Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Truthful mechanisms for two-range-values variant of unrelated scheduling

Changyuan Yu *

Institute for Theoretical Computer Science, Tsinghua University, Beijing, 100084, PR China

ARTICLE INFO

Article history:
Received 8 July 2008
Received in revised form 13 January 2009
Accepted 1 February 2009
Communicated by D.-Z. Du

Keywords:
Truthful mechanism
Approximation algorithm
Scheduling

ABSTRACT

In this paper, we consider a restricted variant of the scheduling problem, where the machines are the strategic players. For this multi-parameter mechanism design problem, the only known truthful mechanisms use task independent allocation algorithms and only have approximation ratio $O(m)$ [N. Nisan, A. Ronen. Algorithmic mechanism design (extended abstract), in: STOC'99: Proceedings of the thirty-first annual ACM symposium on Theory of computing, ACM, New York, NY, USA, 1999. pp. 129–140; A. Mu'alem, M. Schapira, Setting lower bounds on truthfulness: Extended abstract, in: SODA'07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1143–1152; P. Lu, C. Yu, An improved randomized truthful mechanism for scheduling unrelated machines, in: 25th International Symposium on Theoretical Aspects of Computer Science, STACS, 2008, pp. 527–538; P. Lu, C. Yu, Randomized truthful mechanisms for scheduling unrelated machines, in: C.H. Papadimitriou, S. Zhang (Eds.), Proceedings of WINE, in: Lecture Notes in Computer Science, vol. 5385, Springer, 2008, pp. 402–413]. Lavi and Swamy first use the cycle monotone condition and design a 3-approximation truthful mechanism for a two value variant in [R. Lavi, C. Swamy, Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity, in: EC'07: Proceedings of the 8th ACM conference on Electronic commerce, ACM, New York, NY, USA, 2007, pp. 252–261], where the processing time of task j on machine i, say t_{ij}, can only be either a lower value L_j or a higher value H_j. We consider a generalized variant, where t_{ij} lies in $[L_j, L_j(1+\epsilon)] \bigcup [H_j, H_j(1+\epsilon)]$ and ϵ is a parameter satisfying some condition. We consider two special cases, case A when $H_j/L_j > 2, \forall j$ and case B when $H_j/L_j \leq 2, \forall j$, and give randomized truthful mechanisms with approximation ratio $4(1+\epsilon)$ for both cases. Based on these two cases' results, we are also able to deal with the general case of our two-range-values scheduling problem. We use a combination of two mechanisms, which is also a novel method in mechanism design for scheduling problems, and finally we give a randomized truthful mechanism with approximation ratio $7(1+\epsilon)$.

Although the generalization seems a little incremental, we actually use a very novel technique in the key step of proving truthfulness for case A, as well as a new mechanism scheme for case B. Besides, the results in this paper are the first truthful mechanisms with constant approximation ratios when a machine (player) can report infinitely possible values, which is quite different from the two value variant, in which only finite values are available. Furthermore, together with Lavi and Swamy's work, our results suggest that such a task-dependent approach can really do much better for the scheduling unrelated machines problem.

* Tel.: +86 13426333048.
E-mail address: yucy05@mails.tsinghua.edu.cn.

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.02.001

1. Introduction

Mechanism design, as an important area both in game theory and computer science, has received extensive study in the past few years. A mechanism consists of several algorithms, which can acquire information from some players and achieve certain global objectives. However, the players are rational and selfish, so they may misreport their information to the mechanism on their own behalves. To overcome this difficulty, the most common way is to require the "truthfulness" of the mechanism, which implies that for any player, reporting his/her values truthfully to the mechanism will always maximize his/her own utility, no matter how other players act. The most celebrated result in mechanism design is the VCG family mechanisms [31,11,14], which applies when the global objective is to optimize the total sum of all the agents' utilities.

The study of the algorithmic aspects of mechanism design was initiated by Nisan and Ronen in [28]. They focused on the problem of scheduling unrelated machines, where there are n tasks to be allocated to m machines. The machines are selfish players and machine i privately knows $t_{ij}, \forall j$, which is the processing time of task j on machine i. The objective of this problem is to find an allocation so that the maximal completion time (named makespan) is minimized. In [28], they mainly consider how to design truthful mechanisms which can approximate the optimal makespan well. They gave an m-approximation truthful mechanism, and proved no truthful mechanism can have approximation ratio less than 2. This lower bound 2 has been improved to 2.41 for $m \geq 3$ recently in [10], then to 2.61 for sufficiently large m in [16].

Fractional and randomized versions are also studied, however still $O(m)$-approximation mechanisms can be obtained [9,24,25]. They design mechanisms based on a so-call task independent approach, which assigns tasks, one by one, independently. This method already achieves its limit, since almost tight lower bounds are also shown in both cases [9,25].

Archer and Tardos first studied the variant of scheduling related machines in [5]. In such a variant, machine i's private information is simply a single parameter s_i, which specifies the speed of machine i. This variant falls into the one-parameter domain, and becomes easier. There is even a truthful mechanism which can always output an optimal allocation if exponential running time is allowed. For polynomial time approximation mechanisms, a lot of positive results are also obtained. The reason is that in one-parameter domain, truthfulness has an equivalent characterization, value monotone condition, which is easy to apply in the design of mechanism. However, the generalization of value monotonicity in multi-parameter domain, which is called cycle monotonicity, is really difficult to use.

Lavi and Swamy first use this cycle monotonicity in mechanism design for multi-parameter domain. They considered a restricted variant of scheduling unrelated machines, in which t_{ij} is either L_j or H_j, and gave a 3-approximation randomized truthful mechanism [22]. They used an LP rounding idea based on [21]. For the special case when $L_j = L, H_j = H, \forall j$, they gave a deterministic truthful mechanism with approximation ratio 2.

1.1. Our results

In this paper, we generalize Lavi and Swamy's two-value variant such that t_{ij} is in either a relatively lower values range $[L_j, L_j(1+\epsilon)]$ or a relatively higher values range $[H_j, H_j(1+\epsilon)]$, where $H_j > L_j$, and ϵ is some positive parameter satisfying $\epsilon \leq \frac{1}{16mn}$. The information of L_j, H_j and ϵ are all publicly known to the mechanism and the agents.

This general case is difficult, and we consider two special cases corresponding to $H_j/L_j > 2, \forall j$ and $H_j/L_j \leq 2, \forall j$, which we called case A and case B respectively. We present $4(1+\epsilon)$-approximation randomized truthful mechanisms for both special cases, and combine them into an $7(1+\epsilon)$-approximation randomized truthful mechanism for the general case.

For case A, following the idea in [22], we first round the reported values into a two-value case. Then we obtain a (fractional) solution $\tilde{x}$ output by some algorithm with good conditions, and use a *spreading algorithm* to convert it into another fractional solution x. This converting process will make the allocation x satisfy some *separation bound*, which is important in our proof for cycle monotonicity. Finally, we use a rounding algorithm to obtain a random integer solution X.

In Lavi and Swamy's work, each machine can only have finite choices of values, and they prove the cycle monotonicity holds even for every task. In our variant, there are infinitely many possible values, and their method does not work. The idea we use here is to estimate the cycle's sum globally, instead of just considering one task. We first reduce the cycle to a simple case where adjacent nodes are different in some sense. Then we divide the terms into two classes, the gain and the loss term. Finally, we estimate the gain and the loss to the cycle's sum separately, which leads to non-negativeness of the cycle's sum, i.e., the cycle monotonicity condition.

For case B, we first design a fractional truthful mechanism directly, then use rounding. In the fractional truthful mechanism, we assign each task to machines proportionally to the inverse of their bids.

We emphasize that this is the first work with constant approximation ratio in designing truthful mechanisms when each task has infinitely many possible values. Our mechanisms are certainly task-dependent and use LP rounding, which demonstrates the use of fractional variant and the potential of task-dependent mechanisms.

1.2. Related work

As a computational problem, scheduling unrelated machines problem is an NP-hard optimization problem, and there are several polynomial time algorithms achieving the best known approximation ratio 2. In the negative side, unless $P = NP$, it

Please cite this article in press as: C. Yu, Truthful mechanisms for two-range-values variant of unrelated scheduling, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.02.001

is impossible to approximate the optimum within a factor less than 1.5 in polynomial time [23]. If the number of machines is bounded by some constant, Angel, Bampis and Kononov gave an FPTAS [2].

From the mechanism design setting, beyond Nisan and Ronen's work we mentioned above, Christodoulou, Koutsoupias and Kovács gave a fractional truthful mechanism with approximation ratio of $(m+1)/2$, and a lower bound of $2-1/m$ for any fractional truthful mechanisms. They also defined a broad class of allocation algorithms named *task-independent* algorithm, in which tasks are allocated one by one, independently. For the task-independent truthful fractional mechanisms, they proved a tight lower bound of $(m+1)/2$ [9].

Randomized algorithms are usually more powerful, and this is also true for this scheduling problem. Nisan and Ronen provided a randomized truthful mechanism with approximation ratio of 1.75 for two machines [28]. Recently Mu'alem and Schapira gave a lower bound of $2-1/m$ for the randomized truthful mechanisms [26]. They also generalized the 1.75 approximation mechanism for two machines to a $0.875m$-approximation mechanism for m machines. In [24,25], we first improved the upper bound 1.75 to 1.67, then to 1.59 for two machines case, and improved $0.875m$ to $(m+5)/2$. We also show almost tight lower bounds, 1.57 and $(m+1)/2$ respectively, for task independent mechanisms.

For scheduling related machines problem, Archer and Tardos gave a 3-approximation randomized truthful mechanism, and the ratio was later improved to 2 by Archer [3]. Andelman et al. [1] gave the first deterministic polynomial mechanism with an approximation ratio of 5, and the ratio was improved to 3, then to 2.8 both by Kovacs [17,18].

Another important line of work in mechanism design is combinatorial auctions and a lot of work is done (see [27,4,8,6,12,13]). In the study of truthfulness in the general domain, Saks and Yu [30] recently proved that for convex domain the Monotonicity Property characterizes the class of truthful mechanisms, generalizing results of [20,15,7].

2. Preliminary

2.1. Scheduling mechanisms

We first give some definitions and notations used in this paper. Throughout our paper, for a given generic matrix $\mathbf{a}=(a_{ij})$, we will use $\mathbf{a}_i$ to denote the i-th row of matrix $\mathbf{a}$ and $\mathbf{a}_{-i}$ to denote the matrix obtained from deleting i-th row in $\mathbf{a}$. We also use $(\mathbf{v},\mathbf{a}_{-i})$ to denote the matrix, where the i-th row of matrix $\mathbf{a}$ is replaced by the row vector $\mathbf{v}$.

- We use $[n]$ to denote the set of tasks, and $[m]$ to denote the set of machines (players). We use a matrix $\mathbf{t}=(t_{ij})$ to denote an instance, where t_{ij} is the processing time of task j on machine i.
- $\mathbf{t}_i$: the i-th row of the matrix $\mathbf{t}$, which is privately known to machine i, and is called i's type value.
- $\mathbf{b}=(b_{ij})$: the reported values of t_{ij}, called bid matrix. The i-th row $\mathbf{b}_i$ is reported by machine i.
- $M=(X,P)$: a mechanism. $\mathbf{X}(\mathbf{b})=(X_{ij}(\mathbf{b}))$ specifies the allocation, and $\mathbf{P}(\mathbf{b})=(P_i(\mathbf{b}))$ specifies the payments to each agent, where both of them are functions of the bid $\mathbf{b}$.
- $T_i(\mathbf{X})$: machine i's load under allocation X. $T_i(\mathbf{X})=\sum_j X_{ij}t_{ij}$.
- $T(\mathbf{X})$: the makespan of allocation $\mathbf{X}$, which is the maximal load under allocation $\mathbf{X}$. $T(\mathbf{X})=\max_{i\in[m]} T_i(\mathbf{X})$.
- $u_i(\mathbf{t}_i;\mathbf{b})$: the utility (profit) of machine i when its type value is $\mathbf{t}_i$ and the bid matrix is $\mathbf{b}$. $u_i(\mathbf{t}_i;\mathbf{b})=P_i(\mathbf{b})-T_i(\mathbf{X}(\mathbf{b}))$.

In this paper, we consider the following variant.

Two-range-values scheduling: For each task j, t_{ij} is either in a relatively lower values' range $[L_j, L_j(1+\epsilon)]$ or in a relatively higher values' range $[H_j, H_j(1+\epsilon)]$. H_j, L_j and ϵ are publicly known and satisfy $H_j > L_j$, $\epsilon \le \frac{1}{16mn}$. Without loss of generality, we also assume that $\epsilon, L_j > 0$, and $m, n \ge 2$. The objective of scheduling is to minimize the makespan.

Now we specify two special cases, depending on the ratios of H_j/L_j, $j\in[n]$.

- **case A:** $H_j/L_j > 2, \forall j \in [n]$.
- **case B:** $H_j/L_j \le 2, \forall j \in [n]$.

2.2. Truthfulness and characterizations

In such a mechanism design problem, the mechanism's objective is to minimize the makespan. However, each player only wants to maximize his/her utility and may not report the true type values to the mechanism. Therefore in this paper, we are interested in truthful mechanisms, in which telling the true values will maximize utility for each player.

Definition 2.1. A deterministic mechanism $M=(X,P)$ is truthful if, for each player i, to report the type values truthfully will maximize his/her utility, no matter how other players act. Formally, the mechanism $M=(X,P)$ is truthful if and only if, given any instance $\mathbf{t}=(t_{ij})$, for any i, any fixed $\mathbf{b}_{-i}$, any bid $\mathbf{b}_i$, we have:

$$P_i((\mathbf{t}_i,\mathbf{b}_{-i})) - \sum_j X_{ij}((\mathbf{t}_i,\mathbf{b}_{-i}))t_{ij} \ge P_i((\mathbf{b}_i,\mathbf{b}_{-i})) - \sum_j X_{ij}((\mathbf{b}_i,\mathbf{b}_{-i}))t_{ij}. \quad (1)$$

When randomness is involved, there are two versions of truthfulness: in the stronger version (i.e. *universally truthfulness*) the mechanism remains truthful even if the random bits are fixed; in the weaker version (i.e. *truthfulness in expectation*) an agent maximizes his/her expected utility when reporting values truthfully.

As we mentioned before, our problem falls into the multi-parameter domain, in which truthfulness has an equivalent characterization, i.e., the *cycle monotone condition*. This was first observed by Rochets [29] in 1987, and has been used widely in designing truthful mechanisms. But most applications are in one parameter domains, in which this cycle monotone condition boils down to certain value monotone condition and becomes more applicable. For our need and simplicity, we only include the following restatement of cycle monotonicity and the characterization of truthfulness.

Definition 2.2 (*Cycle Monotonicity [29,22]*). An allocation algorithm X satisfies cycle monotonicity if for any agent i, any fixed $\mathbf{b}_{-i}$, and i's bids $\mathbf{b}_i^1, \ldots, \mathbf{b}_i^K$, let $\mathbf{X}^k = \mathbf{X}(\mathbf{b}_i^k, \mathbf{b}_{-i})$, $1 \le k \le K$, and $\mathbf{X}^{K+1} = \mathbf{X}^1$, $\mathbf{b}_i^{K+1} = \mathbf{b}_i^1$. Then we have:

$$\sum_{k=1}^{K} \sum_{j=1}^{n} X_{ij}^{k+1} (b_{ij}^k - b_{ij}^{k+1}) \ge 0. \tag{2}$$

Theorem 2.1 (*[29,15]*). *There is a payment algorithm P, such that the mechanism $M = (X, P)$ is truthful if and only if the allocation algorithm X satisfies the cycle monotonicity.*

For the one-parameter domain, cycle monotonicity boils down to value monotonicity .

Theorem 2.2 (*[27,5]*). *In a one-parameter domain (only one task in scheduling problem), an allocation algorithm X admits a payment algorithm P to make the mechanism $M = (X, P)$ truthful if and only if X is monotone decreasing, which means that for any i, $\mathbf{b}_{-i}$ and $b_i \ge b_i'$, we have $X_i((b_i, \mathbf{b}_{-i})) \le X_i((b_i', \mathbf{b}_{-i}))$.*

The cycle monotonicity enables us to design the truthful mechanisms without considering the payment issue any more. So in this paper, we only give the allocation algorithms when specifying the mechanisms, and prove that they satisfy the cycle monotonicity.

3. Mechanism for case A

In this section, we consider case A: $H_j/L_j > 2, \forall j$. We generalize the black-box fashion technique in [22]. We give a general method to convert any c-approximation algorithm into a $4c$-approximation, truthful-in-expectation mechanism. To be self contained, we introduce Lavi and Swamy's ideas in [22] as follows.

First, we use cycle monotonicity to prove the truthfulness of the mechanism. This enables us to throw away the payment issue, and only focus on the allocation algorithm. Second, we obtain the randomized mechanism by rounding a fractional mechanism. This is built on the following lemma.

Lemma 3.1 (*[22]*). *Let $M = (x, P)$ be a fractional truthful mechanism. Let $\mathcal{R}$ be a randomized rounding algorithm that, given a fractional allocation $\mathbf{x}$, outputs a random integer allocation $\mathbf{X}$ such that $E[X_{ij}] = x_{ij}$ for all i, j. Then there exists a payment algorithm P' such that the mechanism $M' = (X, P')$ is truthful in expectation, where the allocation algorithm X is the combination of two algorithms $\mathcal{R}$ and x, i.e. $X = \mathcal{R} \circ x$.*

In this paper, we use the same rounding algorithm as in [22]. It is a randomized rounding algorithm provided by Kumar, Marathe, Parthasarathy, Srinivasan [19].

Lemma 3.2 (*Kumar et al. [19]*). *Given a fractional assignment $\mathbf{x}$ and a matrix $\mathbf{t}$ specifying the running times, there exists a randomized rounding algorithm $\mathcal{R}$ that yields a randomized integer assignment $\mathbf{X}$ such that, for any i, j, $E[X_{ij}] = x_{ij}$, and with probability* 1,

$$\sum_j X_{ij} t_{ij} < \sum_j x_{ij} t_{ij} + \max_{j: x_{ij} \in (0,1)} t_{ij}.$$

Now all we need is a fractional truthful mechanism. We first give the allocation algorithm $\mathcal{A}_{\mathcal{C}}$ as follows, then we prove that it satisfies cycle monotonicity, hence there exists some payment algorithm P such that $(\mathcal{A}_{\mathcal{C}}, P)$ is a fractional truthful mechanism.

Input: The reported bid matrix $\mathbf{b} = (b_{ij})$.
Output: a fractional allocation $\mathbf{x} = (x_{ij})$.
Allocation algorithm $\mathcal{A}_{\mathcal{C}}$:
(1) Round each b_{ij} down to $\tilde{b}_{ij}$. If $b_{ij} \in [L_j, L_j(1+\epsilon)]$, $\tilde{b}_{ij} = L_j$; otherwise $\tilde{b}_{ij} = H_j$. Let $\tilde{\mathbf{b}} = (\tilde{b}_{ij})$.
(2) Feed the input $\tilde{\mathbf{b}}$ into any c-approximation algorithm $\mathcal{C}$ that satisfies a *load bounding condition*, and get an allocation $\tilde{x}$.
(3) Use a "Spreading Algorithm" $\mathcal{S}$, and convert $\tilde{\mathbf{x}}$ into $\mathbf{x}$.

We explain each step in detail as follows.

(1) By the assumption $H_j/L_j > 2, \epsilon < \frac{1}{16mn}$, two intervals $[L_j, L_j(1+\epsilon)]$, $[H_j, H_j(1+\epsilon)]$ do not intersect, hence the rounded result is unique.

(2) We can use any c-approximation algorithm $\mathcal{C}$ satisfying the following *load bounding condition*: for input $\tilde{\mathbf{b}}$ and output $\tilde{x}$, if $\tilde{x}_{ij} > 0$, then $\tilde{b}_{ij} \le T(\tilde{\mathbf{x}})$. Besides, we do not require that $\tilde{\mathbf{x}}$ to be an integral solution, however the approximation ratio is regarding the integral optimal makespan.

(3) *Spreading Algorithm* $\mathcal{S}$: for each j, let $I_{L_j} = \{i : \tilde{b}_{ij} = L_j\}$, $I_{H_j} = [m] \setminus I_{L_j}$, $l_j = |I_{L_j}|$, $h_j = |I_{H_j}|$. For each i,

- if $\tilde{b}_{ij} = L_j$, let

$$x_{ij} = \tilde{x}_{ij} + \sum_{i' \in I_{L_j}, i' \neq i} \frac{2(\tilde{x}_{i'j} - \tilde{x}_{ij})}{h_j + 2l_j} + \sum_{i' \in I_{H_j}} \frac{2\tilde{x}_{i'j}}{h_j + 2l_j}.$$

- if $\tilde{b}_{ij} = H_j$, let

$$x_{ij} = \sum_{i' \in I_{H_j}} \frac{\tilde{x}_{i'j}}{h_j + 2l_j}.$$

To prove that this allocation algorithm satisfies cycle monotonicity, we need the following lemma. The separation bound produced in the spreading algorithm $\mathcal{S}$ plays an essential role in the proof of the cycle monotonicity.

Lemma 3.3. *The spreading algorithm $\mathcal{S}$ has the following properties:*

- *For any $\tilde{\mathbf{x}}$, $\mathbf{x}$ produced by $\mathcal{S}$ is a feasible allocation.*
- *If $\tilde{b}_{ij} = L_j$, then $x_{ij} \ge 2/(h_j + 2l_j)$; if $\tilde{b}_{ij} = H_j$, then $x_{ij} \le 1/(h_j + 2l_j)$.*
- *(**Separation bound**) For any i and $\mathbf{b}_{-i}$, if i changes $\mathbf{b}_i$ to $\mathbf{b}'_i$, and $\tilde{b}_{ij} \neq \tilde{b}'_{ij}$, then we have:*

$$|x_{ij} - x'_{ij}| \ge \frac{1}{4m}. \tag{3}$$

Proof.

- To show that $\mathbf{x}$ is a feasible solution, we only need to show that, for each j, we have:

$$\sum_i x_{ij} = 1.$$

In fact,

$$\begin{aligned}
\sum_i x_{ij} &= \sum_{i \in I_{L_j}} x_{ij} + \sum_{i \in I_{H_j}} x_{ij} \\
&= \sum_{i \in I_{L_j}} \left(\tilde{x}_{ij} + \sum_{i' \in I_{L_j}, i' \neq i} \frac{2(\tilde{x}_{i'j} - \tilde{x}_{ij})}{h_j + 2l_j} + \sum_{i' \in I_{H_j}} \frac{2\tilde{x}_{i'j}}{h_j + 2l_j} \right) + \sum_{i \in I_{H_j}} \sum_{i' \in I_{H_j}} \frac{\tilde{x}_{i'j}}{h_j + 2l_j} \\
&= \sum_{i \in I_{L_j}} \tilde{x}_{ij} + \sum_{i, i' \in I_{L_j}} \frac{2(\tilde{x}_{i'j} - \tilde{x}_{ij})}{h_j + 2l_j} + l_j \frac{2}{h_j + 2l_j} \sum_{i' \in I_{H_j}} \tilde{x}_{i'j} + \frac{h_j}{h_j + 2l_j} \sum_{i' \in I_{H_j}} \tilde{x}_{i'j} \\
&= \sum_{i \in I_{L_j}} \tilde{x}_{ij} + \sum_{i' \in I_{H_j}} \tilde{x}_{i'j} \\
&= 1.
\end{aligned}$$

- If $\tilde{b}_{ij} = L_j$, then

$$\begin{aligned}
x_{ij} &= \tilde{x}_{ij} + \sum_{i' \in I_{L_j}, i' \neq i} \frac{2(\tilde{x}_{i'j} - \tilde{x}_{ij})}{h_j + 2l_j} + \sum_{i' \in I_{H_j}} \frac{2\tilde{x}_{i'j}}{h_j + 2l_j} \\
&= \frac{h_j + 2}{h_j + 2l_j} \tilde{x}_{ij} + \sum_{i' \neq i} \frac{2}{h_j + 2l_j} \tilde{x}_{i'j} \\
&\ge \sum_{i'} \frac{2}{h_j + 2l_j} \tilde{x}_{i'j} \\
&= \frac{2}{h_j + 2l_j}.
\end{aligned}$$

If $\tilde{b}_{ij} = H_j$, then

$$\begin{aligned} x_{ij} &= \sum_{i' \in I_{H_j}} \frac{1}{h_j + 2l_j} \tilde{x}_{i'j} \\ &\le \frac{1}{h_j + 2l_j}. \end{aligned}$$

- Assuming that $\tilde{b}_{ij} = H_j$, $\tilde{b}'_{ij} = L_j$, then $h'_j = h_j - 1$, $l'_j = l_j + 1$. Noticing $m = h_j + l_j$, we have:

$$\begin{aligned} x'_{ij} - x_{ij} &\ge \frac{2}{h_j - 1 + 2(l_j + 1)} - \frac{1}{h_j + 2l_j} \\ &= \frac{m + l_j - 1}{(m + l_j + 1)(m + l_j)} \\ &\ge \frac{1}{4m}. \quad \square \end{aligned}$$

Now we formally give the following theorem and proof.

Theorem 3.1. *The allocation algorithm $\mathcal{A}_{\mathcal{C}}$ satisfies cycle monotonicity.*

Proof. First, we recall the cycle monotonicity. We need to prove that for any fixed player i, $\mathbf{b}_{-i}$ and any $\mathbf{b}_i^1, \ldots, \mathbf{b}_i^K$, we have

$$\sum_{k=1}^{K} \sum_{j=1}^{n} x_{ij}^{k+1} (b_{ij}^k - b_{ij}^{k+1}) \ge 0.$$

To prove this, we first reduce the cycle to a simple case, then separately estimate the positive terms and negative terms, which we call the gain and loss of this cycle, respectively. We label the cycle's nodes by $1, \ldots, K$, and associate the term $x_{ij}^k (b_{ij}^{k-1} - b_{ij}^k)$ with node k. Sometimes we also label the cycle's nodes with $\tilde{\mathbf{b}}_i^k$, $1 \le k \le K$.

Now we will show the cycle monotonicity. Notice that we only need to consider the case when there is no k such that $\tilde{\mathbf{b}}_i^k = \tilde{\mathbf{b}}_i^{k+1}$. This is because if some such k_0 exists, then $\tilde{\mathbf{x}}^{k_0} = \tilde{\mathbf{x}}^{k_0+1}$, hence $\mathbf{x}^{k_0} = \mathbf{x}^{k_0+1}$, and we have:

$$\begin{aligned} \sum_{k=1}^{K} \sum_{j=1}^{n} x_{ij}^{k+1} (b_{ij}^k - b_{ij}^{k+1}) &= \cdots + \sum_{j=1}^{n} x_{ij}^{k_0} (b_{ij}^{k_0-1} - b_{ij}^{k_0}) + \sum_{j=1}^{n} x_{ij}^{k_0+1} (b_{ij}^{k_0} - b_{ij}^{k_0+1}) + \cdots \\ &= \cdots + \sum_{j=1}^{n} x_{ij}^{k_0+1} (b_{ij}^{k_0-1} - b_{ij}^{k_0} + b_{ij}^{k_0} - b_{ij}^{k_0+1}) + \cdots \\ &= \cdots + \sum_{j=1}^{n} x_{ij}^{k_0+1} (b_{ij}^{k_0-1} - b_{ij}^{k_0+1}) + \cdots . \end{aligned}$$

So we eliminate the node k_0 from the cycle, and keep the cycle's sum unchanged. We can do this until there is no k, such that $\tilde{\mathbf{b}}_i^k = \tilde{\mathbf{b}}_i^{k+1}$.

Starting from node 1, we walk along this cycle clockwise, then for any j, if we see some node k' such that $\tilde{b}_{ij}^{k'-1} = H_j$ and $\tilde{b}_{ij}^{k'} = L_j$, then there must be another node k'' such that $\tilde{b}_{ij}^{k''-1} = L_j$ and $\tilde{b}_{ij}^{k''} = H_j$. We select the first k'' we see after k', and pair (k', k'') together. The summation of this pair's terms equals

$$\begin{aligned} x_{ij}^{k'} \left(b_{ij}^{k'-1} - b_{ij}^{k'} \right) + x_{ij}^{k''} \left(b_{ij}^{k''-1} - b_{ij}^{k''} \right) &\ge x_{ij}^{k'} \left(H_j - L_j (1 + \epsilon) \right) - x_{ij}^{k''} \left(H_j (1 + \epsilon) - L_j \right) \\ &= \left(x_{ij}^{k'} - \alpha x_{ij}^{k''} \right) \left(H_j - L_j (1 + \epsilon) \right). \end{aligned}$$

Here $\alpha = \frac{H_j(1+\epsilon) - L_j}{H_j - L_j(1+\epsilon)}$, and one can easily verify that $\alpha \le 1 + \frac{1}{4mn}$ holds under the assumption on ϵ and H_j/L_j. So we have

$$\begin{aligned} x_{ij}^{k'} - \alpha x_{ij}^{k''} &\ge x_{ij}^{k'} - x_{ij}^{k''} - \frac{1}{4mn} x_{ij}^{k''} \\ &\ge \frac{1}{4m} - \frac{1}{4mn} \cdot \frac{1}{m} \quad \text{(use separation bound)} \\ &\ge \frac{3}{16m} \quad \text{(we assume } m, n \ge 2\text{)}. \end{aligned}$$

Then we estimate the gain of this pair by at least $\frac{3}{16m}((H_j - L_j(1+\epsilon)))$ which we denote by 2Δ. So we can imagine that each node in this pair has a positive term, which contributes at least Δ to the gain of the cycle's sum.

Since any two nodes $k, k+1$ in the cycle satisfy $\tilde{\mathbf{b}}_i^k \neq \tilde{\mathbf{b}}_i^{k+1}$, there is at least one j such that $\tilde{b}_{ij}^k \neq \tilde{b}_{ij}^{k+1}$. So node $k+1$ contributes a gain at least Δ to the cycle's sum. So the whole cycle has a gain at least $K\Delta$.

For the loss of the cycle, it is somewhat easier. Walking from k to $k+1$, there are at most $n-1$ tasks with the same value in $\tilde{\mathbf{b}}_i^k$ and $\tilde{\mathbf{b}}_i^{k+1}$. If i's rounded values on task j are both H_j, then $x_{ij}^{k+1}(b_{ij}^k - b_{ij}^{k+1}) \geq -\frac{1}{m}\epsilon H_j$ because $x_{ij}^{k+1} \leq \frac{1}{h_j+2l_j} \leq \frac{1}{m}$. If i's rounded values on task j are both L_j, then $x_{ij}^{k+1}(b_{ij}^k - b_{ij}^{k+1}) \geq -\epsilon L_j$. So the total loss associated to node $k+1$ is no more than $n \cdot \max\{\frac{1}{m}\epsilon H_j, \epsilon L_j\}$, hence the total loss of the cycle is no more than $Kn \cdot \max\{\frac{1}{m}\epsilon H_j, \epsilon L_j\}$.

To sum up, if we consider the cycle's sum globally, we will see that the gain is more than the loss, because $\Delta \geq \frac{3}{32m}((H_j - L_j(1+\epsilon))) \geq n \cdot \max\{\frac{1}{m}\epsilon H_j, \epsilon L_j\}$. So the allocation algorithm $\mathcal{A}_{\mathcal{C}}$ satisfies the cycle monotonicity. □

Theorem 3.1 combined with Lemmas 3.1 and 3.2, gives the following theorem.

Theorem 3.2. *Given any c-approximation algorithm $\mathcal{C}$ satisfying the load bounding condition, there exists a $4c(1+\epsilon)$-approximation, truthful-in-expectation mechanism $M'_{\mathcal{C}}$.*

Proof. The cycle monotonicity of $\mathcal{A}_{\mathcal{C}}$ indicates that there exists a payment algorithm P such that $M_{\mathcal{C}} = (\mathcal{A}_{\mathcal{C}}, P)$ is a fractional truthful mechanism by Theorem 2.1. Then by Lemmas 3.1 and 3.2, there exists P' such that the mechanism $M'_{\mathcal{C}} = (X, P')$ is truthful in expectation, where the allocation algorithm X is the combination of the rounding algorithm R in Lemma 3.2 and $\mathcal{A}_{\mathcal{C}}$. So we only need to show the approximation ratio.

Since we already proved that the mechanism is truthful in expectation, we can treat the bid $\mathbf{b}$ as the true type values $\mathbf{t}$, and we use $\mathbf{t}$ in the following analysis. We will first show that if we use a c-approximation algorithm $\mathcal{C}$ satisfying the load bounding condition in the mechanism $M'_{\mathcal{C}}$, then $M'_{\mathcal{C}}$'s approximation ratio will be $4c(1+\epsilon)$. Let OPT and $\widetilde{OPT}$ denote the optimum, and $\mathbf{x}^*, \tilde{\mathbf{x}}^*$ be the optimal allocation for the instance $\mathbf{b}$ and $\tilde{\mathbf{b}}$ respectively, $\mathbf{X}$ be $M'_{\mathcal{C}}$'s allocation.

For each i, we have

$$\sum_j X_{ij} t_{ij} \leq \sum_j X_{ij}(1+\epsilon)\tilde{t}_{ij} \leq (1+\epsilon) \cdot \left(\sum_j x_{ij}\tilde{t}_{ij} + \max_{j: x_{ij}>0} \tilde{t}_{ij} \right)$$

where the latter inequality is due to Lemma 3.2.

For $\sum_j x_{ij}\tilde{t}_{ij}$, since for each task j, any machine i gets at most $2/m$ fraction of j from machine r with $\tilde{t}_{rj} \geq \tilde{t}_{ij}$ (see spreading algorithm $\mathcal{S}$ for detail). So we have:

$$\begin{aligned}\sum_j x_{ij}\tilde{t}_{ij} &\leq \sum_j \tilde{x}_{ij}\tilde{t}_{ij} + \frac{2}{m}\sum_{r, r\neq i}\sum_j \tilde{x}_{rj}\tilde{t}_{ij} \\ &\leq 3\max_i \sum_j \tilde{x}_{ij}\tilde{t}_{ij} \\ &\leq 3c\widetilde{OPT}.\end{aligned}$$

Since the approximation algorithm $\mathcal{C}$ satisfies the load bounding condition, we have

$$\max_{j:\tilde{x}_{ij}>0} \tilde{t}_{ij} \leq T(\tilde{\mathbf{x}}) \leq c\widetilde{OPT}.$$

In our spreading algorithm, the allocation can only be spread in a value non increasing direction. Formally, for any task j, machine i can get some fraction of task j from machine r if and only if $\tilde{t}_{rj} \geq \tilde{t}_{ij}$. So if $x_{ij} > 0$, then either $\tilde{x}_{ij} > 0$ or there is some r, such that $\tilde{t}_{rj} \geq \tilde{t}_{ij}$, and $\tilde{x}_{rj} > 0$. We have

$$\max_{j:x_{ij}>0} \tilde{t}_{ij} \leq \max_{j:\tilde{x}_{ij}>0} \tilde{t}_{ij} \leq c\widetilde{OPT}.$$

So we have:

$$\begin{aligned}\sum_j X_{ij}t_{ij} &\leq 4c(1+\epsilon)\widetilde{OPT} \\ &= 4c(1+\epsilon)\max_i \sum_j \tilde{x}^*_{ij}\tilde{t}_{ij} \\ &\leq 4c(1+\epsilon)\max_i \sum_j x^*_{ij}\tilde{t}_{ij} \\ &\leq 4c(1+\epsilon)\max_i \sum_j x^*_{ij}t_{ij} \\ &= 4c(1+\epsilon)OPT.\end{aligned}$$

The second inequality is because $\mathbf{x}^*$ is a feasible solution for the instance $\tilde{\mathbf{t}}$. □

To see the use of Theorem 3.2, we use the following algorithm $\mathcal{C}_1$ as in [22].

Input: bid matrix $\tilde{\mathbf{b}} = (\tilde{b}_{ij})$.
Output: a fractional allocation $\tilde{\mathbf{x}} = (\tilde{x}_{ij})$.
Algorithm $\mathcal{C}_1$:
(1) Compute in polynomial time the smallest T for which the following LP has a feasible solution. Denote it by T^*.

$$\sum_j \tilde{x}_{ij}\tilde{b}_{ij} \le T \quad \forall i$$
$$\sum_i \tilde{x}_{ij} = 1 \quad \forall j$$
$$\tilde{x}_{ij} \ge 0 \quad \forall i, j$$
$$\tilde{x}_{ij} = 0 \quad \text{if } \tilde{b}_{ij} > T$$

(2) Output any feasible solution $\tilde{\mathbf{x}}$ of this LP with $T = T^*$.

This LP has null objective function and the last constraint is used to satisfy the load bounding condition. T^* can be computed by binary search. We can see that $T^* \le \widetilde{OPT}$, where $\widetilde{OPT}$ is the optimum for the two value instance $\tilde{\mathbf{b}}$, since the optimal solution is a feasible solution to the above LP with $T = \widetilde{OPT}$. So this algorithm $\mathcal{A}_{\mathcal{C}_1}$ can be viewed as a 1-approximation algorithm, and we have the following corollary:

Corollary 3.1. *There exists a $4(1+\epsilon)$-approximation truthful-in-expectation mechanism for case A: $H_j/L_j > 2, \forall j$.*

4. Mechanism for case B

In this section, we give a mechanism for case B when $H_j/L_j \le 2, \forall j$. We first design a truthful fractional mechanism directly, and then round it as in case A, using the rounding algorithm in Lemma 3.2.

As before, we only specify the allocation algorithm.

Input: The reported bid matrix $\mathbf{b} = (b_{ij})$.
Output: a fractional allocation $\mathbf{x} = (x_{ij})$.
Allocation algorithm $\mathcal{A}_2$:
For each task j, set $x_{ij} = \frac{\frac{1}{b_{ij}}}{\sum_s \frac{1}{b_{sj}}}$.

Lemma 4.1. *There exists a payment algorithm P such that the fractional mechanism $M_2 = (\mathcal{A}_2, P)$ is truthful.*

Proof. We obtain the algorithm P by specifying a payment algorithm P^j for each task j. For each task j, the allocation algorithm of task j is monotone decreasing. By Theorem 2.2, there exists a payment algorithm P^j for task j, such that the submechanism for task j is truthful. Since an agent's utility equals the sum of utilities gained from each task, and telling the true bids will maximize his/her utilities from each task, the whole mechanism is also truthful. □

By Lemmas 3.1 and 3.2 again, we have the following theorem.

Theorem 4.1. *There exists a payment algorithm P' such that the mechanism $M_2' = (X, P')$ is truthful in expectation, and has an approximation ratio of $4(1+\epsilon)$. Here the allocation algorithm X is the combination of the rounding algorithm in Lemma 3.2 and $\mathcal{A}_2$.*

Proof. Since the truthfulness is already proven, we use t_{ij} and $\mathbf{t}$ instead of b_{ij} and $\mathbf{b}$ in the analysis of the approximation ratio. Let OPT be the optimal solution (or the optimum) and T_i be the tasks allocated on machine i in OPT, then $\sum_{j \in T_i} t_{ij} \le OPT, \forall i$. For task j, let $I_{L_j}, I_{H_j}, l_j, h_j$ be defined as before. Then for any i, r, we have:

$$\begin{aligned}
\frac{x_{ij}t_{ij}}{t_{rj}} &= \frac{\frac{1}{t_{rj}}}{\sum\limits_{s \in I_{L_j}} \frac{1}{t_{sj}} + \sum\limits_{s \in I_{H_j}} \frac{1}{t_{sj}}} \\
&\le \frac{\frac{1}{L_j}}{\frac{l_j}{L_j(1+\epsilon)} + \frac{h_j}{H_j(1+\epsilon)}} \\
&\le \frac{\frac{1}{L_j}}{\frac{l_j}{L_j(1+\epsilon)} + \frac{h_j}{2L_j(1+\epsilon)}} \\
&= \frac{2(1+\epsilon)}{2l_j + h_j} \\
&\le \frac{2(1+\epsilon)}{m}.
\end{aligned}$$

Please cite this article in press as: C. Yu, Truthful mechanisms for two-range-values variant of unrelated scheduling, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.02.001

So for each i, we have:

$$\sum_j x_{ij}t_{ij} = \sum_{r=1}^{m}\sum_{j\in T_r} x_{ij}t_{ij} \le \sum_{r=1}^{m}\sum_{j\in T_r}\frac{2(1+\epsilon)}{m}t_{rj} \le 2(1+\epsilon)OPT.$$

By Lemma 3.2, we have:

$$\sum_j X_{ij}t_{ij} \le \sum_j x_{ij}t_{ij} + \max_{x_{ij}>0} t_{ij}.$$

Since $t_{ij} \le H_j(1+\epsilon) \le 2(1+\epsilon)L_j \le 2(1+\epsilon)OPT, \forall i, j,$

$$\sum_j X_{ij}t_{ij} \le 2(1+\epsilon)OPT + 2(1+\epsilon)OPT = 4(1+\epsilon)OPT. \quad \square$$

5. Mechanism for the general case

For the general case, there are some tasks with $H_j/L_j > 2$, and some tasks with $H_j/L_j \le 2$. We denote them by J_A and J_B respectively. W.l.o.g., $J_A, J_B \ne \emptyset$.

First we give the fractional allocation $\mathcal{A}_3$, then we prove that it satisfies the cycle monotonicity. By rounding this fractional allocation as before, we obtain a truthful in expectation mechanism.

Input: the reported bid matrix $\mathbf{b} = (b_{ij})$.
Output: a fractional allocation $\mathbf{x} = (x_{ij})$.
Allocation algorithm $\mathcal{A}_3$:
divide the tasks into two sets, J_A and J_B,
apply $\mathcal{A}_{\mathcal{C}_1}$ on J_A, $\mathcal{A}_2$ on J_B and obtain allocations
$\mathbf{x}_A, \mathbf{x}_B$, output $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_B)$.

Lemma 5.1. *There exists a payment algorithm P such that the fractional mechanism $M_3 = (\mathcal{A}_3, P)$ is truthful.*

Proof. we notice that any cycle's sum equals the summation of the cycle's sum on task sets J_A and J_B. Since the cycle's sum on both task sets J_A and J_B are non-negative, the cycle's sum on all tasks is non-negative. So the allocation algorithm $\mathcal{A}_3$ satisfies cycle monotonicity, which means that there exists a payment algorithm P such that the fractional mechanism $M_3 = (\mathcal{A}_3, P)$ is truthful. $\square$

By Lemmas 3.1 and 3.2 again, we have the following theorem.

Theorem 5.1. *There exists a payment algorithm P' such that the mechanism $M_3' = (X, P')$ is truthful in expectation, and has an approximation ratio of $7(1+\epsilon)$. Here the allocation algorithm X is the combination of the rounding algorithm in Lemma 3.2 and $\mathcal{A}_3$.*

Proof. Since the truthfulness is already proven, we use t_{ij} and $\mathbf{t}$ instead of b_{ij} and $\mathbf{b}$ in the analysis of the approximation ratio. Let OPT be the optimum, $\mathbf{x}^*$ be the optimal allocation, and $\mathbf{x}_A^*, \mathbf{x}_B^*$ be the restriction of $\mathbf{x}^*$ on task sets J_A, J_B. Let $\mathbf{X}_A, \mathbf{X}_B$ be the restriction of $\mathbf{X}$ on task sets J_A, J_B. For each machine i, let $TA_i^*, TB_i^*, TA_i, TB_i$ denote the load on machine i under $\mathbf{x}_A^*, \mathbf{x}_B^*, \mathbf{X}_A, \mathbf{X}_B$ respectively. We also denote the optimal makespans on task sets J_A and J_B as OPT_A, OPT_B.

By Lemma 3.2, we have:

$$\sum_j X_{ij}t_{ij} \le \sum_j x_{ij}t_{ij} + \max_{x_{ij}>0} t_{ij}$$

$$\begin{aligned}\sum_j X_{ij}t_{ij} &\le \sum_j x_{ij}t_{ij} + \max_{x_{ij}>0} t_{ij} \\ &= \sum_{j\in J_A} x_{ij}t_{ij} + \sum_{j\in J_B} x_{ij}t_{ij} + \max\{\max_{j\in J_A, x_{ij}>0} t_{ij}, \max_{j\in J_B, x_{ij}>0} t_{ij}\} \\ &\le 3(1+\epsilon)OPT_A + 2(1+\epsilon)OPT_B + \max\{(1+\epsilon)OPT_A, 2(1+\epsilon)OPT_B\}.\end{aligned}$$

Since $\mathbf{x}_A^*$ is also a feasible allocation of task set A, we have:

$$OPT_A \le \max_i TA_i^* \le \max_i\{TA_i^* + TB_i^*\} = OPT.$$

Please cite this article in press as: C. Yu, Truthful mechanisms for two-range-values variant of unrelated scheduling, Theoretical Computer Science (2009), doi:10.1016/j.tcs.2009.02.001

Similarly, we have $OPT_B \leq OPT$.

$$\sum_j X_{ij} t_{ij} \leq (1+\epsilon)\big(3OPT_A + 2OPT_B + \max\{OPT_A, 2OPT_B\}\big)$$
$$\leq 7(1+\epsilon)OPT. \quad \square$$

6. Conclusion and open problems

In this paper, we study the two-range-values variant of the scheduling unrelated machines problem from the mechanism design setting. For one special case, we use cycle monotonicity to obtain a general technique converting any c-approximation algorithm into a $4c(1+\epsilon)$-approximation mechanism. We prove the cycle monotonicity via a novel idea, which may be of independent interest. For another special case, we first design a fractional truthful mechanism directly, then use rounding. For both cases, we obtain truthful-in-expectation mechanisms with approximation ratio $4(1+\epsilon)$. Finally, for the general case, we combine two mechanisms into a mechanism which is still truthful in expectation and has approximation ratio $7(1+\epsilon)$. This idea of combination may also be useful for further research.

However, the two-range-values variant is still somewhat restricted. We are interested in a more general variant, such that each task has values in k ranges ($k > 2$) or even a big interval. Of course, the biggest challenge in general scheduling unrelated machines problem is to close the gap between the lower bound 2.61 and the upper bound m of deterministic truthful mechanisms.

Acknowledgments

The author was supported by the National Natural Science Foundation of China Grant 60553001 and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

References

[1] N. Andelman, Y. Azar, M. Sorani, Truthful approximation mechanisms for scheduling selfish related machines, in: V. Diekert, B. Durand (Eds.), STACS 2005, in: LNCS, Springer, Heidelberg, 2005.
[2] E. Angel, E. Bampis, A. Kononov, A FPTAS for approximating the unrelated parallel machines scheduling problem with costs, in: Proceedings of the 9th Annual European Symposium on Algorithms August 2001, in: LNCS, Springer-Verlag, London, 2001.
[3] A. Archer, Mechanisms for discrete optimization with rational agents, Ph.D. Thesis, Ithaca, NY, USA, 2004 (Adviser-Eva Tardos).
[4] A. Archer, C. Papadimitriou, K. Talwar, É. Tardos, An approximate truthful mechanism for combinatorial auctions with single parameter agents, in: SODA'03: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003, pp. 205–214.
[5] A. Archer, É. Tardos, Truthful mechanisms for one-parameter agents, in: FOCS '01: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, 2001, p. 482.
[6] M. Babaioff, R. Lavi, E. Pavlov, Mechanism design for single-value domains, in: AAAI, 2005, pp. 241–247.
[7] S. Bikhchandani, S. Chatterji, R. Lavi, A. Mu'alem, N. Nisan, A. Sen, Weak monotonicity characterizes deterministic dominant strategy implementation, Econometrica 74 (4) (2006) 1109–1132.
[8] P. Briest, P. Krysta, B. Vöcking, Approximation techniques for utilitarian mechanism design, in: STOC'05: Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 2005, pp. 39–48.
[9] G. Christodoulou, E. Koutsoupias, A. Kovács, Mechanism design for fractional scheduling on unrelated machines, in: ICALP, 2007, pp. 40–52.
[10] G. Christodoulou, E. Koutsoupias, A. Vidali, A lower bound for scheduling mechanisms, in: SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1163–1170.
[11] E. Clarke, Multipart pricing of public goods, Public Choice 11 (1) (1971) 17.
[12] S. Dobzinski, N. Nisan, M. Schapira, Approximation algorithms for combinatorial auctions with complement-free bidders, in: STOC'05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 2005, pp. 610–618.
[13] S. Dobzinski, N. Nisan, M. Schapira, Truthful randomized mechanisms for combinatorial auctions, in: STOC'06: Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 2006, pp. 644–652.
[14] T. Groves, Incentives in teams, Econometrica 41 (4) (1973) 617–631.
[15] H. Gui, R. Müller, R. Vohra, Dominant strategy mechanisms with multidimensional types, Discussion Papers 1392, Center for Mathematical Studies in Economics and Management Science, Northwestern University, July 2004.
[16] E. Koutsoupias, A. Vidali, A lower bound of $1+\phi$ for truthful scheduling mechanisms, in: Proceedings, MFCS, 2007, pp. 454–464.
[17] A. Kovács, Fast monotone 3-approximation algorithm for scheduling related machines, in: European Symposium on Algorithms, in: Lecture Notes in Computer Science, vol. 3669, Springer-Verlag, Berlin, 2005, pp. 616–627.
[18] A. Kovács, Fast algorithms for two scheduling problems, Ph.D. Thesis, Universitat des Saarlandes, 2007.
[19] V. Kumar, M. Marathe, S. Parthasarathy, A. Srinivasan, Approximation algorithms for scheduling on multiple machines, in: Proc 46th FOCS, 2005, pp. 254–263.
[20] R. Lavi, A. Mu'alem, N. Nisan, Towards a characterization of truthful combinatorial auctions, in: FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, 2003, p. 574.
[21] R. Lavi, C. Swamy, Truthful and near-optimal mechanism design via linear programming, in: FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, 2005, pp. 595–604.
[22] R. Lavi, C. Swamy, Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity, in: EC '07: Proceedings of the 8th ACM Conference on Electronic Commerce, ACM, New York, NY, USA, 2007, pp. 252–261.
[23] J. Lenstra, D. Shmoys, É. Tardos, Approximation algorithms for scheduling unrelated parallel machines, Math. Program. 46 (3) (1990) 259–271.
[24] P. Lu, C. Yu, An improved randomized truthful mechanism for scheduling unrelated machines, in: 25th International Symposium on Theoretical Aspects of Computer Science, STACS, 2008, pp. 527–538.
[25] P. Lu, C. Yu, Randomized truthful mechanisms for scheduling unrelated machines, in: C.H. Papadimitriou, S. Zhang (Eds.), Proceedings of WINE, in: Lecture Notes in Computer Science, vol. 5385, Springer, 2008, pp. 402–413.

[26] A. Mu'alem, M. Schapira, Setting lower bounds on truthfulness: Extended abstract, in: SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1143–1152.
[27] R. Myerson, Optimal auction design, Math. Oper. Res. 6 (1) (1981) 58–73.
[28] N. Nisan, A. Ronen, Algorithmic mechanism design (extended abstract), in: STOC'99: Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 1999, pp. 129–140.
[29] J.C. Rochet, A necessary and sufficient condition for rationalizability in a quasilinear context, J. Math. Econom. (1987) 16.
[30] M. Saks, L. Yu, Weak monotonicity suffices for truthfulness on convex domains, in: EC '05: Proceedings of the 6th ACM Conference on Electronic Commerce, ACM, New York, NY, USA, 2005, pp. 286–293.
[31] William Vickrey, Counterspeculation, auctions, and competitive sealed tenders, J. Finance 16 (1961) 8–37.