Improved Lower Bounds for Non-Utilitarian Truthfulness*

Iftah Gamzu†

Abstract

One of the most fundamental results in the field of mechanism design states that every utilitarian social choice function admits a mechanism that truthfully implements it. In stark contrast with this finding, when one considers a non-utilitarian social choice function, it turns out that no guarantees can be made, i.e. there are non-utilitarian functions, which cannot be truthfully implemented. In light of this state of affairs, one of the most natural and intriguing objectives of research is to understand the inherent limitations in the infrastructure of truthful mechanisms for non-utilitarian social choice functions.

In this paper, we focus our attention on studying the boundaries imposed by truthfulness for two non-utilitarian multi-parameter optimization problems. The first is the *workload minimization in inter-domain routing* problem, which models one of the most principle problems in the design of routing protocols, and the other is the *unrelated machines scheduling* problem, which is one of the most classical and general variants in the field of scheduling. Our main findings can be briefly summarized as follows:

1. We prove that any truthful deterministic mechanism, and any universal truthful randomized mechanism for the workload minimization in inter-domain routing problem cannot achieve an approximation guarantee that is better than 2. These results improve the current lower bounds of $(1+\sqrt{5})/2 \approx 1.618$ and $(3+\sqrt{5})/4 \approx 1.309$, which are due to Mu'alem and Schapira [SODA '07].
2. We establish a lower bound of $1+\sqrt{2} \approx 2.414$ on the achievable approximation ratio of any truthful deterministic mechanism for the unrelated machines scheduling problem when the number of machines is at least 3. This lower bound is comparable to a recent result by Christodoulou, Koutsoupias and Vidali [SODA '07]. Nevertheless, our approach is considerably simpler, and thus may shed some new light on the foundations of this problem.

*An extended abstract of this paper appeared in *Proceedings of the 5th International Workshop on Approximation and Online Algorithms*, pages 15–26, 2007.

†School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: `iftgam@post.tau.ac.il`.
Supported by the Binational Science Foundation, and by the Israel Science Foundation.

1 Introduction

The problems. We study the *workload minimization in inter-domain routing* problem. As input to this problem, we are given a directed graph $G = (V, E)$, such that $n = |V|$, every edge $e \in E$ has a *cost* $c_e \in \mathbb{R}_+$, and there is a designated *target* vertex $t \in V$. An additional ingredient of the input is a set $\mathcal{R}$ of connection requests in which every request $r \in \mathcal{R}$ is characterized by a pair (s_r, d_r) such that s_r is the *source* vertex of the request, and $d_r \in \mathbb{R}_+$ is the *demand* or traffic intensity associated with the request. The objective is to assign a path from s_r to t, for every request r, on which the request's demand will be sent, such that all the paths constitute a *confluent routing tree.* A confluent routing tree is a tree in which all the traffic arriving at any vertex leaves along a single edge. In particular, the goal is to determine a routing tree in which the workload imposed on the "busiest" vertex is minimized, that is a tree T that minimizes $\max_{u\in V} c_u^T \sum_{r\in R_u^T} d_r$, where R_u^T and c_u^T denote the set of requests that route their demand using a path that goes through u in T and the cost of the single edge that leaves vertex u in T, respectively. Remark that the original formulation of the problem [11] is slightly different, e.g. the problem is defined with respect to an undirected graph in which every vertex has a cost function on its outgoing edges. Nevertheless, it is not hard to validate that both formulations are essentially equivalent.

We also consider the *unrelated machines scheduling* problem. An instance of this problem consists of n machines, and m tasks such that the execution time of task j on machine i is determined by the t_{ij} entry of an $n \times m$ matrix t. The objective is to generate an allocation of the tasks to the machines that minimizes the *makespan*, i.e. the maximum completion time of the machines. This goal is equivalent to generating an $n \times m$ allocation matrix x in which every x_{ij} entry is an $\{0, 1\}$-indicator such that $x_{ij} = 1$ if and only if task j is allocated to machine i, every task is assigned to exactly one machine, i.e. $\sum_{i\in[n]} x_{ij} = 1$ for every $j \in [m]$, and $\max_{i\in[n]} \sum_{j\in[m]} x_{ij}t_{ij}$ is minimized.

The setting. In the present paper, we study the aforementioned problems from an *algorithmic mechanism design* [12] point of view. Algorithmic mechanism design studies the design of protocols or mechanisms for algorithmic problems in scenarios where the input is presented by *strategic agents.* Strategic agent might declare a fallacious input in order to manipulate the protocol in a way that will maximize its own utility. A primary interest of algorithmic mechanism design is in the development of *incentive compatible* or *truthful* protocols, which are robust against manipulation by agents, i.e. every agent is rationally motivated to truthfully report its input. Particulary, in this paper, we concentrate on lower bounding the achievable approximation guarantee of any truthful protocol for the problems under consideration. Note that in the workload minimization in inter-domain routing problem, every vertex is assumed to be controlled by a strategic agent, which may be dishonest about the costs of the vertex's outgoing edges, and in the unrelated machines scheduling problem, every machine is assumed to be controlled by a strategic agent, which may be untruthful about the execution times of the tasks on the corresponding machine, e.g. the agent that controls machine i may be deceitful about the corresponding row vector of execution times $t_i = \langle t_{i1}, \ldots, t_{im}\rangle$.

1.1 Our results

Workload minimization in inter-domain routing. We establish that any truthful deterministic mechanism for the workload minimization in inter-domain routing problem cannot achieve an approximation guarantee that is better than 2. Additionally, we reinforce this inapproximability result by demonstrating that no randomized mechanism, which is truthful in the universal sense, can obtain an approximation ratio better than 2. These results improve upon the lower bounds presented by Mu'alem and Schapira [11], which are $(1 + \sqrt{5})/2 \approx 1.618$ and $(3 + \sqrt{5})/4 \approx 1.309$, respectively.

Unrelated machines scheduling. We prove that any truthful deterministic mechanism for the unrelated machines scheduling problem cannot yield an approximation guarantee that is better than $1 + \sqrt{2} \approx 2.414$ for input instances that have at least 3 machines. This result is equal to the lower bound exhibited recently by Christodoulou, Koutsoupias and Vidali [4]. Notwithstanding, our approach is significantly simpler. In particular, we demonstrate how to bypass the so-called *geometrical structure of mechanisms*, which seems to be imperative in their proof.

1.2 Related work

The workload minimization in inter-domain routing problem models one of the most fundamental problems in the design of routing protocols. Specifically, it captures the problem of establishing a routing tree for a network (e.g. the Internet), in which no single autonomous system (AS) is excessively congested. This problem was introduced by Mu'alem and Schapira [11], who posed it as a natural extension to the *inter-domain routing* problem, which was formulated by Feigenbaum, Papadimitriou, Sami, and Shenker [7]. Mu'alem and Schapira proved that this problem cannot be approximated to within factors of $(1 + \sqrt{5})/2$ and $(3 + \sqrt{5})/4$ by any truthful deterministic mechanism and any randomized mechanism that is truthful in the universal sense, respectively. In addition, they designed a simple truthful deterministic mechanism that obtains an approximation ratio of n. They also considered the single-parameter variant of the problem, in which all the outgoing edges of a vertex have the same cost, and demonstrated that this variant can be solved in an optimal way by a truthful deterministic exponential-time mechanism.

The problem of unrelated machines scheduling is one of the most classical and general variants in the field of scheduling and as such, it has been given extensive attention in past years, both from an algorithmic point of view and from a game-theoretic one. From a pure algorithmic point of view, the problem is known to admit 2-approximation algorithms (see e.g. [10, 14, 1]), and is known to be $\frac{3}{2}$-hard to approximate in polynomial time, unless P = NP [10]. The mechanism design version of the problem originates in the pioneering work of Nisan and Ronen [12]. They proposed a polynomial-time truthful deterministic mechanism, which is a member of the VCG family [15, 5, 8], that achieves an approximation guarantee of n, and also proved that no deterministic mechanism can obtain an approximation ratio that is better than 2. They also conjectured that this gap will be resolved by showing that no deterministic mechanism can attain approximation ratio better than n. Finally, they demonstrated that randomization may help to obtain better outcome by presenting a randomized truthful mechanism for a two machines scenario, which has an approximation ratio of $\frac{7}{4}$. Recently, Mu'alem and Schapira [11] extended the last result, and devised a randomized truthful mechanism for any n machines, which achieves an approximation ratio of $\frac{7n}{8}$. In the same work, they also established a lower bound of $2 - \frac{1}{n}$ on the achievable approximation guarantee of any randomized mechanism. Correspondingly, Christodoulou, Koutsoupias and Vidali [4] proved that any truthful deterministic mechanism cannot yield an approximation guarantee that is better than $1 + \sqrt{2}$ for 3 machines or more. A concurrent line of work, initiated independently by Christodoulou, Koutsoupias and Kovács [3] and Lavi and Swamy [9], studied special variants of the unrelated machines scheduling problem. Christodoulou, Koutsoupias and Kovács considered the fractional variant of the problem, and devised a deterministic polynomial-time truthful mechanism that attains an approximation ratio of $\frac{n+1}{2}$, while proving that this fractional variant cannot be truthfully approximated within a factor better than $2 - \frac{1}{n}$. Lavi and Swamy researched the "low-high" variant of the problem, in which the execution time of every tasks is either "low" or "high", and designed a 3-approximation truthful-in-expectation mechanism. They also presented a truthful deterministic 2-approximation mechanism for the case that all the tasks share the same "low" and "high" values, and demonstrated that in this case no truthful deterministic mechanism can achieve an approximation ratio better than 1.14.

2 Preliminaries

In this section, we exhibit a brief introduction to the field of algorithmic mechanism design, and then turn to describe a key property, which every truthful deterministic mechanism must satisfy. This property is fundamental to our approach as our lower bound proofs are built upon it. Remark that this section strives to provide a succinct description of the relevant definitions and results of Bikhchandani et al. [2], and hence the keen reader is encouraged to refer to the aforesaid paper for a more comprehensive presentation of the underlying concepts.

We begin by outlining the nature of questions that algorithmic mechanism design studies. In an algorithmic mechanism design problem setup, there is a set of n *strategic agents* and a finite set of *outcomes* A. Every agent i has a private *type* represented by a *valuation function* $v_i : A \to \mathbb{R}_+$, where $v_i \in V_i$, and V_i denotes the domain of all valid types of agent i. Note that each agent is only interested to maximize its own gain, and thus may be dishonest when reporting its type. The main interest of algorithmic mechanism design is to generate a *social choice mechanism* that is *truthful*. Essentially, the goal is to design an *allocation algorithm* $f : V_1 \times \ldots \times V_n \to A$, and a *payment scheme* $p : V_1 \times \ldots \times V_n \to \mathbb{R}^n$ such that each agent's dominant strategy is to truthfully report its type to the mechanism $M = (f, p)$.

We now turn to describe a property that every truthful deterministic mechanism must satisfy. Notice that this reduces the goal of establishing a lower bound on the achievable approximation ratio of truthful deterministic mechanisms to that of proving a lower bound for a restricted class of allocation algorithms, which constitute truthful mechanisms. Note that we will henceforth refer to such allocation algorithms as *truthful allocation algorithms*.

Definition 2.1. Let $v = (v_1, \ldots, v_n)$ be a tuple of valuations, and $v' = (v'_i, v_{-i})$ be the tuple of valuations obtained by replacing the valuation function of agent i in v from v_i to v'_i. In addition, let f be an allocation algorithm such that $a = f(v)$ and $b = f(v')$. The allocation algorithm f is said to be *weakly monotone* if

$$v_i(a) + v'_i(b) \leq v'_i(a) + v_i(b) \ ,$$

for every $i \in [n]$, and every valid valuations tuples v and v'.[1]

Theorem 2.2 ([2]). *If $M = (f, p)$ is a truthful mechanism then f must be weakly monotone.*

Remark that Saks and Yu [13] proved that the weak monotonicity property is not only necessary for truthfulness, but for convex domains is also sufficient. Nonetheless, this fact will not be utilized in the context of this paper.

3 Workload Minimization in Inter-Domain Routing

In this section, we study the workload minimization in inter-domain routing problem, and establish a lower bound of 2 on the achievable approximation guarantee of any truthful deterministic mechanism, and any universal truthful randomized mechanism. Prior to describing the finer details of our approach, we provide an interpretation of the weak monotonicity theorem, i.e. Theorem 2.2, to the problem under consideration. Bear in mind that the valuation function of the agent that controls vertex u satisfies $v_u(a) = d \cdot c_e$, where d is the total traffic that goes through u in the routing tree defined by the outcome a, e is the single edge that leaves vertex u in that routing tree, and c_e is its cost.

[1] Note that this definition applies for cases in which the valuation function of every agent represents cost induced on that agent, and the interest of every agent is to minimize its cost, e.g. it applies for the problems under consideration. For cases in which the valuation function of every agent corresponds to profit, the inequality is in the opposite direction, i.e. $\geq$ instead of $\leq$.

Corollary 3.1. *Suppose we are given two input instances for the workload minimization in inter-domain routing problem, which only differ in the cost functions on the edges. In particular, suppose that the cost functions c and c' only disagree on vertex u's outgoing edges costs. Every truthful allocation algorithm must satisfy that if*

- *e and e' are two outgoing edges of u,*
- *d units of traffic are routed through e in the routing tree generated w.r.t. c,*
- *d' units of traffic are routed through e' in the routing tree generated w.r.t. c',*

then

$$d(c_e - c'_e) + d'(c'_{e'} - c_{e'}) \leq 0 \ .$$

We now turn to argue about the deterministic lower bound. In particular, we prove that any truthful allocation algorithm has a "poor" input instance, for which it generates a routing tree whose workload value is bounded away from the optimal workload value by a factor of at least 2.

Theorem 3.2. *The approximation ratio of any truthful deterministic mechanism for the workload minimization in inter-domain routing problem cannot be better than* 2.

Proof. Consider a truthful allocation algorithm for the problem under consideration, and suppose that its input is the directed graph schematically described in Figure 1(a), and the set of requests is $\mathcal{R} = \{r_1, r_2\} = \{(s_1, 1), (s_2, 1)\}$.

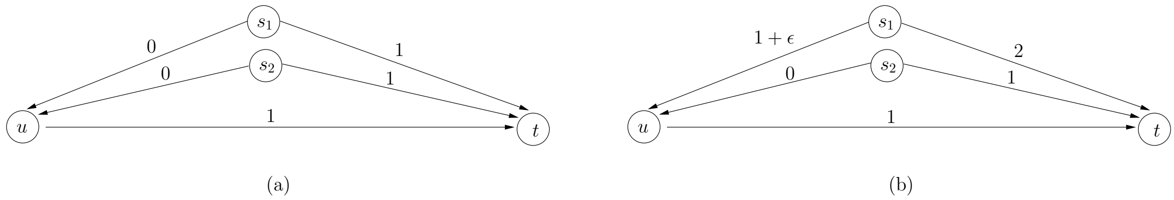

Figure 1: The deterministic lower bound instances.

Notice that if the algorithm routes both requests using the vertex u, the obtained routing tree has a workload value of 2. Also notice that an optimal routing tree for this instance has a workload value of 1, e.g. the routing tree, which consists of the edge set $\{(s_1, t), (s_2, t)\}$. Therefore, in such case, we infer that the algorithm cannot have an approximation ratio better than 2. Consequently, we will assume, throughout the remainder of this proof and without loss of generality, that the algorithm routes the request r_1 through the edge (s_1, t). Now, suppose the algorithm is given as input the directed graph schematically described in Figure 1(b), and the same set of requests as before. Remark that the only difference between this input instance, and the aforementioned input instance is the costs of the edges that leave s_1. Specifically, $c_{(s_1,t)} = 2$, and $c_{(s_1,u)} = 1 + \epsilon$, where $0 < \epsilon < 1$ is constant. We claim the algorithm, given this input instance, must also route the request r_1 through the edge $e = (s_1, t)$. This follows from the observation that if the request r_1 is routed through the edge $e' = (s_1, u)$, we yield a contradiction to Corollary 3.1, since $d = 1$, $d' = 1$, $c_e = 1$, $c'_e = 2$, $c'_{e'} = 1+\epsilon$ and $c_{e'} = 0$. Consequently, the workload value of the routing tree generated by the algorithm is at least 2, whereas the optimal routing tree has a workload value of $1 + \epsilon$, e.g. the routing tree that comprises the edge set $\{(s_1, u), (u, t), (s_2, t)\}$. This establishes that the algorithm cannot have an approximation guarantee better than $\frac{2}{1+\epsilon}$. Since one can select any positive constant $\epsilon \to 0$, the theorem follows. ∎

In the following, we reinforce the last theorem by establishing a lower bound of 2 for universally truthful randomized mechanisms. Note that such mechanisms are defined as a probability distribution over truthful deterministic mechanisms [12, 6]. Our approach is based on Yao's min-imax principle [16]. In the context of our setting, this principle states that the approximation ratio of the best universal truthful randomized mechanism is equal to the approximation ratio of the best deterministic truthful mechanism under a worst-case input distribution. Accordingly, we exhibit a probability distribution over input instances for which any deterministic truthful mechanism cannot attain an approximation guarantee better than 2.

Theorem 3.3. *The approximation ratio of any universal truthful randomized mechanism for the workload minimization in inter-domain routing problem cannot be better than* 2.

Proof. Let I denote the input instance, which consists of the directed graph schematically described in Figure 2, and the set of requests $\mathcal{R} = \{r_1, r_2, \ldots, r_k\} = \{(s_1, 1), (s_2, 1), \ldots, (s_k, 1)\}$. Additionally, let I_j be the input instance that is nearly I, but has different costs to the edges that leave s_j. Specifically, the costs of the corresponding edges in I_j are $c_{(s_j,t)} = 2$, and $c_{(s_j,u)} = 1 + \epsilon$, where $0 < \epsilon < 1$ is constant. Finally, let P be a probability distribution over the set of input instances $\{I, I_1, \ldots, I_k\}$ such that every instance is picked with probability $\frac{1}{k+1}$.

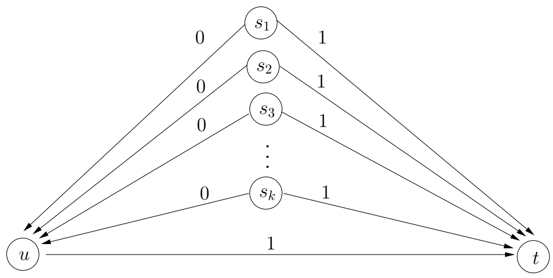

Figure 2: The randomized lower bound instance.

Consider a truthful allocation algorithm for the problem under consideration, and let us analyze its performance on the set of input instances $\{I, I_1, \ldots, I_k\}$ with probability distribution P. We now consider two cases, depending on the structure of the routing tree generated by the algorithm for the input instance I.

Case I: The algorithm does not route any of the requests through u. Notice that the algorithm generates a routing tree $T = \bigcup_{i=1}^{k}(s_i, t)$ whose workload value is optimal for I. However, one can apply arguments similar to those used in Theorem 3.2, and yield that the algorithm cannot obtain an approximation guarantee better than $\frac{2}{1+\epsilon}$ for any of the instances $\{I_1, \ldots, I_k\}$. In particular, one can easily verify that given the input instance I_j, the algorithm must route the request r_j using the edge (s_j, t). Consequently, the expected approximation ratio of the algorithm on the input distribution P is at least $1 \cdot \frac{1}{k+1} + \frac{2}{1+\epsilon} \cdot \frac{k}{k+1} > 2\frac{k}{(k+1)(1+\epsilon)}$. Since one can utilize graph instances for which $k \to \infty$, and may select any constant $\epsilon \to 0$, the theorem follows for this case.

Case II: The algorithm routes $1 \leq q \leq k$ requests through u. Let $Q \subseteq \mathcal{R}$ be the set of requests that the algorithm routes through u. Notice that the workload value of the routing tree generated by the algorithm is q, and accordingly the approximation ratio of the algorithm is q. Additionally, one can apply arguments similar to those used in Theorem 3.2, and yield that the algorithm cannot attain a better than $\frac{2}{1+\epsilon}$-approximation for any of the instances $\{I_j : r_j \in \mathcal{R} \backslash Q\}$. Hence, the expected approximation ratio of the algorithm on the input distribution P is at least $q \cdot \frac{1}{k+1} + \frac{2}{1+\epsilon} \cdot \frac{k-q}{k+1} + 1 \cdot \frac{q}{k+1} > 2\frac{k}{(k+1)(1+\epsilon)}$, and thus the theorem follows also for this case. ■

4 Unrelated Machines Scheduling

In this section, we establish a lower bound of $1+\sqrt{2}$ on the approximation ratio of any truthful deterministic mechanism for the unrelated machines scheduling problem when the number of machines is at least 3. Before we turn to portray the details of our approach, we provide an abstraction of the weak monotonicity theorem, i.e. Theorem 2.2, to the problem under consideration. Remark that the valuation function of the agent that controls machine i satisfies $v_i(a) = \sum_{j\in[m]} x_{ij}t_{ij}$, where x_{ij} indicates if task j is allocated to machine i in the outcome a, and t_{ij} is the execution time of task j on machine i.

Corollary 4.1. *Let t and t' be input matrices for the unrelated machines scheduling problem, which differ only in the execution times of machine i. Every truthful allocation algorithm that generates the allocation matrices x and x' w.r.t. t and t' must satisfy*

$$\sum_{j\in[m]} (x_{ij} - x'_{ij})(t_{ij} - t'_{ij}) \leq 0 \ .$$

In the following, we exploit Corollary 4.1 to derive two simple claims, which will later enable us to demonstrate the desired lower bound. Remark that the first claim extends Lemma 1 of [4].

Claim 4.2. *Let x and x' be allocation matrices generated by a truthful allocation algorithm w.r.t. input matrices t and t', which differ only in the execution times of machine i. If every task realizes one of the following cases*

- Case I: *task j can only be allocated to machine i w.r.t. both t and t'.*[2]
- Case II: $t'_{ij} > t_{ij}$ *and* $x_{ij} = 0$.
- Case III: $t'_{ij} < t_{ij}$ *and* $x_{ij} = 1$.

then x and x' must agree on the allocation of machine i.

Proof. If task j can only be allocated to machine i, then clearly $x_{ij} = x'_{ij} = 1$. Also notice that this implies that the corresponding weak monotonicity term, i.e. $(x_{ij} - x'_{ij})(t_{ij} - t'_{ij})$, equals 0, and hence it does not contribute to the left hand side of the requirement in Corollary 4.1, that is $\sum_{j\in[m]}(x_{ij} - x'_{ij})(t_{ij} - t'_{ij})$.

Focusing on the other two cases, one can easily validate that every corresponding weak monotonicity term can only be nonnegative. For example, if task j satisfies $t'_{ij} > t_{ij}$ and $x_{ij} = 0$ then the corresponding weak monotonicity term reduces to $(0 - x'_{ij})(t_{ij} - t'_{ij})$. This term is nonnegative as $(t_{ij} - t'_{ij}) < 0$, and $x_{ij} \in \{0, 1\}$. Consequently, in order to satisfy the weak monotonicity theorem, it must follow that $x'_{ij} = x_{ij}$ for all the corresponding tasks. ■

Claim 4.3. *Let x and x' be allocation matrices generated by a truthful allocation algorithm w.r.t. input matrices t and t', and let $\{k, \ell\}$ be a set of two tasks. If $t_{ik} = t_{i\ell} = a$ for some $a > 1$, $t'_{ik} = t'_{i\ell} = 1$, all other execution times of t and t' are identical, and x assigns* exactly *one of the tasks in $\{k, \ell\}$ to machine i, then x' must assigns at least one of the tasks in $\{k, \ell\}$ to machine i.*

Proof. Notice that in order to satisfy Corollary 4.1 in the aforesaid settings, x'_{ik} and $x'_{i\ell}$ must fulfill $(1 - x'_{ik} - x'_{i\ell})(a - 1) \leq 0$. Accordingly, x' must assign at least one of the tasks $\{k, \ell\}$ to machine i. ■

[2] We say that task j can *only* be allocated to machine i w.r.t. t if $t_{ij} \neq \infty$, and $t_{\ell j} = \infty$ for every $\ell \in [n] \setminus \{i\}$.

We are now ready to prove the aforementioned lower bound. Essentially, we prove that any truthful allocation algorithm has a "bad" input matrix, for which it generates an allocation whose makespan value is bounded away from the optimal makespan value by a factor of at least $1+\sqrt{2}$. Remark that for ease of presentation, we may apply Claim 4.2 to input matrices t and t', in which some execution times of machine i are the same. Nevertheless, the understanding between us is that there is a tiny change in these execution times, which we neglect in order to keep the expressions simple, that satisfies the restrictions imposed by the claim.

Theorem 4.4. *The approximation ratio of any truthful deterministic mechanism for the unrelated machines scheduling problem with at least* 3 *machines cannot be better than* $1+\sqrt{2}$.

Proof. Consider a truthful allocation algorithm for the unrelated machines scheduling problem, and suppose that its input is the following 3-machines 5-tasks matrix

$$t = \begin{pmatrix} 0 & \infty & \infty & \sqrt{2} & \sqrt{2} \\ \infty & 0 & \infty & \sqrt{2} & \sqrt{2} \\ \infty & \infty & 0 & \sqrt{2} & \sqrt{2} \end{pmatrix} .$$

Note that this input matrix admits two distinct task allocations up to symmetries, i.e. name changes of the machines. These two possible tasks allocations are

$$x = \begin{pmatrix} 0^* & \infty & \infty & \sqrt{2}^* & \sqrt{2}^* \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2} \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix}, \text{ and } y = \begin{pmatrix} 0^* & \infty & \infty & \sqrt{2}^* & \sqrt{2} \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2}^* \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix} ,$$

where every superscript $*$ denotes an assignment of the column corresponding task to the row corresponding machine. We now consider two cases, depending on which allocation is generated by the algorithm.

Case I: x is generated by the algorithm. Lets consider the matrix t', which is identical to t with a single exception that is $t'_{11} = \sqrt{2}$. By Claim 4.2, we know that the allocation generated by the algorithm for the first machine when t' is the input matrix cannot change. Namely, the tasks allocation is

$$\begin{pmatrix} \sqrt{2}^* & \infty & \infty & \sqrt{2}^* & \sqrt{2}^* \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2} \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix} .$$

This allocation has a value of $3\sqrt{2}$, while it is easy to verify that the optimal allocation has a value of $\sqrt{2}$. Consequently, this proves that the algorithm cannot have an approximation ratio better than $3 > 1+\sqrt{2}$.

Case II: y is generated by the algorithm. Lets consider the matrix t' that has the same execution times as t with two exceptions, which are $t'_{14} = t'_{15} = 1$. By Claim 4.3, we know that the allocation generated by the algorithm when t' is the input matrix must assign at least one of the tasks $\{4, 5\}$ to the first machine. Accordingly, the tasks allocation, up to symmetry, is either

$$x' = \begin{pmatrix} 0^* & \infty & \infty & 1^* & 1^* \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2} \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix}, \text{ or } y' = \begin{pmatrix} 0^* & \infty & \infty & 1^* & 1 \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2}^* \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix} .$$

Again, we regard two cases, depending on which allocation is generated.

Case IIa: x' is generated by the algorithm. Lets consider the matrix t'', which is identical to t' with a single difference that is $t''_{11} = \sqrt{2}$. By Claim 4.2, we know that the allocation generated by the algorithm for the first machine when t'' is the input matrix cannot change. Consequently, the tasks allocation is

$$\begin{pmatrix} \sqrt{2}^* & \infty & \infty & 1^* & 1^* \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2} \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix} .$$

Notice that this allocation has a value of $\sqrt{2}+2$, whereas the optimal allocation has a value of $\sqrt{2}$. Therefore, this establishes that the algorithm cannot have an approximation guarantee better than $\frac{\sqrt{2}+2}{\sqrt{2}} = 1+\sqrt{2}$.

Case IIb: y' is generated by the algorithm. Lets consider a two-step transition. First, consider the input matrix t'' that is alike t' with a single exception, which is $t''_{14} = 0$. By Claim 4.2, we know that the allocation generated by the algorithm for the first machine cannot change. Hence, the tasks allocation, up to symmetry, is

$$\begin{pmatrix} 0^* & \infty & \infty & 0^* & 1 \\ \infty & 0^* & \infty & \sqrt{2} & \sqrt{2}^* \\ \infty & \infty & 0^* & \sqrt{2} & \sqrt{2} \end{pmatrix} .$$

Second, consider the input matrix t''', which is identical to the matrix t'' with a single change that is $t'''_{22} = 1$. Again, by Claim 4.2, we know that the allocation generated by the algorithm for the second machine cannot change. Hence, the tasks allocation for the second machine is

$$\begin{pmatrix} 0 & \infty & \infty & 0 & 1 \\ \infty & 1^* & \infty & \sqrt{2} & \sqrt{2}^* \\ \infty & \infty & 0 & \sqrt{2} & \sqrt{2} \end{pmatrix} .$$

This allocation has a value of $1+\sqrt{2}$, while it is easy to validate that the optimal allocation has a value of 1. Thus, this proves that the algorithm cannot have an approximation guarantee better than $1+\sqrt{2}$. ■

Acknowledgments

I would like to thank Yossi Azar for his valuable comments on earlier drafts of this paper. Furthermore, I would like to thank the anonymous referees for their constructive suggestions.

References

[1] Y. Azar and A. Epstein. Convex programming for scheduling unrelated parallel machines. In *Proceedings 37th ACM Symposium on Theory of Computing*, pages 331–337, 2005.

[2] S. Bikhchandani, S. Chatterji, R. Lavi, A. Mu'alem, N. Nisan, and A. Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 74(4):1109–1132, 2006.

[3] G. Christodoulou, E. Koutsoupias, and A. Kovács. Mechanism design for fractional scheduling on unrelated machines. In *Proceedings 34th International Colloquium on Automata, Languages and Programming*, 2007.

[4] G. Christodoulou, E. Koutsoupias, and A. Vidali. A lower bound for scheduling mechanisms. In *Proceedings 18th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1163–1170, 2007.

[5] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[6] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings 38th ACM Symposium on Theory of Computing*, pages 644–652, 2006.

[7] J. Feigenbaum, C. H. Papadimitriou, R. Sami, and S. Shenker. A bgp-based mechanism for lowest-cost routing. *Distributed Computing*, 18(1):61–72, 2005.

[8] T. Groves. Incentives in teams. *Econemetrica*, 41(4):617–631, 1973.

[9] R. Lavi and C. Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *Proceedings 8th ACM Conference on Electronic Commerce*, pages 252–261, 2007.

[10] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(3):259–271, 1990.

[11] A. Mu'alem and M. Schapira. Setting lower bounds on truthfulness. In *Proceedings 18th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1143–1152, 2007.

[12] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[13] M. E. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings 6th ACM Conference on Electronic Commerce*, pages 286–293, 2005.

[14] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(3):461–474, 1993.

[15] W. Vickery. Counterspeculation, auctions and competitive sealed tender. *Journal of Finance*, 16:8–37, 1961.

[16] A. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings 18th Annual IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.