Theoretical Computer Science 414 (2012) 1–8

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

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

Randomized truthful algorithms for scheduling selfish tasks on parallel machines ☆

Eric Angel [a], Evripidis Bampis [b], Nicolas Thibault [c,*]

[a] IBISC, Université d'Évry, France

[b] LIP6 UMR 7606, Université Pierre et Marie Curie (Paris 6), France

[c] ERMES EA 4441, Université Panthéon-Assas (Paris 2), France

ARTICLE INFO

Article history:
Received 15 June 2010
Received in revised form 5 October 2011
Accepted 9 October 2011
Communicated by X. Deng

Keywords:
Scheduling
Algorithmic game theory
Approximation

ABSTRACT

We study the problem of designing truthful algorithms for scheduling a set of tasks, each one owned by a selfish agent, to a set of parallel (identical or unrelated) machines in order to minimize the makespan. We consider the following process: at first the agents declare the length of their tasks, then given these bids, the protocol schedules the tasks on the machines. The aim of the protocol is to minimize the makespan, i.e. the maximum completion time of the tasks, while the objective of each agent is to minimize the completion time of its task and thus an agent may lie if by doing so, his task may finish earlier. In this paper, we show the existence of randomized truthful (non-polynomial-time) algorithms with an expected approximation ratio equal to 3/2 for different scheduling settings (identical machines with and without release dates and unrelated machines) and models of execution (strong or weak). Our result improves the best previously known result Angel et al. (2006) [1] for the problem with identical machines ($P \parallel C_{max}$) in the strong model of execution and reaches, asymptotically, the lower bound of Christodoulou et al. (2007) [5]. In addition, this result can be transformed to a polynomial-time truthful randomized algorithm with an expected approximation ratio $3/2 + \epsilon$ (resp. $\frac{11}{6} - \frac{1}{3m}$) for $Pm \parallel C_{max}$ (resp. $P \parallel C_{max}$).

1. Introduction

Nowadays, there are many systems involving autonomous entities (agents). These systems are organized by protocols, trying to maximize the social welfare in the presence of private information held by the agents. In some settings, the agents may try to manipulate the protocol by reporting false information in order to maximize their own profit. With false information, even the most efficient protocol may lead to unreasonable solutions if it is not designed to cope with the selfish behavior of the agents. In such a context, it is natural to study the efficiency of *truthful* protocols, i.e. protocols that are able to guarantee that no agent has incentive to lie. This approach has been considered in many papers of past few years (see [4] for a recent survey).

In this paper, we study the problem of designing truthful algorithms for scheduling a set of tasks, each one owned by a selfish agent, to a set of parallel (identical or unrelated) machines in order to minimize the makespan. We consider the following process: before the start of the execution, the agents declare the length of their tasks, then given these bids, the

☆ A preliminary version of this paper has been presented in the 9th Latin American Symposium (LATIN), Oaxaca, Mexico, LNCS 6034, 38–48, 2010.

* Corresponding author. Tel.: +33 06 81 50 14 07.
E-mail addresses: Eric.Angel@ibisc.fr (E. Angel), Evripidis.Bampis@lip6.fr (E. Bampis), Nicolas.Thibault@u-paris2.fr, nicolasvincentpierre@gmail.com (N. Thibault).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.10.006

protocol schedules the tasks on the machines. The aim of the protocol is to minimize the makespan, i.e. the maximum completion time of the tasks, while the objective of each agent is to minimize the completion time of its task and thus an agent may lie if by doing so, his task may finish earlier. We focus on protocols without side payments that simultaneously offer a guarantee on the quality of the schedule (its makespan is not arbitrarily far from the optimum) and guarantee that the solution is truthful (no agent can lie and improve his own completion time).

1.1. Formal definition

There are n agents, represented by the set $\{1, 2, \dots, n\}$ and m parallel machines.

Variants of the problem. Depending on the type of the machines and the job characteristics, we consider three different variants of the problem.

- **Identical parallel machines ($P||C_{\max}$).** All the machines are identical and every task i has a private value t_i that represents its length. We assume that an agent cannot shrink the length of his task (otherwise he will not get his result), but if he can decrease his completion time by bidding a value larger than the real one ($b_i \geq t_i$), then he will do so.
- **Identical parallel machines with release dates ($P|r_i|C_{\max}$).** All the machines are identical and every task i has now a private pair (t_i, r_i), where t_i is the length of task i and r_i is its release date. Every task i may bid any pair (b_i, r_i^b) such that $b_i \geq t_i$ and $r_i^b \geq r_i$. A task i may not bid a release date smaller than its real release date i.e. $r_i^b < r_i$, because otherwise, the task may be scheduled before r_i and thus the final schedule may be infeasible.
- **Unrelated parallel machines ($R||C_{\max}$).** The machines are here unrelated. Every task i has a private vector $(t_i^1, \dots, t_i^m)$, where t_i^j, $1 \leq j \leq m$, is the processing time of task i if it is executed on machine j. Every task i bids any vector $(b_i^1, \dots, b_i^m)$ with $b_i^1 \geq t_i^1, \dots, b_i^m \geq t_i^m$.

Models of execution. Before describing the two models of execution that we will consider here, let us recall the process: first, every agent i bids a length b_i for its task and second, given these bids, the protocol constructs a schedule. An agent could either underbid, or overbid. However, if, say, agent i underbids by declaring a length $b_i < t_i$, then the execution of his task will never be completed because in the schedule an interval of only b_i units of time will be reserved for his task. Therefore, no agent has incentive to underbid in such a context and so we may consider that the agents may lie, only by overbidding, i.e. that for every agent i, his bid b_i is always greater than or equal to its real length t_i. Let us now define the models of execution that we will study.

- *The strong model of execution:* task i bids any value $b_i \geq t_i$ and its execution time is t_i (i.e. task i is completed t_i units of time after it starts even if i bids $b_i \neq t_i$).
- *The weak model of execution:* i bids any value $b_i \geq t_i$ and its execution time is b_i (i.e. task i is completed b_i units of time after it starts).

Notice that an alternative way to define these two models would be to say that in the strong (respectively, weak) model of execution, task i is allowed to both under- and overbid and is completed $\min\{b_i, t_i\}$ (respectively, $\max\{b_i, t_i\}$) units of time after its starting time. However, for simplicity reasons, and given that the agents will never have incentive to underbid, we only consider the possibility of overbidding in the following.
Another thing that should be clear is the motivation of examining these two models. The strong model corresponds to a situation where the agents have the opportunity to follow "on screen" the execution of their task (and thus get the result at the completion of the "real" part of their task), while the weak model corresponds to an execution in a batch mode where the agents get the result at the completion of their task (both "real" and possibly "fake" parts included).
Notation. By C_i, we denote the completion time of task i. The objective of the protocol is to determine a schedule of the tasks minimizing the maximum completion time of the tasks or makespan, denoted in what follows by $C_{\max}$. We say that an algorithm is *truthful*, if and only if, for every task i, $1 \leq i \leq n$ and for every bid b_j, $j \neq i$, the completion time of task i is minimum when i bids $b_i = t_i$. In other words, an algorithm is truthful if truth-telling is the best strategy for a player i regardless of the strategy adopted by the other players.

1.2. Related works

The works that are more closely related to ours are those in [1–3,5]. In the paper by Auletta et al. [3], the authors consider the variant of the problem of m related machines in which the individual function of each task is the completion time of the machine on which it is executed, while the global objective function is the makespan. They consider the strong model of execution by assuming that each task may declare an arbitrary length (smaller or greater than its real length) while the load of each machine is the sum of the true lengths of the tasks assigned to it. They provide equilibria-truthful mechanisms that use payments in order to retain truthfulness. In [1], the authors consider a different variant with m identical machines in which the individual objective function of each task is its completion time and they consider the strong model of execution (but here the tasks may only report values that are greater than or equal to their real lengths). Given that for this variant the

SPT (Shortest Processing Time) algorithm[1] is truthful, they focus on the design of algorithms with a better approximation ratio than that of the SPT algorithm. The rough idea of their approach is a randomized algorithm in which they combine the LPT (Longest Processing Time) algorithm,[2] which has a better approximation ratio than SPT but is not truthful, with a schedule (DSPT) based on the SPT algorithm where some ''unnecessary'' idle times are introduced between the tasks. These unnecessary idle times are introduced in the SPT schedule in order to penalize more the tasks that report false information. Indeed, in the DSPT schedule such a task is doubly penalized, since not only is its execution delayed by the other tasks but also by the introduced idle times. In such a way, it is possible to find a probability distribution over the deterministic algorithms, LPT and DSPT which produces a randomized algorithm that is proved to be truthful and with an (expected) approximation ratio of $2-\frac{1}{m+1}(\frac{5}{3}+\frac{1}{3m})$, i.e. better than the one of SPT which is equal to $2-\frac{1}{m}$. An optimal truthful randomized algorithm and a truthful randomized PTAS for identical parallel machines in the weak model of execution appeared in [2]. The idea of these algorithms is to introduce fake tasks in order to have the same completion time in all the machines and then to use a random order in each machine for scheduling the tasks allocated to it (including the eventual fake one). These results have been also generalized in the case of related machines and the on-line case with release dates. Another related work, presented in [5], gives some new lower and upper bounds. More precisely, the authors proved that there is no truthful deterministic (resp. randomized) algorithm with an approximation ratio smaller than $2-1/m$ (resp. $3/2-1/2m$) for the strong model of execution. They also provide a lower bound of 1.1 for the deterministic case in the weak model (for $m\geq 3$) and a deterministic $\frac{4}{3}-\frac{1}{3m}$ truthful algorithm based on the idea of bloc schedule where after inserting fake tasks in order to have the same completion time in all the machines, instead of using a random order on the tasks of each machine, the authors proposed to take the mirror of the LPT schedule.

1.3. Our contribution

In the first part of the paper, we consider the strong model of execution. Our contribution is a new truthful randomized non-polynomial algorithm that we call Starting Time Equalizer (STE), presented in Section 2, whose approximation ratio for the makespan is $\frac{3}{2}$ for $P||C_{\max}$. This new upper bound asymptotically closes the gap between the lower bound $\frac{3}{2}-\frac{1}{2m}$ of [5] and the previously best known upper bound of $2-\frac{1}{m+1}\left(\frac{5}{3}+\frac{1}{3m}\right)$ for this problem [1]. We also give two polynomial-time variants of algorithm STE, respectively with approximation ratios $\frac{3}{2}+\epsilon$ for $Pm||C_{\max}$ and $\frac{11}{6}+\frac{1}{3m}$ for $P||C_{\max}$ (we underline that $\frac{11}{6}+\frac{1}{3m}$ is better than the previous upper bound of $2-\frac{1}{m+1}\left(\frac{5}{3}+\frac{1}{3m}\right)$ for m large). In the second part of the paper, we consider the weak model of execution. We give in Section 3.1, a new truthful randomized non-polynomial algorithm, called Mid-Time Equalizer (MTE) for the off-line problem with release dates, where the private information of each task is not only each length, but also its release date ($P|r_i|C_{\max}$). Finally, we consider the case of scheduling a set of selfish tasks on a set of unrelated parallel machines ($R||C_{\max}$) for the weak model of execution (Section 3.2) where we propose a new truthful randomized non-polynomial algorithm that we call Completion Time Equalizer (CTE). Table 1 gives a summary of the upper and lower bounds on the approximation ratio of truthful algorithms for the considered problems (with † we give the results obtained in this paper).

The lower bounds for truthful deterministic algorithms in the weak model for $P|r_i|C_{\max}$ and $R||C_{\max}$ are simple implications of the lower bound for truthful deterministic algorithms solving $P||C_{\max}$. Up to our knowledge, there is no interesting lower bounds for truthful randomized algorithms (resp. upper bound for truthful deterministic algorithms) for $R||C_{\max}$ and $P|r_i|C_{\max}$ (resp. $R||C_{\max}$). The upper bound $2-\frac{1}{m}$ for $P|r_i|C_{\max}$ in the weak model holds only if we consider that each task can be identified by an identification number (ID). With this assumption, we just have to consider the on-line algorithm which schedules the tasks when they become available with (for instance) the smallest ID first. This algorithm is then trivially truthful, because task i will not have incentive of bidding ($b_i>t_i$, $r_i^b>r_i$) (b_i has no effect on the way in which tasks are scheduled and bidding $r_i^b>r_i$ can only increase C_i). Moreover, as this algorithm is a particular case of Graham's list scheduling (LS) algorithm with release dates, it is $(2-\frac{1}{m})$-competitive (because algorithm LS is $(2-\frac{1}{m})$-competitive for $P|\textit{on-line-list}|C_{\max}$, [7]).

2. Strong model of execution

Identical machines

2.1. Algorithm STE

We consider in this section the problem with identical machines ($P||C_{\max}$) in the strong model. Every task i has a private value t_i that represents its length and it has to bid any value $b_i\geq t_i$.

[1] Where the tasks are scheduled greedily following the increasing order of their lengths (its approximation ratio is $2-1/m$).
[2] Where the tasks are scheduled greedily following the decreasing order of their lengths (its approximation ratio is $4/3-1/(3m)$).

Table 1
Bounds for m parallel machines.

<table>
<tr><td></td><td colspan="2">Deterministic</td><td colspan="2">Randomized</td></tr>
<tr><td></td><td>Lower bound</td><td>Upper bound</td><td>Lower bound</td><td>Upper bound</td></tr>
<tr><td>$P||C_{\max}$
strong model</td><td rowspan="2">$2-\frac{1}{m}$ [5]</td><td rowspan="2">$2-\frac{1}{m}$ [6]</td><td rowspan="2">$\frac{3}{2}-\frac{1}{2m}$ [5]</td><td>$\frac{3}{2}$ †
(non-polynomial algorithm)
$\frac{11}{6}+\frac{1}{3m}$ †
(polynomial algorithm)</td></tr>
<tr><td>$Pm||C_{\max}$
strong model</td><td>$\frac{3}{2}+\epsilon$ †
(polynomial algorithm)</td></tr>
<tr><td>$P||C_{\max}$
weak model</td><td rowspan="3">if $m=2$ then
$1+\frac{\sqrt{105}-9}{12}>1.1$
if $m\geq 3$ then
$\frac{7}{6}>1.16$ [5]</td><td>$\frac{4}{3}-\frac{1}{3m}$ [5]</td><td>1 [2]</td><td>1 [2]</td></tr>
<tr><td>$R||C_{\max}$
weak model</td><td>unknown</td><td rowspan="2">unknown</td><td>$\frac{3}{2}$ †
(non-polynomial algorithm)</td></tr>
<tr><td>$P|r_i|C_{\max}$
weak model</td><td>$2-\frac{1}{m}$ [7]</td><td>$\frac{3}{2}$ †
(non-polynomial algorithm)</td></tr>
</table>

Algorithm STARTING TIME EQUALIZER (STE)

1. Let $C_{\max}^{OPT}$ be the makespan of an optimal schedule OPT for $P||C_{\max}$.
   Let OPT_j be the sub-schedule of OPT on machine j.
   Let $b_{j_1} \leq \cdots \leq b_{j_k}$ be the bids (sorted by increasing order) of the k tasks in OPT_j.
2. Construct schedule S_1 as follows: for every machine j $(1 \leq j \leq m)$, every task j_i $(1 \leq i \leq k)$ in OPT_j is executed on machine j by starting at time $\sum_{l=i+1}^{k} b_{j_l}$.
3. Construct schedule S_2 as follows: for every machine j $(1 \leq j \leq m)$, every task j_i $(1 \leq i \leq k)$ in OPT_j is executed on machine j by starting at time $C_{\max}^{OPT} - \sum_{l=i+1}^{k} b_{j_l}$.
4. Choose schedule S_1 or S_2 each with probability $1/2$.

Fig. 1 illustrates the construction of schedules S_1 and S_2 in algorithm STE on machine j.

The main idea of algorithm STE is to make equal the expected starting times of all the tasks. More precisely, we prove below that the expected starting time of every task in the final schedule constructed by STE, which is the average between its starting time in S_1 and its starting time in S_2, will be equal to $\frac{C_{\max}^{OPT}}{2}$ (i.e. the same value for every task). This property will be used in the proof of Theorem 1 to show that STE is truthful. In the example given in Fig. 1, the expected starting time of the four tasks is $\frac{C_{\max}^{OPT}}{2}$ and it is equal to 5.5.

Theorem 1. *STE is a randomized, truthful and $\frac{3}{2}$-approximate algorithm in the strong model of execution for $P||C_{\max}$.*

Proof. As STE is a randomized algorithm, to prove it is truthful, we have to show that the expected completion time of each task is minimum when it tells the truth. By definition of STE, the expected completion time C_i of any task i is the average between its completion time in schedule S_1 and its completion time in schedule S_2. In the strong model of execution, every

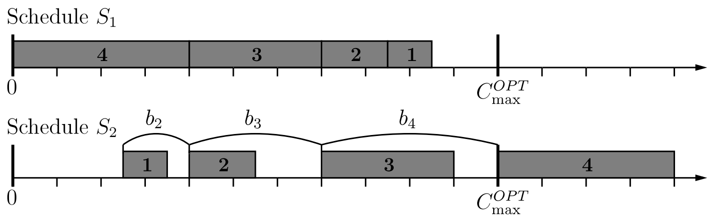

Fig. 1. An illustration of execution of algorithm STE on machine j. We give an example of schedules S_1 and S_2 with four tasks in OPT_j such that $b_{j_1} = 1$, $b_{j_2} = 1.5$, $b_{j_3} = 3$, $b_{j_4} = 4$ and $C_{\max}^{OPT} = 11$.

task i is completed t_i units of time after its starting time. Thus,

$$C_i = \frac{1}{2}\left(\left(t_i + \sum_{l=i+1}^{k} b_{j_l}\right) + \left(t_i + C_{\max}^{OPT} - \sum_{l=i+1}^{k} b_{j_l}\right)\right) = t_i + \frac{C_{\max}^{OPT}}{2}.$$

For every task i, the completion time of task i is $C_i = t_i + \frac{C_{\max}^{OPT}}{2}$ and it reaches its minimum value when i tells the truth because t_i does not depend on the bid b_i and because $C_{\max}^{OPT}$ obviously does not decrease if i bids $b_i > t_i$ instead of $b_i = t_i$. Thus, STE is truthful in the strong model of execution. Given that STE is truthful, we may consider in the following that for every i, we have $b_i = t_i$. Given also that STE is a randomized algorithm choosing with probability 1/2 schedule S_1 and with probability 1/2 schedule S_2, its approximation ratio will be the average between the approximation ratios of schedules S_1 and S_2. In S_1, all tasks end before or at time $C_{\max}^{OPT}$. Thus, as for every i, $b_i = t_i$, $C_{\max}^{OPT}$ is the makespan of an optimal solution computed with the true types of the agents, S_1 is optimal. In S_2, on every machine j, all tasks end before or at time $C_{\max}^{OPT}$ except task j_k, which finishes at time $C_{\max}^{OPT} + t_{j_k}$. Given that $t_{j_k} \leq C_{\max}^{OPT}$, all tasks in S_2 end before or at time $2C_{\max}^{OPT}$. Thus, S_2 is 2-approximate. Hence, the expected approximation ratio of STE is $\frac{1}{2}(1+2) = \frac{3}{2}$. □

2.2. Polynomial-time variants of algorithm STE

Given that algorithm STE requires the computation of an optimal solution for $P||C_{\max}$ and as this problem is NP-hard, it is clear that STE cannot be executed in polynomial time. Nevertheless, it is interesting for two reasons. First, it asymptotically closes the gap between the lower bound $\frac{3}{2} - \frac{1}{2m}$ of any truthful algorithm and the previously best known upper bound of $2 - \frac{1}{m+1}\left(\frac{5}{3} + \frac{1}{3m}\right)$. Second, by using approximated solutions instead of the optimal one, we can obtain polynomial-time variants of STE. To precise these variants, we first need to define what we call an *increasing* algorithm.

Definition (*Increasing Algorithm [2]*)**.** Let H and H' be two sets of tasks ordered in non-increasing order with respect to their execution times $\{T_1, \ldots, T_n\}$ and $\{T'_1, \ldots, T'_n\}$ respectively. We denote by $H \leq H'$ the fact that for every $1 \leq i \leq n$, we have $l(T_i) \leq l(T'_i)$ (where $l(T)$ is the length of task T). An algorithm A is increasing if for every pair of sets of tasks H and H' such that $H \leq H'$, it constructs schedules such that $C_{\max}(H) \leq C_{\max}(H')$ (where $C_{\max}(X)$ is the makespan of the solution constructed by algorithm A for the set of tasks X).

As LPT (Longest Processing Time) is an increasing algorithm (see [2]) and as there exists an increasing PTAS for $Pm||C_{\max}$ (see [2]), we get the following two theorems.

Theorem 2. *By using LPT instead of an optimal algorithm, we obtain a polynomial-time, randomized, truthful and* $(\frac{11}{6} - \frac{1}{3m})$*-approximate variant of STE in the strong model of execution for* $P||C_{\max}$.

Theorem 3. *By using the increasing PTAS in [2] instead of an optimal algorithm, we obtain a polynomial-time, randomized, truthful and* $(\frac{3}{2} + \epsilon)$*-approximate variant of STE in the strong model of execution for* $Pm||C_{\max}$.

Theorem 2 (resp. Theorem 3) can be proved in a similar way as in Theorem 1. Indeed, as the completion time of each task will be $C_i = t_i + \frac{C_{\max}^{LPT}}{2}$ (resp. $C_i = t_i + \frac{C_{\max}^{PTAS}}{2}$) instead of $C_i = t_i + \frac{C_{\max}^{OPT}}{2}$ and as LPT (resp. the PTAS in [2]) is increasing, the variant of STE in Theorem 2 (resp. Theorem 3) is truthful. Moreover, as LPT is $(\frac{4}{3} - \frac{1}{3m})$-approximate for $P||C_{\max}$ (resp. the PTAS in [2] is $(1+\epsilon)$-approximate for $Pm||C_{\max}$), we obtain that the expected approximation ratio of the variant of STE in Theorem 2 (resp. Theorem 3) is $\frac{1}{2}(\frac{4}{3} - \frac{1}{3m} + \frac{4}{3} - \frac{1}{3m} + 1) = \frac{11}{6} - \frac{1}{3m}$ (resp. $\frac{1}{2}(1 + \epsilon + 1 + \epsilon + 1) = \frac{3}{2} + \epsilon$).

3. Weak model of execution

3.1. Identical machines with release dates

We consider in this section $P|r_i|C_{\max}$ in the weak model. Every task i has now a private pair (t_i, r_i) (its type), where t_i is the length of task i and r_i its release date. Each task i may bid any pair (b_i, r_i^b) such that $b_i \geq t_i$ and $r_i^b \geq r_i$. Notice here that

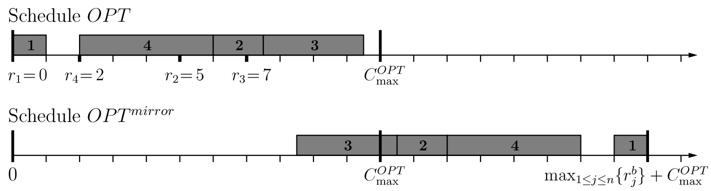

Fig. 2. An illustration of execution of algorithm MTE on machine m_i. We give an example of schedules OPT and OPT^{mirror} with four tasks on machine m_i such that $(b_1 = 1, r_1 = 0)$, $(b_2 = 1.5, r_2 = 5)$, $(b_3 = 3, r_3 = 7)$, $(b_4 = 4, r_4 = 2)$, $\max_{1\le j\le n}\{r_j^b\} = 8$ and $C_{\max}^{OPT} = 11$.

we consider that task i may not bid a release date smaller than its real release date i.e. $r_i^b < r_i$, because otherwise, the task may be scheduled before r_i in the final schedule and thus, the final schedule may be infeasible.

Algorithm MID-TIME EQUALIZER (MTE)

1. Let $C_{\max}^{OPT}$ be the makespan of an optimal schedule OPT for $P|r_i|C_{\max}$.
   Let m_i be the machine where Task i is executed in OPT.
   Let $C_i(OPT)$ be the completion time of Task i in OPT.
2. Construct Schedule OPT^{mirror} in which every task i is executed on machine m_i and start at Time $\max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT} - C_i(OPT)$.
3. Choose Schedule OPT or OPT^{mirror} each with probability 1/2.

Fig. 2 illustrates the construction of schedules OPT and OPT^{mirror} in algorithm MTE on any machine m_i.

The main idea of algorithm Mid-Time Equalizer (MTE) is to make equal the expected time at which every task has executed half of its total length. More precisely, we prove below that the expected mid-time of every task in the final schedule constructed by MTE is the average between its mid-time in OPT and in OPT^{mirror} and it is equal to $\frac{1}{2}\left(\max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT}\right)$ (i.e. the same value for every task). This property will be used in the proof of Theorem 4 in order to show that MTE is truthful in the weak model of execution. In the example given in Fig. 2, the expected mid-time of the four tasks is $\frac{1}{2}\left(\max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT}\right)$ and it is equal to $\frac{8+11}{2} = 9.5$.

Note that as we consider that for every i, we have $r_i^b \ge r_i$, we get $\max_{1\le j\le n}\{r_j^b\} \ge \max_{1\le j\le n}\{r_j\}$. Moreover, as $C_i(OPT) \le C_{\max}^{OPT}$, every task i starts in schedule OPT^{mirror} at time $\max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT} - C_i(OPT) \ge \max_{1\le j\le n}\{r_j\} \ge r_i$. Thus, schedule OPT^{mirror} respects all the constraints of the release dates.

Theorem 4. *MTE is a randomized, truthful and $\frac{3}{2}$-approximate algorithm in the weak model of execution for $P|r_i|C_{\max}$.*

Proof. Let us prove that the expected completion time of every task is minimum when it tells the truth. By definition of MTE, the expected completion time C_i of any task i is the average between its completion time $C_i(OPT)$ in schedule OPT and its completion time $C_i(OPT^{mirror})$ in schedule OPT^{mirror}. In the weak model of execution, every task i is completed b_i units of time after its starting time. Thus, we have

$$\begin{aligned} C_i &= \frac{1}{2}\left(C_i(OPT) + \max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT} - C_i(OPT) + b_i\right) \\ &= \frac{1}{2}\left(\max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT} + b_i\right). \end{aligned}$$

For every task i, its completion time $C_i = \frac{1}{2}\left(\max_{1\le j\le n}\{r_j^b\} + C_{\max}^{OPT} + b_i\right)$ reaches its minimum value when i tells the truth (i.e. when i bids simultaneously $b_i = t_i$ and $r_i^b = r_i$), because

- for every $r_i^b \ge r_i$, both $C_{\max}^{OPT}$ and b_i obviously do not decrease if i bids $(b_i > t_i, r_i^b)$ instead of $(b_i = t_i, r_i^b)$, and
- for every $b_i \ge t_i$, both $\max_{1\le j\le n}\{r_j^b\}$ and $C_{\max}^{OPT}$ obviously do not decrease if i bids $(b_i, r_i^b > r_i)$ instead of $(b_i, r_i^b = r_i)$.

It is then clear that MTE is truthful and thus we may consider in what follows that for every i, we have $b_i = t_i$ and $r_i^b = r_i$. The expected approximation ratio of MTE will be the average between the approximation ratios of OPT and OPT^{mirror}. In OPT, all tasks end before or at time $C_{\max}^{OPT}$. Thus, as for every i, $b_i = t_i$, $C_{\max}^{OPT}$ is the makespan of an optimal solution computed with the types of the agents, and thus, OPT is optimal. In OPT^{mirror}, all tasks end before or at time $\max_{1\le j\le n}\{r_j\} + C_{\max}^{OPT}$ (because for every i, $r_i^b = r_i$ by definition of MTE). Given that $\max_{1\le j\le n}\{r_j\} \le C_{\max}^{OPT}$, all tasks in OPT^{mirror} terminate before or at time $2C_{\max}^{OPT}$. Thus, OPT^{mirror} is 2-approximate. Hence the expected approximation ratio of algorithm MTE is $\frac{1}{2}(1+2) = \frac{3}{2}$. □

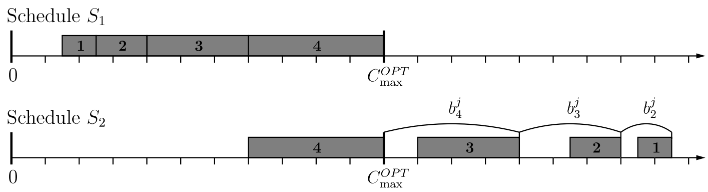

Fig. 3. An illustration of execution of algorithm CTE on machine j. An example of schedules S_1 and S_2 is given with four tasks in OPT_j such that $b_{j_1}^j = 1$, $b_{j_2}^j = 1.5$, $b_{j_3}^j = 3$, $b_{j_4}^j = 4$ and $C_{\max}^{OPT} = 11$.

3.2. Unrelated machines

We consider in this section the case with unrelated machines ($R||C_{\max}$) in the weak model of execution. Here, every task i has a private vector $(t_i^1, \ldots, t_i^m)$ (his type), where t_i^j $(1 \le j \le m)$ is the processing time of i if it is executed on machine j. Every task i bids any vector $(b_i^1, \ldots, b_i^m)$ with $b_i^1 \ge t_i^1, \ldots, b_i^m \ge t_i^m$.

Algorithm COMPLETION TIME EQUALIZER (CTE)
1. Let $C_{\max}^{OPT}$ be the makespan of an optimal schedule OPT for $R\|\|C_{\max}$. Let OPT_j be the sub-schedule of OPT on Machine j. Let $b_{j_1}^j \le \cdots \le b_{j_k}^j$ be the bids (sorted by increasing order) of the k tasks in OPT_j.
2. Construct schedule S_1 as follows: for every machine j $(i \le j \le m)$, every task j_i $(1 \le i \le k)$ in OPT_j is executed on machine j by starting at time $C_{\max}^{OPT} - \sum_{l=i}^{k} b_{j_l}^j$.
3. Construct schedule S_2 as follows: for every machine j $(i \le j \le m)$, every task j_i $(1 \le i \le k-1)$ in OPT_j is executed on machine j by starting at time $C_{\max}^{OPT} - b_{j_i}^j + \sum_{l=i+1}^{k} b_{j_l}^j$ and task j_k is executed on machine j by starting at time $C_{\max}^{OPT} - b_{j_k}^j$.
4. Choose schedule S_1 or S_2 each one with probability 1/2.

Fig. 3 illustrates the construction of schedules S_1 and S_2 in algorithm CTE on machine j.

The intuitive idea of algorithm Completion Time Equalizer is to make equal the expected completion times of the tasks. More precisely, the expected completion time of every task in the final schedule constructed by CTE is the average between its starting time in S_1 and its starting time in S_2 and it is equal to $C_{\max}^{OPT}$ (i.e. the same for all the tasks). This property will be used in the proof of Theorem 5 to show that CTE is truthful in the weak model of execution. For instance, in the example given in Fig. 3, the expected completion time of the four tasks is $C_{\max}^{OPT}$ and it is equal to 11.

Theorem 5. *CTE is a randomized, truthful and $\frac{3}{2}$-approximate algorithm in the weak model of execution for $R||C_{\max}$.*

Proof. We first show that the expected completion time of each task is minimum when it tells the truth. By definition of CTE, the expected completion time C_i of any task i is the average between its completion time in Schedule S_1 and its completion time in Schedule S_2. In the weak model of execution, each task i is completed b_i units of time after its starting time on machine j. Thus, we have

$$C_i = \frac{1}{2}\left(\left(b_i^j + C_{\max}^{OPT} - \sum_{l=i}^{k} b_{j_l}^j\right) + \left(b_i^j + C_{\max}^{OPT} - b_i^j + \sum_{l=i+1}^{k} b_{j_l}^j\right)\right) = C_{\max}^{OPT}.$$

For every task i, $C_i = C_{\max}^{OPT}$ reaches its minimum value when i tells the truth because $C_{\max}^{OPT}$ obviously does not decrease if for any i, j, task i bids $b_i^j > t_i^j$ instead of $b_i^j = t_i^j$. Hence, CTE is truthful and so we can consider in the following that for every i, j, we have $b_i^j = t_i^j$. In schedule S_1, all tasks finish before or at time $C_{\max}^{OPT}$. Thus, as for every i, j, $b_i^j = t_i^j$, $C_{\max}^{OPT}$ is the makespan of an optimal solution computed with the types of the agents, S_1 is optimal. In S_2, on each machine j, all tasks end before or at time $C_{\max}^{OPT} + \sum_{l=2}^{k} b_{j_l}^j$. As $\sum_{l=2}^{k} b_{j_l}^j \le C_{\max}^{OPT}$, all tasks in S_2 end before or at time $2C_{\max}^{OPT}$. Thus, S_2 is 2-approximate. Finally, the expected approximation ratio of algorithm CTE is $\frac{1}{2}(1+2) = \frac{3}{2}$. □

References

[1] E. Angel, E. Bampis, F. Pascual, Truthful algorithms for scheduling selfish tasks on parallel machines, Theoretical Computer Science (short version in WINE 2005) 369 (2006) 157–168.

[2] E. Angel, E. Bampis, F. Pascual, A. Tchetgnia, On truthfulness and approximation for scheduling selfish tasks, Journal of Scheduling (2009) doi:10.2007/s10951-009-0118-8.

[3] V. Auletta, R. De Prisco, P. Penna, P. Persiano, How to route and tax selfish unsplittable traffic, in: 16th ACM Symposium on Parallelism in Algorithms and Architectures, 2004, pp. 196–204.

[4] R. Mueller, B. Heydenreich, M. Uetz, Games and mechanism design in machine scheduling — an introduction, Research Memoranda 022, Maastricht: METEOR, Maastricht Research School of Economics of Technology and Organization, 2006.

[5] G. Christodoulou, L. Gourvès, F. Pascual, Scheduling selfish tasks: about the performance of truthful algorithms, in: 13th International Computing and Combinatorics Conference, in: Lecture Notes in Computer Science, vol. 4598, Springer, 2007, pp. 187–197.

[6] G. Christodoulou, E. Koutsoupias, A. Nanavati, Coordination mechanisms, in: Proceedings of the 31st International Colloquium on Automata, Languages, and Programming, ICALP, in: LNCS, vol. 3142, Springer, 2004, pp. 345–357.

[7] R.L. Graham, Bounds for certain multiprocessing anomalies, Bell System Technical Journal 45 (1966) 1563.