack it is an NP-hard problem [20].

A different class of algorithms that naturally leads to computationally tractable ranking algorithms are greedy algorithms. We give here a brief sketch of this approach, and provide further details in Appendices B, C and D. For matroids there exist forward- and backward-feasible greedy ranking algorithms that compute an optimal solution [16]. The standard greedy algorithm for knapsack orders players in decreasing order by their value divided by size, and then adds the players in this order as long as they still fit into the knapsack; this gives a 2-approximation. An alternative greedy algorithm adds elements in non-decreasing order of value as long as they fit, and obtains a $((1-\lambda)\mu)^{-1}$-approximation (recall that μ, λ are the size ratios of the smallest and largest elements and of the knapsack itself). It is not difficult to see that both these algorithms correspond to forward-feasible ranking algorithms, and can be turned into backward-feasible ranking algorithms by going through the players in reverse order and rejecting them as long as the remaining players do not fit into the knapsack. For matchings, adding edges by non-increasing value unless one of their endpoints is already matched yields a 2-approximation. It is not possible to turn this process around into a backward-feasible ranking algorithm.

3.2. Composition of Ranking Algorithms

We distinguish between compositions of two forward-feasible ranking algorithms, and two backward-feasible ones. We call the former *forward composition* and the latter *backward composition*. A crucial ingredient to both is the following definition of a composition rule.

Definition 3.4. A *composition rule* for a ranking algorithm for buyers and a ranking algorithm for sellers is a boolean function, which receives as input a buyer-seller pair (i, j) composed by querying the two ranking algorithms, the pair's value and cost v_i, c_j, and their query histories, and outputs either 1 ("the pair is accepted") or 0 ("the pair is rejected").

Specific composition rules that we will use in this paper are the t-threshold composition rule, the lookback composition rule, and the lookback t-threshold composition rule defined next.

Definition 3.5. The *t-threshold* composition rule accepts a buyer-seller pair (i, j) if and only if the pair's gain from trade $v_i - c_j$ is at least t, where t is a non-negative threshold in $\mathbb{R}$.

Definition 3.6. A *lookback* composition rule is any rule that decides whether to accept or reject a buyer-seller pair (i, j) without observing their value and cost v_i, c_j, but rather observing only the history of values and costs of previously-queried players.

Definition 3.7. The *lookback t-threshold* composition rule is a lookback composition rule that accepts a buyer-seller pair (i, j) if and only if some part of the history contains a previously-queried pair (i', j'), whose gain from trade $v_{i'} - c_{j'}$ is at least t, where t is a non-negative threshold in $\mathbb{R}$.

We are now ready to formally define forward and backward composition. Intuitively, the main difference between forward and backward composition is in the information that is available to the composition rules they use.

Definition 3.8. The *forward composition* of two forward-feasible ranking algorithms (also see Algorithm 2 in Appendix A) greedily determines an allocation as follows:

1. It queries the output streams of both forward-feasible ranking algorithms. If either ranking algorithm returns no player, then it stops and rejects all remaining players.
2. Otherwise it applies the composition rule to the resulting buyer-seller pair to decide whether or not to accept it based on its value and cost and the history of previous queries.
3. If it accepts the pair, then it continues with Step 1. Otherwise, it stops and rejects all remaining players.

Definition 3.9. The *backward composition* of two backward-feasible ranking algorithms (also see Algorithm 3 in Appendix A) greedily determines an allocation as follows:

0. (*Preprocessing.*) It rejects the first $n - \min\{n - \ell_B, m - \ell_S\}$ buyers, and the first $m - \min\{n - \ell_B, m - \ell_S\}$ sellers by querying the output streams (where recall that ℓ_B, ℓ_S are the ranks of the largest-rank buyer and seller, respectively, that must be rejected for feasibility). The remaining players now form a feasible set.
1. It queries the output streams of both backward-feasible ranking algorithms.
2. It applies the composition rule to the resulting buyer-seller pair to decide whether or not to reject it based on its value and cost and the history of previous queries, excluding preprocessing queries from Step 0.[11]
3. If it rejects the pair, then it continues with Step 1. Otherwise, it stops and accepts all remaining players.

Observation 3.10. *Forward and backward composition leads to a feasible set of accepted buyers and sellers.*

We present two illustrative examples that demonstrate what our composition framework is able to achieve in simple settings.

[11]The exclusion of queries carried out during preprocessing is so that the composition rule will only take into account pairs that could have potentially traded. This is necessary to achieve the budget balance property—see Section 6.

Example 3.11 (VCG Mechanism via Forward Composition)**.** Consider an unconstrained double auction setting. For such a setting, the trivial forward-feasible ranking algorithm ranks the players from high to low quality (i.e., from high to low value or from low to high cost). Observe that the VCG double auction is precisely a forward composition of the trivial forward-feasible ranking algorithms using the 0-threshold composition rule and applying threshold payments. Indeed, it sorts the players from high to low quality and greedily accepts trading pairs (i, j) while their gain from trade $v_i - c_j$ is positive.

Example 3.12 (McAfee's Trade Reduction Mechanism via Backward Composition)**.** In the same unconstrained double auction setting, the trivial backward-feasible ranking algorithm ranks the players in reverse order, from low to high quality (i.e., from low to high value or from high to low cost). McAfee's trade reduction double auction is precisely a backward composition of the trivial backward-feasible ranking algorithms using the lookback 0-threshold composition rule and applying threshold payments. Indeed, it sorts the players from low to high quality and greedily rejects trading pairs, until it has rejected a pair (i, j) whose gain from trade $v_i - c_j$ is non-negative.

More generally, the forward-feasible greedy ranking algorithms *for matroids* combined with the 0-threshold rule and threshold payments are in fact the VCG mechanism, while the backward-feasible greedy ranking algorithms for matroids combined with the lookback 0-threshold rule and threshold payments are precisely McAfee's mechanism. The next example illustrates this for McAfee's mechanism.

Example 3.13 (Trade Reduction on Matroids)**.** Consider a scenario with 4 buyers and values $8, 5, 3, 2$, and 3 sellers with costs $1, 2, 4$. Suppose we can accept at most two sellers for trade (e.g., because they are firms which need a facility to produce the good and there is space for at most two facilities). Note this is a matroid constraint, with a simple matroid called "2-uniform" [36]. The backward-feasible greedy ranking algorithm for matroids would return buyers in order $4, 3, 2, 1$ with $\ell_B = 0$, and sellers in order $3, 2, 1$ with $\ell_S = 1$. So in the preprocessing step of the backward composition, buyers $4, 3$ and seller 3 would be rejected. The lookback 0-threshold composition rule would proceed by also rejecting buyer-seller pair $(2, 2)$. Now for the first time there exists a pair rejected by the composition rule with positive gain from trade, and the mechanism would stop and accept the remaining buyer-seller pair $(1, 1)$.

Even more generally, the composition framework enables us to generalize the VCG and trade reduction mechanisms to accommodate feasibility constraints beyond matroids, as long as the constraints have appropriate forward- and backward-feasible ranking algorithms. This leads to Definitions 4.4 (VCG-style double auction) and 4.10 (trade reduction-style double auction) below.

Running Examples. Our composition framework leads to a wealth of double auction mechanisms via forward or backward composition. Specifically, any of the ranking algorithms described in the previous subsection can be combined with the composition rules described above. However, not all combinations will yield DSIC or WGSP mechanisms or succeed in obtaining a good fraction of the optimal welfare. Our goal in the next few sections will be to develop the theory that explains which properties of the ranking algorithms and the composition rule guarantee that the resulting double auction mechanism has these properties.

4. Incentives

Recall that a composition theorem relates the properties of ranking algorithms and a composition rule to those of the composed double auction mechanism. This section presents our composition theorems for the double auction properties of DSIC and WGSP. In Section 4.1 we focus on the DSIC property. Our DSIC composition theorem applies equally well to both forward and backward compositions; for simplicity of presentation we state it for forward compositions. In Section 4.2 we focus on the WGSP property. Our WGSP composition theorem applies only to backward compositions—this is one of the significant differences between the forward and backward approaches. We discuss the implications of both composition theorems for our three running examples.

4.1. DSIC Composition Theorem

To state the DSIC composition theorem we shall need the following definitions, related to the monotonicity notion from Definition 2.1.

Definition 4.1. A ranking algorithm is *rank monotone* if the rank of a player changes monotonically with his quality as follows:

- In a forward-feasible ranking algorithm for buyers, $v_i < v'_i \implies r_i(\vec{v}) \geq r_i(v'_i, v_{-i})$ for every $\vec{v}, i$.
- In a forward-feasible ranking algorithm for sellers, $c_j < c'_j \implies r_j(\vec{c}) \leq r_j(c'_j, c_{-j}))$ for every $\vec{c}, j$.
- In a backward-feasible ranking algorithm for buyers, $v_i < v'_i \implies r_i(\vec{v}) \leq r_i(v'_i, v_{-i})$ for every $\vec{v}, i$.
- In a backward-feasible ranking algorithm for sellers, $c_j < c'_j \implies r_j(\vec{c}) \geq r_j(c'_j, c_{-j})$ for every $\vec{c}, j$.

Definition 4.2. A ranking algorithm is *consistent* if players' ranks are consistent with their qualities as follows:

- In a forward-feasible ranking algorithm for buyers, for every $i, i' < \infty$, if buyer i's rank is lower than that of buyer i' then $v_i \geq v_{i'}$.
- In a forward-feasible ranking algorithm for sellers, for every $j, j' < \infty$, if seller j's rank is lower than that of seller j' then $c_j \leq c_{j'}$.
- In a backward-feasible ranking algorithm for buyers, let ℓ_B be the rank of the largest-rank buyer to discard for feasibility; for every i, i' with rank $> \ell_B$, if buyer i's rank is lower than that of buyer i' then $v_i \leq v_{i'}$.
- In a backward-feasible ranking algorithm for sellers, let ℓ_S be the rank of the largest-rank seller to discard for feasibility; for every j, j' with rank $> \ell_S$, if seller j's rank is lower than that of seller j' then $c_j \geq c_{j'}$.

What is the relation between rank-monotonicity (Definition 4.1) and consistency (Definition 4.2)? Neither implies the other: A forward-feasible ranking algorithm for buyers which outputs only buyer 1 and only if his value is *lower* than a threshold t is consistent but not rank monotone. A forward-feasible ranking algorithm for buyers which always outputs buyer 1 and then buyer 2, even when $v_2 > v_1$, is rank-monotone but not consistent.

Definition 4.3. Consider a composition rule, and let the following be two different inputs to it: two buyer-seller pairs (i, j) and (i', j'), with qualities (v_i, c_j) and $(v_{i'}, c_{j'})$, and histories (h_i, h_j) and $(h_{i'}, h_{j'})$, respectively. Assume that the second input dominates the first in terms of quality, i.e., $v_{i'} \geq v_i$ and $c_{j'} \leq c_j$. The composition rule is *monotone for forward composition* if for any such two inputs where h' is a prefix of h, if it accepts the pair (i, j) then it accepts the pair (i', j'). The composition rule is *monotone for backward composition* if for any such two inputs where h is a prefix of h', if it accepts the pair (i, j) then it accepts the pair (i', j').

An example of a composition rule that is monotone for forward or backward composition is the t-threshold composition rule, since it ignores the histories and accepts whenever $v_i - c_j \geq t$, which implies that $v_{i'} - c_{j'} \geq t$. The following definition relates the above concepts to the VCG double auction mechanism, generalized to accommodate feasibility constraints.[12]

Definition 4.4. A *VCG-style* double auction mechanism is a forward composition of consistent, rank-monotone ranking algorithms using the 0-threshold composition rule and applying threshold payments.

We are now ready to state our composition theorem; we state here the version for forward composition, but it applies equally well to backward composition by an analogous argument.

Theorem 4.5. *A forward composition of consistent, rank monotone ranking algorithms using a composition rule that is monotone for forward composition and applying threshold payments is a DSIC double auction mechanism.*

[12] The welfare of a VCG-style mechanism as defined in Definition 4.4 is not necessarily optimal. Since we focus on computationally tractable mechanisms, this is as can be expected—maximizing welfare subject to feasibility constraints is not always computationally tractable (assuming P≠NP).

Proof. We apply the characterization of DSIC double auctions in Proposition 2.3 to show that the composition is DSIC. We only need to show that the allocation rule is monotone—this also means that the payments are well-defined. Fix value and cost profiles $\vec{v}, \vec{c}$. We argue that an accepted buyer who raises his value remains accepted; a similar argument shows that an accepted seller who lowers his cost remains accepted, thus completing the proof.

Denote the accepted buyer by i, and the seller with whom i trades by j. Consider the application of the composition rule to the pair (i, j), and denote by (v_i, c_j) and (h_i, h_j) the pair's qualities and histories, respectively. By assumption, the composition rule accepts (i, j). By rank monotonicity of the forward-feasible ranking algorithm for buyers, when i raises his value to $v'_i > v_i$, his rank weakly decreases. Let j' be the seller with whom i is considered for trade by the composition rule after he raises his value and his rank decreases. Consider the application of the composition rule to the pair (i, j'), and denote by $(v'_i, c_{j'})$ and $(h'_i, h_{j'})$ the pair's qualities and histories, respectively. Then by consistency of the forward-feasible ranking algorithm for sellers, $c_{j'} \leq c_j$. Since the composition rule is monotone for forward composition, and since the history h'_i is a prefix of h_i, the pair (i, j') must be accepted for trade as well by the composition rule. □

The following is an immediate corollary of Theorem 4.5.

Corollary 4.6. *Let $t \in \mathbb{R}$ be a threshold. Every forward composition of consistent, rank-monotone ranking algorithms using the t-threshold composition rule and applying threshold payments is DSIC. In particular, VCG-style double auctions are DSIC.*

Necessity of the Conditions. The following examples show that monotonicity of the ranking algorithms and the composition rule is necessary for monotonicity of the double auction's allocation rule and DSIC, and demonstrate why consistency of the ranking algorithms is required.

Example 4.7 (Necessity of Rank Monotonicity)**.** Let $n = m = 1$. Let the type spaces be $[v_1^0, v_1^1] = [1, v_{\max}]$ and $[c_1^0, c_1^1] = [0, 0]$. Consider a forward composition using the 0-threshold composition rule of a forward-feasible ranking algorithm for buyers that outputs buyer 1 if and only if $v_1 < v_{\max}$, and a forward-feasible ranking algorithm for sellers that outputs seller 1. Observe that the ranking algorithm for buyers is consistent but not rank monotone, and the ranking algorithm for sellers is consistent and rank monotone. Let buyer 1's value be $v_{\max}$ (seller 1's cost is determined by his type space). Then no players are paired and accepted unless buyer 1 shades his bid and reports a value $< v_{\max}$, in which case he is paired with seller 1 and accepted.

Example 4.8 (Necessity of Composition Rule Monotonicity)**.** Let $n = m = 1$. Let the type spaces be $[v_1^0, v_1^1] = [1, v_{\max}]$ and $[c_1^0, c_1^1] = [0, 0]$. Consider a forward composition of forward-feasible ranking algorithms for buyers resp. sellers that output buyer 1 resp. seller 1, using a composition rule that accepts a buyer-seller pair (i, j) if and only if the gain from trade $v_i - c_j$ is $\in [1, v_{\max})$. Let buyer 1's value be $v_{\max}$ (seller 1's cost is determined by his type space). Then buyer 1 is paired with seller 1, and the pair is rejected unless buyer 1 shades his bid and reports a value $< v_{\max}$, in which case the pair is accepted.

Example 4.9 (Necessity of Ranking Algorithm Consistency)**.** Let $n = m = 2$. Let the type spaces be $[v_1^0, v_1^1] = [4, 4]$, $[v_2^0, v_2^1] = [1, 2]$, $[c_1^0, c_1^1] = [3, 3]$, and $[c_2^0, c_2^1] = [0, 0]$. Consider a forward composition using the 0-threshold composition rule of a forward-feasible ranking algorithm for buyers that outputs buyer 1 and then buyer 2 if $v_2 < 2$ and outputs only buyer 2 otherwise, and a forward-feasible ranking algorithm for sellers that always outputs seller 1 and then seller 2. Observe that the ranking algorithm for buyers is both rank monotone and consistent, and the ranking algorithm for sellers is rank monotone but not consistent. Let buyer 2's value be 2 (buyer 1's value and the sellers' costs are determined by their type spaces). Then buyer 2 is paired with seller 1 and rejected unless he shades his bid and reports a value < 2, in which case he is paired with seller 2 and accepted.

Running Examples. What implications does the DSIC composition theorem have for our running examples? We sketch here how it applies to all three examples. The forward-feasible ranking algorithms that are based on computing the feasible set of buyers (resp. sellers) with maximum value (resp. minimum cost) are rank-monotone and consistent if the buyers (resp. sellers) are returned from highest to lowest value (resp. lowest to highest cost). An analogous argument applies to the backward-feasible ranking algorithms based on this approach. The greedy algorithms discussed before all lead to rank-monotone ranking algorithms and, with the exception of the greedy algorithm which ranks buyers by value divided by size, also to consistent ranking algorithms.

4.2. WGSP Composition Theorem

For our WGSP composition theorem we leverage the framework of *deferred-acceptance algorithms* [32]. We first explain deferred-acceptance algorithms and their relation to one-sided auctions, and then we discuss how they can be used in the context of ranking algorithms for double auctions.

Deferred-acceptance algorithms are described in Algorithm 1. Applied to a set of players, they output an ordered stream R of rejected players, and thus can form the basis of one-sided auctions as follows: A *deferred-acceptance auction* for sale (procurement) is a one-sided mechanism whose allocation rule runs a maximization (minimization) deferred-acceptance algorithm to get the set R of players to reject. Non-rejected (or "active") players are accepted. By monotonicity of the scoring functions (Algorithm 1), the allocation rule is monotone and so we can set the payment rule to apply threshold payments.

Deferred-acceptance algorithms can also form the basis of backward-feasible ranking algorithms, as follows: A *deferred-acceptance ranking* algorithm for buyers (sellers) is a ranking algorithm that first runs a maximization (minimization) version of a deferred-acceptance algorithm to get R. It sets the first part of the output stream of buyers (sellers) to R, and lets the rank of the largest-rank buyer (seller) to reject be $\ell = |R|$. We require that the result is backward-feasible. The second part of the output stream is obtained by sorting the buyers (sellers) not in R from low to high value (high to low cost). Observe that, by construction and by monotonicity of the scoring functions, deferred-acceptance rankings are rank-monotone and consistent.

ALGORITHM 1: Deferred-Acceptance Algorithm—Maximization and Minimization Versions

```
Input: Set of players P and their bid profile b⃗
Possibly, access to a feasibility set system (P, I_P) (represented in a computationally tractable way)
Initialize A = P    % Set of active players—initially all players
Initialize R = ()   % Ordered stream of rejected players

while A ≠ ∅
  % Let s_i^A(b_i, b_{-A}) for every i ∈ A be a scoring function such that:
  % Player i's score depends on his own bid b_i, the bids of the inactive players b_{-A}, and the set of active players A
  % Player i's score does not depend on the bids of the set of active players A \ {i}
  % Player i's score may possibly depend on the feasibility set system (P, I_P)
  % s_i^A(·,·) is non-negative and weakly increasing in its first argument
  assign every active player i ∈ A the score s_i^A(b_i, b_{-A})

  maximization (minimization) version: if all scores are ∞ (0)
    stop algorithm

  maximization (minimization) version: let i* be a player with lowest finite (highest nonzero) score
  A = A \ {i*}
  append i* to R
end  % while
```

The following definition relates the above concepts to the trade reduction mechanism of McAfee, generalized to accommodate feasibility constraints.

Definition 4.10. A *trade reduction-style* mechanism is a backward composition of deferred-acceptance ranking algorithms using the lookback 0-threshold composition rule and applying threshold payments.

We are now ready to state our WGSP composition theorem.

Theorem 4.11. *A backward composition of deferred-acceptance ranking algorithms using a lookback composition rule and applying threshold payments is a WGSP double auction mechanism.*

Proof. It is sufficient to show that the allocation rule of the backward composition in the theorem statement can be implemented by a deferred-acceptance algorithm applied to the set of all players $N \cup M$: Consider the one-sided deferred-acceptance auction for sale based on this algorithm; by construction its allocation rule is identical to the original allocation rule of the backward composition, and both mechanisms apply threshold payments. Thus the incentives of the players in both mechanisms are identical. The theorem then follows from Corollary 1 of [32], by which every deferred-acceptance auction is WGSP.[13]

Consider the backward composition of deferred-acceptance ranking algorithms. We begin by transforming the deferred-acceptance ranking algorithm for sellers into a deferred-acceptance ranking algorithm for *pseudo*-buyers, which runs a maximization rather than minimization version of a deferred-acceptance algorithm:

- The original sellers become pseudo-buyers by multiplying their original costs by -1 and treating the result as the values of the pseudo-buyers for being accepted. In other words, if seller j incurs a cost of c_j when accepted, then the corresponding jth pseudo-buyer's value for being accepted is $-c_j$. Note that pseudo-buyers have non-positive values.
- We define new scoring functions for the deferred-acceptance ranking algorithm for the pseudo-buyers. Consider the new scoring function $s_j^P(b_j, b_{-P})$, where j is a pseudo-buyer, P is the set of active pseudo-buyers, and b_j, b_{-P} are the non-positive reported values of pseudo-buyers. Then $s_j^P(b_j, b_{-P})$ is defined from the original scoring function $\hat{s}_j^P$ of seller j, as follows:

$$s_j^P(b_j, b_{-P}) = \begin{cases} \infty, & \text{if } \hat{s}_j^P(-b_j, -b_{-S}) == 0 \\ -\hat{s}_j^P(-b_j, -b_{-S}) + C, & \text{otherwise} \end{cases}$$

  where C is a large enough constant to make the scores positive (C depends on the original scoring functions and on the maximum cost $\overline{c}$). Observe that every new scoring function s_j^P is weakly increasing in its first argument, as required for a deferred-acceptance algorithm (Algorithm 1).

It is not hard to see that the maximization deferred-acceptance algorithm for pseudo-buyers with the new scoring functions is equivalent in its output to the minimization deferred-acceptance algorithm for sellers with the original scoring functions (the equivalence is by replacing the pseudo-buyers with the corresponding sellers). Thus the resulting deferred-acceptance ranking algorithms are equivalent in their output streams.

Our goal for the remainder of the proof is to use the deferred-acceptance ranking algorithms for buyers and for pseudo-buyers, in particular the original scores $s_i^B(v_i, v_{-B})$ for buyers together with the new scores for pseudo-buyers as well as the ranks ℓ_B, ℓ_P of the largest-rank buyer and pseudo-buyer to reject for feasibility, in order to obtain a maximization deferred-acceptance algorithm for the set of buyers and pseudo-buyers. This algorithm will be equivalent to the allocation rule of the backward composition in the theorem statement (in the sense that the two are identical after replacing the pseudo-buyers with the corresponding sellers).

To achieve the above, we need to define scoring functions such that Algorithm 1 will implement the backward composition whose steps are described in Algorithm 3. In defining the scoring functions we shall utilize the lookback composition rule of the backward composition, and the fact that scoring functions are allowed to depend on the set of active players. The scoring functions are as follows:

- Implementing the preprocessing (Step 0): If the number of inactive buyers is less than ℓ_B, i.e., the set B of active buyers is infeasible, the functions set the scores of all active buyers to $s_i^B(v_i, v_{-B})$, and the scores of all active pseudo-buyers to ∞. This means that the player who will be removed next from the set of active players is the buyer with the lowest score.

  Once the set of active buyers B is feasible, if the number of inactive pseudo-buyers is less than ℓ_P, i.e., the set of active pseudo-buyers P is infeasible, the functions set the scores of all active pseudo-buyers to $s_j^P(b_j, b_{-P})$, and the scores of all active buyers to ∞. This means that the player who will be removed next from the set of active players is the pseudo-buyer with the lowest score.

[13] Note that while the deferred-acceptance framework of [32] primarily focuses on finite bid spaces, some of the results including Corollary 1 apply to infinite bid spaces as well, as in our setting (see Footnote 17 of [32]).

- Maintaining the balance $|B| = |P|$: If there are more active pseudo-buyers than buyers, i.e., $|B| < |P|$, the functions set the scores of all active pseudo-buyers to $b_j + C$, where C is a large enough constant to make the scores positive, and the scores of all active buyers to ∞. This means that the player who will be removed next from the set of active players is the pseudo-buyer with the lowest value. On the other hand, if $|B| > |P|$, the functions set the scores such that the player who will be removed next is the buyer with the lowest value. This is an accurate implementation of the composition whose deferred-acceptance ranking algorithms are consistent.
- Implementing the composition rule (Step 2): The interesting case is when the set of active buyers B is feasible, the set of active pseudo-buyers P is feasible, and $|B| = |P|$. In this case the scores should implement the decision of the lookback composition rule for the next buyer-seller pair. Here we use the fact that the decision only depends on previously-rejected players, excluding those rejected during the preprocessing. Since the scoring functions are allowed to depend on the reports of the inactive players, and in particular can identify the players rejected during the preprocessing, they can simulate the decision of the lookback composition rule and assign scores accordingly:
  - If the decision is to reject the next pair, the functions set the scores such that the player who will be removed next from the set of active players is the buyer with the lowest value—again using the consistency of the deferred-acceptance ranking algorithms. (The matching pseudo-buyer with the lowest value will then be removed as part of maintaining the balance.)
  - If the decision is to accept the next pair, the functions set the scores of all active players to ∞. This brings the deferred-acceptance algorithm to stop, rejecting all players previously removed from the set of active players.

This completes the construction of the scoring functions. It is not hard to check that they are weakly increasing in their first argument, thus providing a deferred-acceptance implementation of the backward composition, as required. □

The following is an immediate corollary of Theorem 4.11.

Corollary 4.12. *Let $t \in \mathbb{R}$ be a threshold. Every backward composition of deferred-acceptance ranking algorithms using the lookback t-threshold composition rule and applying threshold payments is WGSP. In particular, trade reduction-style double auctions are WGSP.*

Two further corollaries apply to backward compositions of deferred-acceptance ranking algorithms using a lookback composition rule, after restricting attention to finite bid spaces: (i) Such double auctions can be implemented as clock auctions (by Proposition 3 in [32]); (ii) For every such double auction, consider a double auction that uses the same allocation rule but charges first-price payments, then it has a complete-information Nash equilibrium in which the allocation and payments are identical to the DSIC outcome of the backward composition double auction with threshold payments (by Proposition 6 in [32]).

Necessity of Backward Composition. We demonstrate that none of the strong properties shown in this subsection are shared by forward compositions. In Appendix E we further show that WGSP (like budget balance) cannot be achieved by forward compositions without losing a great deal in welfare.

Consider for example an unconstrained double auction setting and the VCG double auction mechanism: forward composition of the trivial forward-feasible ranking algorithms (greedy ranking by quality), using the 0-threshold composition rule and applying threshold payments. We show that this double auction, while DSIC, is not WGSP, cannot be implemented by a clock auction and does not have first-price equivalence.

Example 4.13. Let $n = m = 1$. Let the type spaces be $[v_1^0, v_1^1] = \{1, \dots, 4\}$ and $[c_1^0, c_1^1] = \{1, \dots, 4\}$. If buyer 1's value is 3 and seller 1's cost is 2 then the above double auction accepts both players, charges the buyer 2 and pays the seller 3. If buyer 1's value is 4 and seller 1's cost is 1 then the double auction accepts both players, charges the buyer 1 and pays the seller 4.

- *Failure of WGSP*: Even if the true value and cost are 3 and 2, respectively, both players are strictly better off by reporting 4 and 1 instead.

- *Failure of implementability as a clock auction*: A clock auction proposes a sequence of offers to each player, where the offers decrease in their attractiveness. A player may decline his offer and exit the clock auction, or stay active by accepting his offer. The final allocation and prices are determined by the set of final offers and active players upon termination of the clock auction. To be equivalent to the above double auction, the clock auction's allocation and prices should be the same as those of the double auction.

  If the true value and cost are 4 and 1, the clock auction's final offer for the buyer must be a price of 1 and its final offer for the seller must be a payment of 4, and once these offers are accepted the clock auction should stop. But this means that even if the true value and cost are 3 and 2, the players payments will still be 1 and 4, thus differing from those of the above double auction.

- *Failure of first-price equivalence*: Consider the allocation rule of the double auction coupled with first prices (i.e., a pay-as-bid payment rule). If the true value and cost are 3 and 2, to achieve the same payments as the original double auction the buyer's bid must be 2 and the seller's bid must be 3, but then the allocation rule rejects both players.

Running Examples. As in the case of the DSIC composition theorem, we ask what implications does the WGSP composition theorem have for our running examples, and sketch how it applies to all three examples.

The greedy algorithm for matroids can be implemented as a deferred-acceptance algorithm (see Appendix B for details). The greedy algorithm for knapsack which sorts buyers by non-increasing value can also be implemented as a deferred-acceptance algorithm (see Appendix C for details). For matchings, neither the optimal nor the greedy by weight algorithm can be implemented via deferred-acceptance; we therefore design a new deferred-acceptance algorithm for matchings (see Appendix D for details).

5. Welfare

In this section we discuss the welfare guarantees of double auction mechanisms arising from compositions, and the implications for the three running examples.

Throughout this section we will consider fixed feasibility constraints as given by the set systems $\mathcal{I}_N$, $\mathcal{I}_M$ and the requirement that a set of buyers $B \subseteq N$ and a set of sellers $S \subseteq M$ is feasible if $B \in \mathcal{I}_N$, $S \in \mathcal{I}_M$ and $|B| \leq |S|$.

Our goal will be to relate the welfare $W(\vec{v}, \vec{c})$ achieved by a fixed double auction mechanism obtained through composition to the optimal welfare $OPT(\vec{v}, \vec{c})$ for value and cost profile $(\vec{v}, \vec{c})$. Specifically, we will identify properties related to the composition rule and the ranking algorithms that guarantee that the resulting double auction mechanism achieves optimal or near-optimal welfare on every input.

Recall that since we assume downward-closed feasibility sets, the optimal welfare will be achieved by a set of buyers and a set of sellers of equal cardinality. In our analysis, we will use $s^*(\vec{v}, \vec{c})$ to denote the cardinality of the pair (B^*, S^*) with the maximum number of trades among all pairs $(B, S) \in \mathcal{I}_N \times \mathcal{I}_M$ with $|B| = |S|$ achieving optimal welfare $\mathrm{OPT}(\vec{v}, \vec{c})$.

5.1. Welfare Composition Theorem

To state our welfare composition theorem we need a few parameters that quantify relevant properties of the composition rule and the ranking algorithms and how hard the problem instance is.

The first pair of parameters is related to how "exhaustive" the composition rule is. The first parameter $s(\vec{v}, \vec{c})$ denotes the number of buyer-seller pairs that the double auction mechanism accepts on input $(\vec{v}, \vec{c})$. The second parameter $s'(\vec{v}, \vec{c})$ denotes the optimal number of buyer-seller pairs that a composition mechanism based on the same ranking algorithms but with a potentially different composition rule could have accepted. In other words, $s'(\vec{v}, \vec{c})$ is the number of buyer-seller pairs that the 0-threshold rule would accept. Note that for unconstrained double auction settings it is possible that

$$\forall \vec{v}, \vec{c} : s'(\vec{v}, \vec{c}) = s^*(\vec{v}, \vec{c});$$

and this is also the case for some constrained settings such as matroids.

In our analysis we will focus on cases where $s'(\vec{v}, \vec{c}) \geq 1$ and $s(\vec{v}, \vec{c}) \leq s'(\vec{v}, \vec{c})$. The former means that a forward (or backward) composition based on these ranking algorithms could have accepted at least one buyer-seller pair with non-negative gain from trade. The latter means that the double auction mechanism under consideration does not accept buyer-seller pairs with negative gain from trade.

The next two parameters $\alpha \geq 1$ and $\beta \geq 1$ quantify how close to the optimal solution the one-sided ranking algorithms are *at any point q of their execution*, in the worst case over all inputs $\vec{v}$ resp. $\vec{c}$. Intuitively, such an "any time" guarantee is necessary as the final number of accepted buyers depends on the interaction of the two ranking algorithms and is therefore extrinsic to the buyer-ranking algorithm (and similarly for the sellers).

Formally, let $card(\mathcal{I}_N) = \max_{B\in\mathcal{I}_N}|B|$ and $card(\mathcal{I}_M) = \max_{S\in\mathcal{I}_M}|S|$ denote the cardinality of the largest feasible buyer resp. seller set. For every $q \in [card(\mathcal{I}_N)]$, denote by $v_{\text{OPT}}(q)$ the value of the feasible solution of at most q buyers that maximizes total value. For every $q \in [card(\mathcal{I}_M)]$, denote by $c_{\text{OPT}}(q)$ the cost of the feasible solution of at least q sellers that minimizes total cost. That is,

$$v_{OPT}(q) = \max_{B\in\mathcal{I}_N,|B|\leq q}\sum_{i\in B} v_i \quad \text{and} \quad c_{OPT}(q) = \min_{S\in\mathcal{I}_M,|S|\geq q}\sum_{j\in S} c_j.$$

For a given forward-feasible ranking algorithm, we append 0's to the buyer stream if it has length less than $card(\mathcal{I}_N)$ and we append $\bar{c}$'s to the seller stream if it has length less than $card(\mathcal{I}_M)$. We denote by $v_{\text{ALG}}(q)$ (resp. $c_{\text{ALG}}(q)$) the value (resp. cost) achieved by greedily allocating to the first $q \in [card(\mathcal{I}_N)]$ buyers (resp. $q \in [card(\mathcal{I}_M)]$ sellers) in the modified output stream. For a given backward-feasible ranking algorithm, the definitions are the same except that the last q buyers (resp. sellers) in the feasible part of the output stream are considered. A ranking algorithm for buyers is a *uniform α-approximation* if for every value profile $\vec{v}$ and every $q \leq card(\mathcal{I}_N)$,

$$v_{\text{ALG}}(q) \geq \frac{1}{\alpha}\cdot v_{\text{OPT}}(q).$$

A ranking algorithm for sellers is a *uniform β-approximation* if for every cost profile $\vec{c}$ and every $q \leq card(\mathcal{I}_M)$,

$$c_{\text{ALG}}(q) \leq \beta\cdot c_{\text{OPT}}(q).$$

The closer $\alpha \geq 1$ and $\beta \geq 1$ are to 1, the better the ranking algorithms.

The final parameter $\gamma(\vec{v},\vec{c})$ measures how difficult the problem instance $(\vec{v},\vec{c})$ is; and is a standard tool with mixed-sign objective functions [cf. 40]. It measures how close the optimal solution $OPT(\vec{v},\vec{c})$ is to zero. Recall the definition of $s^*(\vec{v},\vec{c})$ from above, then $OPT(\vec{v},\vec{c}) = v_{OPT}(s^*(\vec{v},\vec{c})) - c_{OPT}(s^*(\vec{v},\vec{c}))$. Let $\gamma(\vec{v},\vec{c}) = v_{OPT}(s^*(\vec{v},\vec{c}))/c_{OPT}(s^*(\vec{v},\vec{c}))$. Clearly, $\gamma \geq 1$ as $v_{OPT}(s^*(\vec{v},\vec{c})) \geq c_{OPT}(s^*(\vec{v},\vec{c}))$. For $\gamma(\vec{v},\vec{c}) = 1$ we have $OPT(\vec{v},\vec{c}) = 0$; hence we focus on the case where $\gamma(\vec{v},\vec{c}) > 1$ below. Intuitively, the closer $\gamma(\vec{v},\vec{c})$ is to 1, the closer the optimal welfare is to 0, and the harder it is to achieve a good relative approximation.

Theorem 5.1. *Consider input $(\vec{v},\vec{c})$ for which $OPT(\vec{v},\vec{c}) > 0$. The forward (backward) composition of two consistent, forward-feasible (backward-feasible) ranking algorithms that are uniform α- and β-approximations, using a composition rule that accepts the $s(\vec{v},\vec{c}) \leq s'(\vec{v},\vec{c})$ lowest (resp. highest) ranking buyer-seller pairs, achieves welfare at least*

$$\frac{s(\vec{v},\vec{c})}{s'(\vec{v},\vec{c})}\cdot\frac{\frac{\gamma(\vec{v},\vec{c})}{\alpha}-\beta}{\gamma(\vec{v},\vec{c})-1}\cdot \text{OPT}(\vec{v},\vec{c}).$$

Note that if $\alpha = \beta = 1$, then the second term in the approximation factor vanishes. For general α and β the bound degrades gracefully from this ideal case, in the sense that the dependence on the approximation ratios $\frac{1}{\alpha}$ and β is linear.

Proof. Our goal is to show that

$$v_{ALG}(s(\vec{v},\vec{c})) - c_{ALG}(s(\vec{v},\vec{c})) \geq \frac{s(\vec{v},\vec{c})}{s'(\vec{v},\vec{c})}\cdot\frac{\frac{\gamma(\vec{v},\vec{c})}{\alpha}-\beta}{\gamma(\vec{v},\vec{c})-1}\cdot\Big(v_{OPT}(s^*(\vec{v},\vec{c})) - c_{OPT}(s^*(\vec{v},\vec{c}))\Big).$$

Since the double auction is composed of forward-feasible (backward-feasible) consistent ranking algorithms, we can number the buyers and sellers from the beginning (end) of the respective streams by $1, 2, \ldots$ such that $v_1 \geq v_2 \geq \cdots \geq v_{s'(\vec{v},\vec{c})}$ and $c_1 \leq c_2 \leq \cdots \leq c_{s'(\vec{v},\vec{c})}$. Using this notation,

$$v_{ALG}(s(\vec{v},\vec{c})) - c_{ALG}(s(\vec{v},\vec{c})) = \sum_{i=1}^{s(\vec{v},\vec{c})}(v_i - c_i) \text{ and } v_{ALG}(s'(\vec{v},\vec{c})) - c_{ALG}(s'(\vec{v},\vec{c})) = \sum_{i=1}^{s'(\vec{v},\vec{c})}(v_i - c_i).$$

Another implication of the fact that the double auction is composed of consistent ranking algorithms is that the gain from trade is non-increasing. That is, $i < j$ implies $v_i - c_i \geq v_j - c_j$. Hence for all s such that $s(\vec{v}, \vec{c}) < s \leq s'(\vec{v}, \vec{c})$ we have $v_s - c_s \leq \frac{1}{s(\vec{v},\vec{c})} \sum_{i=1}^{s(\vec{v},\vec{c})} (v_i - c_i)$. It follows that

$$\begin{aligned} v_{ALG}(s(\vec{v}, \vec{c})) - c_{ALG}(s(\vec{v}, \vec{c})) &= \sum_{i=1}^{s'(\vec{v},\vec{c})} (v_i - c_i) - \sum_{i=s(\vec{v},\vec{c})+1}^{s'(\vec{v},\vec{c})} (v_i - c_i) \\ &\geq \sum_{i=1}^{s'(\vec{v},\vec{c})} (v_i - c_i) - \left(s'(\vec{v}, \vec{c}) - s(\vec{v}, \vec{c})\right) \cdot \frac{1}{s(\vec{v}, \vec{c})} \sum_{i=1}^{s(\vec{v},\vec{c})} (v_i - c_i) \\ &= \left(v_{ALG}(s'(\vec{v}, \vec{c})) - c_{ALG}(s'(\vec{v}, \vec{c}))\right) - \left(\frac{s'(\vec{v}, \vec{c})}{s(\vec{v}, \vec{c})} - 1\right) \cdot \left(v_{ALG}(s(\vec{v}, \vec{c})) - c_{ALG}(s(\vec{v}, \vec{c}))\right). \end{aligned}$$

Rearranging this shows

$$v_{ALG}(s(\vec{v}, \vec{c})) - c_{ALG}(s(\vec{v}, \vec{c})) \geq \frac{s(\vec{v}, \vec{c})}{s'(\vec{v}, \vec{c})} \cdot \left(v_{ALG}(s'(\vec{v}, \vec{c})) - c_{ALG}(s'(\vec{v}, \vec{c}))\right). \tag{1}$$

Recall that $s^*(\vec{v}, \vec{c})$ is defined as the number of trades in a solution that maximizes welfare, while $s'(\vec{v}, \vec{c})$ is the number of trades that maximizes welfare for the given ranking algorithms. By the definition of $s'(\vec{v}, \vec{c})$ all trades up to and including $s'(\vec{v}, \vec{c})$ are beneficial, and then either one of the streams reached its end or the subsequent trades are no longer beneficial. Hence, by the definition of v_{ALG} and c_{ALG},

$$v_{ALG}(s'(\vec{v}, \vec{c})) - c_{ALG}(s'(\vec{v}, \vec{c})) \geq v_{ALG}(s^*(\vec{v}, \vec{c})) - c_{ALG}(s^*(\vec{v}, \vec{c})). \tag{2}$$

Finally, we use that the ranking algorithms are uniform α- and β-approximations and the definition of $\gamma(\vec{v}, \vec{c})$ to deduce that

$$\begin{aligned} v_{ALG}(s^*(\vec{v}, \vec{c})) - c_{ALG}(s^*(\vec{v}, \vec{c})) &\geq \frac{1}{\alpha} \cdot v_{OPT}(s^*(\vec{v}, \vec{c})) - \beta \cdot c_{OPT}(s^*(\vec{v}, \vec{c})) \\ &= \left(\frac{\gamma(\vec{v}, \vec{c})}{\alpha} - \beta\right) \cdot c_{OPT}(s^*(\vec{v}, \vec{c})) \\ &= \frac{\frac{\gamma(\vec{v},\vec{c})}{\alpha} - \beta}{\gamma(\vec{v}, \vec{c}) - 1} \cdot \left(v_{OPT}(s^*(\vec{v}, \vec{c})) - c_{OPT}(s^*(\vec{v}, \vec{c}))\right). \end{aligned} \tag{3}$$

Combining inequalities (1)–(3) completes the proof. □

We obtain the following corollaries for VCG- and trade reduction style-mechanisms with the 0-threshold or the lookback 0-threshold composition rule.

Corollary 5.2. *Consider input $(\vec{v}, \vec{c})$ for which $OPT(\vec{v}, \vec{c}) > 0$. Consider the forward composition of two forward-feasible, consistent ranking algorithms that are uniform α- and β-approximations. The 0-threshold rule accepts the $s'(\vec{v}, \vec{c})$ lowest ranking buyer-seller pairs. Hence its welfare is at least*

$$\frac{\frac{\gamma(\vec{v},\vec{c})}{\alpha} - \beta}{\gamma(\vec{v}, \vec{c}) - 1} \cdot OPT(\vec{v}, \vec{c}).$$

Corollary 5.3. *Consider input $(\vec{v}, \vec{c})$ for which $OPT(\vec{v}, \vec{c}) > 0$. Consider the backward composition of two backward-feasible, consistent ranking algorithms that are uniform α- and β-approximations. The lookback 0-threshold rule accepts the $s'(\vec{v}, \vec{c}) - 1$ lowest ranking buyer-seller pairs. Hence its welfare is at least*

$$\left(1 - \frac{1}{s'(\vec{v}, \vec{c})}\right) \cdot \frac{\frac{\gamma(\vec{v},\vec{c})}{\alpha} - \beta}{\gamma(\vec{v}, \vec{c}) - 1} \cdot OPT(\vec{v}, \vec{c}).$$

When $\alpha = \beta = 1$, these two corollaries specialize to the traditional guarantees of the VCG and trade reduction mechanisms. For general α and β these bounds again degrade gracefully from this ideal case as the dependence on the approximation ratios $\frac{1}{\alpha}$ and β is again linear.

Running Examples. Since all ranking algorithms that we have not yet ruled out are consistent, it is the uniform approximation property that we have to check. The greedy algorithms for matroids are not only optimal, but also uniformly so (as we show in Appendix B). Similarly, the algorithm for knapsacks that ranks by weight is a uniform $((1-\lambda)\mu)^{-1}$ approximation (as we show in Appendix C). For matchings, we show that the algorithm that we propose is a uniform 2-approximation (see Appendix D).

6. Budget Balance

This section studies the budget balance properties of compositions, and derives implications for the running examples. The budget balance composition theorem is as follows. We say that a backward composition *reduces an efficient trade* if there is a buyer-seller pair with non-negative gain from trade that is rejected by the composition rule (Step 2 of Algorithm 3).[14] Then:

Theorem 6.1. *A backward composition of deferred-acceptance ranking algorithms using a lookback composition rule that reduces at least one efficient trade and applying threshold payments is a BB double auction mechanism.*

Proof. Without loss of generality, denote the buyers in the output stream of the deferred-acceptance ranking algorithm for buyers by $1, 2, \ldots, n$, and the sellers in the output stream of the deferred-acceptance ranking algorithm for sellers by $1, 2, \ldots, m$. Recall that a backward composition is a composition of two backward-feasible ranking algorithms; the streams returned by such algorithms each have a player, denoted by ℓ_B and ℓ_S respectively for buyers and sellers, such that the following holds: for every $i \geq \ell_B$, the set of buyers $N \setminus \{1, \ldots, i\}$ is feasible, and for every $j \geq \ell_S$, the set of sellers $M \setminus \{1, \ldots, j\}$ is feasible.

Let (ℓ'_B, ℓ'_S) be a buyer-seller pair with non-negative gain from trade that is reduced by the composition rule. Since players $1, \ldots, \max\{\ell_B, \ell_S\}$ from both streams are rejected in the preprocessing stage of the backward composition (Algorithm 3), it must be the case that both ℓ'_B and ℓ'_S are strictly greater than $\max\{\ell_B, \ell_S\}$, i.e., they appear in their respective streams after the players rejected in the preprocessing stage. Denote the value of ℓ'_B by v', and the cost of ℓ'_S by c'. Clearly, $v' \geq c'$.

Our goal is to show that every buyer whom the composition accepts pays at least v'. A symmetric argument shows that every seller whom the composition accepts is paid at most c'. Since $v' \geq c'$ this is sufficient to establish the property of budget balance. In fact, due to threshold payments, it is enough to show that every buyer i whom the composition accepts will be rejected if he reports a value lower than v'.

Consider an accepted buyer i. By the greediness of backward composition, which repeatedly rejects until the first buyer-seller pair is accepted, it must be the case that $i > \ell'_B$, i.e., buyer i appears after the reduced buyer ℓ'_B in the buyer ranking. By consistency of the deferred-acceptance ranking algorithm for buyers, i's original report is thus at least v'. What changes if i reports a value lower than v'? By consistency, the buyer ranking must change in this case, and we denote by r the new rank of buyer i. We distinguish two cases:

- $r \leq \max\{\ell_B, \ell_S\}$. That is, the new rank of i is smaller than the original rank of the largest-ranked buyer to discard for feasibility. We now exploit a property of deferred-acceptance algorithms together with consistency to establish that in the new buyer stream where buyer i appears in rank r, the first $r-1$ buyers have not changed and are still buyers $1, ..., r-1$. The property we use is that an active player's bid does not affect the scores of any other active player in the deferred-acceptance algorithm. Since we know that rejecting buyers $1, \ldots, r-1$ is not enough for feasibility, buyer i is necessarily rejected.

- $r > \max\{\ell_B, \ell_S\}$. As above, the first $r-1$ buyers in the new buyer stream have not changed and are still buyers $1, \ldots, r-1$. Therefore, by consistency and since i reports a value lower than v', it must hold that $r \leq \ell'_B$. We

[14] In the proof we use the property of backward composition by which the composition rule does not take into account efficient trades reduced in the preprocessing stage (Step 0 of Algorithm 3). The following example shows why this is necessary: Consider a setting with two buyers and two sellers. The value profile is $(8, 4)$ and the cost profile is $(6, 5)$. It is only feasible to accept up to one buyer, and the deferred-acceptance ranking algorithm for buyers scores the first buyer by his value and the second buyer by 3 times his value. The sellers are unconstrained and ranked by the trivial backward-feasible ranking algorithm (which clearly has a deferred-acceptance implementation). Thus the buyer stream is $1, 2$ and the seller stream is $1, 2$, and the preprocessing step rejects buyer 1 and seller 1, a pair with positive gain from trade. If this pair were part of the history, the pair buyer 2 and seller 2 would be accepted. But this pair has negative gain from trade and so the resulting mechanism is either not IR or not BB.

now use the fact that originally the buyer-seller pair (ℓ'_B, ℓ'_S) was reduced. This means that the decision of the lookback composition rule given the history up to and including rank $r-1$ is to reject, and so buyer i is rejected.

This completes the proof. $\square$

Thus, when uniform 1-approximate ranking algorithms are available, the trade reduction mechanism is BB. For cases where either $\alpha > 1$ or $\beta > 1$, Theorem 6.1 shows that suitable generalizations of this mechanism are BB.

Running Examples. Our BB composition theorem applies to all ranking algorithms that are implementable within the deferred-acceptance framework: the greedy algorithm for matroids, the greedy by weight algorithm for knapsack, and the new matching algorithm that we describe in Appendix D.

7. Lower Bounds

This section investigates the interplay between welfare on one hand and incentives and budget balance on the other. We prove lower bounds on the welfare achievable by double auctions (compositions or not) that are either WGSP or DSIC and BB. We show the lower bounds for the most basic setting, the unconstrained double auction setting.

7.1. Lower Bound Subject to WGSP

Our lower bound for WGSP mechanisms applies to deterministic, anonymous double auction mechanisms. A double auction mechanism for problem instance $\mathcal{I}_N, \mathcal{I}_M$ is *anonymous* if any renaming of the players does not change the players' payoffs. Formally, denote the utility of player $i \in N \cup M$ given input $(\vec{v}, \vec{c})$ by $u_i(\vec{v}, \vec{c})$. Then, for every permutation π of the buyers and sellers such that $\pi(i) \in N$ for all $i \in N$ and $\pi(i) \in M$ for all $i \in M$ and all inputs $(\vec{v}, \vec{c})$ it holds that $u_i((v_i)_{i\in N}, (c_i)_{i\in M}) = u_{\pi(i)}((v_{\pi(i)})_{i\in N}, (c_{\pi(i)})_{i\in N})$. While natural double auctions for our setting are anonymous, it would also be interesting to extend our lower bound to non-anonymous mechanisms.

Theorem 7.1. *Consider an unconstrained double auction setting. That is, $\mathcal{I}_N = 2^N$ and $\mathcal{I}_M = 2^M$ and therefore $B \subseteq N$ and $S \subseteq M$ are feasible whenever $|B| \leq |S|$. Let $\hat{c} > \hat{v} > 0$. Consider valuations $v_i \in [0, \hat{v}]$ for all $i \in N$ and costs $c_j \in [0, \hat{c}]$ for all $j \in M$. Recall that $s^*(\vec{v}, \vec{c})$ denotes the maximum number of trades in a welfare-maximizing solution for input $(\vec{v}, \vec{c})$. Then no deterministic, anonymous double auction mechanism that is WGSP can guarantee a worst-case approximation guarantee strictly better than*

$$1 - \frac{1}{s^*(\vec{v}, \vec{c})}.$$

Proof. Assume by contradiction that there is a deterministic, anonymous double auction mechanism that is WGSP and achieves a strictly better worst-case approximation guarantee. Then there must be an $\epsilon > 0$ such that for all inputs $(\vec{v}, \vec{c})$ the mechanism achieves welfare at least

$$W(\vec{v}, \vec{c}) \geq \left(1 - \frac{1}{s^*(\vec{v}, \vec{c})} + \epsilon\right) \cdot OPT(\vec{v}, \vec{c}).$$

Consider the following class of inputs $(\vec{v}_{s,v}, \vec{c}_{s,c})$ parameterized by integer s such that $1 \leq s \leq \min(n, m)$ and $v, c > 0$. The first s buyers have a value of v and the remaining buyers have a value of 0. Similarly, the first s sellers have a cost of c and the remaining sellers have a cost of $\hat{c}$.

Then for any fixed s we have $s^*(\vec{v}_{s,v}, \vec{c}_{s,c}) = s$ if $v \geq c$ and $s^*(\vec{v}_{s,v}, \vec{c}_{s,c}) = 0$ otherwise. For $v > c$ we get a contradiction to the claimed welfare guarantee if not all of the first s buyers and s sellers trade. Hence in this case exactly these buyers and sellers must win. Similarly, for $v < c$ we get a contradiction to the claimed welfare guarantee if any buyer-seller pair is accepted for trade. Hence in this case all players must lose. For ease of presentation we will assume that for $v = c$ all players with non-zero value/cost win.

Since the double auction mechanism is anonymous we know that winning buyers (sellers) with the same value (cost) must make (receive) identical payments. In particular, if all buyers (sellers) win and have the same value (cost) then all buyers (sellers) must make (receive) identical payments.

We claim that for any fixed s and all $v \geq c > 0$ the double auction mechanism must set the payments $p_B(\vec{v}_{s,v}, \vec{c}_{s,c})$ of the first s buyers and the payments $p_S(\vec{v}_{s,v}, \vec{c}_{s,c})$ to the first s sellers to $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) = c$ and $p_S(\vec{v}_{s,v}, \vec{c}_{s,c}) = v$. The arguments for the buyers and the sellers are symmetric, and so we only present the argument for the buyers.

We first show that the payments for buyers with values $v = c$ must be $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) = c$. If the payments are $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) > c$, we get a contradiction to WGSP, because the buyers currently have utility $v - p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) < 0$ and could jointly deviate to $v' < c$ which would make them lose and pay nothing for a utility of zero. If the payments are $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) < c$, then in an instance $(\vec{v}_{s,v'}, \vec{c}_{s,c})$ where the first s buyers have values v' and the first s sellers have costs c such that $c > v' > p_B(\vec{v}_{s,v}, \vec{c}_{s,c})$, the buyers could jointly deviate and report a value of c. Before the deviation they are losing and not paying anything for a utility of zero, after the deviation they are winning and paying $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) < v'$ which gives them a strictly positive utility.

Next we show that the payments for buyers with values $v > c$ must be $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) = c$. If the payments are $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) > c$, then these buyers could strictly gain by a group deviation to c. This would strictly improve their utility from $v - p_B(\vec{v}_{s,v}, \vec{c}_{s,c})$ to $v - c$, where we use that they pay exactly c if they report a value of c. If the payments are $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) < c$, then in an instance $(\vec{v}_{s,v'}, \vec{c}_{s,c})$ where the first s buyers have values $v' = c$ and the first s sellers have costs c, the buyers could strictly gain by a group deviation to v because this will lower their payment from c to $p_B(\vec{v}_{s,v}, \vec{c}_{s,c}) < c$, where we again use that for $v' = c$ each buyer has to pay c.

The statement of the theorem follows from this partial characterization of the payments by considering an input $(\vec{v}_{s,v}, \vec{c}_{s,c})$ from the restricted class of inputs described above with $s \geq 1$ and $v \geq c > 0$ and a group deviation of the s buyers with value v and the s sellers with cost c to v', c' such that $v' > v \geq c > c'$ because this—as we have just shown— will strictly reduce the payments of the buyers from c to c' and strictly increase the payments to the sellers from v to v'. □

From Corollary 5.3 we know that we can achieve this lower bound via the backward composition of uniformly 1-approximate, deferred-acceptance ranking algorithms with the lookback 0-threshold rule. We conclude that whenever the trade reduction mechanism can be implemented in this manner, it achieves optimal worst-case welfare subject to WGSP.

7.2. Lower Bound Subject to DSIC and BB

Next we show a lower bound that applies to all deterministic double auction mechanisms resulting from composition or not that are DSIC and BB.

Theorem 7.2. *Consider an unconstrained double auction setting. That is,* $\mathcal{I}_N = 2^N$ *and* $\mathcal{I}_M = 2^M$ *and therefore* $B \subseteq N$ *and* $S \subseteq M$ *are feasible whenever* $|B| \leq |S|$. *Let* $\hat{c} > \hat{v} > 0$. *Consider valuations* $v_i \in [0, \hat{v}]$ *for all* $i \in N$ *and costs* $c_j \in [0, \hat{c}]$ *for all* $j \in M$. *Recall that* $s^*(\vec{v}, \vec{c})$ *denotes the maximum number of trades in a welfare-maximizing solution for input* $(\vec{v}, \vec{c})$. *Let* $k = \min\{n, m\}$. *Then for no* $\epsilon > (k-1)/(2k^2-1)$ *there exists a deterministic double auction mechanism that is DSIC and BB and achieves a worst-case approximation guarantee of*

$$1 - \frac{1}{s^*(\vec{v}, \vec{c})} + \epsilon.$$

Note that $(k-1)/(2k^2-1) = o(1)$ meaning that $(k-1)/(2k^2-1) \to 0$ as $k \to \infty$. So asymptotically the theorem establishes a lower bound of $1 - 1/s^*(\vec{v}, \vec{c})$.

Proof. For contradiction, assume that there is a DSIC and BB double auction that achieves a strictly better worst-case approximation guarantee. Then there must be an $\epsilon > (k-1)/(2k^2-1)$ such that the welfare on any input $(\vec{v}, \vec{c})$ is at least

$$W(\vec{v}, \vec{c}) \geq \left(1 - \frac{1}{s^*(\vec{v}, \vec{c})} + \epsilon\right) \cdot OPT(\vec{v}, \vec{c}).$$

Fix some s such that $1 \leq s \leq k = \min\{n, m\}$ and consider an input $(\vec{v}_{s,y}, \vec{c}_{s,x})$ in which the first s buyers have a value of y and the first s sellers have a cost of x where $0 < x < y$, while all other buyers have a value of 0 and all other sellers have a cost of $\hat{c}$. For this input it is optimal to accept all buyers with value y and all sellers with cost x, so $s^*(\vec{v}_{s,y}, \vec{c}_{s,x}) = s$.

Consider a unilateral deviation by some buyer with value y to value y' such that $y \geq y' \geq x$. We claim that the buyer must remain winning as long as

$$y' > \frac{\left(1 - \frac{1}{s} - \epsilon(s-1)\right) y + \epsilon s x}{1 - \frac{1}{s} + \epsilon}. \tag{4}$$

To see this observe that in the altered input $(\vec{v}_{s,y,y'}, \vec{c}_{s,x})$ we still have $s^*(\vec{v}_{s,y,y'}, \vec{c}_{s,x}) = s$. Moreover, $\text{OPT}(\vec{v}_{s,y,y'}, \vec{c}_{s,x}) = (s-1)(y-x) + (y'-x)$. So if the buyer who deviated to y' would not win, then the welfare achieved by the double auction would be at most $(s-1)(y-x)$. But then, together with inequality (4), this would contradict the claimed approximation ratio on input $(\vec{v}_{s,y,y'}, \vec{c}_{s,x})$.

Now consider a unilateral deviation by some seller with cost x to cost x' such that $y \geq x' \geq x$. Then an analogous argument shows that the buyer must remain winning as long as

$$x' < \frac{\left(1 - \frac{1}{s} - \epsilon(s-1)\right) x + \epsilon s y}{1 - \frac{1}{s} + \epsilon}. \tag{5}$$

Together with the DSIC requirement these arguments show that the payments of the buyers with value y in the original instance are at most the RHS of inequality (4) and the payments to the sellers with cost x in the original instance are at least the RHS of inequality (5). We obtain a contradiction to BB if the former is smaller than the latter. This is the case for

$$\epsilon > \frac{s-1}{2s^2 - s},$$

which we assumed to be the case for $s = k$. □

From Corollary 5.3 we know that we can achieve this asymptotic lower bound via the backward composition of uniformly 1-approximate, deferred-acceptance ranking algorithms with the lookback 0-threshold rule. Hence whenever the trade reduction mechanism can be implemented in this manner, it is not only worst-case optimal subject to WGSP but also subject to DSIC and BB.

8. Conclusion and Discussion

Motivated by the complexity of double auction design, we proposed a modular approach to the design of double auctions that decomposes the design task into the tasks of designing greedy algorithms for either side of the market and a composition rule. Focusing on the unit-demand and unit-supply case, we proved a number of composition theorems for (approximate) efficiency, DSIC or WGSP, and BB, which relate the properties of the double auction to the properties of the modules used in its construction.

We instantiated our approach for three different feasibility structures—matroids, knapsacks and matchings. For matroids we showed that both the VCG mechanism and a natural analog of McAfee's trade reduction mechanism can be implemented via composition. For the other settings we obtained VCG- and trade reduction-style mechanisms with approximate-efficiency guarantees. We also identified a sense in which our guarantees are the best possible, subject to strong incentive or budget balance constraints.

The main future research direction arising from this work is how to extend our results to more general double auction settings. Below we list three ways in which real-world scenarios can be more complicated than our model. While our current analysis does not apply to such scenarios, recent developments in the study of double auctions have resulted in a more robust theory that begins to address some of these complications (see Section 1.4). Together with the tools of greediness and modularity developed in our paper, we believe this provides a promising foundation for the study of feasibility constraints in general double auction environments.

In particular, a first generalization of our model is to consider other desirable objectives of double auction design, such as pay-as-bid implementations or revenue guarantees. An interesting question here is to explore the trade-off between accepting fewer buyer-seller pairs for trade for a higher per-pair revenue, and accepting more buyer-seller pairs at a lower per-pair revenue.

A second generalization is to consider more complex valuations, as suggested in the conclusion to McAfee's work [30]: buyers who wish to buy multiple units, as can be found in several relevant double auction applications; buyers with non-private valuations; and buyers facing a choice between different kinds of goods. Similar extensions apply to seller preferences.

A third generalization is to retain the strategic and non-strategic properties considered here, but to consider more complex, cross-market feasibility constraints. A potential starting point could be a setting with single-minded buyers who each want to buy from a certain set of sellers, each of whom produces a single unit of a unique good.

Acknowledgments

We greatly benefited from discussion with Moshe Babaioff, Gabriel Carroll, Fuhito Kojima, Paul Milgrom and Ilya Segal, as well as from the helpful comments of the anonymous referees. This research is supported in part by NSF Awards CCF-1016885 and CCF-1215965, an ONR PECASE Award, an SNF Postdoctoral Fellowship, and the Hsieh Family Stanford Interdisciplinary Graduate Fellowship.

References

[1] G. Aggarwal and J. D. Hartline. Knapsack auctions. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[2] M. Babaioff and N. Nisan. Concurrent auctions across the supply chain. *Journal of Artififical Intelligence Research*, 21:595–629, 2004.

[3] M. Babaioff, N. Nisan, and E. Pavlov. Mechanisms for a spatially distributed market. *Games and Economic Behavior*, 66(2):660–684, 2009.

[4] C. Y. Baldwin and K. B. Clark. *Design Rules: The Power of Modularity*. MIT Press, 2000.

[5] S. Bikhchandani, S. de Vries, J. Schummer, and R. V. Vohra. An ascending Vickrey auction for selling bases of a matroid. *Operations Research*, 59(2):400–413, 2011.

[6] L. Blumrosen and S. Dobzinski. Reallocation mechanisms. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 2014.

[7] A. Borodin and B. Lucier. Price of anarchy for greedy auctions. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 537–553, 2010.

[8] J. Bredin and D. C. Parkes. Models for truthful online double auctions. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*, pages 50–59, 2005.

[9] G. Carroll. The efficiency-incentive tradeoff in double auction environments. Working paper, 2013.

[10] L. Y. Chu. Truthful bundle/multiunit double auction. *Management Science*, 55(7):1184–1198, 2009.

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

[12] M. W. Cripps and J. M. Swinkels. Efficiency of large double auctions. *Econometrica*, 74(1):47–92, 2006.

[13] K. Deshmukh, A. V. Goldberg, J. D. Hartline, and A. R. Karlin. Truthful and competitive double auctions. In *Proceedings of the 11th Annual European Symposium on Algorithms*, pages 361–373, 2002.

[14] D. E. Drake and S. Hougardy. A simple approximation algorithm for the weighted matching problem. *Information Processing Letters*, 85(4): 211–213, 2003.

[15] P. Dütting, V. Gkatzelis, and T. Roughgarden. The performance of deferred-acceptance auctions. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 2014.

[16] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[18] D. Fudenberg, M. Mobius, and A. Szeidl. Existence of equilibrium in large double auctions. *Journal of Economic Theory*, 133(1):550–567, 2007.

[19] D. Gale and L. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:915, 1962.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, CA, USA, 1979.

[21] M. Gonen, R. Gonen, and E. Pavlov. Generalized trade reduction mechanisms. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 20–29, 2007.

[22] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[23] J. W. Hatfield and P. R. Milgrom. Matching with contracts. *American Economic Review*, 95(4):913–935, 2005.

[24] L. Hurwicz. On informationally decentralized systems. In C. B. McGuire and R. Radner, editors, *Decision and Organization*, pages 297–336. University of Minnesota Press, Minneapolis, MN, USA, 1972.

[25] R. Juarez. Group strategyproof cost sharing: The role of indifferences. *Games and Economic Behavior*, 82:218–239, 2013.

[26] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[27] F. Kojima and T. Yamashita. Double auction with interdependent values: Incentives and efficiency. Working paper, 2013.

[28] K. Leyton-Brown. The viability of exact feasibility checking. Presentation, 2013.

[29] K. Leyton-Brown. Feasibility checking for spectrum repacking: Methodologies and test results. Presentation, 2014.

[30] R. P. McAfee. A dominant strategy double auction. *Journal of Economic Theory*, 56(2):434–450, 1992.

[31] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. *Games and Economic Behavior*, 67(1):125–155, 2009.

[32] P. Milgrom and I. Segal. Deferred-acceptance auctions and radio spectrum reallocation. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 2014. Online version available at `http://www.stanford.edu/~isegal/heuristic.pdf`, accessed June 2015.

[33] A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. *Games and Economic Behavior*, 64: 612–631, 2008.

[34] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29:265–281, 1983.

[35] T.-D. Nguyen and T. Sandholm. Optimizing prices in descending clock auctions. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 2014.

[36] J. G. Oxley. *Matroid Theory*. Oxford University Press, 1992.

[37] R. Preis. Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science*, pages 259–269, 1999.

[38] M. Pycia. Stability and preference alignment in matching and coalition formation. *Econometrica*, 80(1):323–362, 2012.

[39] G. C. Robinson and D. J. A. Welsh. The computational complexity of matroid properties. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87(1):29–45, 1980.

[40] T. Roughgarden and M. Sundararajan. Quantifying inefficiency in cost-sharing mechanisms. *Journal of the ACM*, 56(4):article 23, 2009.

[41] A. Rustichini, M. A. Satterthwaite, and S. R. Williams. Convergence to efficiency in a simple market with incomplete information. *Econometrica*, 62(5):1041–63, 1994.

[42] M. A. Satterthwaite and S. R. Williams. The rate of convergence to efficiency in the buyer's bid double auction as the market becomes large. *Review of Economic Studies*, 56(4):477–98, 1989.

[43] M. A. Satterthwaite and S. R. Williams. The optimality of a simple market mechanism. *Econometrica*, 70(5):1841–1863, 2002.

[44] A. Schrijver. *Combinatorial Optimization*. Springer, 2002.

[45] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[46] R. V. Vohra. *Mechanism Design: A Linear Programming Approach*. Cambridge University Press, 2011.

A. Pseudocode for Forward and Backward Composition

In this appendix, Algorithms 2 and 3 describe the allocation rules for forward and backward composition, respectively, as defined in Section 3.2.

ALGORITHM 2: The Allocation Rule of Forward Composition

Input: Value profile $\vec{v}$ and cost profile $\vec{c}$
Access to two forward-feasible ranking algorithms, one for buyers and one for sellers, and to a composition rule
Initialize $A = \emptyset$ % Accepted players

% Repeatedly query the ranking algorithms
repeat
  query both ranking algorithms on inputs $\vec{v}, \vec{c}$, respectively, to compose the next buyer-seller pair i, j % Step 1
  if one stream or both run out of players
    stop algorithm % All players not currently in A are rejected

  run the composition rule on i, j, their value and cost, and their query histories % Step 2
  if i, j accepted
    $A = A \cup \{i, j\}$ % Add i, j to the set of accepted players
    iterate % Go to Step 1 and query the next players in the stream
  else
    stop algorithm % The algorithm stops as soon as it rejects the first pair of players; all players not currently in A are rejected
end % repeat

B. Ranking Algorithms for Matroids

In this appendix we present additional details for the greedy by weight algorithm for matroids. We first show that the greedy algorithm is a uniform 1-approximation; afterwards we show how to implement it within the deferred-acceptance framework. This means that the ranking algorithms described for completeness (for buyers) in Algorithms

ALGORITHM 3: The Allocation Rule of Backward Composition

Input: Value profile $\vec{v}$ and cost profile $\vec{c}$
Access to two backward-feasible ranking algorithms, one for buyers and one for sellers, and to a composition rule
Initialize $R_B = \emptyset; R_S = \emptyset$ % Rejected buyers and sellers

% Preprocessing—reject players until the remaining buyer and seller sets are both feasible and have equal size % Step 0
for $k = 1$ to $n - \min\{n - \ell_B, m - \ell_S\}$ % ℓ_B, ℓ_S are the ranks of the largest-rank buyer and seller to discard for feasibility
  query the ranking algorithm for buyers on input $\vec{v}$ for the next buyer i
  $R_B = R_B \cup \{i\}$; % Reject buyer i
end % for
for $k = 1$ to $m - \min\{n - \ell_B, m - \ell_S\}$ % ℓ_B, ℓ_S are the ranks of the largest-rank buyer and seller to discard for feasibility
  query the ranking algorithm for sellers on input $\vec{c}$ for the next seller j
  $R_S = R_S \cup \{j\}$; % Reject seller j
end % for

% Repeatedly query the ranking algorithms
repeat
  query both ranking algorithms on inputs $\vec{v}, \vec{c}$, respectively, to compose the next buyer-seller pair (i, j) % Step 1
  run the composition rule on (i, j), their value and cost, and their query histories *excluding* preprocessing queries % Step 2
  if (i, j) rejected
    $R_B = R_B \cup \{i\}; R_S = R_S \cup \{j\}$ % Reject buyer i and seller j
    iterate % Go to Step 1 and query the next players in the streams
  else
    stop algorithm % The algorithm stops as soon as it accepts the first pair of players; all players in $R_B \cup R_S$ are rejected
end % repeat

4 and 5 are uniform 1-approximations, and that Algorithm 5 can be implemented as a deferred-acceptance ranking algorithm.[15]

ALGORITHM 4: Forward-feasible Ranking Algorithm for Matroids (Presented for Buyers)

Input: Value profile $\vec{v}$, without loss of generality assumed to be sorted $v_1 \geq \cdots \geq v_n$
Access to a matroid feasibility set system $(N, \mathcal{I}_N)$ (represented in a computationally tractable way)
Initialize $S = ()$ % Output stream

for $i = 1$ to n % Go over buyers from high to low quality
  if $S \cup i \in \mathcal{I}_N$ % If adding buyer i to the stream preserves its independence
    append i to S
end % for

Proposition B.1. *The ranking algorithms for matroids based on the greedy one-sided algorithm are uniform* 1-*approximations.*

Proof. That the greedy algorithm finds a maximum weight basis of any matroid is a well known fact [16]. The claim of uniform 1-approximation follows from the fact that if we restrict the independent sets to sets of size at most k, the matroid structure is preserved [e.g., 44]. □

Proposition B.2. *The backward-feasible ranking algorithm for matroids based on the greedy by weight one-sided algorithm can be implemented as a deferred-acceptance algorithm.*

[15]For Algorithm 5 we remark that representation of the input by a feasibility oracle is sufficient to check whether a given element forms a circuit with a given set in a computationally tractable way, e.g., by checking whether the element is in the set's closure [39].

ALGORITHM 5: Backward-feasible Ranking Algorithm for Matroids (Presented for Buyers)

```
Input: Value profile v⃗, without loss of generality assumed to be sorted v_1 ≥ ··· ≥ v_n
Access to a matroid feasibility set system (N, I_N) (represented in a computationally tractable way)
Initialize S = ()    % Output stream
Initialize ℓ = 0    % Rank of largest-ranked buyer to discard for feasibility

% First loop
for i = n down to 1    % Go over buyers from low to high quality
  if i forms a circuit with buyers from N \ (S ∪ {i})    % Consider the buyers not in the stream
    append i to S    % i forms a circuit (minimal dependent set) with buyers not in the stream
    ℓ = ℓ + 1
end   % for

% Second loop
for i = n down to 1    % Go over buyers from low to high value
  if i ∉ S    % Append every remaining buyer not yet in the stream
    append i to S
end   % for
```

Proof. The greedy algorithm repeatedly accepts the element with highest weight that does not violate the feasibility constraint. This can be turned around by rejecting the element with lowest weight that forms a minimal dependent set (circuit) with the yet-unrejected elements [see also 5, 32]. Forming a circuit is a structural property of the unrejected elements, i.e., no knowledge of their weights is required. The reverse greedy algorithm can therefore be implemented as a deferred-acceptance algorithm. The implementation for sellers uses the following scores for active players in A:

$$s_i^A(b_i, b_{-A}) = \begin{cases} b_i & \text{if } i \text{ forms a minimal dependent set with a subset of } A \setminus \{i\}, \\ \infty & \text{otherwise.} \end{cases} \qquad \square$$

C. Ranking Algorithms for Knapsacks

In this appendix we present additional details for the greedy by weight algorithm for knapsacks. We first show that the greedy algorithm is a uniform $1/(1-\lambda)\mu$-approximation, where $\lambda \le 1$ is the ratio of the largest element's size to the knapsack size, $\mu \le 1$ is the ratio of the smallest element's size to the largest element's size and it is assumed that $1/\mu$ is integral. Afterwards we show how to implement the greedy algorithm within the deferred-acceptance framework. It follows that there exist ranking algorithms that are uniform $1/(1-\lambda)\mu$-approximations, in paricular a backward-feasible one that can be implemented as a deferred-acceptance ranking algorithm.

Proposition C.1. *The ranking algorithms for knapsacks based on the greedy by weight one-sided algorithm are uniform* $\frac{1}{(1-\lambda)\mu}$*-approximations.*

Proof. The proof is by induction on the maximum allowed number k of elements in the knapsack. For every $k \ge 0$ and for every knapsack instance Q, denote by A_k^Q the solution for Q of size at most k found by the greedy algorithm, and by O_k^Q the optimal solution for Q of size at most k. Fix $k > 0$. Induction hypothesis: for every instance Q it holds that $w(A_{k-1}^Q)/(1-\lambda)\mu \ge w(O_{k-1}^Q)$, where $w(A_{k-1}^Q)$ and $w(O_{k-1}^Q)$ denote the total weight of solutions A_{k-1}^Q and O_{k-1}^Q. The hypothesis is easy to verify for $k = 1$.

Consider a knapsack instance Q' in which all elements fit into the knapsack. Without loss of generality we assume that sizes are normalized such that the size of the knapsack is 1, and that the elements are ordered from high to low weight. Thus the first step of the greedy algorithm is to place element 1 into the knapsack. We define a residual instance Q where the allowed number of elements is $k-1$, and the size of the knapsack decreases by the size of element 1. In addition, the elements whose sizes are larger than the residual knapsack are removed from the residual element set. By the greediness of the algorithm, we know that $w(A_k^{Q'}) = w(A_{k-1}^Q) + w_1$.

Assume that the optimal solution to Q' with up to k elements does not include element 1, by how much is $w(O_{k-1}^Q)$ decreased relative to $w(O_k^{Q'})$ due to placing element 1 in the knapsack? There are two sources of loss. First, element

1 takes up place in the knapsack according to its size, and thus excludes elements from the optimal solution. Second, there may be elements removed from the residual element set since they no longer fit into the knapsack, so that the element sets available to $O_k^{Q'}$ and to O_{k-1}^{Q} may differ.

We start from the second source of loss. Observe that the minimum size of an element removed from the residual element set is $1 - \lambda$, and the maximum weight of such an element is w_1. There are at most $1/(1 - \lambda)$ such elements in the optimal solution to Q'. If their aggregate size is at least the size of element 1, removing them from the optimal solution also makes room for element 1, and so in this case

$$w(O_{k-1}^{Q}) \geq w(O_k^{Q'}) - w_1 \frac{1}{1-\lambda} \geq w(O_k^{Q'}) - w_1 \frac{1}{(1-\lambda)\mu}.$$

On the other hand, if the aggregate size of elements to remove from the residual set is less than the size of element 1, then there are at most $\lambda/(1 - \lambda)$ such elements (using that by normalization, the size of element 1 is at most λ). In addition, there is the first source of loss, from excluding a maximum number of $1/\mu$ additional elements from the optimal solution to make room for element 1 (we use here the assumption that $1/\mu$ is integral). Thus, in this case,

$$w(O_{k-1}^{Q}) \geq w(O_k^{Q'}) - w_1 \left(\frac{1}{\mu} + \frac{\lambda}{1-\lambda} \right) \geq w(O_k^{Q'}) - w_1 \frac{1}{(1-\lambda)\mu},$$

where in the second inequality we used that $\lambda \geq \lambda\mu$.

Putting everything together and using the induction assumption we get that

$$w(O_k^{Q'}) \leq w(O_{k-1}^{Q}) + w_1 \frac{1}{(1-\lambda)\mu} \leq w(A_{k-1}^{Q}) \frac{1}{(1-\lambda)\mu} + w_1 \frac{1}{(1-\lambda)\mu} \leq w(A_k^{Q'}) \frac{1}{(1-\lambda)\mu},$$

thus verifying the hypothesis for k and completing the proof. □

Proposition C.2. *The backward-feasible ranking algorithm for knapsacks based on the greedy by weight one-sided algorithm can be implemented as a deferred-acceptance algorithm.*

Proof. The reverse greedy algorithm repeatedly rejects the element with lowest weight until the unrejected elements fit into the knapsack. The sizes are a structural property and so it can be implemented as a deferred-acceptance algorithm. The implementation for sellers uses the following scores for active players in A:

$$s_i^A(b_i, b_{-A}) = \begin{cases} b_i & \text{if the total size of active players exceeds the size of the knapsack,} \\ \infty & \text{otherwise.} \end{cases}$$ □

D. Ranking Algorithms for Matchings

In this appendix we present the novel backward-feasible algorithm for matchings. We describe the algorithm and analyze its approximation guarantee in Section D.1. Afterwards, in Section D.2, we show how to implement it within the deferred-acceptance framework.

D.1. Backward-Feasible Ranking Algorithm

We describe a backward greedy algorithm that is based on an idea of [37]. Our description (cf. Algorithm 6) follows that of [14]. Unlike the algorithm of Drake and Hougardy our algorithm is randomized.

Our algorithm starts with an arbitrary node and then grows a path of locally heaviest edges. If such a path cannot be extended any further, it restarts this process at an arbitrary node. In the end—with probability 1/2—it takes all even edges along the paths. Otherwise, it takes all odd edges. Once the remaining edges are feasible, we can continue to output edges in reverse order of their weight for the sake of consistency. This gives rise to a backward-feasible ranking algorithm.

Proposition D.1. *The above backward-feasible ranking algorithm based on the path growing algorithm is consistent, rank monotone, and a uniform* 2*-approximation.*

ALGORITHM 6: Path Growing Algorithm

Input: Graph $G = (V, E)$, weights $w(e) \geq 0$ for all edges $e \in E$
Output: Matching M
Set $M_1 = \emptyset$, $M_2 = \emptyset$, $i = 1$;
while $E \neq \emptyset$ **do**
    Choose $x \in V$ of degree at least 1 arbitrarily;
    while *x has a neighbor* **do**
        Let (x, y) be the heaviest edge incident to x;
        Add (x, y) to M_i;
        Set $i = 3 - i$;
        Remove x from G;
        Set $x = y$;
    end
end
Output M_1 with probability $1/2$, otherwise output M_2;

Proof. The algorithm is consistent because it outputs the elements of the set M_i that has been picked in order of their weights.

It is rank monotone because by increasing its bid a player can only enter and not drop out of either M_1 or M_2.

For the approximation guarantee we assign each edge to some node in the graph in the following way. Whenever a node is removed, all edges that are currently incident to that node are assigned to it. To prove the factor 2, we consider an optimal solution of cardinality k. Each of these edges is assigned to a node. If we consider the edges adjacent to these nodes that were added to $M_1 \cup M_2$, then from the fact that we picked the locally heaviest edges we know that their total weight is at least the weight of the optimal edges. The claim now follows from the fact that we pick each of these edges (or a better one) with probability $1/2$. □

D.2. Implementation as Deferred-Acceptance Algorithm

The randomization can be implemented by tossing a fair coin at the beginning of the algorithm, and by choosing M_1 if the coin shows heads and by choosing M_2 if the coin shows tails. The path growing part of the algorithm can be implemented as described in [15]. Once the set of active edges becomes feasible, we can continue to score by weight in order to maintain consistency.

E. Impossibility Result for Forward Composition

In this appendix we present an impossibility result, which shows that DSIC double auction mechanisms that are the result of forward composition are particularly ill-equipped to achieve either WGSP or budget balance while maintaining a non-trivial efficiency guarantee.

Proposition E.1. *Consider a double auction setting with $n = m = 2$ and no feasibility constraints. For every forward composition of consistent ranking algorithms that is DSIC, there exist value and cost profiles for which either the budget deficit is arbitrarily high and the mechanism is not WGSP, or the welfare is arbitrarily small with respect to* OPT.

Proof. Let H be an arbitrarily large constant. We show there exist value and cost profiles such that either the budget deficit is at least $H/8$ and the mechanism is not WGSP, or the welfare is at most an $8/H$-fraction of the welfare achievable by the trade reduction double auction.

We define the following value and cost profiles:

$$\begin{array}{llll} \vec{v}^{\,1} = (H, H) & \vec{v}^{\,2} = (\frac{H}{4}, \epsilon) & \vec{v}^{\,3} = (H, \epsilon) & \vec{v}^{\,4} = (\frac{3H}{8}, \epsilon) \\ \vec{c}^{\,1} = (\frac{3H}{4}, H - \epsilon) & \vec{c}^{\,2} = (0, 0) & \vec{c}^{\,3} = (\frac{5H}{8}, H - \epsilon) & \vec{c}^{\,4} = (0, H - \epsilon). \end{array}$$

Observe that for profile pairs $(\vec{v}^{\,1}, \vec{c}^{\,1}), (\vec{v}^{\,2}, \vec{c}^{\,2})$, the trade reduction double auction with greedy ranking algorithms achieves welfare of at least $H/4$, and for the profile pair $(\vec{v}^{\,4}, \vec{c}^{\,3})$ its welfare is zero. It is also both BB and WGSP for all profile pairs.

We first show that for all the above value (cost) profiles, we can assume that the ranking algorithms rank first the buyer (seller) with higher value (lower cost). This holds trivially for $\vec{v}^{\,1}$ and $\vec{c}^{\,2}$ given that the ranking algorithms have non-empty outputs (otherwise, the welfare is 0 and we are done). The other profiles are all of the form $\vec{v} = (v_h, v_\ell)$ where $v_h - v_\ell \geq H/4 - \epsilon$ and $v_\ell \geq \epsilon$; and $\vec{c} = (c_\ell, c_h)$ where $c_h - c_\ell \geq H/4 - \epsilon$ and $c_h \leq H - \epsilon$. If the buyer ranking algorithm given $\vec{v}$ does not rank first the higher buyer, by consistency it does not rank this buyer at all, and so the welfare can be arbitrarily small with respect to the welfare of the trade reduction double auction (e.g., when $\vec{v}$ is paired with cost profile $(v_\ell - \epsilon, v_\ell - \epsilon)$). The argument for the seller ranking algorithm is similar (e.g., when $\vec{c}$ is paired with value profile $(c_h + \epsilon, c_h + \epsilon)$).

Now consider profile pairs $(\vec{v}^{\,1}, \vec{c}^{\,1})$ and $(\vec{v}^{\,2}, \vec{c}^{\,2})$. Let (v, c) be the value and cost of the first buyer-seller pair that the composition rule considers; observe in both cases it has either $v = H$ or $c = 0$. By DSIC, the composition rule for the first buyer-seller pair is equivalent to setting a threshold $t_B = t_B(c)$ on the buyer's value, and a threshold $t_S = t_S(v)$ on the seller's cost. What are the possible thresholds $t_B(0), t_S(H)$? If either $t_B(0) > H/4$ or $t_S(H) < 3H/4$ then the first buyer-seller pair is rejected, and the maximum welfare from the second buyer-seller pair is ϵ, completing the proof. It is left to reason about the case in which $t_B(0) \leq H/4$ and $t_S(H) \geq 3H/4$. We now show that if this is the case then there is a large budget deficit for the profile pair $(\vec{v}^{\,3}, \vec{c}^{\,2})$, and in addition the WGSP property is violated.

Given $(\vec{v}^{\,3}, \vec{c}^{\,2})$, the first buyer-seller pair that the composition rule considers has value and cost $(H, 0)$, which clears both thresholds and is accepted for trade. Threshold payments imply a deficit of $\geq H/2$, and since the most that the second buyer-seller pair can contribute to covering this deficit is ϵ, the total budget deficit is $\gg H/8$ for small enough ϵ.

We conclude by showing a violation of WGSP. Consider the profile pair $(\vec{v}^{\,4}, \vec{c}^{\,3})$; the first buyer-seller pair has value and cost $(3H/8, 5H/8)$. If it is accepted then the welfare is negative and the proof is complete. Otherwise, consider a group deviation to the profile pair $(\vec{v}^{\,3}, \vec{c}^{\,4})$. The first buyer-seller pair then has reported value and cost $(H, 0)$ and is accepted with payments $t_B(0) \leq 2H/8, t_S(H) \geq 6H/8$. This deviation is strictly preferable to both players in the deviating pair, completing the proof. $\square$