On Characterizations of Truthful Mechanisms for Combinatorial Auctions and Scheduling

Shahar Dobzinski
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem, Israel
shahard@cs.huji.ac.il

Mukund Sundararajan
Department of Computer Science
Stanford University
470 Gates Building, 353 Serra Mall, Stanford, CA 94305.
mukunds@cs.stanford.edu

ABSTRACT

We characterize truthful mechanisms in two multi-parameter domains. The first characterization shows that every mechanism for combinatorial auctions with two subadditive bidders that always allocates all items is an affine maximizer. The second result shows that every truthful machine scheduling mechanism for 2 unrelated machines that yields a finite approximation of the minimum makespan, must be task independent. That is, the mechanism must determine the allocation of each job separately.

The characterizations improve our understanding of these multi-parameter settings and have new implications regarding the approximability of central problems in algorithmic mechanism design.

Categories and Subject Descriptors

F.2.8 [**Analysis of Algorithms and Problem complexity**]: Miscellaneous

General Terms

Theory

Keywords

Characterizations, Combinatorial Auctions, Scheduling, Incentive Compatibility

1. INTRODUCTION

1.1 Background

Mechanism design discusses the design of protocols that achieve specific outcomes even when players are self-interested and have private information. The central positive result in mechanism design is the widely applicable VCG mechanism. However, VCG mechanisms do have significant limitations. Though they can be used to maximize the welfare of the players (or minor variants of this objective), often we are interested in other objectives; for instance, in scheduling domains we may be interested in minimizing the makespan. Furthermore, even in settings in which welfare maximization is our desired goal, implementing VCG may not be computationally feasible. The obvious question is: Are there other types of truthful mechanisms that do not suffer from these limitations?

1.2 Implementability in Multi-Parameter Domains

Consider a setting with n selfish players. We also have a set of alternatives (outcomes) $\mathcal{A}$. Each player has a different value for each alternative. In other words, each player i has some valuation function $v_i : \mathcal{A} \to \mathbb{R}$ (this information is private to the player i), that gives a value for each possible alternative. These valuations may also have additional structure, for instance in combinatorial auctions players never prefer an allocation in which they receive some bundle S over an allocation in which they get some superset of S. One of the central questions in mechanism design is whether a given social choice function $f : \Pi_{i=1}^{n} V_i \to \mathcal{A}$ can be implemented truthfully; that is can we design a payment rule that will incentivize selfish players to always report their true values.

For single-parameter domains, when players values are described by a single number, the characterization of Myerson [15] tells us that a social choice function is implementable if and only if it is monotone; for a monotone social choice function, a player's winnings can only increase when his bid increases (for a fixed set of bids of the other players).

Unfortunately, much less is understood about exactly which social choice functions are implementable in multi-parameter domains, where a player's private information is no longer describable by a single real number. The main characterization result in such domains is Roberts' Theorem. Roughly, it says that if the v_i's can be arbitrary with no structure (unrestricted valuations), only a very small subset of functions are implementable. These functions are called affine maximizers. An affine maximizer has the following form: $\arg\max_{a\in\mathcal{A}} w_i v_i(a) + c_a$. Notice that the c_a's and the w_i's are predetermined constants and do not depend on the valuations of the players. Interestingly, this is precisely the set of functions that can be implemented by (weighted) VCG.

Unfortunately, unrestricted valuations are not applicable in most settings (for instance in combinatorial auctions bidders have no externalities and only care about the items they

EC'08, July 8–12, 2008, Chicago, Illinois, USA.

get) and Roberts' theorem does not apply; Indeed, several non VCG mechanisms are known [2, 8]. Further, VCG is known to have severe limitations. We discuss these limitations for two settings that we consider in this paper:

- **Combinatorial Auctions:** In a combinatorial auction n bidders are competing on a set M, $|M| = m$, of heterogeneous items. Each bidder i has a valuation function v_i assigns a non-negative value to every possible bundle of items. The valuations are monotone and normalized ($v_i(\emptyset) = 0$). The goal is to find an allocation $(S_1, \ldots, S_n)$ that maximizes the welfare: $\Sigma_i v_i(S_i)$. Using VCG to achieve an approximation ratio better than $O(\sqrt{m})$ requires exponential communication, thus the VCG mechanism is not computationally feasible. Can a truthful, polynomial time mechanism achieve this ratio[1]?

- **Scheduling Unrelated Machines:** Here we have n machines and m jobs. The time it takes for machine i to process job j is $v_i(\{j\})$, which is also the processing cost. The goal is to minimize the makespan, the time it takes for all the jobs to be processed (the completion time of the last machine). The VCG mechanisms provides an easy upper bound of n for this problem. Nisan and Ronen [17] prove a lower bound of 2, for all truthful mechanisms, even for computationally unbounded ones. Recently, [4, 3] improves this ratio to to 2.61.

1.2.1 The Role of Characterizations

Characterizations identify precisely which social choice functions can be implemented truthfully; they often yield intuitive goals for a mechanism designer to achieve. For example, in a single-parameter setting checking for monotonicity is often simpler than proving truthfulness from scratch. Characterizations also help us prove lower bounds on approximation ratios of truthful mechanisms. For instance, in combinatorial auctions we want to understand the best approximation ratio that can be achieved by polynomial-time mechanisms; [13, 6] suggest we first prove that all mechanisms that provide a good approximation ratio must use the VCG payment scheme (i.e., the algorithms are *maximal-in-range*. See [16, 6]), and then to prove that this class of algorithms cannot provide a good approximation to the welfare in polynomial time.

Some partial success of proving lower bounds using characterizations was already achieved. The most notable example is in the setting of multi-unit auctions (see [7] for a definition). In this setting Lavi, Mu'alem, and Nisan [13] prove that every $(2-\epsilon)$-approximation mechanism for 2 players that *always allocates all items* must be an affine maximizer; [7] proves that affine maximizers cannot provide a better than 2 approximation in polynomial time. On the other hand, ignoring incentives issues, an FPTAS exists.

1.3 Our Results

Combinatorial auctions with subadditive and XOS bidders have received substantial amount of attention, both from an algorithmic point of view and a mechanism design standpoint (e.g., [9, 10, 6, 5]). A valuation v is called subadditive if for each two bundles S and T we have that $v(S) + v(T) \geq V(S \cup T)$. The definition of XOS valuations (every XOS valuation is also subadditive) is more involved, and we postpone it to the appendix. Our first result characterizes the class of truthful mechanisms for combinatorial auctions with subadditive bidders:

Theorem: Let f be a deterministic mechanism for combinatorial auctions for 2 subadditive (or XOS) bidders with range of size at least $m + 2$, that always allocates all items. Then, f is an affine maximizer.

It is known that the range of any approximation algorithm that provides an approximation ratio better than 2 is larger than $m+2$ [6]. We can thus use the above theorem and [6] to conclude that any mechanism for 2 subadditive bidders that always allocates all items and achieves an approximation ratio better than 2 must run in exponential time. We prove similar results for the class of XOS bidders. Hence, we get another separation between the power of polynomial time algorithms and the power of truthful polynomial time approximation mechanisms: ignoring incentives, a $\frac{4}{3}$-approximation algorithm for 2 XOS bidders exists (for n players the approximation ratio of the algorithm approaches $\frac{e}{e-1} \approx 1.58$).

Unlike LMN [13], who prove a similar (but not identical) result for multi-unit auctions using new machinery, we extend and modify Roberts' original proof [18, 12]. The LMN approach relies on the assumption that if a bidder bids high enough then he receives all the items. This property is not necessarily true for approximation algorithms for combinatorial auctions with subadditive bidders; It looks that the LMN approach does not suffice to prove our result. Furthermore, we have found Roberts' approach easier to understand and work with than the LMN approach.

We then switch gears to scheduling unrelated machines. All previously known truthful mechanisms for machine scheduling that yield a finite approximation to the minimum makespan objective are task-independent; i.e they decide the allocation for each job separately. Our second main result claims that this is all we can do, at least for 2 machines:

Theorem: Let f be a mechanism for the scheduling problem for 2 machines that provides a finite approximation ratio. Then, f is task independent.

We believe that the characterization is of interest from a technical point of view: we are not aware of any previous characterization for multi-parameter domain that results in non affine maximizers. Our characterization also holds for randomized mechanisms that are truthful in the universal sense; if such a mechanism provides a finite approximation ratio, then every mechanism in its support must be task-independent. Thus, we can optimize over this class of randomized mechanisms in order to determine the correct approximation ratio.

Is there any hope of extending our characterization to more than 2 machines? Unfortunately, we show that there exists a non-local mechanism for $n > 2$ machines that provides a finite approximation ratio (slightly worse than n). Our 2-machine characterization may, however, serve as a starting point for characterizations in settings with 3 machines or more.

Finally, in Section 5, we find that our characterizations carry over to settings with more than two players under an

[1]We limit this discussion to deterministic algorithms. See [5, 8, 14] for randomized ones.

additional condition on the social choice function called *stability*; Informally, stability says that if we change the value of one player and its allocation remains the same, then all the other players' allocations must also remain the same. The condition clarifies the role of the "two players, all items are always allocated" assumption in essentially all successful characterizations, including [13] and ours. Our results highlight the need for mechanism design techniques that explicitly *break* this condition.

1.3.1 Open Questions

One open question is to characterize combinatorial auctions without the assumption that all items are always allocated, and to characterize truthful mechanisms for scheduling for more than 2 machines. The assumption that all items are always allocated looks technical from an algorithmic standpoint; without loss of generality every approximation algorithm always allocates all items. However, when incentives issues are taken into consideration, this assumption implies the implicit existence of externalities: if one bidder does not get an item then the other does. Obtaining characterizations without this kind of assumptions seems to be a major obstacle.

Organization of the Paper

After a short preliminaries section, in Section 3 we prove our characterization of combinatorial auctions with subadditive (and XOS) bidders. In Section 4 we characterize scheduling domains. Finally, Section 5 discusses extensions of our work to settings with more than two players.

2. PRELIMINARIES

As we focus on settings with two bidders (machines or bidders), it is convenient to let the letters v and u denote the two bidders valuations. The notation $f(v,u) = (S, S')$, says that for the valuation v, u, bidder (machine) 1 is allocated the bundle S, and bidder (machine) 2 is allocated the bundle S'.

Let v, v' and u be valuations. A mechanism f is *weakly monotone* if for all valuations v, v' and u, if $f(v,u) = a$, and $f(v', u) = b$, then $v(a) - v(b) \leq v'(a) - v'(b)$. A mechanism is called *strongly monotone* if for all valuations it holds that $v(a) - v(b) < v'(a) - v'(b)$, if $a \neq b$. In a domains when a player is interested in minimizing his cost rather than maximizing his value (like scheduling) the direction of the inequalities is reversed.

It is well known that truthful mechanisms must guarantee the following for any fixed player. For a fixed allocation, the payment of a player does not depend on the player's own valuation (though it may depend on other players' valuations). Further, the selected allocation must maximize the player's utility (as a function of the payments and the player's valuation). For instance, the payment of player 1 does not depend on its valuation v, but only on the bundle assigned to it and the other player's valuation, u; for every bundle T and every u there exist payments $p^1_T(u)$ such that if the mechanism allocates the bundle S to player 1 then it pays $p^1_S(u)$. Moreover if the mechanism allocates S to player 1, then $S \in \arg\max_T (v(T) - p^1_T(u))$.

3. COMBINATORIAL AUCTIONS WITH SUBADDITIVE BIDDERS

3.1 The Characterization

THEOREM 3.1. *Let f be a truthful combinatorial auction for 2 bidders with subadditive valuations, which always allocates all items. If the range of f is of size at least $m+2$ then f must be an affine maximizer. The same characterization holds if the bidders have XOS valuations instead.*

We note that if f provides an approximation ratio better than 2 then f must select from at least $m + 2$ distinct outcomes (in fact exponential in m) [6]. In the appendix we deal with the necessary changes to the prove the result for XOS valuations.

In this extended abstract we prove the theorem only for the case where f is scalable; as a consequence, we essentially prove that f is a weighted welfare maximizer (i.e., the constants c_a in the definition of affine maximizers are 0). A social choice function f is called *scalable* if for every $\alpha > 0$, and valuations v, u we have that $f(v, u) = f(\alpha \cdot v, \alpha \cdot u)$. Adding this assumption does not change the basic logic of the proof, but rather allows us to simplify some arguments. We defer the proof without this assumption to the full version of the paper. To simplify our proof even more, we assume that the domain is open in the sense defined in the appendix (such a domain suffices for proving lower bounds). In the appendix we show that for such domains, we can assume that f satisfies strong monotonicity without loss of generality.

The reader is encouraged to read the examples in Subsection 3.2 in parallel to the proof. We discuss there how removing various conditions breaks our characterizations. These examples also clarify the steps of the proof.

Basic Definitions and Proof Structure

DEFINITION 3.2. *Let $(S, M \setminus S)$ and $(T, M \setminus T)$ be two allocations in the range of f. $\alpha = (\alpha_1, \alpha_2) \in \mathbb{R}^2$ is in $P(S,T)$ if there exist two valuations v and u such that $f(v,u) = (S, M \setminus S)$, and $\alpha_1 = v(S) - v(T)$ and $\alpha_2 = u(M \setminus S) - u(M \setminus T)$.*

The sets $P(S,T)$ have a geometric interpretation and our proof asserts that these sets have a specific shape. Suppose we plot $P(S,T)$ on the cartesian plane. We can partition the plane into three regions. $P(S,T)$, $\overline{P}(S,T)$, and an invalid region. We say that a point $\alpha = (\alpha_1, \alpha_2)$ is *invalid* if there are no two valuations v and u such that $\alpha_1 = v(S) - v(T)$ and $\alpha_2 = u(M \setminus S) - u(M \setminus T)$. As an example for a point that is invalid, consider two bundles S, T, $S \subseteq T$: bidder 1 prefers T over S and bidder 2 prefers $M \setminus S$ over $M \setminus T$, because of the monotonicity of the valuations; thus in this case the only valid points are in the north-west quadrant. For S and T where the symmetric difference is not empty, all points in $P(S,T)$ are valid. Finally, a point belongs to $\overline{P}(x,y)$ if it is valid and not in $P(x,y)$.

We now discuss the connection between the structure of P's and affine maximizers. Suppose f is a weighted welfare maximizer. Then, there exist constants w_1, w_2 for the two bidders such that for any two valuations v, u and $f(v) = (S, M \setminus S)$, and for any other outcome $(T, M \setminus T)$,

$$w_1 \cdot v(S) + w_2 \cdot u(M \setminus S) \geq w_1 \cdot v(T) + w_2 \cdot u(M \setminus T)$$

A rearrangement of the above expression will give us the following:

$$w_1 \cdot (v(S) - v(T)) \geq w_2 \cdot (u(M \setminus T) - u(M \setminus S))$$

The form of the above inequality implies that the following conditions must necessarily hold for any affine maximizer. Next to each condition we list the lemma that establishes the condition for any mechanism that satisfies the requirements of our characterization.

1. We must be able to arrive at the allocation $f(v, u)$, for any valuations v, u, by using the following process of "pairwise elections" between the allocations. Initially, all allocations are available. At every step, we consider an arbitrary pair of available allocations, $(S, M \setminus S)$ and $(T, M \setminus T)$ and rule out one of the two possibilities $f(v, u) = (S, M \setminus S)$, or $f(v, u) = (T, M \setminus T)$ by a deterministic rule that considers *only* the vector (indexed by machines) of differences $(v(S) - v(T), u(M \setminus S) - u(M \setminus T))$ (For an affine maximizer, this rule is given by the inequality above). We stop when one allocation remains. Lemma 3.3 establishes this condition; in essence, $P(S, T)$ is precisely the set of differences for which the allocation $(S, M \setminus S)$ 'beats' the allocation $(T, M \setminus T)$.

2. The slopes of the line must be identical for all choices of S and T. Lemma 3.5 proves that all the P's are identical, up to their invalid regions.

3. Further, if we plot the sets $P(S, T)$ and $\overline{P}(S, T)$ on the cartesian plane, they must be separated by a line (through the origin for weighted welfare maximizers). Lemma 3.8 establishes this.

Finally, taken together, the above conditions imply that the social choice function must be an affine maximizer: As all the P's are identical, the line that separates P from $\overline{P}$ must be identical. The slope of this line determines the weights w_1 and w_2 for the two players.

A technical note about the proof: When we have to define a new valuation, we sometimes explicitly define the values of only some of the bundles. The value of the rest of the bundles is the minimal value possible to ensure the monotonicity of the valuation[2].

The Consistency of $P(S, T)$

LEMMA 3.3. *Let S, T be two different bundles. Suppose v and u are two valuations such that $f(v, u) = (S, M \setminus S)$. Let $\alpha = (v(S) - v(T), u(M \setminus S) - u(M \setminus T))$. Let v' and u' be two valuations where $v'(S) - v'(T) = v(S) - v(T)$ and $u'(M \setminus S) - u'(M \setminus T) = u(M \setminus S) - u(M \setminus T)$. Then, $f(v', u') \neq (T, M \setminus T)$.*

PROOF. (of Lemma 3.3) Assume $S \neq \emptyset$, otherwise interchange the roles of the players. The proof is by contradiction. Suppose $f(v', u') = (T, M \setminus T)$. Define v'' as follows, where $c > 0$ is large enough to enforce subadditivity constraints:

$$v''(U) = \begin{cases} 0, & U = \emptyset, \\ v(S) + c, & U = S \\ v(T) + c, & U = T \\ c, & U \neq \emptyset, \quad U \not\subseteq S, T, \quad S, T \not\subseteq U. \end{cases}$$

By strong monotonicity and as all items are allocated, $f(v, u) = f(v'', u) = (S, M \setminus S)$, because $v''(S) - v(S) \geq v''(U) - v(U)$, for all bundles U. (Notice the use of the assumption that all items are allocated. If bidder 1 is allocated S then bidder 2 must be allocated $M \setminus S$). Similarly, $f(v', u') = f(v'', u') = (T, M \setminus T)$.

As $f(v'', u) = (S, M \setminus S)$ and $f(v'', u') = (T, M \setminus T)$, by strong-montonicity, $u(M \setminus S) - u(M \setminus T) > u'(M \setminus S) - u'(M \setminus T)$. But this contradicts the antecedent of the lemma. □

The Closure Lemma

LEMMA 3.4 (CLOSURE). *Let $\alpha \in P(S, T)$. Let $\epsilon = (\epsilon_1, \epsilon_2)$, where $\epsilon_1 \geq 0, \epsilon_2 \geq 0$. If $\alpha + \epsilon$ is valid for (S, T) then $\alpha + \epsilon \in P(S, T)$. As a corollary, if $\alpha \in \overline{P}(S, T)$ and $\alpha - \epsilon$ is valid for (S, T) then $\alpha - \epsilon \in \overline{P}(S, T)$.*

PROOF. Let v and u be two valuations where $\alpha = (v(S) - v(T), u(M \setminus S) - u(M \setminus T))$, and $f(v, u) = (S, M \setminus S)$. Define v'' and u'', for large enough c:

$$v'(U) = \begin{cases} 0, & U = \emptyset, \\ c + v(U) + \epsilon_1, & U = S \\ c + v(T), & U = T \\ c, & U \neq \emptyset, U \not\subseteq S, T \; S, T \not\subseteq U. \end{cases}$$

$$u'(U) = \begin{cases} 0, & U = \emptyset, \\ c + u(U) + \epsilon_2, & U = M \setminus S \\ c + u(T), & U = M \setminus T \\ c, & U \neq \emptyset, U \not\subseteq M \setminus S, M \setminus T, \\ & M \setminus S, M \setminus T \not\subseteq U. \end{cases}$$

By strong monotonicity and since all items are allocated, $f(v, u) = f(v', u) = f(v', u') = (S, M \setminus S)$. Thus, $\alpha + \epsilon \in P(S, T)$. Finally, we note that if S is the empty set, then a small technicality arises, as the above transformation is not allowed, since we assume that the valuations are normalized and the value of the empty set cannot be larger than 0. In this case we set each bundle U to be $v'(U) = \min(v(U), v(M) - \epsilon_1)$. If $S = M$, then u' is defined similarly: $u'(U) = \min(u(U), u(M) - \epsilon_2)$. □

The P*'s are Identical*

We now prove that that all the P's are identical, up to invalid points. Formally speaking, we only prove that the *interiors* of the P's are identical, where the interior is defined in the usual topological sense. We denote the interior of $P(S, T)$ by $\dot{P}(S, T)$.

LEMMA 3.5. *Suppose that for each $P(S, T)$, and for every $\epsilon > 0$ there exists $\beta \in P(S, T)$, $|\beta| < \epsilon$. Let S, T, R, W be different bundles. Then, $\dot{P}(S, T) \cap \overline{P}(R, W) = \emptyset$.*

Notice that scalability implies that we indeed have that for each $P(S, T)$, and for every $\epsilon > 0$ there exists $\beta \in P(S, T)$, $|\beta| < \epsilon$.

PROOF. (of Lemma 3.5) It is enough to prove the following two claims:

[2] Our characterization requires that in all valuations each two different bundles get different values. Hence,"noise" should be added to the valuations below, and to some of the valuations used in the sequel. We omit this noise from the description of the valuations to enhance readability.

CLAIM 3.6. *Let S, T and W be different bundles. $\dot{P}(S,T) \cap \overline{P}(S,W) = \emptyset$.*

PROOF. Let $\alpha \in \dot{P}(S,T)$. If α is not valid for (S,W) then the claim is trivially true for this α. Therefore, we assume that α is valid for (S,W) and prove that $\alpha \in \dot{P}(S,W)$. Let v,u be two valuations such that $f(v,u) = (S, M \setminus S)$, and $(v(S) - v(T), u(M \setminus S) - u(M \setminus T)) = \alpha$.

We prove the claim by considering several different cases, depending on the value of α. There are 4 possible cases, and we omit the description of one case that is symmetric to case 3.

Finally, before turning to the case analysis, let us note that we assume that neither S, T, nor W are the empty set (or that they equal the bundles of all items). We explain the technical difficulty in this case and how to overcome it at the end of the proof.

Case 1: $\alpha_1 \geq 0$**,** $\alpha_2 \geq 0$

Let v,u be two valuations such that $f(v,u) = (S, M \setminus S)$ which are "small enough", $v(M), u(M) << \epsilon$, for some $\epsilon > 0$. Define the following valuations, where c is chosen to be large enough to enforce subadditivity constraints (notice that in this case due to the value of α, we have, for example, that W is not a subset of S):

$$v'(U) = \begin{cases} 0, & U = \emptyset, \\ v(U) + c + \alpha_1, & U \subseteq S \\ v(U) + c, & S \nsubseteq U. \end{cases}$$

$$u'(U) = \begin{cases} 0, & U = \emptyset, \\ u(U) + c + \alpha_2, & U \subseteq M \setminus S \\ u(U) + c, & M \setminus S \nsubseteq U. \end{cases}$$

We now prove that $f(v,u) = f(v',u) = f(v',u') = (S, M \setminus S)$. First $f(v,u) = f(v',u) = (S, M \setminus S)$ by strong monotonicity and since for each U we have that $v'(S) - v'(U) \geq v(S) - v(U)$. A similar argument shows that $f(v',u) = f(v',u') = (S, M \setminus S)$.

In particular, we proved that $\alpha \in P(S,W)$, since $(v'(S) - v'(W), u'(M \setminus S) - u'(M \setminus W))$ is of distance of less than $\epsilon > 0$ from α, for any $\epsilon > 0$, and since the process can be repeated for any $\epsilon > 0$.

Case 2: $\alpha_1 < 0$**,** $\alpha_2 < 0$

Roughly speaking, we will define v'', u'' where $(S, M \setminus S)$ "beats" $(T, M \setminus T)$, and $(T, M \setminus T)$ "beats" $(W, M \setminus W)$. Hence $f(v'', u'') = (S, M \setminus S)$. Of course, we will make sure that $(v''(S) - v''(W), u''(M \setminus S) - u''(M \setminus W)) = \alpha$.

Let v', u' be two valuations such $f(v', u') = (T, M \setminus T)$, and $|v'|, |u'| < \epsilon$, for some $\epsilon > 0$.

We now define v''. We start with $v'' = v$, and alter v''. Set $v''(W) = v(T) - (v'(T) - v'(W))$. Set the value of any bundle $U \neq S, T, W$ to the minimum value possible by the monotonicity constraints of the valuations. Finally, for all $U \neq \emptyset$ raise $v''(U)$ by some large enough constant c, to enforce subadditivity constraints. Define u'' similarly, by starting with $u'' = u$, and setting $u''(M \setminus W) = u(M \setminus T) - (u'(M \setminus T) - u'(M \setminus W))$.

We now prove that $f(v'', u'') = (S, M \setminus S)$. We first claim that the resulting output is not $(U, M \setminus U)$, for any $U \subseteq S$. To see this let $\beta = (v(S) - v(U), u(M \setminus S) - u(M \setminus U))$, and thus $\beta \in P(S,U)$. However $(v''(S) - v''(U), u''(M \setminus S) - u''(M \setminus U)) = \beta + \epsilon'$, for some $\epsilon' > 0$. Thus, by Lemmas 3.3 and 3.4, $f(v'', u'') \neq (S, M \setminus S)$.

A similar argument shows that $f(v'', u'') \neq (U, M \setminus U)$, for $U \neq S, T$, $U \nsubseteq S$. For this we let $\beta = (v'(T) - v'(U), u'(M \setminus T) - u'(M \setminus U))$, and recall that $f(v', u') = (T, M \setminus T)$.

Thus, the only two alternatives that we have to consider are $f(v'', u'') = (S, M \setminus S)$, or $f(v'', u'') = (T, M \setminus T)$. However, since $\alpha \in P(S,T)$, it must be the case that $f(v'', u'') = (S, M \setminus S)$ (by Lemma 3.3).

As in the previous case, we proved that $\alpha \in P(S,W)$, since $(v'(S) - v'(W), u'(M \setminus S) - u'(M \setminus W))$ is of distance of less than $\epsilon > 0$ from α, for any $\epsilon > 0$.

Case 3: $\alpha_1 \geq 0$**,** $\alpha_2 < 0$

Let v', u' be two valuations such $f(v', u') = (T, M \setminus T)$, and $|v'|, |u'| < \epsilon$, for some $\epsilon > 0$. We start by defining v'' and u''. v'' is defined as in case 1, and u'' is defined as in case 2. Using arguments similar to the previous cases, we have that $(T, M \setminus T)$ "beats" every alternative $(U, M \setminus U)$, $U \nsubseteq S$, because $f(v', u') = (T, M \setminus T)$. $(S, M \setminus S)$ "beats" every alternative $(U, M \setminus U)$, $S \subset U$, because $f(v,u) = (S, M \setminus S)$. Finally, $(S, M \setminus S)$ beats $(T, M \setminus T)$ because $\alpha \in P(S,T)$.

Let us note how to handle the case where either S, T, nor W are the empty set (or that they equal the bundles of all items). The difficulty is that in this case we cannot raise the appropriate bundle by c, to make it subadditive. First, consider the case where $T = \emptyset$ (or $M \setminus T = \emptyset$). The proof is similar to the proof in case 2, with some minor changes. The first change is that we build the valuations in a way such that the alternative S "beats" every alternative that contains $S \setminus W$. The second change is that instead of raising all bundles by some large enough constant c, it is enough to raise only bundles $U \supseteq S \setminus W$ (for v), and $U \subseteq (M \setminus S) \cup W$ (for u).

If $T = \emptyset$ then the treatment is similar. Finally, if $S = \emptyset$ then we define the valuations similarly to this case in Lemma 3.4. $\square$

CLAIM 3.7. *Let S, T and U be different bundles. $\dot{P}(S,T) \cap \overline{P}(U,T) = \emptyset$.*

PROOF. The proof is similar to the proof of the previous claim and is omitted from this extended abstract. $\square$

Let S, T be some bundles where S is not contained in T and T is not contained in S such that the outcomes $(S, M \setminus S)$ and $(T, M \setminus T)$ are in the range of f; by assumptions, the range of f has size at least $m + 2$, and such outcomes must exist. Notice that all points in the plane are valid for (S,T). Thus we have that $P(S,T)$ and each $P(U,W)$ are identical, up to invalid points. The same holds for $\overline{P}(S,T)$ and $\overline{P}(U,W)$. We now have that all the P's are the same, up to invalid points.

The P's are Separated by a Line

LEMMA 3.8. *Suppose there exist two bundles S and T be such that $(S, M \setminus S)$ and $(T, M \setminus T)$ are in the range and neither of the bundles is contained in the other. Then, $P(S,T)$ is separated by a line from $\overline{P}(S,T)$.*

PROOF. First, observe that if neither of the bundles is contained in the other, then the whole plane is valid for (S,T). Let α be some point that is on the border of $P(S,T)$ and $\overline{P}(S,T)$ (notice that such point must exist, otherwise either $(S, M \setminus S)$ or $(T, M \setminus T)$ are not in the range). Notice that by scalability $x \cdot \alpha$ is also on the border, for all x. Finally, observe that the closure lemma separates the plane into two separate regions: if β is above the $x \cdot \alpha$ line then it is in $P(S,T)$, if it is below the line, then it is in $\overline{P}(S,T)$. □

As we observed before, all the P's are equal, up to invalid points. By this and the previous lemma, all the P's are separated by the same line, up to invalid points. This is enough to prove affine maximization.

3.2 Some Non-Affine Maximizers Mechanisms

In this section, we discuss the tightness of our characterization. That is, we mention examples of mechanisms that do not obey the conditions of the characterization, and thus break it. The examples serve to clarify the intermediate steps in the proof of Theorem 3.1.

EXAMPLE 3.9 (INDIFFERENT DICTATOR). *Consider the following mechanism for combinatorial auctions with two bidders, which does not always allocate all the items. Fix two items a, b. The mechanism always allocates all the items other than b to the second bidder. If the first bidder's value for item a is an even integer, then item b is allocated to the second bidder. Otherwise b is not allocated at all. Note that the first bidder is never allocated any items and the second bidder cannot determine its allocation; we may term the first bidder as an* Indifferent Dictator. *It is easy to see that the mechanism is truthful but not an affine maximizer.*

Recall that the characterization in Lemma 3.3 says that we can decide the "winning" outcome by a process of pair-wise *winner determination; in each step, we look* only *at the vector of relative preferences (indexed by the bidders) between the two allocations under consideration. The mechanism breaks this property. There exist pairs of valuation profiles, where the relative preferences of the two bidders between the two allocations are identical, but the allocations chosen differ.*

We now describe a mechanism, called *Serial*, that satisfies Lemma 3.3, but breaks Lemma 3.5. Informally, Serial is closer to being an affine maximizer than the Indifferent Dictator.

EXAMPLE 3.10 (SERIAL). *The mechanism works as follows: Bidder* 1 *selects its most profitable bundle, where each item has a fixed price of p; Bidder* 2 *then does the same with the left-over items. Note that the mechanism does not allocate all items and is obviously truthful.*

Unlike the Indifferent Dictator mechanism, Serial satisfies Lemma 3.3. Recall that Lemma 3.3 says that we can decide the "winning" allocation pair-wise*, focussing* only *at the vector of relative preferences (indexed by the bidders) between the two allocations. However Serial is not an affine maximizer as it does not satisfy Lemma 3.5.*

4. SCHEDULING MECHANISMS

We now prove that every mechanism that provides a finite approximation ratio for minimizing the makespan with two machines is a task independent mechanism. The characterization only requires the mechanisms to have finite approximation ratio. We prove this by first characterizing machine scheduling mechanisms that satisfy a property called decisiveness. We conclude the section with other applications of our characterization.

4.1 The Characterization

We say that an allocation $(S, M \setminus S)$ is *in the range* of v if there exists some u such that $f(v,u) = (S, M \setminus S)$. A mechanism is called *decisive* if for every valuation v of one machine all allocations are in the range of v. Unlike the characterization from the previous section, the characterization from this section is stated as a property of the payments offered by the mechanism. Recall (see the preliminaries) that for any truthful mechanism payments to machine i for a specific allocation do not vary with machine i's value, and that the allocation chosen maximizes the machine's utility subject to its bid. Also, note that the utility of a machine is the payment it receives minus the time it spends executing the jobs it is allocated.

Let $T_1 = \{v_1, \ldots v_t\}, T_2 = \{v'_1, \ldots v'_2\}$, $|T_1| = |T_2| = t$, be two sets of valuations. We say that T_1 and T_2 *agree on* j if for each k we have that $v_k(\{j\}) = v'_k(\{j\})$. Also, we denote by $p^i_{j|S}(v) = p^i_{S \cup \{j\}}(v) - p^i_S(v)$ the marginal payment for job j given bundle S (and v). We sometimes omit the superscript from the expression $p^1_S(u)$ if it is clear from the context. The main result of this section is that decisive mechanisms induce a pricing scheme with a special additive structure:

DEFINITION 4.1. *A scheduling mechanism f is called* local *if the following holds for any machine i. For every bundle S there exists a constant C^i_S such that for any valuation v_{-i} of the other machines, $p^i_S(v_{-i}) = C^i_S + \sum_{j \in S} p^i_{j|\emptyset}(v_{-i})$.*

A mechanism is called task independent *[3] if we have that $C^i_S = 0$, for all bundles S.*

Note that for any task-independent mechanism, the allocation can be determined on a job-by-job basis: there is a payment for each job j, and the payment for bundle S is simply the sum of payments for items in S. As an example for a truthful mechanism that is local but not task independent, consider an affine maximizer where the additive weights for some allocations are non-zero. The main result of this section shows that every decisive mechanism is local. In the next subsection we show that every mechanism that provides a finite approximation ratio is task independent.

We restrict our attention to valuations that are strictly positive; note that this does not affect our ability to reason about properties of mechanisms such as *approximation ratios.* We assume that f satisfies strong monotonicity, the appendix shows that this is essentially without loss of generality.

THEOREM 4.2. *Let f be a decisive mechanism for scheduling* 2 *machines. Then, f is local.*

We first show that marginal payments, to a machine, for adding job j to a set S depends only on the valuation of the other machine for the job j. W.l.o.g, the lemma is from the perspective of the second machine.

LEMMA 4.3. *Let S be a bundle and $j \notin S$ a job. Let v and v' be two valuations that agree on j. Then, for every bundle S and job j, $j \notin S$: $p_{j|S}(v) = p_{j|S}(v')$.*

PROOF. Assume to the contrary that there exist valuations v and v' with that agree on j and $p_{j|S}(v) \neq p_{j|S}(v')$, w.l.o.g. $p_{j|S}(v) > p_{j|S}(v')$.

We now show that there exist valuations u and u' such that $u(\{j\}) > u'(\{j\})$ and $f(v,u) = (S, M \setminus S)$, $f(v', u') = (S \cup \{j\}, M \setminus S \setminus \{j\})$.

We define the valuation u, for positive c, δ and ϵ_1, in the following way:

$$u(\{t\}) = \begin{cases} c, & t \in S \\ \delta, & t \in M \setminus S \\ p_{j|S}(v) - \epsilon_1 & t = j. \end{cases}$$

By decisiveness of f, $M \setminus S$ is in the range of v; hence, for any positive ϵ_1 there exist large enough c and small enough δ such that the allocation $M \setminus S$ maximizes the machine's utility and hence $f(v, u) = (S, M \setminus S)$.

Similarly we can show that there exists u' with $u'(\{j\}) = p_{j|S}(v') + \epsilon_2$, for any small positive ϵ_2 such that $f(v, u') = (S \cup \{j\}, M \setminus S \setminus j)$. Note that for small enough ϵ_1, ϵ_2, $u(\{j\}) > u'(\{j\})$.

Now define v'', u'' as follows:

$$v''(\{t\}) = \begin{cases} \min(v(\{t\}), v'(\{t\})), & t \in S \\ \max(v(\{t\}), v'(\{t\})), & t \notin S, t \neq j \\ v'(\{j\}), & t = j. \end{cases}$$

$$u''(\{t\}) = \begin{cases} \min(u(\{t\}), u'(\{t\})), & t \notin S \\ \max(u(\{t\}), u'(\{t\})), & t \in S, t \neq j \\ \frac{u'(\{j\})+u(\{j\})}{2}, & t = j. \end{cases}$$

By strong monotonicity and since all jobs are allocated: $f(v, u) = f(v'', u) = f(v'', u'')$. For instance, using strong monotonicity, the allocation to the first machine must remain the same when its bid from v to v''; further, as all jobs must be allocated, the allocation to the other machine cannot change either. Similarly, $f(v', u') = f(v'', u') = f(v'', u'')$. This completes the contradiction as $f(u, v) \neq f(u', v')$. □

The following lemma almost finishes the characterization:

Lemma 4.4. *For any job j and set S such that $j \notin S$, $C^{j,i}_S = p_{j|S}(v) - p_{j|\emptyset}(v)$ is constant.*

Proof. If $S = \emptyset$ then the lemma is trivial. Let j' be some item in S. Observe that by definition:

$$\begin{aligned} p_S(v) &= p_{S\setminus\{j\}\setminus\{j'\}}(v) + \\ &\quad p_{j|S\setminus\{j\}}(v) + p_{j'|S\setminus\{j'\}\setminus\{j\}}(v) \\ &= p_{S\setminus\{j\}\setminus\{j'\}}(v) + p_{j'|S\setminus\{j\}'}(v) + \\ &\quad p_{j|S\setminus\{j'\}\setminus\{j\}}(v) \end{aligned}$$

Thus we have that

$$p_{j|S\setminus\{j\}}(v) - p_{j|S\setminus\{j'\}\setminus\{j\}}(v) = p_{j'|S\setminus\{j'\}}(v) - p_{j'|S\setminus\{j'\}\setminus\{j\}}(v)$$

By Lemma 4.3 and the decisiveness of f, $(p_{j'|S\setminus\{j\}'}(v) - p_{j'|S\setminus\{j'\}\setminus\{j\}}(v))$ does not depend on $v(\{j\})$. In particular, it is equal in every two valuations that agree on j. To complete the proof, notice that we can express $p_{j|S'}(v) - p_{j|\emptyset}(v)$ as a telescoping sum of differences of the type $p_{j|S}(v) - p_{j|S\setminus j}(v)$. □

To see that for every bundle S there exists a constant C_S such that $p^i_S(v_{-i}) = C^i_S + \sum_{j\in S} p^i_{j|\emptyset}(v_{-i})$, we start by arbitrarily ordering the items in S: $j_1, j_2, \ldots, j_{|S|}$. Let S_k denote the set of the first k items. By definition we have that $p^i_S(v_{-i}) = \sum_{k=1}^{|S|} p^i_{j|S_k}(v_{-i})$. By the last lemma we can rewrite $p^i_S(v_{-i}) = \sum_{k=1}^{|S|} (p^i_{j|\emptyset}(v_{-i}) + C^{i,k}_{S_k})$. Now let $C^i_S = \sum_{k=1}^{|S|} C^i_{S_k}$.

4.2 Applications

4.2.1 Makespan Minimization

We first prove that if f is a mechanism for 2 machines that provides a finite approximation ratio to the makespan, then f must be decisive. By our characterization we have that f must be local. We then use this to show that f must be task independent. Finally, we characterize all universally truthful mechanisms that provide a finite approximation ratio by observing that all mechanisms in the support of a universally truthful mechanism (that provides a finite approximation ratio) must provide a finite approximation ratio alone. Hence, each such mechanism must be task independent.

Theorem 4.5. *Let f be a mechanism for minimizing the makespan for 2 machines that provides a finite approximation ratio. Then, f is task independent.*

Proof. To prove the theorem we first prove that f is decisive, hence by our characterization it is local. We then show that f is task independent.

Lemma 4.6. *Let f be a mechanism for minimizing the makespan for 2 machines that provides a finite approximation ratio. Then, f is decisive.*

Proof. Let v be a valuation such that S is not in its range. Define the following valuation (where $\epsilon > 0$ is some small constant):

$$u'(\{t\}) = \begin{cases} \epsilon, & t \in S \\ \infty, & t \notin S. \end{cases}$$

Let $f(v, u) = (M \setminus T, T)$. Notice that in order to obtain a finite approximation ratio and because S is not in the range T is strictly contained in S. Thus S cannot be the empty set. Define v' as follows:

$$v'(\{t\}) = \begin{cases} \epsilon, & t \in M \setminus S \\ v'(t), & t \in S. \end{cases}$$

Strong monotonicity guarantees that $f(v', u) = (M \setminus T, T)$. Notice that the allocation $(M \setminus S, S)$ provides a makespan of $m \cdot \epsilon$, while in $(M \setminus T, T)$ machine 1 gets at least one job that is not in $M \setminus S$ and thus have a processing time much larger than ϵ. Thus, f does not provide a finite approximation ratio for the instance (v', u). □

We complete the proof of Theorem 4.5. It suffices to show that for each machine i and bundle S, $C^i_S = 0$.

Normalize $p^i_\emptyset(v)$ to 0, and $C^i_\emptyset = 0$. First, we argue that for all bundles S, $p_S(v)$ tends to 0 as $v(S)$ tends to 0; If not there exists some bundle S for which this is not true. In particular, by decisiveness it must be the case that $p_M(v) \geq p_S(v)$ for all v (otherwise, M will never be selected since the valuations are strictly positive and this implies $u(S) < u(M)$). Set $v(\{k\}) = \epsilon$ for each job j. Also set $u(k) = p_M(v)/2m$. To get a finite approximation, all jobs must be allocated to the first machine. However, this is not the utility maximizing allocation for the second machine (allocating all items to u gives it a strictly positive profit), contradicting the truthfulness of f.

Suppose that for some S it holds that $C^i_S < 0$. In this case the mechanism is not decisive as the allocation $(S, M \setminus S)$ will never be the output if $v(\{k\}) = \epsilon$ for all jobs k, and sufficiently small $\epsilon > 0$, similarly to before.

Finally, suppose there exists some $C^i_S > 0$. Define valuation u where for each $j \in S$, $u(\{j\}) = C^i_S/|S| - \epsilon$, and $u(\{j\}) = \epsilon$ for $j \notin S$. Choose some $v(\{j\}) = \epsilon$ for each job j. Observe, that second machine 2 is assigned at least one job $j \in S$ in $f(v,u)$. Hence, f provides a makespan of at least $C^i_S/|S| - \epsilon$, while the optimal makespan is $m \cdot \epsilon$. The theorem follows for sufficiently small ϵ. □

We now take advantage of the above characterization to study randomized mechanisms.

PROPOSITION 4.7. *Let A be a universally truthful mechanism for some minimization problem that provides in expectation a finite approximation ratio. Then, every mechanism in A's support must provide a finite approximation ratio.*

PROOF. Let f be a mechanism in A's support that does not provide a finite approximation ratio. There exists some instance I where the approximation ratio A provides on I is ∞. In particular, observe that $A(I) = \infty$, as there is a non-zero probability that f will be selected as the realization of A. □

We therefore have that each mechanism in the support of a universally truthful mechanism (that provides a finite approximation ratio) must be task independent. Thus, in order to determine the exact approximation ratio of universally truthful randomized mechanisms for 2 machines, all that remains is to optimize over the class of mechanisms that are distribution over task independent mechanisms, which we defer to future work.

4.2.2 Weighted Sum of Completion Times

In this scheduling problem we are also given a weight w_j for each job j. Fix some schedule, and let T_j denote the time job j is completed in this schedule. The goal is to find the schedule that minimizes $\Sigma_j w_j T_j$. A lower bound of 1.17 is known for this problem [1], even if the machines are uniformly related.

A proof identical to the proof of Lemma 4.6 shows that every truthful mechanism for minimizing the weighted sum of completion times must be decisive. The characterization proceeds similarly and we have that every truthful approximation mechanism must be local, and even task independent.

Now it is easy to show that every deterministic mechanism for minimizing the weighted sum of completion times cannot achieve an approximation ratio better than 2, even for 2 machines.

4.3 A Non-local Mechanism for more than 2 Machines

Unfortunately, the above characterization cannot be extended to more than 2 machines: there exists a truthful mechanisms for $n > 2$ machines with an approximation ratio of $\Theta(n)$, and is neither local nor task independent.

Consider the following mechanism: for each job j let i_j be the machine with the highest cost for j. Let a be a different, arbitrary job. If $v_{i_j}(a) \geq 10$ or $i_j = 1$, then allocate job j to the machine with the lowest cost. Otherwise, allocate job j to the machine with the lowest cost among the other $n-1$ machines, where machine 1 cost for this purpose is $v_1(\{j\})/1.001$. This is a truthful, non-local mechanism, with a finite approximation ratio (of $\Theta(n)$).

5. STABLE MECHANISMS

In this section we extend our characterizations to settings where there are more than two players/machines and not all items are allocated (in the case of combinatorial auctions). The characterizations instead use a condition called stability; the conditions clarifies the role of the condition that there are two players and all items are allocated.

DEFINITION 5.1. *Let f be a mechanism for a scheduling or a combinatorial auctions domain. The mechanism f is* stable *if for each bidder i, and each two of his possible valuations v, v' the following holds: let $f(v, v_{-i}) = (S_1, S_n \ldots, S_n)$ and $f(v', v_{-i}) = (S'_1, \ldots S'_n)$. If $S_i = S'_i$, then $S_k = S'_k$, for all $k \neq i$.*

Thus, for any stable mechanism, if we change the valuation of one of the players and that player's allocation does not change, then no other player's allocation changes. The assumption that there are two players and all items are allocated implies stability; if we always allocate all items then an outcome is uniquely identified by the allocation to one of the two players, and, if one player's allocation does not change with a change in its report, the other player's allocation does not change either. We now state the two characterizations. The second characterization additionally requires that the social choice function is scalable, without which there exist truthful mechanisms that are not affine maximizers (see the serial mechanism from Section 3.2). A social choice function satisfies scalability if the outcome does not change when all valuations are scaled uniformly. The proofs of the theorems are straightforward extensions of the proofs of Theorems 3.1 and 4.2.

THEOREM 5.2. *Every stable and scalable mechanism with a large enough range for combinatorial auctions where bidders have subadditive valuations is an affine maximizer.*

THEOREM 5.3. *Every stable mechanism for the machine scheduling problem that yields a finite approximation for the minimum makespan objective is task independent.*

Finally we note that stability is implied by a (stronger) condition called *Independence of Irrelevant Alternatives* (IIA) used in characterizations by [13]. Stability is arguable more illuminating than IIA from the perspective of designing social choice rules. Note that stability is satisfied, upto tie-breaking, by all social choice functions that optimize some function of the players' realized valuations. We cannot make a similar claim for IIA.

These characterizations constrain the types of objective functions we may optimize. Further, as discussed in the introduction, optimizing over affine maximizers might be computationally hard. For scheduling, local mechanisms are $\Omega(n)$-approximate. Thus it is important to understand the capabilities of techniques that are not based on optimizing some objective and explicitly break the stability condition; e.g., the random sampling techniques of [11, 8].

Acknowledgements

We thank Aleksandra Korolova, Ron Lavi, and Tim Roughgarden for helpful discussions.

This work was done while the first author was visiting Stanford University. The first author was supported by the Adams Fellowship Program of the Israel Academy of

Sciences and Humanities, and by a grant from the Israeli Academy of Sciences. The second author is supported by NSF Award CCF-0448664 and a Stanford Graduate Fellowship.

6. REFERENCES

[1] Aaron Archer and Eva Tardos. Truthful mechanisms for one-parameter agents. In *FOCS'01*.

[2] Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK 03*.

[3] George Christodoulou, Elias Koutsoupias, and Annamária Kovács. Mechanism design for fractional scheduling on unrelated machines. In *ICALP'07*.

[4] George Christodoulou, Elias Koutsoupias, and Angelina Vidali. A lower bound for scheduling mechanisms. In *SODA '07*.

[5] Shahar Dobzinski. Two randomized mechanisms for combinatorial auctions. In *APPROX-RANDOM*, 2007.

[6] Shahar Dobzinski and Noam Nisan. Limitations of vcg-based mechanisms. In STOC'07.

[7] Shahar Dobzinski and Noam Nisan. Mechanisms for multi-unit auctions. In EC'07.

[8] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In STOC'06.

[9] Uriel Feige. On maximizing welfare where the utility functions are subadditive. In STOC'06.

[10] Uriel Feige and Jan Vondrak. Approximation algorithms for allocation problems: Improving the factor of 1-1/e. In FOCS'06.

[11] Andrew Goldberg, Jason Hartline, Anna Karlin, Mike Saks, and Andrew Wright. Competitive auctions. Games and Economic Behaviour, 2006.

[12] Ron Lavi. Computationally efficient approximation mechanisms. In Algorithmic Game Theory, edited by Noam Nisan and Tim Roughgarden and Eva Tardos and Vijay Vazirani.

[13] Ron Lavi, Ahuva Mu'alem, and Noam Nisan. Towards a characterization of truthful combinatorial auctions. In *FOCS'03*.

[14] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. In FOCS 2005.

[15] R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[16] Noam Nisan and Amir Ronen. Computationally feasible vcg-based mechanisms. In *EC'00*.

[17] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *STOC*, 1999.

[18] Kevin Roberts. The characterization of implementable choise rules. In Jean-Jacques Laffont, editor, *Aggregation and Revelation of Preferences. Papers presented at the first European Summer Workshop of the Economic Society*, pages 321–349. North-Holland, 1979.

APPENDIX

A. STRONG MONOTONICITY

We justify why it suffices to focus on social choice functions that are strongly monotone. We start with combinatorial auctions with subadditive valuations.

A.1 Combinatorial Auctions

We focus on domains that are *open* in the following sense (this is a slight change from the definition of [13].)

Definition A.1. *A domain V of valuations for combinatorial auctions is* open *if for each $v \in V$ the value of each bundle S can be decreased or increased by ϵ, while the value of the other bundles does not increase by ϵ or more, for some $\epsilon > 0$, and the altered valuation is still in V.*

Intuitively, a domain is open if both the monotonicity and the subadditivity constraints hold with strict inequality. The next lemma shows that if the domain is open and all the implementable social choice functions that satisfy strong monotonicity are affine maximizers, then all the implementable social choice functions are affine maximizers (since every implementable social choice function satisfies weak monotonicity).

Lemma A.2 (essentially from [13]). *Let f be an an implementable social choice function that satisfies weak monotonicity that is not an affine maximizer that is defined on an open domain. Then, there exists a strongly monotone social choice function that is also not an affine maximizer that is defined on the same domain.*

Arguably, our main interest in characterizations follow from our desire to prove lower bound on approximation ratios. We now show to transform every set of subadditive or XOS valuations to an open set of valuations, with "minimal" changes. We note that all known lower bounds goes through after this transformation. For subadditive valuations, we change the valuation so that no two bundles have the same value (by increasing the value of the bundles by a different, small enough ϵ for each bundle). To ensure that the subadditivity constraints are strict, we then raise the value of each bundle (except for the empty set) by the same ϵ, for some small $\epsilon > 0$.

For XOS valuations the transformation is essentially the same. Here we raise the value of each bundle by raising the value of each item in the corresponding clause.

A.2 Machine Scheduling

We now justify why it suffices to focus on strongly monotone social choice functions for machines scheduling. Fix an implementable allocation rule f. Recall that any such allocation rule satisfies weak monotonicity and vice-versa. Now consider a valuation profile v, u for which we do not have strong monotonicity, we show that there is an almost equivalent valuation profile for which we definitely have strong monotonicity.

Consider a valuation profile v, u for which we possibly not have strong monotonicity. Let $f(v, u) = (S, M \setminus S)$. Define, for small positive ϵ, the valuation profile v', u', with $v'(j) = v(j) - \epsilon$, for $j \in S$ and $v'(j) = v(j) + \epsilon$ for $j \notin S$; also $u'(j) = u(j) + \epsilon$, for $j \in S$ and $v'(j) = v(j) - \epsilon$ for $j \notin S$. First, note that for sufficiently small ϵ, the profile v', u' is strictly positive because the valuation profile v, u is strictly positive. Second, by weak monotonicity and as all jobs are allocated, we can prove that $f(v', u') = (S, M \setminus S)$.

It is easy to check that if strong monotonicity does not hold for the valuation profile v', u', then weak mendicity is broken for the profile v', u', contradicting the truthfulness of f. For instance, suppose there exists a v'' such that $f(v'', u) = (S'', M \setminus S'')$ and $v''(S'') + v(S) = v(S'') + v''(S)$ (the inequality '>' is ruled out because f is weakly monotone) and $S'' \neq S$; then it is easy to see that $v''(S'') + v'(S) < v'(S'') + v''(S)$.

B. COMBINATORIAL AUCTIONS WITH XOS VALUATIONS

B.1 Definition and Representation of XOS

Let us start with formally defining the XOS class. First, recall that a valuation is called *additive* if for all $S \subseteq M$, $v(S) = \Sigma_{j \in S} v(\{j\})$. Since an additive valuation is completely defined by the values $b_1, ..., b_m$ it assigns to items $1, ..., m$ respectively, it can be represented by the following *clause*:

$$(x_1 : b_1 \vee x_2 : b_2 \vee ... \vee x_m : b_m)$$

We can now define XOS valuations:

Definition B.1. *A valuation v is said to be XOS if there is a set of additive valuations $\{a_1, ..., a_t\}$, such that $v(S) = \max_k\{a_k(S)\}$ for all $S \subseteq M$. We denote XOS valuations by*

$$\begin{aligned}(x_1 &: a_1(\{1\}) \vee ... \vee x_m : a_1(\{m\})) \\ &\oplus \cdots \oplus (x_1 : a_t(\{1\}) \vee ... \vee x_m : a_t(\{m\}))\end{aligned}$$

where each of the clauses connected by the $\oplus$ sign represents an additive valuation.

We call the clause of an additive valuation a, for which $a(S) = \max_k\{a_k(S)\}$, the *maximizing clause* for S in v (if there are several such clauses we arbitrarily choose one).

B.2 The Characterization

In this subsection we remark how the valuations should be altered in the proof of Section 3 in order for the characterization to hold also for XOS valuations.

The only change that is required is in the definition of the valuations. Whenever the value of the bundle S is set to t (in v we add the clause $(\vee_{j \in S} j : ((v(S) + c)/|S|))$ to v, where c is sufficiently large constant that is the same for all clauses in v. Two problems arise: first, we might not be able to define the value of some other bundle $T \subseteq S$ as the value of T might be "dominated" by the clause of S. Notice that this problem does not arise if c is large enough: "according" to the S clause the difference between the value of S and the sum of values of items in T is at least $c/|S|$ (since there exists some $j \in S, j \notin T$, while $v(S) - v(T) << c/|S|$.

The second problem we might encounter is due to the "balancing": we start with an arbitrary clause $x_1 : b_1 \vee x_2 : b_2 \vee ... \vee x_m : b_m)$ and transform it to a clause of the form $(\vee_{j \in S} j : ((v(S) + c)/|S|))$, where all items have the value. The next proposition, needed in the proof of Lemma 3.5 if the valuations are XOS (needed in the construction of cases 2 and 3), shows some connection between $P(S,T)$, and $P(S,U)$, where $U \subseteq T$.

Proposition B.2. *If $\alpha \in P(S,T)$, then for every $U \subseteq T$, $(\frac{|U|(\alpha_1+c)}{|S|}, \frac{|M \setminus U|(\alpha_2+c)}{|M \setminus S|}) \in P(S,U)$, for large enough values of c.*

Proof. Take some v, u, such that $f(v,u) = (S, M \setminus S)$, and $\alpha = (v(S) - v(U), u(M \setminus S) - u(M \setminus U))$. Define v' and u':

$$\begin{aligned} v' &= (\vee_{j \in S} j : ((v(S) + c)/|S|)) \oplus \\ &\quad (\vee_{j \in T} j : ((v(T) + c)/|T|)) \end{aligned}$$

$$\begin{aligned} u' &= (\vee_{j \in M \setminus S} j : ((v(S) + c)/|M \setminus S|)) \oplus \\ &\quad (\vee_{j \in M \setminus T} j : ((v(M \setminus T) + c)/|M \setminus T|)) \end{aligned}$$

Notice that $f(v', u') = (S, M \setminus S)$. In particular this implies that for each $U \subseteq T$ the allocation $(T, M \setminus T)$ is not chosen. Also, $(v'(S) - v'(U), v'(M \setminus S) - v'(M \setminus U) = \frac{|U|(\alpha_1+c)}{|S|}, \frac{|M \setminus U|(\alpha_2+c)}{|M \setminus S|})$, as needed. □