A Sufficient Condition for Truthfulness with Single Parameter Agents

Nir Andelman *
School of Computer Science
Tel-Aviv University
andelman@cs.tau.ac.il

Yishay Mansour†
School of Computer Science
Tel-Aviv University
mansour@cs.tau.ac.il

ABSTRACT

We consider the task of designing truthful mechanisms for single parameter agents. We prove a general sufficient condition for truthfulness when each agent's valuation function for each possible outcome is a one-dimensional function of its type, continuous everywhere and differentiable almost everywhere. For certain types of natural valuation functions, our condition is also necessary. Our condition extends both the Mirrlees-Spence condition [25, 17], applicable only for differentiable real allocations, and Archer and Tardos' single parameter characterization [4], which assumes an agent's valuation is linear in its type.

We demonstrate the simplicity of testing our condition by showing that classical criteria for truthfulness in combinatorial problems such as auctions and machine scheduling can be derived from our condition. In addition, we use our condition to derive results for new single parameter problems, which have not been previously analyzed.

We also consider combinatorial problems where the true types of agents affect the valuation of each other, such as in machine scheduling with selfish jobs. In such cases there are only degenerate dominant strategy mechanisms. We show that the same condition can be used to design mechanisms which are ex-post truthful, meaning that the outcome where all agents cooperate and report their true type is a Nash equilibrium. We demonstrate the power of this condition by applying it on the problem of machine scheduling with strategic job owners, previously presented in [5]. We give a constant approximation ratio algorithms for the original problem and to the double setting where both jobs and machines are strategic.

*Partially supported by a grant from the Israel Science Foundation.

†The work was done while the author was a fellow in the Institute of Advance studies, Hebrew University. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, by a grant no. 1079/04 from the Israel Science Foundation and an IBM faculty award. This publication only reflects the authors' views.

EC'06, June 11–15, 2006, Ann Arbor, Michigan.

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Nonnumerical Algorithms and Problems; G.2.1 [**Descrete Mathematics**]: Mathematics of Computing

General Terms

Algorithms, Theory, Economics

Keywords

Game theory, Algorithmic mechanism design, Approximation Algorithms

1. INTRODUCTION

1.1 Background

At the core of Mechanism Design is the desire to define a system such that selfish agents interacting with the system would reach the desired outcome. Example of such setting is a centralized decision maker with regulatory power (government, employer, etc.) that wishes to optimize an objective function by aggregating signals on private information collected from agents (buyers, sellers, workers, etc.). The private information of an agent, which is also referred as its type, may include its demand curve, income reports, potential labor power, etc. The information (e.g., bids) that the agents submit signals their types, however they are expected to strategically choose their signal such that they will affect the decision making in a manner that will optimize their own utility.

Mechanism design aims to give the agents an incentive to simply reveal their information to the decision maker, and avoid strategic signaling. This goal is achieved by assigning payments from or to the mechanism, which encourage the agents' cooperation by making truthtelling a (weakly) dominant strategy. I.e., for each agent, regardless of the other agents' actions, reporting its type is a best response strategy. Settings where each agent has a dominant strategy are more convenient to the central decision maker, since it is likely that all agents will indeed follow their dominant strategies, and this frees the central decision maker from considering the implications of strategic behaviors of the agents. By the revelation principle, if dominant strategy mechanisms exist, it is sufficient to consider mechanisms where truthtelling is the dominant strategy (see, e.g., [16]). Under the assumption that each agent is aware of its own type, truthtelling is the simplest strategy for an agent to follow, which motivates making it a dominant strategy.

Algorithmic mechanism design [19] extends mechanism design into the realm of discrete algorithms, by considering computational limitations on the algorithmic process that calculates the decision rule. Much of the algorithmic mechanism design literature is devoted to studying a wide range of different utility and objective functions and design appropriate efficient mechanisms for them.

One can hope that rather than verifying that a given algorithm and a payment scheme construct together a truthful mechanism, we would rather have a criteria for verifying whether a given algorithm can be melded into a truthful mechanism by adding a proper payment scheme (i.e., if it is rationalizable). Such criteria was given by Rochet in [22]. Additionally, if a decision rule is rationalizable, there is a mathematical formulation to the suitable payment functions. The drawback of Rochet's characterization of rationalizable decision rules is that it does not provide a computationally practical method for testing rationalizability. A simpler condition, referred as *weak monotonicity* [14] or as *2-cycle inequality* [10], is a necessary condition for rationalizability, but not a sufficient condition, as demonstrated in [14].

Given a restricted domain of single parameter agents, we search for a sufficient condition, which is not a necessary one in general, although in some settings we can show it is also necessary. On one hand, we aim for a condition that is simple enough to be easily applied to test rationalizability. On the other hand, the condition should be general enough to be useful in constructing truthful mechanisms. We focus on mechanisms for single parameter agents, which although restricted, include a wide range of mechanisms, and weak monotonicity alone does not imply rationalizability by itself.

In the case that the output of the mechanism is continuous, the *Mirrlees-Spence condition* [25, 17] is a simple necessary and sufficient condition that characterizes all truthful single parameter mechanisms. The requirement that the output space is continuous is a severe limitation. In algorithmic mechanism design, the outcome space for many problems is discrete. (Example of such problems include auctions, machine scheduling, routing, etc.) In such settings the Mirrlees-Spence condition is inapplicable, since in discrete spaces the preliminary requirements required for applying this condition immediately fail. Currently, algorithmic mechanism design requires deriving rationalizability conditions for each model, separately. A replacement to the Mirrlees-Spence condition that holds for discrete models would simplify this process in single parameter setups.

Archer and Tardos [4] suggested a simple characterization for single parameter agents, where the valuation (or cost) of an agent is linear with respect to its type, and therefore the type can be considered as representing a cost (or value) per unit. Unlike the Mirrlees-Spence condition, this characterization does not require the allocation function to be continuous, only monotone in the type. However, it significantly limits the structure of the valuation function.

1.2 Our Results

We present a generalization to both the Mirrlees-Spence condition, and Archer and Tardos' characterization, which relaxes prior assumptions on the structure of the valuation functions and the decision rule. We define a property named *Halfway Monotone Derivative (HMD)*, prove that all HMD algorithms are rationalizable and characterize the structure of the payment function. We also prove that for some valuation functions, HMD is also a necessary condition.

We apply our condition on several sample algorithmic problems and derive simple conditions for truthfulness, as well as simple structures of the payment functions. Specifically, we show that in all single commodity auctions, the critical value condition is equivalent to the HMD condition. We show that in machine scheduling problems where agents (machines) have a cost per unit of work function, the monotonicity condition of Archer and Tardos [4] is an outcome of the HMD condition. We define a setting where an agent's utility from getting service before some deadline is proportional to the time before the deadline that the service was given, and analyze conditions for rationalizability. We analyze a single parameter version of an auction with limit constraints [2] and give a piecewise monotonicity condition for rationalizability. We then present rationalizable approximation algorithms for this problem.

We then extend our work to models where the valuation of each agent is also affected by the true types of the other agents. Since it is not possible to achieve a reasonable truthful implementation in dominant strategies, we settle with an *ex-post truthful* implementation, where there is a Nash equilibrium where all agents reveal their true type. We show that the HMD condition holds for this model, and then apply it on the problem of scheduling on related machines with strategic job owners, which is a dual problem to scheduling on strategic related machines [4]. We show that the monotonicity condition in [5] can be derived using HMD, and give a 2 approximation which is ex-post truthful, the first constant approximation ratio for the problem. Finally, we consider the problem where both machines and jobs are strategic, and show that the 5 approximation in [1] is ex-post truthful for this problem.

1.3 Related Work

Provided that each agent's valuation function depends only on its type and the outcome, a truthful mechanism for any optimization problem where the objective function of the algorithm is to maximize the sum of utilities (maximize the social welfare) is the *VCG* mechanism [7, 9, 26], which can also be generalized to any maximization problem of an affine function. However, for other objective functions VCG is not applicable. Additionally, optimally solving an affine maximization problem may be computationally hard, which motivates developing truthful mechanisms that only approximate the optimal solution. However, the VCG mechanism does not cope well with approximate solutions [20].

If the valuation functions are unrestricted and there are at least 3 possible outcomes, Roberts [21] proved that a mechanism is truthful is and only if it is a weighted VCG mechanism. In particular, the allocation must maximize an affine function of the valuations.

Given that the valuation function of the agents is taken from some restricted domain, several papers have attempted to characterize function spaces for which weak monotonicity would also be a sufficient condition for truthfulness [10, 14]. Saks and Yu [23] have generalized these results to any finite outcome space and valuation functions defined on convex domains (in contrast, we do not require the outcome space to be finite).

Although weak monotonicity is a simple condition, in practice it is not always easy to prove for given a given algorithm. We believe that a weaker sufficient yet not necessary condition may be easier to prove, yet most practical algorithms will not stumble into the pitfall of rationalizable algorithms that this condition does not hold for them.

2. PRELIMINARIES

2.1 Mechanisms

The system consists of a decision rule (an algorithm) and n agents. Each agent submits a bid (signal) $b_i \in T$, and then an outcome $\omega \in \Omega$ is calculated by an algorithm $A(b)$, where b denotes the bid vector.

The bid vector without the i-th bid is denoted by b_{-i}. Additionally, (β, b_{-i}) denotes the bid vector b with the i-th bid replaced with β. When it is clear from the context that A and b_{-i} are fixed, we shall let $\omega_{b_i} = A(b_i, b_{-i})$ denote the outcome when agent i bids b_i.

DEFINITION 2.1. A decision rule *is a function* $A : T^n \mapsto \Omega$ *that given a vector* b *of* n *bids returns an outcome* $\omega \in \Omega$. A payment scheme P *is a set of payment functions* $P_i : \Omega \times T^n \mapsto \mathbb{R}$, *where* P_i *determines the payment of agent* i *to the mechanism, given the output* ω *and the bid vector* b. A mechanism $M = (A, P)$ *is a combination of a decision rule* A *and a payment scheme* P.

We note that payments may be nagative, meaning that the mechanism pays the agents.

2.2 Utilities

Each agent has a private value $t_i \in T_i$, which is called its *type*. To simplify the notation, we will assume that all agents have their types taken from the same space as the bids (signaling) space, T.

Each agent has a valuation function $v_i : \Omega \times T \mapsto \mathbb{R}$ that reflects the utility from an outcome $\omega \in \Omega$, given that the type of the agent is t_i. The agents have a *quasi-linear* utility, meaning that their utility can be shifted linearly by monetary payments. Therefore, the utility of agent i from an outcome ω and a payment P_i is $u_i(\omega, t_i, P_i) = v_i(\omega, t_i) - P_i$.

In some cases we will be interested in the partial derivative of a valuation function by the agent's type. Therefore, for simplicity, we denote $v_i' = \frac{\partial v_i}{\partial t_i}$.

2.3 Truthfulness

For truthful mechanisms we will concentrate on payment functions of the form $P_i : \Omega \times T^{n-1} \mapsto \mathbb{R}$, which don't depend on the i-th bid, because it is easy to see that truthful mechanisms must have payments of this form.

DEFINITION 2.2. *Algorithm* A admits a truthful payment *if there exists a payment scheme* P *such that for any set of fixed bids* b_{-i}, *and for any two types* $s, t \in T$

$$v_i(\omega_t, t) - P_i(t, b_{-i}) \geq v_i(\omega_s, t) - P_i(s, b_{-i}) \quad (1)$$

In this case, A *is also called* rationalizable*, and the mechanism* $M = (A, P)$ *is called* truthful.

In words, a mechanism is truthful if for every agent, reporting its true type as a bid is a weakly dominant strategy. A mechanism is *strongly truthful* if truthtelling is the only weakly dominant strategy.

If an algorithm A is randomized, then it can be viewed as a distribution over deterministic algorithms. Therefore, a randomized mechanism $M = (A, P)$, which is a distribution over truthful mechanisms is also truthful (this property is also referred as *universal truthfulness*). A weaker notion of truthfulness is to carry the randomization into the mechanism, and require truthtelling to always be a best response strategy in expectation over the random choices of the algorithm. A mechanism of this type is *truthful in expectation*.

Rochet [22] presented a necessary and sufficient condition for rationalizable decision rules. We refer to a slightly different presentation of this condition, which appears in [10]:

THEOREM 2.3. *[22, 10] Given an agent* i *and having all other bids* b_{-i} *held fixed, let* $G(i, b_{-i}) = (V, E)$ *be a weighted directed graph such that* $V = T$, $E = T \times T$ *and the weight of every edge is* $w(s, t) = v_i(\omega_t, t) - v_i(\omega_s, t)$. *An allocation algorithm admits a truthful payment if and only if for every agent* i *and for every vector of fixed bids* b_{-i}, *the graph* $G(i, b_{-i})$ *has no finite negative cycles.*

In addition to Theorem 2.3, if an decision rule is rationalizable, the following is a suitable payment function for the i-th agent: For every vector of fixed bids b_{-i}, choose an arbitrary type t_0. The payment from agent i to the mechanism if it bids t is:

$$p(t, b_{-i}) = \inf \left\{ \sum_{j=0}^{k} w(t_j, t_{j+1}) | k \geq 0, \quad \begin{matrix} t_1, \ldots, t_{k+1} \in T \\ t_{k+1} = t \end{matrix} \right\} \quad (2)$$

In words, the payment function is the infimum on all finite paths from an arbitrary type t_0 to the actual bid t.

Theorem 2.3 does not provide an efficient computational method for testing whether an algorithm is rationalizable. A simpler condition, which should be easier to test, is whether the graph contains a negative cycle of length 2. Formally, a graph $G(i, b_{-i})$ does not have negative cycles of length 2 if and only if for every two types $t, s \in T$,

$$v_i(\omega_t, t) - v_i(\omega_s, t) \geq v_i(\omega_t, s) - v_i(\omega_s, s) \quad (3)$$

This necessary (but not sufficient) condition is referred as *weak monotonicity* [14] or as *2-cycle inequality* [10].

2.4 Single Parameter

DEFINITION 2.4. *An agent* i *is a* single parameter *agent with respect to* Ω *if there exists an interval* $S_i \subseteq \mathbb{R}$ *and a bijective transformation* $r_i : T \mapsto S_i$ *such that for any* $\omega \in \Omega$, *the function* $\hat{v}_i(\omega, s_i)$ *is continuous and differentiable almost everywhere in* s_i, *where* $\hat{v}_i(\omega, s_i) = v_i(\omega, {r_i}^{-1}(s_i))$.

The purpose of $r_i(\cdot)$ is to overcome the difficulty of having different representations for the same type space. The single parameter property should be indifferent to the chosen representation. For simplicity, we shall assume the type representation allows ignoring $r_i(\cdot)$. For example, if the type space is $\mathbb{R}$ itself, or an interval $T = [t_0, t_1]$, and the utility function is continuous and differentiable almost everywhere in T for every $\omega \in \Omega$, then $r_i(\cdot)$ is the identity function and can be ignored. We therefore, slightly abuse the notation and assume $v_i = \hat{v}_i$.

3. HALFWAY MONOTONE DERIVATIVE

The *Mirrlees-Spence condition* [25, 17, 22] can be used to characterize all truthful single parameter mechanisms, where the notion of single parameter is more restricted than the one we use. Assume the output space Ω is a continuous interval in $\mathbb{R}$, the type space T is a segment $T = [t_0, t_1]$. The condition requires that v_i is twice differentiable, and that

$$\forall \omega \in \Omega, t \in T \quad \frac{\partial^2}{\partial t \partial \omega} v_i(\omega, t) > 0 \tag{4}$$

then the mechanism is truthful if and only if ω is non-decreasing in the bid b_i of the agent.

Our condition is a generalization of this notion of the Mirrlees-Spence condition, as it relaxes the assumptions on Ω and the differentiability of the valuation and decision functions. It is a sufficient but not necessary condition, which we call *Halfway Monotone Derivative (HMD)* condition.

DEFINITION 3.1. *A valuation function v_i satisfies* Halfway Monotone Derivative *(HMD) condition with respect to a given decision rule if for every fixed bid vector b_{-i},* one of the following *holds:*

1. *For every two types $s, t \in T$ such that $s < t$, $\forall u \geq s$ we have, $v_i'(\omega_s, u) \leq v_i'(\omega_t, u)$, except for a set of measure zero.*
2. *For every two types $s, t \in T$ such that $s < t$, $\forall u \leq t$ we have, $v_i'(\omega_s, u) \leq v_i'(\omega_t, u)$, except for a set of measure zero.*

The set of points where the inequality may not hold is required since the valuation function may not be differentiable everywhere. Note that by integrating the derivatives in the definition over $[s, t]$, HMD also implies weak monotonicity.

The following condition gives a characterization to a family of rationalizable decision rules.

THEOREM 3.2. *A single parameter decision rule $A(b)$: $T^n \mapsto \Omega$ is rationalizable when all valuation functions are HMD.*

Proof: We shall prove only that the first condition of HMD leads to rationalizability, as the proof for the second condition is analogous. Assume by way of contradiction that the condition holds yet A is not rationalizable. By Theorem 2.3 there is a some graph $G(i, b_{-i})$ which has a negative cycle $t_0, t_1, \ldots, t_k, t_{k+1} = t_0$. We shall first show that the graph must also contain a negative 2-cycle, and later infer that the condition is violated. If $k = 1$ then clearly a negative 2-cycle exists. Otherwise, let t be the node in the cycle such that $\forall 0 \leq i \leq k,\ t \leq t_i$, and let s and u be the nodes in the cycle which have cycle arcs in and from t, respectively. Since t is the node of minimum value in the cycle, we have $t \leq u$ and $t \leq s$. The length of the path from s to u through t is:

$$\begin{aligned} & w(s,t) + w(t,u) \\ & = v_i(\omega_t, t) - v_i(\omega_s, t) + v_i(\omega_u, u) - v_i(\omega_t, u) \\ & = v_i(\omega_s, u) - v_i(\omega_s, t) - v_i(\omega_t, u) + v_i(\omega_t, t) \\ & \quad - v_i(\omega_s, u) + v_i(\omega_u, u) \\ & = \int_t^u v_i'(\omega_s, x)dx - \int_t^u v_i'(\omega_t, x)dx \\ & \quad - v_i(\omega_s, u) + v_i(\omega_u, u) \\ & = \int_t^u \left(v_i'(\omega_s, x) - v_i'(\omega_t, x)\right) dx + w(s,u) \geq w(s,u) \end{aligned}$$

The last integral is non-negative since $t \leq u$ and since $v_i'(\omega_s, x) \geq v_i'(\omega_t, x)$ for every $x \geq t$ (in particular, $t \leq x \leq u$), due to the first HMD condition. From the inequality we infer that a shorter negative cycle with $k-1$ nodes can be constructed with a shortcut from s to u. By induction, we get that the graph has a negative 2-cycle. Let s and u be the nodes in this cycle, and assume without loss of generality that $s < u$. We infer from HMD, that:

$$\begin{aligned} w(s,u) \quad + \quad & w(u,s) \\ = \quad & v_i(\omega_u, u) - v_i(\omega_s, u) + v_i(\omega_s, s) - v_i(\omega_u, s) \\ = \quad & \int_s^u v_i'(\omega_u, x)dx - \int_s^u v_i'(\omega_s, x)dx \\ = \quad & \int_s^u (v_i'(\omega_u, x) - v_i'(\omega_s, x))dx \geq 0, \end{aligned}$$

which contradicts the cycle being of negative length. □

For some types of valuation functions, HMD is also a necessary condition. The following Theorem proves this property for a simple case where the partial derivative is a constant in the agent's type (proof omitted).

THEOREM 3.3. *If for every agent i, fixed bid vector b_{-i}, and bid b_i, $v_i'(\omega_{b_i}, x)$ does not depend on x, then HMD is a necessary and sufficient condition for rationalizability.*

In Sections 4.1 and 4.2 we demonstrate the importance of Theorem 3.3 by showing that classical truthfulness results are special cases of the theorem. In contrast to this result, we give in Section 4.4.3 an example of a truthful mechanism where neither HMD condition holds.

We now show a simple structure for the payment function in truthful HMD mechanisms:

THEOREM 3.4. *A suitable payment scheme for agent i in a single parameter rationalizable decision rule $A : T^n \mapsto \Omega$ that is HMD is*

$$P_i(t, b_{-i}) = c(b_{-i}) + v_i(\omega_t, t) - \int_{t_0}^t v_i'(\omega_x, x)dx \tag{5}$$

Where b_{-i} is held fixed, $t_0 \in T$ is an arbitrary type and c is an arbitrary function of b_{-i}.

Proof: The following proof holds for the first HMD condition. The proof for the second HMD condition is analogous. We first prove the Theorem for $t_0 = \inf T$, and later extend the result for any arbitrary $t_0 \in T$. For now we assume that $\inf T \in T$. At the end of the proof we refer to the case where $\inf T \notin T$.

By Theorem 2.3, a suitable payment function is

$$P_i(t, b_{-i}) = \inf \left\{ \sum_{j=0}^k w(t_j, t_{j+1}) | k \geq 0, \quad \begin{matrix} t_1, \ldots, t_{k+1} \in T \\ t_{k+1} = t \end{matrix} \right\} \tag{6}$$

We call an arc (t_j, t_{j+1}) a *forward arc* if $t_j < t_{j+1}$ and a *backward arc* if $t_j > t_{j+1}$. We first prove that within a path, adding an intermediate node inside a forward arc cannot make it longer. Let (t_j, t_{j+1}) be a forward arc in a given path. Let $s \in [t_j, t_{j+1}]$ be an intermediate node. We have that

$$\begin{aligned}
&w(t_j, s) + w(s, t_{j+1}) \\
&= v_i(\omega_s, s) - v_i(\omega_{t_j}, s) + v_i(\omega_{t_{j+1}}, t_{j+1}) - v_i(\omega_s, t_{j+1}) \\
&= -\int_s^{t_{j+1}} v_i'(\omega_s, x)dx + \int_s^{t_{j+1}} v_i'(\omega_{t_j}, x)dx \\
&\quad + v_i(\omega_{t_{j+1}}, t_{j+1}) - v_i(\omega_{t_j}, t_{j+1}) \\
&= \int_s^{t_{j+1}} \left(-v_i'(\omega_s, x) + v_i'(\omega_{t_j}, x)\right) dx + w(t_j, t_{j+1}) \\
&\leq w(t_j, t_{j+1})
\end{aligned}$$

The last inequality is because the integrated expression is negative within the integral's range, due to the first HMD condition. For the first forward arc leaving the interval $[t_0, t]$ (if such an arc exists) we add an intermediate node at t. The resulting path is not longer, and all the nodes outside the interval $[t_0, t]$ are contained in a cycle, starting and ending at t (as the entire path ends at t). Since there are no negative cycles in the graph, we can remove the cycle and end with a shorter path that uses only nodes within the interval $[t_0, t]$.

In a similar technique we remove all the backward arcs (t_j, t_{j+1}) such that $t_{j+1} < t_j$. Let $l > j$ be the index of the first node t_l that appears after t_{j+1} such that $t_l \geq t_j$. The arc that ends at t_j must be a forward arc (i.e. $t_{l-1} < t_l$), since by definition of l, $t_{l-1} \leq t_j$. By adding t_j as an intermediate node within the arc (t_{l-1}, t_l), we get a cycle from t_j to itself, which can be removed. Removing the cycle also removes the backward arc (t_j, t_{j+1}).

We now have that the payment is the infimum on paths from t_0 to t that only use forward arcs. By adding intermediate points to all arcs, we get that

$$\begin{aligned}
&P_i(t, b_{-i}) = \lim_{\Delta \to 0} \sum_{j=t_0/\Delta}^{t/\Delta} w(j\Delta, (j+1)\Delta) \\
&= \lim_{\Delta \to 0} \sum_{j=t_0/\Delta}^{t/\Delta} \left(v_i(\omega_{(j+1)\Delta}, (j+1)\Delta) - v_i(\omega_{(j+1)\Delta}, j\Delta)\right) \\
&= v_i(\omega_t, t) - v_i(\omega_{t_0}, t_0) \\
&\quad - \lim_{\Delta \to 0} \sum_{j=t_0/\Delta}^{t/\Delta} \left(v_i(\omega_{(j+1)\Delta}, j\Delta) - v_i(\omega_{j\Delta}, j\Delta)\right) \\
&= v_i(\omega_t, t) - v_i(\omega_{t_0}, t_0) \\
&\quad - \lim_{\Delta \to 0} \sum_{j=t_0/\Delta}^{t/\Delta} \int_{j\Delta}^{(j+1)\Delta} v_i'(\omega_{j\Delta}, x)dx \\
&= v_i(\omega_t, t) - v_i(\omega_{t_0}, t_0) - \int_{t_0}^{t} v'(\omega_x, x)dx
\end{aligned}$$

Since $v_i(\omega_{t_0}, t_0)$ does not depend on t it can be replaced by any other function $c(b_{-i})$ without affecting the truthfulness. Additionally, the integral can be changed to $\int_{\hat{t}}^{t} v'(\omega_x, x)dx$, where $\hat{t}$ is an arbitrary type, since this change only subtracts $\int_{t_0}^{\hat{t}} v'(\omega_x, x)dx$ from the payment, which does not depend on t and therefore is part of $c(b_{-i})$.

If $t_0 = \inf T \notin T$, we can naturally add t_0 to T by setting $v_i(\omega, t_0) = \lim_{t \to t_0} v_i(\omega, t)$ and $v_i(\omega_{t_0}, s) = \lim_{t \to t_0} v_i(\omega_t, s)$. If these limits are not well defined we can settle for any arbitrary values. Since t_0 is not a legal bid nor a legal type, these values affect neither the rationalizability of the algorithm, nor the value of the payment function. □

We describe a large class of cases where the integral in Theorem 3.4 reduces into an alternative representation that may be more convenient to implement. In many combinatorial problems the outcome space Ω is finite or countable, and combinatorial algorithms have the following property: For each agent i, when b_{-i} is fixed, the type space T can be divided into a finite or countable number of intervals, such that for each interval, the algorithm outputs the same outcome for any bid within the interval. We denote such algorithms as *piecewise continuous*. Let $T_{b_{-i}}$ denote the set of endpoints of these intervals. For any endpoint $s \in T_{b_{-i}}$, let s^- and s^+ denote the intervals whose upper and lower endpoints are s, respectively. Clearly, for any endpoint s, either $s \in s^-$ or $s \in s^+$. Given an interval $[t_1, t_2]$, $T_{b_{-i}} \cap [t_1, t_2]$ denotes the set of endpoints whose surrounding intervals intersect with $[t_1, t_2]$ (i.e., all the endpoints within (t_1, t_2), and possibly t_1 and t_2, if they are endpoints and $t_1 \in t_1^+$, $t_2 \in t_2^-$).

COROLLARY 3.5. *A suitable payment scheme for agent i in a single parameter piecewise continuous rationalizable decision rule $A : T^n \mapsto \Omega$ that is HMD is*

$$P_i(t, b_{-i}) = c(b_{-i}) + \begin{cases} \sum_{s \in T_{b_{-i}} \cap [t_0, t]} v_i(\omega_{s^+}, s) - v_i(\omega_{s^-}, s) & t > t_0 \\ \sum_{s \in T_{b_{-i}} \cap [t, t_0]} v_i(\omega_{s^-}, s) - v_i(\omega_{s^+}, s) & t < t_0 \\ 0 & t = t_0 \end{cases}, \quad (7)$$

where b_{-i} is held fixed, $t_0 \in T$ is an arbitrary type.

In words, the non constant part of the payment is given by accumulating the values of the transitions between two consecutive outcomes, over all the endpoints between t_0 and t.

Example: To illustrate our payment function consider the following case for player i, when all the other bids b_{-i} are fixed. If $b_i = x \in \mathbb{R}^+$ then $A(x, b_{-i}) == \omega_{\lceil x \rceil}$. The valuation is $v_i(\omega_{\lceil x \rceil}, t) = t + \lceil x \rceil$. Choosing $t_0 = 0$ and $c(b_{-i}) = 0$, the payment in this case would be $P_i(x) = \lceil x \rceil$. If we change the valuation function to $v_i(\omega_{\lceil x \rceil}, t) = t \cdot \lceil x \rceil$ then the payment would be $P_i(x) = \sum_{k=1}^{\lceil x \rceil - 1} k = O(x^2)$.

4. HMD APPLICATIONS

We show applications of the HMD condition for several single parameter mechanisms. In Sections 4.1 and 4.2 we rederive results from [4, 3] by applying the HMD condition to some well known mechanisms, and showing that conditions derived for these mechanisms are actually special cases of Theorem 3.3. In the rest of this section we present new single parameter mechanisms and demonstrate the usage of the HMD condition for them (due to space constraints several proofs were omitted)

4.1 Single Commodity Auction

For simplicity, we concentrate on single commodity auctions, where each agent has a unit demand. However, the results hold for a more general setup of an auction with known single minded bidders [18], which is a multi commodity combinatorial auction where each agent is interested in a specific bundle of commodities, which is publicly known.

In an auction, an agent's type is a private value t_i, which is the value the agent associates with a good that it is bidding on. From the point of view of a specific agent, there are two possible outcomes: winning and losing. When winning, the value of the agent is t_i, and in the losing outcome, the value is 0.

The second price auction [26] is a classical method for auctioning a single commodity: The highest bidder wins and pays the bid of the second highest bid. Although it is a popular example, it is not the only truthful mechanism for auctioning.

A well known result on rationalizable deterministic auctions is the existence of a *critical value* for each bidder, unless the bidder has no winning bid. A bidder's critical value is determined by the bids of the other agents. The bidder wins if its bid is above the critical value and loses if the bid is under it. Given that losing bidders pay 0, the winning bidders pay exactly their critical value. The second price auction is truthful since the critical value of each bidder is the highest bid among the other bidders.

The HMD condition is equivalent to the critical value condition in auctions, and extends to randomized auctions:

CLAIM 4.1. *In deterministic auctions the critical value condition is equivalent to HMD.*

In randomized auctions, the relevant outcome for an agent is a probability $0 \le p \le 1$ for winning the good. The randomized auctioning can be viewed as a distribution over deterministic auctions. If for each and every one of these deterministic auctions we have a critical value, then we have a universal truthfulness. Otherwise, we can settle for truthfulness in expectation. In this case, the following holds:

CLAIM 4.2. *The following are equivalent:*

1. *A randomized auction is rationalizable ,*
2. *each bidder's probability for winning is weakly monotone in its bid,*
3. *HMD conditions hold.*

4.2 Machine Scheduling

A central mechanism assigns n jobs to m machines, with the goal of minimizing the *makespan*, which is the longest completion time (also known as $Q\|C_{max}$). Machines are related by speeds $s_1, \ldots, s_m$, and the jobs have processing requirements $p_1, \ldots, p_n$. Therefore, running the i-th job on the j-th machine requires $\frac{p_i}{s_j}$ time. It is sometimes more convenient to refer to a machine's cost per unit of work $c_j = \frac{1}{s_j}$ rather than to its speed.

If all speeds and weights are known to the mechanism, then it is known that the optimization problem is NP-Complete [8], however there is a $PTAS$ to this problem [11]. Since the problem is strongly NP hard, this is the best approximation result possible, unless $P = NP$. In case the number of machines is constant then there exists an $FPTAS$ to the problem [12].

4.2.1 Mechanism Design for Machine Scheduling

A machine j incurs a cost proportional to the amount of work assigned to it, i.e. $L_j = (\sum_{i \in I_j} p_i)c_j$, where I_j is the set of jobs assigned to machine j. If the machines' speeds are only known privately to the owners of the machines, then the owners (agents) may report false values to the mechanism in order to lower their costs.

Archer and Tardos [4] have shown that a scheduling algorithm is rationalizable if and only if it is monotone. Monotonicity means the amount of work assigned to a machine cannot decrease if it raises its speed, while the rest of the inputs remain constant. Although traditional algorithms for job assignment are not monotone, several monotone algorithms have been given to this problem.

There is a mechanism that solves the optimization problem optimally and is monotone. Obviously, its running time is not polynomial, unless $P = NP$. However, there is a 2-approximation polynomial time randomized mechanism that is monotone in expectation, and therefore achieves truthfulness in expectation [3] . There is also a 3-approximation polynomial time deterministic mechanism that is truthful [13], improving a previously known 5-approximation [1]. If the number of machines is constant, there is a monotone $FPTAS$ [1] .

The monotonicity condition holds for any problem where the agents have a cost function which is linear in the amount of work assigned to them. This has been demonstrated in [4] for strategic versions of various allocation problems such as maximal flow and facility location. We show that the monotonicity condition is equivalent to HMD.

CLAIM 4.3. *A scheduling algorithm is monotone iff it is HMD.*

By Theorem 3.4, when choosing $t_0 = \infty$ (or any large enough type that assures a zero assignment) and $c = 0$, the payment from the machine to the mechanism is

$$\begin{aligned} P_j(b_j, b_{-j}) &= -b_j A_j(b_j, b_{-j}) - \int_{\infty}^{b_j} [-A_j(x, b_{-j})]dx \\ &= -b_j A_j(b_j, b_{-j}) - \int_{b_j}^{\infty} A_j(x, b_{-j})dx \qquad (8) \end{aligned}$$

where $A_j(x, b_{-j})$ is the amount of work assigned to machine j when the bid vector is (x, b_{-j}). Since the payment is negative, the mechanism actually pays the machine $-P_j$, which is exactly the payment function derived in [4].

The natural dual problem for strategic machine scheduling is scheduling with strategic job owners, which is discussed in Section 5.

4.3 Scheduling with Deadlines

The following problem is an example for single parameter agents, where the derivative of the valuation function is not a constant (but still simple): n agents apply to get service from a central mechanism. An agent's type is a deadline $t_i \in \mathbb{R}^+$, which it must be served by to have a positive valuation. The output is a service time $\omega_i \in \mathbb{R}^+ \cup \{\infty\}$. The valuation function of an agent is $v_i(\omega_i, t_i) = \max\{0, t_i - \omega_i\}$, i.e., it linearly decreases until it drops to 0 at the deadline, and then remains fixed. The infinity outcome represents the case where an agent is never served.

Since the HMD condition relates only to the valuation function, we don't consider here the objective function of the mechanism, or the set of feasible outcomes, which are needed for the algorithmic design, but settle on deriving the rationalizability condition:

THEOREM 4.4. *Given that a server never serves an agent after its declared deadline, then it is rationalizable if and* *only if for each agent, either* $\omega_{b_i} = \infty$ *for every* b_i*, or it has a critical time* c_i *such that if* $b_i < c_i$ *then* $\omega_{b_i} = \infty$*, and if* $b_i > c_i$ *then* $\omega_{b_i} < c_i$*.*

The assumption that an agent is never served after its deadline simplifies the condition for rationalizability. It is also a reasonable assumption since a cooperative agent is indifferent between late service and denial of service.

4.4 Auctions with Limit Constraints

A *limit constraint* [1] is a special case of submodular [15] valuations. Given n items and m agents (bidders), p_{ij} denotes the valuation of the i-th agent for the j-th item. Each agent has also a limit constraint t_i. Given a bundle of items I, the valuation of the i-th agent of the bundle is $v_i(I, t_i) = \min\{t_i, \sum_{j\in I} p_{ij}\}$. For simplicity we assume that $\max_j\{p_{ij}\} \leq t_i \leq \sum_j p_{ij}$. We also assume that the allocation algorithm does not have to allocate all of the items. We define the objective function of the mechanism to be the total valuation of all agents.

This optimization problem is known to be NP-Complete, but has several approximations: A simple greedy algorithm gives a 2-approximation for a wider class of valuations (submodular valuations) [15], an LP-rounding gives a 1.58 approximation, and there is a PTAS when the number of bidders is constant [2]. All of these algorithms assume that the bidders are non strategic.

4.4.1 Strategic Limits

The following is a single parameter problem: Assume that all valuations are known, but the limits are privately held by the agents. Although this is a somewhat artificial setting, it demonstrates the HMD condition well: We will show that the valuation function is very similar to the valuation function in Section 4.3, and the rationalizability conditions are different.

DEFINITION 4.5. *An allocation scheme for auctions with limit constraints is* piecewise monotone *if for every agent* i *and every limit* t_0 *such that* $v_i(\omega_{t_0}, t_0) = t_0$*, then for every* $t_1 > t_0$*,* $\omega_{t_1} \geq \omega_{t_0}$*.*

In words, whenever the value of the allocation meets or exceeds the limit, it sets a lower bound on the value of the allocation for higher bids. However, in between these points, the allocation need not be monotone.

THEOREM 4.6. *Any piecewise monotone allocation rule is rationalizable.*

Proof: From the point of view of the i-th agent, the exact allocation is not important, only the total value of items assigned to him (denoted by ω), and the limit constraint. Having ω fixed, $v_i(\omega, t_i) = t_i$ if $t_i < \omega$ and $v_i(\omega, t_i) = \omega$ otherwise. The derivative is therefore 1 if $t_i < \omega$ and 0 otherwise.

We now show that piecewise monotonicity leads to the first HMD condition: Given a limit constraint b_0, first assume that the outcome ω_0 does not exceed b_0 in its total value. For any type $t > b_0$, $v'_i(\omega_0, t) = 0$, so b_0 does not induce any constraints on the outcomes for larger bids. If ω_0 exceeds b_0 in its total value, then the derivative is 1 until ω_0, therefore to fulfill the HMD condition, for any type $t > b_0$ the outcome must be at least ω_0. This is achieved due to piecewise monotonicity, and therefore we also have the first HMD condition, which by Theorem 3.2 is a sufficient condition for rationalizability. □

Piecewise monotonicity also has a dual version that leads to the second HMD condition: Let ω_0 be the value of the allocation resulting from bidding a limit constraint of b_0. If $\omega_0 < b_0$ then the dual piecewise monotonicity condition requires that for any $b_1 < b_0$, the value of the outcome ω_1 will fulfill $\omega_1 \leq \omega_0$. If ω_0 exceeds the limit constraint, there are no additional requirements. We omit the proof that this property implies the second HMD condition.

Since finding the optimal allocation is NP-Hard, we show that there are piecewise monotone (hence, rationalizable) allocation algorithms that output approximate allocations. The approximation algorithms for the non-strategic form of the problem [2, 15] are not piecewise monotone. The next subsection presents approximation algorithms that are also examples of the following two special subcases of piecewise monotone algorithms:

COROLLARY 4.7. *Any allocation algorithm that never exceeds the limit constraint is rationalizable.*

COROLLARY 4.8. *Any allocation algorithm for which the total value of items it allocated to each agent is weakly monotone increasing in its limit constraint is rationalizable.*

4.4.2 Piecewise Monotone Algorithms

Under the restriction of never exceeding the limit constraint, the allocation problem reduces to a subcase of the MAX-GAP (Generalized Assignment Problem). MAX-GAP is a generalization of the Multiple Knapsack problem [6], where each item has a size and profit, and the items are packed into bins (knapsacks) of limited capacity. The goal is to maximize the total profit of the packed items, under the capacity constraints. In the MAX-GAP problem, each item has a different size and profit for each bin. MAX-GAP is NP-Hard to solve, but there is a 2-approximation, based on a reduction to MIN-GAP [24], which is a similar problem where items have a cost instead of a profit. An instance to an auction with limit constraints is reduced to a MAX-GAP instance by having a bin for each agent, with a capacity equal to the limit constraint of that agent. The value of an assigning the j-th item to the i-th item is both the size and the profit of packing the j-th item in the i-th bin. The 2-approximation of the MAX-GAP problem is preserved under the auction with limit constraint problem, since the approximation is in comparison to a fractional assignment, which is optimal in both problems. By Corollary 4.7, this approximation algorithm is also rationalizable. Choosing that an agent with an empty assignment won't pay the mechanism, the payment function in Corollary 3.5 reduces to paying the exact value of the allocated items.

It may seem at first that the greedy algorithm [15, 2] that greedily allocates each item to the highest bidder (with respect to the valuations given the current partial allocation),

[1]In [15, 2] the class of limited valuations is called budget constrained, or budget limited. In non-strategic settings, maximizing valuations can be equivalently restated as a problem of maximizing payments. However, in strategic models, payments are different than valuations. The usage of the term 'budget' for these valuations is misleading in the strategic context and therefore was altered here

Input: a matrix $\{p_{ij}\}$ of values of items for agents
a bid vector $b = (b_1, b_2, .., b_m)$ of limit constraints, such that $b_i \geq p_{ij}$, $\forall i, j$

Output: An allocation $\{S_i\}_{i=1}^m$ of items to agents

1. $I \leftarrow \{1, 2, \ldots, m\}$, $S_i \leftarrow \emptyset$, for $1 \leq i \leq m$
2. For $j \leftarrow 1$ to n;
   (a) If $I = \emptyset$ Return $\{S_i\}_{i=1}^m$
   (b) $i \leftarrow \arg\max_{i \in I}\{p_{ij}\}$
   (c) $S_i \leftarrow S_i \cup \{j\}$
   (d) If $\sum_{j \in S_i} p_{ij} > b_i$ Then $I \leftarrow I \setminus \{i\}$
3. Return the allocation $\{S_i\}_{i=1}^m$

Algorithm 1: Lazy Greedy

would be rationalizable since it appears to be monotone in the limit constraint. The following example disproves this intuitive claim:

EXAMPLE 4.1. *Consider 3 items* $\{1, 2, 3\}$ *and 2 bidders, with valuations* $\{3, 5, 6\}$ *and* $\{0, 4, 2\}$ *respectively. If both bidders have a limit constraint of* 6 *then a greedy assignment gives the first bidder items* 1 *and* 3*. If the first bidder changes his limit constraint to* 9 *his allocation will be items* 1 *and* 2*. Week monotonicity is violated, since:*

$$v_1(\{1,3\}, 6) - v_1(\{1,2\}, 6) = 6 - 6$$
$$= 0 < 1 = 9 - 8 = v_1(\{1,3\}, 9) - v_1(\{1,2\}, 9)$$

Algorithm *Lazy Greedy* outputs a weakly monotone assignment and requires only $O(mn)$ operations, which is linear in the input size. Lazy Greedy allocates each item to the highest bidder, ignoring the limit constraint until after the assignment. If the limit constraint of an agent has been exceeded, then the agent is removed from the rest of the allocation process. Ignoring the limit constraint until after the allocation is required for weak monotonicity: An agent that raises its limit constraint will receive at least the same items. By Corollary 4.8, this implies rationalizability. When the limit is raised to a critical value needed to win another item, the extra valuation of this item is zero. Therefore, the payment function in Corollary 3.5 reduces to a constant payment, regardless of the actual assignment.

The analysis of Lazy Greedy is not given due to space constraints. However, the following holds for Lazy Greedy:

THEOREM 4.9. *The approximation ratio of Lazy Greedy is* 3.

One may argue that both algorithms presented here give a weak incentive to cooperation, since the mechanisms are not strongly truthful. There are strategies different than truthtelling, that may change the allocation, without affecting the utility. In the MAX-GAP based mechanism, since the payment equals the valuation, there is no incentive to avoid underbidding. The Lazy Greedy based mechanism does not give an incentive to avoid overbidding, since the payment is constant.

This difficulty is an outcome of the structure of the valuation functions, but can be overridden by a convex combination of these mechanisms, i.e. a randomized mechanism that with probability p applies the MAX-GAP reductions, and with probability $1 - p$ applies Lazy Greedy. This mechanism is universally truthful (a distribution over truthful mechanisms), has an approximation ratio of 3 in the worst case, and $3 - p$ in expectation.

4.4.3 A Negative Example

We note that unlike previous examples, HMD is not a necessary condition here. For example, consider VCG [26, 7, 9] mechanisms. Since the objective function is to maximize the total valuation, these mechanisms are truthful (and therefore, any algorithm that outputs an optimal allocation is rationalizable).

If there is more than one optimal outcome, VCG may choose among them arbitrarily. However, An inconsistent choice rule between optimal outcomes in different settings may lead to violation of both versions of the HMD condition. An example of such a case follows:

EXAMPLE 4.2. *Consider 2 bidders and 5 identical items. Both bidders have a valuation of* 1 *for each item. Given that the second bidder reports a limit of* 2*, consider the following allocation function: If the first bidder bids* 2*, he receives two items, if he bids* 4*, he receives four items, and for any other bid he receives three items. All other items are allocated to the second bidder. One can verify that these allocations maximize the total valuation, and are therefore consistent with the VCG mechanism.*

However, the decrease in the value of the allocation when bidding 2 *instead of* 1 *violates the first HMD condition, while the increase while lowering the bid from* 5 *to* 4 *violates the second HMD condition. Therefore, there exists a truthful VCG-based mechanism which does not comply to either HMD conditions.*

5. EX-POST EQUILIBRIUM

The theory of mechanism design assumes that the valuation function of each agent depends on its true type and on the output of the mechanism, which may depend on the bids of the other agents. The true types of all other agents, have no effect on the valuation function, once the outcome is fixed. In such setups, it is reasonable to try and construct a mechanism where truthtelling is a dominant strategy, since each agent is invariant to whether the other agents bid truthfully or strategically.

This assumption fails to hold when valuation functions are affected by the true types of other bidders and not only their actual bids and the mechanism outcome. When an agent lies about its type, it not only affects the outcome, but also inserts great uncertainty into the system, since the valuations of the other agents depend on its true type.

Since having truthtelling to be a dominant strategy may be an unreasonable aim, we settle on designing mechanisms which are ex-post truthful, meaning that there is an equilibrium where all agents reveal their true type. Relaxing the dominance requirement is necessary due to the complexity of the valuation functions, which depend on the private parameters of the other agents. Formally, the valuation function of an agent is now $v_i : \Omega \times T^n \mapsto \mathbb{R}$. The outcome space is now $\Omega \times T^{n-1}$ instead of Ω, i.e. a combination of the algorithm's decision and the true types of the other agents. We define an *ex-post truthful mechanism* as follows:

DEFINITION 5.1. *A mechanism* $M = (A, P)$ *is* ex-post truthful *if for every agent i, for every fixed set of bids for the other agents* b_{-i} *and for every types* $s, t \in T$

$$\begin{aligned} v_i(A(t, b_{-i}), (t, b_{-i})) - P_i(A(t, b_{-i}), b_{-i}) \\ \geq v_i(A(s, b_{-i}), (t, b_{-i})) - P_i(A(s, b_{-i}), b_{-i}) \end{aligned} \quad (9)$$

In words, a mechanism is ex-post truthful if any setting where all agents report their true type is a Nash Equilibrium. For any agent, assuming that all other agents report their true values, its valuation function reduces to a function that does not depend directly on the other agents' types, as those are assumed to be equal to the bids, and are now part of the extended outcome space. Let $\Omega' \subset \Omega \times T^{n-1} = \{(\omega, b_{-i}) | \exists b_i,\ A(b_i, b_{-i}) = \omega\}$ denote the set of possible outcomes if indeed the other agents bid their types. Replacing Ω with Ω', we can now extend the HMD conditions for valuation functions which depend on the types of the other agents, and give a ex-post truthful characterization and payment functions similar to Theorems 3.2 and 3.4.

THEOREM 5.2. *If a decision rule A supports the HMD condition then* $M = (A, P)$ *is an ex-post truthful mechanism, where*

$$\begin{aligned} P_i(t, b_{-i}) &= c(b_{-i}) + v_i(A(t, b_{-i}), (t, b_{-i})) \\ &- \int_{t_0}^{t} v_i'(A(x, b_{-i}), (x, b_{-i}))dx \end{aligned} \quad (10)$$

HMD is a necessary condition if for each agent, the partial derivative of the valuation function by the agent's type does not depend on the type.

5.1 Scheduling with Strategic Job Owners

Consider a machine scheduling setting where the machine speeds are publicly known, but the job owners have to report their processing requirements. Each job owner has a negative utility from waiting until the completion of the job (we assume that there is also a positive utility from having the job executed, which would motivate job owners to participate, but we ignore this issue here). The valuation of having a job with a processing requirement p_i executed on a machine with a cost per unit of c_j (i.e., a speed of $s_j = 1/c_j$) is $v_i = -c_j(W_{ij} + p_i)$, where W_{ij} denotes the total processing requirements of jobs that are executed on machine j, not including the i-th job [2]

Clearly, W_{ij} depends on the actual precessing requirements, which are unknown to the mechanism, and only assumed to be equal to the bids.

First, we like to show that unless the scheduling algorithm is degenerate, there is no dominant strategy for the job owners. A degenerate scheduler is a scheduler that always assigns the same jobs together on the same machine (or on a machine with the same speed), regardless of their bids, given that the jobs are assigned on the same machine for some bid (jobs that are always assigned alone can change machines). Clearly, degenerate scheduling may be considerably inefficient.

Assume that given the bids of the other agents, if job owner j bids its true type b_j then it is assigned to a machine together with the jobs in set S_1, and if the bid is $\hat{b}_j$ then the assignment is to a (possibly different) machine with the set S_2 of jobs. Assume that there is a job $i \in S_1$ such that $i \notin S_2$. Since the real processing requirement of job i is unknown to the mechanism and may be arbitrary large, any payment to job owner j may be insufficient to make j favor reporting b_j over $\hat{b}_j$, which avoids sharing a machine with the i-th job. Therefore truthtelling cannot be a dominant strategy in this case. A similar result can be shown if $S_1 = S_2$ but are non empty, and $s_{j_1} \neq s_{j_2}$. This motivates searching for mechanisms that are ex-post truthful.

Observe that the partial derivative is of the i-th valuation function is $v_i' = -c_j$. By Theorem 5.2, the necessary and sufficient HMD condition is that when a job raises its declared processing requirement (having all other jobs fixed and bidding truthfully), the job cannot move to a slower machine. Deriving this condition from HMD simplifies the proof that appears in [5]. Surprisingly, the constraint does not depend on W_{ij}. This is because if indeed all other agents bid truthfully, then $c_j W_{ij}$ is known to the mechanism and its effect can be reversed by setting the payment appropriately.

Follows is a construction of a fractional job assignment from [4], which is used to both lower bound the makespan of the optimal integral assignment, and serves as a starting point for constructing rationalizable scheduling algorithms for strategic machines [4, 3, 1].

1. Sort the jobs in non increasing order
2. Set a threshold of maximal load
3. Assign jobs to machines, from fastest to slowest, splitting a job whenever the threshold is reached

The threshold that serves as a lower bound for the optimal integral allocation is the smallest value that is large enough such that there is enough volume to allocate all the jobs, and no fraction of a job is allocated to a machine that cannot run the full job within the threshold limit. It is known that this threshold can be computed efficiently, in polynomial time.

By rounding each fractionally assigned job toward the faster machine we get a 2-approximation, since each machine gets at most one extra fraction, and the load it incurs is lower than the threshold. Although this approximation is not rationalizable for strategic machines, it can be used to construct an ex-post mechanism for strategic job owners.

THEOREM 5.3. *There exists an ex-post mechanism for machine scheduling with strategic job owners to minimize the makespan that is a 2-approximation*

Proof: We show that the approximation described above is monotone in the jobs, meaning that when a job increases in processing requirement, it cannot move to a slower machine. The existence of an ex-post mechanism follows from Theorem 5.2.

Given an increase in the processing requirement of the i-th job, we observe the effect of the increase in stages: As long as the order of the sorted jobs does not change, the threshold of the fractional assignment can only increase, increasing the capacity of every machine. The total processing

[2]The definition of W_{ij} given here assumes that the jobs are executed in parallel. Alternatively, one can consider serial execution, and define W_{ij} as the total processing requirements of all the jobs assigned before the i-th job, on the j-th machine.

requirement of jobs 1 to $i-1$ is unchanged, and therefore the machine where the first fraction (whether there are one or two fractions) of the i-th job is allocated is either a faster machine, or the same one. The integral solution rounds to the fastest machine with a fraction of the job, which is the machine that holds the first fraction, and therefore the chosen machine cannot be slower.

In breakpoints where the increase in the processing requirement of the i-th job cause it to swap places in the sorted job vector with the job before it, the same allocation is generated, except for swapping between two jobs. Obviously, the i-th job is shifted toward the faster machines, and cannot be allocated to a slower one. □

This is an improvement to [5], which presented non constant approximation algorithms, unless the ratio between the fastest and slowest machine speeds is small. Closing the gap between this upper bound and the lower bound of approximately 1.28 in [5] remains an open problem.

The scheduling mechanism described above is polynomial in its running time, since the assignment algorithm runs in polynomial time, and the payment function requires repeating the algorithm for each job in all the breakpoints that cause the job to change a machine, or change the jobs that are assigned with it to the same machine. Finding all breakpoints in polynomial time is explained in [4].

If both jobs and machines are strategic, then any ex-post truthful mechanism must fulfill a dual HMD condition, for both jobs and machines. Among the several rationalizable approximation schemes for strategic machines, the 5 approximation in [1] is also monotone from the jobs' perspective. Using similar ideas to those of [1] one can derive the following theorem.

THEOREM 5.4. *There exists an ex-post truthful mechanism for uniformly related machine scheduling, where both machines and job owners are strategic, which achieves a* 5 *approximation.*

6. REFERENCES

[1] N. Andelman, Y. Azar, and M. Sorani. Truthful approximation mechanisms for scheduling selfish related machines. In *Proc. 22nd Ann. Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 69–82, 2005.

[2] N. Andelman and Y. Mansour. Auctions with budget constraints. In *Proc. 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 26–38, 2004.

[3] A. Archer. *Mechanisms for Discrete Optimization with Rational Agents*. PhD thesis, Cornell University, 2004.

[4] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. 42nd Ann. Symp. on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[5] V. Auletta, R. D. Prisco, P. Penna, and P. Persiano. How to route and tax selfish unsplittable traffic. In *Proc. 16th Ann. ACM Symp. on Parallel Algorithms (SPAA)*, pages 196–205, 2004.

[6] C. Chekuri and S. Khanna. A ptas for the multiple knapsack problem. In *Proc. 11th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 213–222, 2000.

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

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-completeness*. Freeman, San-Francisco, 1979.

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

[10] H. Gui, R. Müller, and R. V. Vohra. Dominant strategy mechanisms with multidimensional types. working paper.

[11] D. S. Hochbaum and D. B. Shmoys. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM J. Comput.*, 17(3):539–551, 1988.

[12] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *J. of the Assoc. for Comp. Machinery*, 23:317–327, 1976.

[13] A. Kovács. Fast monotone 3-approximation algorithm for scheduling related machines. In *Proc. 13th Ann. European Symp. on Algorithms (ESA)*, pages 616–627, 2005.

[14] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proc. 44th Symp. on Foundations of Computer Science (FOCS)*, pages 574–583, 2003.

[15] B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proc. ACM Conf. on Electronic Commerce (EC)*, pages 18–28, 2001.

[16] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University press, 1995.

[17] J. A. Mirrlees. Optimal tax theory: A synthesis. *J. of Public Economics*, 6:327–358, 1976.

[18] A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proc. 18th National Conf. on Artificial Intelligence and 14th Conf. on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 379–384, 2002.

[19] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proc. 31st Ann. ACM Symp. on Theory of Computing (STOC)*, pages 129–140, 1999.

[20] N. Nisan and A. Ronen. Computationally feasible vcg mechanisms. In *Proc. ACM Conf. on Electronic Commerce (EC)*, pages 242–252, 2000.

[21] K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–348. North Holland, 1979.

[22] J. C. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *J. of Mathematical Economics*, 16:191–200, 1987.

[23] M. E. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proc. ACM Conf. on Electronic Commerce (EC)*, pages 286–293, 2005.

[24] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993.

[25] M. Spence. Competitive and optimal responses to signals: An analysis of efficiency and distribution. *J. of Economic Theory*, 7:296–332, 1974.

[26] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *J. of Finance*, 16:8–37, 1961.