On 2-Player Randomized Mechanisms for Scheduling

Pinyan Lu

Microsoft Research Asia
pinyanl@microsoft.com

Abstract. In this paper, we study randomized truthful mechanisms for scheduling unrelated machines. We focus on the case of scheduling two machines, which is also the focus of many previous works [12,13,6,4]. For this problem, [13] gave the current best mechanism with an approximation ratio of 1.5963 and [14] proved a lower bound of 1.5. In this work, we introduce a natural technical assumption called *scale-free*, which says that the allocation will not change if the instance is scaled by a global factor. Under this assumption, we prove a better lower bound of $\frac{25}{16}(= 1.5625)$. We then study a further special case, namely scheduling two tasks on two machines. For this setting, we provide a correlation mechanism which has an approximation ratio of 1.5089. We also prove a lower bound of 1.506 for all the randomized scale-free truthful mechanisms in this setting.

1 Introduction

Mechanism design has become an active area of research both in Computer Science and Game Theory. In the mechanism design setting, players are selfish and wish to maximize their own utilities. To deal with the selfishness of the players, a mechanism should both satisfy some game-theoretical requirements such as *truthfulness* and some computational properties such as *good approximation ratios*. The study of their algorithmic aspect was initiated by Nisan and Ronen in their seminal paper "Algorithmic Mechanism Design" [15]. The focus of this paper was on the scheduling problem on unrelated machines, for which the standard mechanism design tools (e.t. VCG mechanisms [5,7,16]) do not suffice. They proved that no deterministic mechanism can have an approximation ratio better than 2 for this problem. This bound is tight for the case of two machines. However if we allow randomized mechanisms, this bound can be beaten. In particular they gave a 1.75-approximation randomized truthful mechanism for the case of two machines. This bound has since been improved to 1.6737 [12] and then to 1.5963 [13] by Lu and Yu. In [14], Mu'alem and Schapira proved a lower bound of 1.5 for this setting. The focus of this paper is to explore the exact bound between 1.5 and 1.5963.

In [13], Lu and Yu also proved a lower bound of $\frac{11}{7}$ (≈ 1.5714) for all the *task independent* truthful mechanisms. A recent work [6] by Dobzinski and Sundararajan showed that any truthful mechanism for two machines with a finite

S. Leonardi (Ed.): WINE 2009, LNCS 5929, pp. 30–41, 2009.

approximation ratio is *task-independent*. However the definitions of these two "task-independence" are not identical. The lower bound of [13] requires a strong version of "task-independence" and the characterization theorem in [6] only works for a weak version of "task-independence". Formal definitions of these two versions of "task independence" are given in the next section. This gives an interesting open problem: is there any weak task-independent mechanism which can beat all the strong task independent ones (In particular has an approximation ratio which is better than $\frac{11}{7}$)? We note that all the previous know mechanisms in this setting are strong task-independent[15,12,13]. Roughly speaking, in a strong task-independent mechanism, the random bits used by the allocation algorithm for different tasks are independent, while in weak task-independent mechanisms, they may have some correlation. In section 4, we provide such a correlation mechanism. This is the first truthful mechanism for this problem which is not strong task-independent. This mechanism provides an approximation ratio of 1.5089 for the case of two task, which is strictly better than all the strong task independent mechanisms in the same setting. We note that the lower bound of $\frac{11}{7}$ already holds even for the special case of scheduling two tasks in two machines.

The main focus of this paper is on the lower bound side. We introduce a natural assumption called *scale-free*, which says that the allocation will not change if a instance of the problem is scaled by a global factor. The property of *scale-free* is very natural for an allocation algorithm since a global factor only reflects the unit used for the running times. For example, if we change the unit from "hour" to "min", we will scale the instance by a factor of 60, a reasonable allocation algorithm should be identical on these two instances (since they are in fact the same instance). We provide a refined characterization for all the scale-free truthful mechanisms with finite approximation ratio. Based on this characterization, we prove a lower bound of 1.5625 using Yao's min-max principle [17]. We design a distribution of instances and argue that any scale-free deterministic truthful mechanisms cannot get an expected approximation ratio which is better than 1.5625. In order to get a better lower bound, we use a limitation argument and this value of 1.5625 holds when the number of tasks approaches infinity. So this lower bound only works for instances with a sufficiently large number of tasks. As we have a better mechanism for scheduling 2 tasks, we also study the lower bound of this special case under the assumption of scale-free. The instances used in the general lower bound cannot give a bound which is better than 1.5 when each of them only contains 2 tasks. So we choose a more carefully designed instances distribution to get a lower bound of 1.506. All these lower bound suggests that the lower bound of 1.5 may not be tight. However, it remains open to prove a better lower bound without any assumption.

A lot of technical effort in this work is given to parameter optimization both for the mechanism design part and lower bound proof part. Such optimization is also critical in this problem since the gap between the known upper bound and lower bound is already quite tiny. For example, for the 2 task case, the approximation ratio of the correlation mechanism we provided is 1.5089, while the lower bound is 1.5. There is only a gap of 0.0089. A very carefully designed instance distribution gives a better lower bound of 1.506.

Despite the fact that our lower bounds rely on a technical assumption, we feel it is interesting for several reasons. First we think the assumption of scale-free is very natural, it is hard to imagine that some scale dependant mechanism can beat all the scale free mechanisms. So we conjecture that these lower bounds are also true for all the randomized truthful mechanisms. On the other hand, if one believes that a better mechanism exists, one has to look for really new mechanisms which are not scale-free. In both cases, we believe that our work in this paper is an important step toward the exact bound.

1.1 Related Work

Scheduling unrelated machines is one variant of the most fundamental scheduling Problems. For this NP-hard optimization problem, there is a polynomial-time 2-approximation algorithm, and unless $P = NP$, it is impossible to approximate the optimum within a factor less than $3/2$ in polynomial time[11]. However there is no corresponding payment strategy to make the above allocation algorithm truthful.

In the mechanism design setting, Lavi and Swarmy considered a restricted variant, where each task j only has two values of running time (small time L_j or big time H_j), and gave a 3-approximation randomized truthful mechanism [10]. They first use the cycle monotonicity in designing mechanisms and applied the LP rounding idea based on [9].

For the lower bounds side, Christodoulou, Koutsoupias and Vidali improved the lower bound from 2 to $1 + \sqrt{2}$ when the number of machines is at least 3 [3], and then to 2.618 when the number of machines is sufficiently large [8]. In a recent beautiful work by Ashlagi, Dobzinski and Lavi, an optimal lower bound (m) was proved for all anonymous truthful mechanisms[1].

Christodoulou, Koutsoupias and Vidali gave a characterization for all truthful mechanisms in the same setting as this paper, including those with unbounded approximation ratio [4].

In [2], Christodoulou, Koutsoupias and Kovács considered the fractional version of this problem, in which each task can be split among the machines. For this version, they gave a lower bound of $2 - 1/m$ and an upper bound of $(m + 1)/2$, where m is the number of machines. We remark that these two bounds are closed for the case of two machines as in the integral deterministic version. So to explore the exact bound for the randomized version seems very interesting and desirable.

2 Notations and Preliminaries

In this section we review some definitions and results on mechanism design and the scheduling problem. In the following, for a generic matrix $\mathbf{a} = (a_{ij})$, we use a_i to denote the i-th row of the matrix, and $\mathbf{a}_{-i}$ to denote the matrix obtained from $\mathbf{a}$ deleting $\mathbf{a}_i$. We also use $(\mathbf{v}, \mathbf{a}_{-i})$ to denote the matrix obtained from $\mathbf{a}$ by replacing $\mathbf{a}_i$ with vector $\mathbf{v}$. We use $\mathbb{R}_+$ to denote the set of non-negative real numbers.

In a scheduling problem, there are n tasks and m machines (in this paper, we mainly consider the case where $m = 2$), where each machine $i \in [m]$ needs t_{ij} units of time to perform task $j \in [n]$. We usually use the matrix $\mathbf{t} = (t_{ij})$ to denote an instance of the scheduling problem. In this paper, we consider that each machine is controlled by a strategic player. We assume that player i privately knows $\mathbf{t}_i$, and we call the vector $\mathbf{t}_i$ player i's type. After each player i declares his/her type, an allocation algorithm X will decide an allocation of all the tasks. We assume that all the players are selfish and want to perform the least amount of tasks as possible, so players may misreport their types. We use $\mathbf{b}_i \in \mathbb{R}_+^n$ to denote player i's reported type, and call it player i's bid. Obviously $\mathbf{b}_i$ may not equal to $\mathbf{t}_i$ if that helps player i's interest. To avoid this lying issue, we introduce the payment algorithm P into a mechanism. Formally, a mechanism $M = (X, P)$ consists of two parts:

- **An allocation algorithm:** the allocation algorithm X, given the input of players bid matrix $\mathbf{b} = (\mathbf{b}_1, \cdots, \mathbf{b}_m)$, outputs an allocation denoted by a matrix $\mathbf{x} = (x_{ij})$. x_{ij} is 1 if task j is assigned to machine i, and 0 otherwise. Every task must be completely assigned, hence $\sum_{i\in[m]} x_{ij} = 1$, $\forall j \in [n]$.
- **A payment algorithm:** the payment algorithm P, given the input of players bid matrix $\mathbf{b}$, outputs a vector $\mathbf{p} = (p_1, \cdots, p_m)$, where p_i denotes the money that player i receives from the mechanism.

A mechanism is deterministic if both its allocation and payment algorithms are deterministic. If at least one of the algorithms uses random bits, the mechanism is called randomized.

Now we specify the utility of each player. We use the quasi linear utility, which means the utility u_i of player i with type $\mathbf{t}_i$ over an allocation $\mathbf{x}$ and money p_i is defined as:

$$u_i(\mathbf{x}, p_i|\mathbf{t}_i) = p_i - \sum_{j\in[n]} x_{ij} t_{ij}.$$

In deterministic mechanisms, both $\mathbf{x}$ and p_i are functions of bid matrix $\mathbf{b}$, we can also write the utility as

$$u_i(\mathbf{b}|\mathbf{t}_i) = p_i(\mathbf{b}) - \sum_{j\in[n]} x_{ij}(\mathbf{b}) t_{ij}.$$

Recalling that we want to solve the issue of lying about types, we are interested in truthful mechanisms. A mechanism $M = (X, P)$ is truthful if for each player i, reporting his/her true type will maximize his/her own utility. Formally, for any i, any bids $\mathbf{b}_{-i}$ of all other players, we have

$$u_i((\mathbf{t}_i, \mathbf{b}_{-i})|\mathbf{t}_i) \geq u_i((\mathbf{b}_i, \mathbf{b}_{-i})|\mathbf{t}_i), \quad \forall \mathbf{b}_i \in \mathbb{R}_+^n$$

In randomized mechanisms, both x_{ij} and p_i are random variables. There are two versions of truthfulness for randomized mechanisms. The stronger version is *universally truthful*, which requires the mechanism to be truthful when fixing all the random bits. The weaker version is *truthful in expectation*, which only requires that for each player, reporting his/her true type will maximize his/her own expected utility. In this paper, we focus on the stronger version, universally truthful.

For a truthful mechanism $M = (X, P)$, we may assume that all the players will report their true types, hence $\mathbf{b} = \mathbf{t}$. Now, how can we evaluate the performance of the mechanism's allocation algorithm X? We consider the makespan, which is the maximum load of all the machines. Given input $\mathbf{t}$, the makespan of mechanism M is denoted by $l_M(\mathbf{t})$, and $l_M(\mathbf{t}) = \max_{i\in[m]} \sum_{j\in[n]} x_{ij}t_{ij}$. We use $l_{opt}(\mathbf{t})$ to denote the optimum, and $l_{opt}(\mathbf{t}) = \min_x \max_{i\in[m]} \sum_{j\in[n]} x_{ij}t_{ij}$. A mechanism M is called a c-approximation mechanism if for any instance $\mathbf{t}$, we have $l_M(\mathbf{t}) \leq c{\cdot}l_{opt}(\mathbf{t})$. For randomized mechanism M, we require $E[l_M(\mathbf{t})] \leq c{\cdot}l_{opt}(\mathbf{t})$, where the expectation is over the random bits used in the mechanism.

Definition 1 (Task-Independent Mechanisms)**.** *A deterministic mechanism M is task-independent, if for any two bid matrices $\mathbf{b}$, $\mathbf{b}'$ such that $b_{ij} = b'_{ij}$ for all $i \in [m]$, the allocation of task j is identical, i.e. $x_{ij}(\mathbf{b}) = x_{ij}(\mathbf{b}'), \forall i \in [m]$.*

For randomized mechanisms, there are also two versions of task-independence. One is a weak task-independent mechanism, which is a distribution of several task-independent deterministic mechanisms. The other is a strong task-independent mechanism, which satisfies that not only does the allocation of task j not change as long as j's column of $\mathbf{b}$ does not change, but also all the random variables x_{ij} are independent between different tasks.

We quote a theorem from [6] (Theorem 4.5 in [6]), which gives a characterization for truthful mechanisms for scheduling two machines.

Theorem 1 ([6])**.** *Let M be a mechanism for minimizing the makespan for 2 machines that provides a finite approximation ratio. Then M is task independent.*

This theorem implies that any randomized truthful mechanism with a finite approximation ratio is weak task-independent. In [13], Lu and Yu proved a lower bound of $\frac{11}{7}$ (≈ 1.5714) for all the *strong task-independent* truthful mechanisms. Given these two facts, we have the following interesting open question:

Question 1. Does there exist a weak task-independent randomized truthful mechanism which provides a better approximation ratio ($< \frac{11}{7}$)?

Definition 2 (Scale-Free Mechanisms)**.** *We call an allocation algorithm scale-free if for any instance $\mathbf{b}$ and any non-zero constant λ, the outputs of the algorithm on the input $\mathbf{b}$ and $\lambda\mathbf{b}$ are identical. A mechanism is called scale-free if its allocation algorithm is scale-free. A randomized mechanism is called scale-free if it is a distribution of deterministic scale-free mechanisms.*

Together with the properties of scale-free and task-independent, the allocation of a task j only depends on the ratio of the two bids $\frac{b_{1j}}{b_{2j}}$ for this task. Then using the monotone theorem of truthful mechanism [15], we have the following characterization of scale-free task-independent truthful mechanisms.

Theorem 2. *A deterministic scale-free task-independent truthful mechanism for scheduling two unrelated machines is of the following form: there are n thresholds α_j ($j \in [n]$) for n tasks. For every task $j \in [n]$, the mechanism allocates it to the first machine iff $b_{1j} < \alpha_j b_{2j}$ (or $b_{1j} \leq \alpha_j b_{2j}$).*

For randomized mechanisms, these thresholds are random variables that do not depend on player's bid. In strong task-independent mechanisms, these random variables are further required to be independent. While in weak task-independent mechanisms, they may have some correlation.

Our lower bounds in this paper are proved by Yao's min-max principle [17], which is a typical tool used to prove lower bounds of randomized mechanisms (algorithms, protocols, etc). Based on the characterization Theorem 1, We state the principle in our setting as following.

Lemma 1. *Given a distribution of instances, if any (scale-free) task independent deterministic mechanism cannot have an expected approximation ratio better than α, then α is a lower bound for all the (scale-free) universal truthful randomized mechanisms.*

3 A Lower Bound of 1.5625

In this section, we prove a lower bound of 1.5625 for all the randomized scale-free truthful mechanisms. Let k be an integer and $a > 1$ be a parameter specified later. We consider the following instances distribution. There are $k+1$ instances, each containing $k+1$ tasks. All the instances have equal probability, i.e. a probability of $\frac{1}{k+1}$. The i-th ($1 \leq i \leq k+1$) instance is as following: the running times of the i-th task are ka and ka^2 for the first and second machines respectively; and the running times of the other k tasks are 1 and a for the first and second machines respectively.

For the i-th instance, the optimal solution is to allocate the i-th task to the first machine and the remaining k tasks to the second machine; the optimal makespan is ka for every instance. Now we consider the performance of a deterministic scale-free task independent mechanism on these instances. Since in every instance and for every task, the ratio of two running times is the same (i.e. equal to a), every deterministic scale-free task-independent mechanism will allocate the same task (tasks with the same number) in different instances in the same way. This means that if the mechanism assigns task 1 to the first machine in the first instance, then it must assigns task 1 to the first machine in all the instances. Now we assume that the mechanism assign t tasks in the first instance to the first machine, then the behavior of the mechanism on all these instances is completely fixed. By the symmetry of the tasks, w.o.l.g, we can assume that the mechanism assigns the first t tasks to the first machine. Now we can calculate the expected approximation ratio of the mechanism on this distribution of instances.

For the first t instances, the makespan is the load of the first machine, which is $ka+(t-1)\times 1 = ka+t-1$. For the other $k+1-t$ instances, the makespan is the load of the second machine, which is $ka^2 + (k+1-t-1)a = ka^2 + (k-t)a$.

Therefore the expected approximation ratio R of the mechanism on these instances is

$$\frac{t(ka+t-1)}{ak(k+1)}+\frac{(k+1-t)(ka^2+(k-t)a)}{ak(k+1)} = \frac{a+1}{ak(k+1)}(t^2-(ak+1)t+a(k^2+k)).$$

For any fixed k and $a > 1$, this value R is a quadratic polynomial of t. So we have

$$R \geq \frac{a+1}{ak(k+1)}(a(k^2+k) - \frac{(ak+1)^2}{4}).$$

For a sufficiently large k, the ratio in the RHS approachs the ratio of the k^2 terms which is $a+1-\frac{a(a+1)}{4}$. When $a = \frac{3}{2}$, this expression reaches its maximum value $\frac{25}{16} = 1.5625$.

By Yao's min-max principle, this instance's distribution gives a lower bound of 1.5625. We remark that this lower bound only occurs for a sufficiently large number of tasks.

Theorem 3. *Any randomized scale-free truthful mechanism for scheduling two unrelated machines can not have an approximation ratio which is better than* 1.5625*.*

4 Correlation Gives Better Mechanisms

In this and the next sections, we study a further restricted case, namely scheduling two tasks on two unrelated machines. This seems a very special setting, but we believe it is still very interesting for several seasons. First, we will prove that previous lower bounds (1.5 in general and 1.57 for strong task-independent mechanisms) both hold even for this special case. Second, from a pure mathematical point of view, this is the simplest non-trivial setting, however the exact bound for this simplest case is still unclear. Third, the techniques and ideas developed here for studying this special setting may extend to more general settings. For example, the characterization in [4] is first proved for the 2 task case and then extends to many tasks.

The proof for the lower bound of 1.5 in [14] requires at least 3 tasks. Here we prove that this is also true for two tasks.

Lemma 2. *Any randomized truthful mechanisms for scheduling two tasks on two machines cannot have an approximation ratio that is better than* 1.5*.*

Proof. We consider a distribution of the following two instances, each with probability of $\frac{1}{2}$.

	task 1	task 2
machine 1	1	1
machine 2	1	2

,

	task 1	task 2
machine 1	1	2
machine 2	1	1

.

Any deterministic task-independent mechanism will assign task 1 in these two instances in the same way. By symmetry, we can assume that the mechanism assigns task 1 to machine 1. Then the makespan of the mechanism for the first instance is at least 2 no matter how it allocates task 2. However, the optimal makespan is 1; therefore, the expected approximation ratio of this mechanism on these two instances is at least $\frac{2+1}{2} = 1.5$. By Yao's min-max principle, 1.5 is a lower bound for all the randomized truthful mechanisms.

The proof for a lower bound of $\frac{11}{7}$ in [13] only uses instances with 2 tasks, so this bound also holds for the this special setting.

Lemma 3. *Any strong task-independent randomized truthful mechanism scheduling two tasks on two machines cannot have an approximation ratio that is better than* $\frac{11}{7}(\approx 1.5714)$.

Given these two lower bounds. It is interesting to see if this bound of $\frac{11}{7}$ can be beaten. The answer is affirmed by the following correlation mechanism, which also partially answers Question 1.

Let $f : \mathbb{R}^+ \rightharpoonup [0, 1]$ be a non-decreasing monotone function, where $\mathbb{R}^+ = \{x \in R | x \geq 0\}$, $f(0) = 0$, $\lim_{x\to\infty} f(x) = 1$ and $f(x) + f(1/x) = 1$. The correlation mechanism for scheduling two tasks on two machines is described in Figure 1.

Input: The reported bid matrix b.
Output: A randomized allocation x
and a payment $p = (p_1, p_2)$.
Allocation and Payment Algorithm:
$x_{1j} \leftarrow 0, x_{2j} \leftarrow 0, j = 1, 2.$
$p_1 \leftarrow 0; p_2 \leftarrow 0.$
Choose $\alpha \in R^+$ randomly according to function f.
if $b_{11} < \alpha b_{21}$,
    $x_{11} \leftarrow 1, p_1 \leftarrow p_1 + \alpha b_{21};$
else
    $x_{21} \leftarrow 1, p_2 \leftarrow p_2 + \alpha^{-1} b_{11}.$
if $b_{22} < \alpha b_{12}$,
    $x_{22} \leftarrow 1, p_2 \leftarrow p_2 + \alpha b_{12};$
else
    $x_{12} \leftarrow 1, p_1 \leftarrow p_1 + \alpha^{-1} b_{22}.$

Fig. 1. The Correlation Mechanism

It is easy to show that this mechanism is universally truthful for any function f with the properties listed above. When the random variable α is fixed, it is a task-independent mechanism and for each task it is simply a weighted VCG mechanism. The main new idea in this mechanism is that there are some correlation of randomness for different tasks. Here the random variable α is used both in the mechanisms for the first task and the second task. The intuitive argument is like this. If $\alpha > 1$, then the mechanism is biased to the first machine for the first task and to the second machine for the second task. If $\alpha < 1$, it is the other way round. The different bias for different tasks makes the allocation more balanced. Here the requirement of $f(x)+f(1/x)=1$ makes the mechanism symmetrical for the two machines and for the two tasks.

Now we analyze the performance formally. We consider the following generic instance.

	task 1	task 2
machine 1	b_{11}	b_{12}
machine 2	b_{21}	b_{22}

The expected makespan t of the correlation mechanism on this instance is

$$t = (b_{11}+b_{12})Pr(\alpha > \frac{b_{11}}{b_{21}}, \alpha \le \frac{b_{22}}{b_{12}}) + (b_{21}+b_{22})Pr(\alpha \le \frac{b_{11}}{b_{21}}, \alpha > \frac{b_{22}}{b_{12}})$$
$$+ \max(b_{11}, b_{22})Pr(\alpha > \frac{b_{11}}{b_{21}}, \alpha > \frac{b_{22}}{b_{12}}) + \max(b_{12}, b_{21})Pr(\alpha \le \frac{b_{11}}{b_{21}}, \alpha \le \frac{b_{22}}{b_{12}}).$$

Since all the probabilities in the above expression can be expressed by function values of f, its performance can be estimated at least numerically (and by computer) when the function is given. Here we specify the following simple function f so that the analysis can be done analytically (and by hand). It is a case-by-case analysis and is omitted here due to space limitation.

$$f(x) = \begin{cases} 1, & x \ge A, \\ \frac{1}{2} + \frac{x-1}{2(A-1)}, & 1 \le x < A, \\ \frac{1}{2} - \frac{\frac{1}{x}-1}{2(A-1)}, & \frac{1}{A} \le x < 1, \\ 0, & 0 \le x < \frac{1}{A}. \end{cases} \tag{1}$$

Despite the complicated appearance in the above expression, this function is a simple and natural one. A is a threshold, when input is beyond that, the function value is always 1. $f(x)+f(1/x)=1$ requires that $f(1)=\frac{1}{2}$. The function between 1 and A is simply the the line segment connecting these two end points $(1, \frac{1}{2})$ and $(A, 1)$. The function below 1 is determined by the function above 1 and the requirement $f(x)+f(1/x)=1$.

Theorem 4. *By using the function as* (1) *, where* $A = -\frac{1}{2}+\sqrt{3}+\frac{1}{2}\sqrt{25-12\sqrt{3}}$ ($\approx$2.26), *the approximation ratio of the Correlation Mechanism is* $\frac{1}{6}(\sqrt{25-12\sqrt{3}}+7)$ (≈ 1.5089).

We remark that the function of (1) is only used to illustrate the idea of correlation mechanisms. It is by no means the best choice. However its bound (1.5089) is already very close to the lower bound (1.506) we will prove in the next section.

5 The Ratio of 1.5 Is Not Achievable

Given the succuss of using correlation in the previous section, and also noticing that the instances used to prove the lower bound in Section 3 cannot get any thing beyond 1.5 for the case of two tasks, one may make a point that we can choose some suitable function f in the correlation mechanism to achieve an approximation ratio of exactly 1.5. In this section, we prove that this is impossible.

Theorem 5. *For any non-decreasing monotone function* $f : \mathbb{R}^+ \rightharpoonup [0,1]$ *which satisfies* $\forall x \in \mathbb{R}^+, f(x) + f(1/x) = 1$, *there exists* $c, d \in R^+$ *such that* $c \geq 1$, $c \geq d$ *and*

$$c + f(c) + df(c) - cf(c) - df(d) > 1.5.$$

Proof. We assume for contradiction that there exists a function f such that for all $c, d \in R^+$ satisfying $c \geq 1$, $c \geq d$, we have

$$c + f(c) + df(c) - cf(c) - df(d) \leq 1.5. \tag{2}$$

For any fixed d, let $c \to \infty$, we have $c + f(c) + df(c) - cf(c) \geq 1 + d$. So for any $x \in R^+$, we have

$$f(x) \geq 1 - \frac{1}{2x}.$$

Using this and the fact that $\forall x \in R^+, f(x) + f(1/x) = 1$, we have

$$f(x) \leq \frac{x}{2}.$$

Let $d = 1/c$ in (2), we have

$$c + f(c) + \frac{f(c)}{c} - cf(c) - \frac{1}{c}(1 - f(c)) \leq 1.5.$$

This implies

$$(c-2)(c+1)f(c) \geq (c-2)(c+1/2).$$

So for $c > 2$ we have $f(c) \geq 1 - \frac{1}{2(c+1)}$ and for $1 \leq c < 2$ we have $f(c) \leq 1 - \frac{1}{2(c+1)}$. Together with the fact that f is a non-decreasing monotone, these two inequalities enforce that $f(2) = 1 - \frac{1}{2\times(2+1)} = \frac{5}{6}$.

Now let $c = 2$ in (2), we have

$$2 + \frac{5}{6} + \frac{5}{6}d - 2 \times \frac{5}{6} - df(d) \leq 1.5.$$

This implies that for any $x \leq c = 2$, $f(x) \geq \frac{5}{6} - \frac{1}{3x}$. But f cannot simultaneously satisfy this and $f(x) \leq \frac{x}{2}$. For example choosing $x = \frac{4}{5}$, $f(\frac{4}{5}) \geq \frac{5}{6} - \frac{1}{3x} = \frac{5}{12}$, but on the other hand $f(\frac{4}{5}) \leq \frac{x}{2} = \frac{2}{4} < \frac{5}{12}$, a contradiction.

We can improve this theorem by proving that any scale-free mechanisms cannot achieve 1.5 even for this very special cases.

Theorem 6. *Any randomized scale-free truthful mechanism for scheduling two tasks on two unrelated machines cannot have an approximation ratio that is better than* 1.506*.*

Proof. We use a matrix $\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ to denote the following instance with two machines and two tasks

	task 1	task 2
machine 1	b_{11}	b_{12}
machine 2	b_{21}	b_{22}

.

Now we consider the following distribution of 12 instances:

– $\begin{bmatrix} 1 & 1 \\ a & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 \\ 1 & a \end{bmatrix}$, $\begin{bmatrix} a & 1 \\ 1 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & a \\ 1 & 1 \end{bmatrix}$, each with probability p_1;
– $\begin{bmatrix} 1 & b \\ \frac{1}{b} & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & \frac{1}{b} \\ b & 1 \end{bmatrix}$, $\begin{bmatrix} b & 1 \\ 1 & \frac{1}{b} \end{bmatrix}$, $\begin{bmatrix} \frac{1}{b} & 1 \\ 1 & b \end{bmatrix}$, each with probability p_2;
– $\begin{bmatrix} 1 & b \\ \frac{1}{a} & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & \frac{1}{a} \\ b & 1 \end{bmatrix}$, $\begin{bmatrix} b & 1 \\ 1 & \frac{1}{a} \end{bmatrix}$, $\begin{bmatrix} \frac{1}{a} & 1 \\ 1 & b \end{bmatrix}$, each with probability p_3.

a, b, p_1, p_2, p_3 are parameters to be specified later and satisfy $1 \leq b \leq 2 \leq a$ and $p_1 + p_2 + p_3 = 1/4$.

For these instances, the possible running-time-ratios of tasks are only $1/a$,$1/b$, 1,a,b. For each task, the mechanism can choose a threshold (only 6 possible different thresholds). But by symmetry, we can always assume that the thresholds for the first task are above 1 (so there are 3 possible different thresholds). Overall of there are 18 possible different mechanisms. We can choose the parameters such that the expected approximation ratios of them are all larger than a given value, then this is our lower bound.

By choosing $a = 2.125, b = 1.88, p_1 = 0.1346, p_2 = 0.0796, p_3 = 0.0358$, we can get a lower bound of 1.506. We can prove this formally and also argue that these are the best parameters we can choose. The details are omitted.

6 Conclusion and Discussion

The main results of this paper are two new lower bounds and one new upper bound. Two direct interesting open questions are to get rid of the technical assumption for these lower bounds and to generalize the correlation mechanism to general cases. It is quite surprising that the exact bound for this simple 2-player mechanism has not been settled after a couple of work. We recall that quite simple mechanisms and relatively easy lower bound proofs already match both in the corresponding deterministic and fraction version. We believe that our work in this paper is an important step toward the final answer.

In the general case (m machines), the gap between the best lower bounds (constants) and the best upper bounds ($\Theta(m)$) is huge both in deterministic and randomized versions. Any improvement in either direction is highly desirable. We hope that the technique and ideas we and others developed for this special case can extend to the general case.

Acknowledgement

I am grateful to Wei Chen, Yajun Wang, and Changyuan Yu for helpful discussion on this topic.

References

1. Ashlagi, I., Dobzinski, S., Lavi, R.: An Optimal Lower Bound for Anonymous Scheduling. In: Proceedings of EC 2009, pp. 169–176 (2009)
2. Christodoulou, G., Koutsoupias, E., Kovács, A.: Mechanism Design for Fractional Scheduling on Unrelated Machines. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 40–52. Springer, Heidelberg (2007)
3. Christodoulou, G., Koutsoupias, E., Vidali, A.: A lower bound for scheduling mechanisms. In: The Proceedings of SODA 2007, pp. 1163–1170 (2007)
4. Christodoulou, G., Koutsoupias, E., Vidali, A.: A Characterization of 2-Player Mechanisms for Scheduling. In: Halperin, D., Mehlhorn, K. (eds.) Esa 2008. LNCS, vol. 5193, pp. 297–307. Springer, Heidelberg (2008)
5. Clarke, E.H.: Multipart Pricing of Public Goods. Public Choice 11, 17–33 (1971)
6. Dobzinski, S., Sundararajan, M.: On Characterizations of Truthful Mechanisms for Combinatorial Auctions and Scheduling. In: Proceedings of EC 2008, pp. 38–47 (2008)
7. Groves, T.: Incentives in Teams. Econometrica 41, 617–631 (1973)
8. Koutsoupias, E., Vidali, A.: A Lower Bound of $1 + \phi$ for Truthful Scheduling Mechanisms. In: Kučera, L., Kučera, A. (eds.) MFCS 2007. LNCS, vol. 4708, pp. 454–464. Springer, Heidelberg (2007)
9. Lavi, R., Swamy, C.: Truthful and Near-Optimal Mechanism Design via Linear Programming. In: Proceedings of FOCS 2005, pp. 595–604 (2005)
10. Lavi, R., Swamy, C.: Truthful mechanism design for multi-dimensional scheduling via cycle-monotonicity. In: Proceedings EC 2007, pp. 252–261 (2007)
11. Lenstra, J.K., Shmoys, D.B., Tardos, É.: Approximation algorithms for scheduling unrelated parallel machines. Math. Program. 46(3), 259–271 (1990)
12. Lu, P., Yu, C.: An Improved Randomized Truthful Mechanism for Scheduling Unrelated Machines. In: Proceedings of STACS 2008, pp. 527–538 (2008)
13. Lu, P., Yu, C.: Randomized Truthful Mechanisms for Scheduling Unrelated Machines. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 402–413. Springer, Heidelberg (2008)
14. Mu'alem, A., Schapira, M.: Setting lower bounds on truthfulness. In: Proceedings of SODA 2007, pp. 1143–1152 (2007)
15. Nisan, N., Ronen, A.: Algorithmic mechanism design (extended abstract). In: Proceedings of STOC 1999, pp. 129–140 (1999)
16. Vickrey, W.: Counterspeculation, Auctions, and Competitive Sealed Tenders. Journal of Finance 16, 8–37 (1961)
17. Yao, A.: Probabilistic computations: Toward a unified measure of complexity. In: Proceedings of FOCS 1977, pp. 222–227 (1977)