

The Anarchy of Scheduling Without Money

Yiannis Giannakopoulos¹, Elias Koutsoupias², and Maria Kyropoulou²

¹ University of Liverpool, Liverpool, UK

ygiannak@liverpool.ac.uk

² University of Oxford, Oxford, UK

{elias,kyropoul}@cs.ox.ac.uk

Abstract. We consider the scheduling problem on n strategic unrelated machines when no payments are allowed, under the objective of minimizing the makespan. We adopt the model introduced in [Koutsoupias 2014] where a machine is *bound* by her declarations in the sense that if she is assigned a particular job then she will have to execute it for an amount of time at least equal to the one she reported, even if her private, true processing capabilities are actually faster. We provide a (non-truthful) randomized algorithm whose pure *Price of Anarchy* is arbitrarily close to 1 for the case of a single task and close to n if it is applied independently to schedule many tasks. Previous work considers the constraint of truthfulness and proves a tight approximation ratio of $(n + 1)/2$ for one task which generalizes to $n(n + 1)/2$ for many tasks. Furthermore, we revisit the truthfulness case and reduce the latter approximation ratio for many tasks down to n , asymptotically matching the best known lower bound. This is done via a detour to the relaxed, fractional version of the problem, for which we are also able to provide an optimal approximation ratio of 1. Finally, we mention that all our algorithms achieve optimal ratios of 1 for the social welfare objective.

1 Introduction

We consider a variant of the scheduling problem proposed by Koutsoupias [11] where no payments are allowed and the machines are bound by their declarations. In particular, the goal is to allocate a set of tasks to strategic unrelated machines while minimizing the makespan. The time/cost needed by a machine to execute a task is private information of the machine. Each machine is rational and selfish, and will misreport its costs in an attempt to minimize its own overall running time, under the assumption that if she is allocated a task, she will execute it for at least the declared cost (more specifically, for the maximum among her true and reported execution times). We are interested in designing allocation protocols that do not use payments and the stable outcomes are not far from the non-strategic, centrally enforced optimum makespan.

The field of Mechanism Design [17] focuses on the implementation of desired outcomes. Given the strategic behaviour of the players who provide the input

Supported by ERC Advanced Grant 321171 (ALGAME) and EPSRC grant EP/M008118/1. A full version of this paper can be found in [8].

and a specific objective function that measures the quality of the outcome, the challenge is to design mechanisms which are able to elicit a desired behaviour from the players, while at the same time optimizing that objective value. A primary designer goal that has been extensively studied is that of *truthfulness*, under the central solution concept of dominant strategies: a player should be able to optimize her own individual utility by reporting truthfully, no matter what strategies the other players follow. However, achieving this is not always compatible with maintaining a good objective value [9,21]. The introduction of *payments* was suggested as a means towards achieving these goals as a carefully designed payment scheme incentivizes the players to make truthful declarations. The goal now becomes to design such algorithms (termed *mechanisms*) which utilize monetary compensations in order to impose truthful behaviour while optimizing the objective function [16].

There are many situations, though, where the use of payments might be considered unethical [17], illegal (e.g. organ donations) or even just impractical. For this reason researchers have started turning their attention to possible ways of achieving truthfulness without the use of payments. In such a setting, in order to circumvent Social Choice impossibility results (e.g. the seminal Gibbard-Satterthwaite [9,21] theorem) domains with richer structure have to be considered. Procaccia and Tennenholtz [20] were the first to consider achieving truthfulness without using payments, by sacrificing the optimality of the solution and settling for just an approximation, in the context of facility location problems. Similar questions have been considered in the context of inter-domain routing [14], in assignment problems [6], and in the setting of allocating items to two players (with the use of a certain artificial currency) [10]. Moreover, (exact, as opposed to approximate) mechanism design without money has a rich history in the social choice literature.

Clearly, truthfulness is a property desired by any mechanism designer; if the mechanism can ensure that no player can benefit from misreporting, the designer knows what kind of player behaviour and outcome to expect. Moreover, the focus on truthful mechanisms has been largely motivated by the Revelation Principle stating that essentially every equilibrium state of a mechanism can be simulated by a truthful mechanism which achieves the same objective. However this is no longer possible in the variant we examine here, due to the fact that the players are bound by their declarations and thus don't have quasi-linear utilities. So, it is no longer without loss of generality if we restrict attention to truthful mechanisms. For mechanisms that are not truthful, *Price of Anarchy* (PoA) [12] analysis is the predominant, powerful tool for quantifying the potential suboptimality of the outcomes/equilibria; it measures the impact the lack of coordination (strategic behaviour) has on the solution quality, by comparing it to the optimal, non-strategic solution.

Scheduling is one of the most influential problems in Algorithmic Game Theory and has been studied extensively. In its most general form, the goal is to schedule m tasks to n parallel machines with arbitrary processing times, in order to minimize the makespan. In the front where payments are allowed and truthfulness

comes at no extra cost given the strategic nature of the machines Nisan and Ronen [16] first considered the mechanism design approach of the problem. They prove that the well known VCG mechanism achieves an n -approximation of the optimal makespan, while no truthful deterministic mechanism can achieve approximation ratio better than 2. The currently known best lower bound is 2.61 [13] while Ashlagi et al. [2] prove the tightness of the upper bound for anonymous mechanisms. With respect to randomized (truthful in expectation) mechanisms as well as fractional ones, the best known bounds are $(n + 1)/2$ and $2 - 1/n$ [5, 15]. We note that the aforementioned lower bounds disregard computational feasibility and simply rely on the requirement for truthfulness.

In an attempt to get positive results when payments are not allowed in the scheduling context, Koutsoupias [11] first considered the plausible assumption that the machines are bound by their declarations. This was influenced by the notion of impositions that appeared in [7, 18] and was applied in facility location as well as digital goods pricing settings. The notion of winner imposition fits within the framework of approximate mechanism design without payments. A more powerful framework that is also very much related to this assumption is the notion of verification that appears in [3, 16, 19]. The mechanisms in this context are allowed to use payments and simply give or deny payments to machines after they discover their true execution costs. Relevant works include [1, 4] where the scheduling problem of selfish tasks is considered again under the assumption that the players who control the tasks are bound by their declarations.

Our Results. In this work we adopt the model of [11]. For the case of scheduling a single task Koutsoupias [11] proved that the approximation ratio of any mechanism is at least $(n + 1)/2$ and gave a mechanism matching this bound, where n is the number of machines. When applied to many tasks, this mechanism immediately implies a $n(n + 1)/2$ approximation ratio for the makespan objective. In Sect. 3 we provide a (non-truthful) algorithm which performs considerably better than the best truthful mechanism; even the worst pure equilibrium/outcome of our algorithm achieves an optimal makespan, i.e. our algorithm has a pure PoA of 1. If we run this algorithm independently for each job, we get a task-independent and anonymous algorithm, yielding a PoA of n for any number of tasks. Next, revisiting truthfulness, in Sect. 4 we also show that the mechanism inspired by the LP relaxation of the problem is provably truthful and provides an n -approximation ratio when interpreted as a randomized mechanism, while achieving an optimal approximation ratio 1 for the fractional scheduling problem of divisible tasks. This almost matches the lower bound of $(n + 1)/2$ for truthful mechanisms known from [11]. Finally, in Sect. 5 we briefly study the more optimistic objective of minimizing the makespan at the best possible equilibrium (instead of the worst one used in the Price of Anarchy metric) and show that the natural greedy algorithm achieves an optimal Price of Stability. Due to lack of space some proofs appear only in the full version [8].

2 Model and Notation

We have a set $N = \{1, 2, \dots, n\}$ of unrelated parallel machines and m tasks/jobs that need to be scheduled to these machines. Throughout the text we assume that vector \mathbf{t} denotes the true execution times, i.e. $t_{i,j}$ is the time machine i needs to execute task j . This is private knowledge of each machine i . Let $\hat{\mathbf{t}}$ denote the corresponding (not necessarily true) *declarations* of the machines for these costs.

A (randomized) *allocation protocol* takes as input the machines' declarations $\hat{\mathbf{t}}$ and outputs an allocation \mathbf{A} of tasks to machines where $A_{i,j}$ is a 0–1 random variable indicating whether or not machine i gets allocated task j and \mathbf{a} is the corresponding probability distribution of allocation, i.e. $a_{i,j} = \Pr[A_{i,j} = 1]$ where of course $\sum_{i=1}^n a_{i,j} = 1$ for any task j .

If a machine i is allocated some task j , we assume that the machine will execute the task for time $\max\{t_{i,j}, \hat{t}_{i,j}\}$. So, the expected cost/workload of machine i is defined as

$$C_i(\hat{\mathbf{t}}|\mathbf{t}_i) = \sum_{j=1}^m a_{i,j}(\hat{\mathbf{t}}) \max\{\hat{t}_{i,j}, t_{i,j}\}, \tag{1}$$

while the *makespan* is computed as the average maximum execution time

$$\mathcal{M}(\hat{\mathbf{t}}|\mathbf{t}_i) = \mathbb{E}_{\mathbf{A} \sim \mathbf{a}} \left[\max_{i=1, \dots, n} \sum_{j=1}^m A_{i,j} \max\{\hat{t}_{i,j}, t_{i,j}\} \right].$$

To simplify notation, whenever the true execution times \mathbf{t} are clear from the context we will drop them and simply use $C_i(\hat{\mathbf{t}})$ and $\mathcal{M}(\hat{\mathbf{t}})$.

The allocation protocol is called *truthful*, or truthful mechanism, if it does not give incentives to the machines to misreport their true execution costs. Formally, for every machine i and declarations vector $\hat{\mathbf{t}}$,

$$C_i(\mathbf{t}_i, \hat{\mathbf{t}}_{-i}) \leq C_i(\hat{\mathbf{t}}),$$

where $(\mathbf{x}_i, \mathbf{y}_{-i})$ denotes the vector of declarations where machine i has deviated to \mathbf{x}_i while all other machines report costs as in \mathbf{y} . The *approximation ratio* measures the performance of truthful mechanisms and is defined as the maximum ratio, over all instances, of the objective value (makespan) under that mechanism over the optimal objective value achievable by a centralized solution which ignores the truthfulness constraint.

If an allocation protocol is not truthful (we simply refer to it as algorithm), we measure its performance by the quality of its Nash equilibria; the states from which no player has the incentive to unilaterally deviate. The *Price of Anarchy* (PoA) is established as a meaningful benchmark and captures the maximum ratio, over all instances, of the objective value of the worst equilibrium over that of the optimal centralized solution that ignores the machines' incentives. For most part of this paper we restrict attention to pure Nash equilibria where the machines make deterministic reports about their execution costs, and we will from now on refer to them simply as *equilibria*. Then, the corresponding

benchmark is called pure PoA. A more optimistic benchmark is the *Price of Stability* (PoS) which compares the objective value of the best equilibrium to the value of the optimal centralized solution.

The makespan objective is inherently different if we consider *divisible tasks*, i.e. fractional allocations. In that case, each machine is allocated a portion of each task by the protocol and the makespan is computed as the maximum of the execution times of the machines, namely

$$\mathcal{M}^f(\mathbf{t}) = \max_{i=1\dots n} \sum_{j=1}^m \alpha_{i,j} t_{i,j}$$

where $\alpha_{i,j} \in [0, 1]$ is the *fraction* of task j allocated to machine i . Again, it must be that $\sum_{i=1}^n \alpha_{i,j} = 1$ for any task j . Notice here that each fractional algorithm with allocation fractions α naturally gives rise to a corresponding randomized integral algorithm with allocation probabilities $\mathbf{a} = \alpha$, whose makespan is within a factor of n from the fractional one¹, i.e. for any cost matrix \mathbf{t}

$$\mathcal{M}^f(\mathbf{t}) \leq \mathcal{M}(\mathbf{t}) \leq n \cdot \mathcal{M}^f(\mathbf{t}). \quad (2)$$

Except when clearly stated otherwise, in this paper we deal with the integral version of the scheduling problem.

Social Welfare. An alternative objective, very common in the Mechanism Design literature, is that of optimizing *social welfare*, i.e. minimizing the combined costs of all players: $\mathcal{W}(\hat{\mathbf{t}}) = \sum_{i=1}^n C_i(\hat{\mathbf{t}})$. It is not difficult to see³ that the makespan and social welfare objectives are within a factor of n away, whatever the allocation algorithm \mathbf{a} and the input costs $\hat{\mathbf{t}}$ might be:

$$\mathcal{M}(\hat{\mathbf{t}}) \leq \mathcal{W}(\hat{\mathbf{t}}) \leq n \cdot \mathcal{M}(\hat{\mathbf{t}}). \quad (3)$$

Also notice that for the special case of a single task, since the job is eventually allocated entirely to some machine, the two objectives coincide no matter the number of machines n , i.e. $\mathcal{M}(\hat{\mathbf{t}}) = \mathcal{W}(\hat{\mathbf{t}})$. Because of that and the linearity of the social welfare with respect to the players' costs, it is easy to verify that all algorithms we present in this paper achieve optimal ratios of 1 for that objective, both with respect to equilibrium/PoA and truthfulness analysis (e.g. Theorems 2 and 5). We will not mention that explicitly again in the remaining of the paper and rather focus on the more challenging for our scheduling problem objective of makespan minimization.

3 Price of Anarchy

For clarity of exposition, we first describe our scheduling algorithm in the special case of just $n = 2$ machines (and one task) before presenting the algorithm for

¹ This is due to the fact that for any random variables Y_1, Y_2, \dots, Y_n it is $\mathbb{E}[\max_i Y_i] \leq \mathbb{E}[\sum_i Y_i] = \sum_i \mathbb{E}[Y_i] \leq n \max_i \mathbb{E}[Y_i]$, and also $\max_i \mathbb{E}[Y_i] \leq \mathbb{E}[\max_i Y_i]$ due to the convexity of the max function.

the general case of $n \geq 1$. Since we treat the case of only one task in this section, we use \hat{t}_i and t_i to denote the declared and the true execution time of machine i , respectively, and use a_i to denote i 's allocation probability.

3.1 Warm Up: The Case of Two Machines

To simplify notation, throughout this section we will assume without loss of generality that $\hat{t}_1 \leq \hat{t}_2$, i.e. the input to our algorithm is sorted in nondecreasing order. Notice that the true bids $\mathbf{t} = (t_1, t_2)$ do not have to preserve this ordering, since the highest bidding machine might very well in reality have the fastest execution capabilities.

Our algorithm for the case of two machines, parametrized by two constants $L > 2$, $c > 1$, and denoted by $\mathcal{A}_{L,c}^{(2)}$ is defined by the allocation probabilities in Fig. 1. Whenever parameter c is insignificant in a particular context², we will just use $\mathcal{A}_L^{(2)}$.

	a_1	a_2
if $\hat{t}_1 = \hat{t}_2$	$\frac{1}{2}$	$\frac{1}{2}$
if $\hat{t}_1 < \hat{t}_2 < c \cdot \hat{t}_1$	$\frac{1}{L}$	$1 - \frac{1}{L}$
if $c \cdot \hat{t}_1 \leq \hat{t}_2$	$1 - \frac{1}{L} \frac{\hat{t}_1}{\hat{t}_2}$	$\frac{1}{L} \frac{\hat{t}_1}{\hat{t}_2}$

Fig. 1. Algorithm $\mathcal{A}_{L,c}^{(2)}$ for scheduling a single task to two machines, parametrized by $L > 2$ and $c > 1$. The probability that machine $i = 1, 2$ gets the task is denoted by a_i , and \hat{t}_1, \hat{t}_2 are the reported execution times by the machines.

The main result of this section is the following theorem, showing that by choosing parameter L arbitrarily high, the above algorithm can achieve an optimal Price of Anarchy:

Theorem 1. *For the case of one task and two machines, algorithm $\mathcal{A}_L^{(2)}$ has a (pure) Price of Anarchy of $1 + \frac{1}{L}$ (for any $L > 2$).*

We break down the proof of Theorem 1 in distinct claims.

Claim 1. *At any equilibrium $\hat{\mathbf{t}}$ the ratio of the two bids must be at least c , i.e. $\hat{t}_2 \geq c \cdot \hat{t}_1$.*

² In such case, as it is for example in the statement of Theorem 1, one can simply pick e.g. $c = 1 + \frac{1}{L}$.

Proof. Without loss assume $\hat{t}_1 \neq 0$, since otherwise the claim is trivially true. First, assume for a contradiction that $\hat{t}_1 < \hat{t}_2 < c \cdot \hat{t}_1$. Then the machine with largest report would have an incentive to deviate to bid $t_2 = \max\{c\hat{t}_1, t_2\}$:

$$C_2(\hat{\mathbf{t}}) = \left(1 - \frac{1}{L}\right) \max\{\hat{t}_2, t_2\} > \frac{1}{L}\hat{t}_1 = \frac{\hat{t}_1}{Lt'_2} \max\{t'_2, t_2\} = C_2(\hat{t}_1, t'_2)$$

where the inequality holds since $L > 2$ and the final two equalities hold because for the deviating bid it is $t'_2 \geq t_2, c\hat{t}_1$. Thus $\hat{\mathbf{t}} = (\hat{t}_1, \hat{t}_2)$ could not have been an equilibrium under the assumption that $\hat{t}_1 < \hat{t}_2 < c \cdot \hat{t}_1$.

A similar contradiction can be obtained for the remaining case of $\hat{t}_1 = \hat{t}_2$. In this case, both machines have an incentive to deviate to a bid $t'_1 = \frac{\hat{t}_1}{c} < \hat{t}_1$, since

$$C_1(\hat{\mathbf{t}}) = \frac{1}{2} \max\{\hat{t}_1, t_1\} \geq \frac{1}{2} \max\{t'_1, t_1\} > \frac{1}{L} \max\{t'_1, t_1\} = C_1(t'_1, \hat{t}_2).$$

Claim 2. *At any equilibrium $\hat{\mathbf{t}}$ the machine with the largest report will never have underbid, i.e. $\hat{t}_2 \geq t_2$.*

Proof. Assume for a contradiction that $\hat{t}_2 < t_2$. Then

$$C_2(\hat{\mathbf{t}}) = \frac{\hat{t}_1}{L\hat{t}_2} \max\{\hat{t}_2, t_2\} = \frac{\hat{t}_1}{L\hat{t}_2} t_2 > \frac{\hat{t}_1}{Lt_2} t_2 = C_2(\hat{t}_1, t_2),$$

the first equality holding due to Claim 1 and the last one because $t_2 > \hat{t}_2 \geq \hat{t}_1$.

Claim 3. *At any equilibrium $\hat{\mathbf{t}}$ the smallest bid is given by $\hat{t}_1 = \min\{t_1, \frac{\hat{t}_2}{c}\}$.*

Proof. Assume for a contradiction that $\hat{t}_1 \neq t'_1 = \min\{t_1, \frac{\hat{t}_2}{c}\}$. Then, we will show that the lowest bidding machine would have an incentive to deviate from \hat{t}_1 to t'_1 .

Indeed, first consider the case when $\hat{t}_1 < t'_1$. Then

$$C_1(\hat{\mathbf{t}}) = \left(1 - \frac{\hat{t}_1}{L\hat{t}_2}\right) \max\{\hat{t}_1, t_1\} > \left(1 - \frac{t'_1}{L\hat{t}_2}\right) \max\{t'_1, t_1\} = C_1(t_1, \hat{t}_2).$$

In the remaining case of $\hat{t}_1 > t'_1 = \min\{t_1, \frac{\hat{t}_2}{c}\}$, because of Claim 1 it must be that $t'_1 = t_1 < \hat{t}_1 \leq \frac{\hat{t}_2}{c}$, thus

$$C_1(\hat{\mathbf{t}}) = \left(1 - \frac{\hat{t}_1}{L\hat{t}_2}\right) \max\{\hat{t}_1, t_1\} = \left(1 - \frac{\hat{t}_1}{L\hat{t}_2}\right) \hat{t}_1 > \left(1 - \frac{t_1}{L\hat{t}_2}\right) t_1 = C_1(t_1, \hat{t}_2).$$

where the inequality holds since $x \mapsto \left(1 - \frac{x}{y}\right)x$ is a strictly increasing function for $x \in [0, \frac{y}{2}]$, and indeed $t_1 < \hat{t}_1 < \hat{t}_2 < \frac{L\hat{t}_2}{2}$.

Claim 4. *At any equilibrium $\hat{\mathbf{t}}$ bidding must preserve the relative order of the true execution times, i.e. $t_1 \leq t_2$.*

Proof. For a contradiction assume that $t_2 < t_1$, and first consider the case when $t_2 < \hat{t}_1$. If we pick $t'_2 \in \left(\frac{\hat{t}_1}{c}, \hat{t}_1\right)$, we have

$$C_2(\hat{\mathbf{t}}) = \frac{\hat{t}_1}{L\hat{t}_2} \max\{\hat{t}_2, t_2\} = \frac{\hat{t}_1}{L} > \frac{1}{L} \max\{t'_2, t_2\} = C_2(\hat{t}_1, t'_2).$$

For the remaining case of $\hat{t}_1 \leq t_2 < t_1$, first note that if $t_1 \leq \frac{\hat{t}_2}{c}$ then by Claim 3 we would immediately derive that $\hat{t}_1 = t_1$, which is a contradiction. Hence, we can assume that $\hat{t}_1 = \frac{\hat{t}_2}{c} < t_1$. Then, if $\hat{t}_2 > t_1$ we have that

$$C_1(\hat{\mathbf{t}}) = \left(1 - \frac{\hat{t}_1}{L\hat{t}_2}\right) \max\{\hat{t}_1, t_1\} = \left(1 - \frac{\hat{t}_1}{L\hat{t}_2}\right) t_1 > \frac{1}{L} t_1 = C_1(t_1, \hat{t}_2),$$

the inequality holding because $\frac{\hat{t}_1}{\hat{t}_2} \frac{1}{L} \leq \frac{1}{L} < \frac{1}{2}$, and if $\hat{t}_2 \leq t_1$ then, in the same way, for $t'_1 = \max\{t_1, c\hat{t}_2\}$

$$C_1(\hat{\mathbf{t}}) > \frac{1}{L} t_1 \geq \frac{\hat{t}_2}{L} = \frac{\hat{t}_2}{L\hat{t}_1} \max\{t'_1, t_1\} = C_1(t_1, \hat{t}_2).$$

Proof (Proof of Theorem 1). Claims 1–4 imply that the makespan (and thus also the social cost since we have a single task) of any allocation at equilibrium can be bounded by

$$\begin{aligned} \mathcal{M}(\hat{\mathbf{t}}) &= \left(1 - \frac{\hat{t}_1}{L\hat{t}_2}\right) \max\{\hat{t}_1, t_1\} + \frac{\hat{t}_1}{L\hat{t}_2} \max\{\hat{t}_2, t_2\} \\ &\leq \max\{\hat{t}_1, t_1\} + \frac{1}{L} \hat{t}_1 \\ &\leq \left(1 + \frac{1}{L}\right) t_1, \end{aligned}$$

where t_1 is the optimal makespan.

Also, it is important to mention that it can be verified that there exists at least one (pure Nash) equilibrium, e.g. reporting $\hat{t}_1 = t_1$ and $\hat{t}_2 = \max\{Lc \cdot t_1, t_2\}$.

3.2 The General Case

The algorithm for two machines (and a single task) can be naturally generalized to the case of any number of machines $n \geq 2$. Due to lack of space we only give the definition of the algorithm here and the proof can be found in the full version of the paper [8]. We note that the essence of the techniques and the core ideas we presented in Sect. 3.1 carry over to the general case.

To present our algorithm $\mathcal{A}_{L,c}$ we first need to add some notation. We use \hat{t}_{\min} and \hat{t}_{\sec} to denote the smallest and second smallest declarations in $\hat{\mathbf{t}}$, and N_{\min}, N_{\sec} the corresponding sets of machine indices that make these declarations. (If $N = N_{\min}$, i.e. all machines make the same declaration we define $\hat{t}_{\sec} = \hat{t}_{\min}$). Also let $n_{\min} = |N_{\min}|$ and $n_{\sec} = |N_{\sec}|$.

	$i \in N_{\min}$	$i \in N_{\sec}$	$i \in N \setminus (N_{\min} \cup N_{\sec})$
if $\hat{t}_{\min} = \hat{t}_{\sec}$	$\frac{1}{n}$	$\frac{1}{n}$	$\frac{1}{n}$
if $\hat{t}_{\min} < \hat{t}_{\sec} < c \cdot \hat{t}_{\min}$	$(\frac{1}{L}) / n_{\min}$	$(1 - \frac{1}{L}) / n_{\sec}$	0
if $\hat{t}_{\sec} \geq c \cdot \hat{t}_{\min}$	$\left(1 - \sum_{k \in N \setminus N_{\min}} \frac{\hat{t}_{\min}}{L \cdot \hat{t}_k}\right) / n_{\min}$	$\frac{\hat{t}_{\min}}{L \cdot \hat{t}_i}$	$\frac{\hat{t}_{\min}}{L \cdot \hat{t}_i}$

Fig. 2. Algorithm $\mathcal{A}_{L,c}$ for scheduling a single task to $n \geq 2$ machines, parametrized by $L > 2(n - 1)$ and $c > 1$. The first and second highest reported execution times by the machines are denoted by \hat{t}_{\min} and \hat{t}_{\sec} respectively, while N_{\min}, N_{\sec} denote the corresponding sets of machine indices, and n_{\min}, n_{\sec} their cardinalities.

Our main algorithm $\mathcal{A}_{L,c}$ for the case of one task and n machines, parametrized by $L > 2(n - 1), c > 1$, is defined by the allocation probabilities a_i for each machine $i \in N$ given in Fig. 2.

As the following theorem suggests, by picking a high enough value for L the above algorithm can achieve an optimal performance under equilibrium:

Theorem 2. *For the problem of scheduling one task without payments to $n \geq 2$ machines, algorithm \mathcal{A}_L has a (pure) Price of Anarchy of $1 + \frac{n}{L}$ (for any $L > 2(n - 1)$).*

Multiple Tasks. It is not difficult to extend our single-task algorithm and the result of Theorem 2 to get a task-independent, anonymous algorithm with a pure PoA of n for any number of tasks $m \geq 1$: simply run \mathcal{A}_L independently for each job. Then, the equilibria of the extended setting correspond exactly to players not having an incentive to deviate for any task/round, and the approximation ratio of $1 + \frac{n}{L}$ with respect to the minimum cost $\min_i t_{i,j}$ at every such round $j = 1, \dots, m$, guarantees optimality with respect to the social welfare and thus provides indeed a worst-case n -approximation for the makespan objective (see Eq. 3).

4 Truthful Mechanisms

In this section we turn our attention to truthful algorithms for many tasks and provide a mechanism that achieves approximation ratio n , almost matching the $(n + 1)/2$ known lower bound on truthfulness [11]. The best known ratio before our work was $n(n + 1)/2$, achieved by running the algorithm of Koutsoupias [11] independently for each task. Unfortunately this guarantee turns out to be tight for the particular algorithm (see the full version [8] for a bad instance), thus here we have to devise more involved, non task-independent mechanisms.

4.1 The LP Mechanism

It is a known fact that the LP relaxation of a problem can be a useful tool for designing mechanisms (both randomized and fractional). We recall that the LP relaxation for the scheduling problem is as shown in Fig. 3.

$\begin{aligned} & \text{minimize } \mu \\ \forall j : & \quad \sum_{i=1}^n \alpha_{i,j} = 1 \quad (\text{each task is allocated entirely}) \\ \forall i : & \quad \mu - \sum_{j=1}^m \alpha_{i,j} t_{i,j} \geq 0 \quad (\text{the cost of each machine does not exceed makespan}) \\ \forall i, j : & \quad \alpha_{i,j} \geq 0 \quad (\text{the allocation probabilities are positive}) \end{aligned}$
--

Fig. 3. The LP relaxation for the scheduling problem. Our LP mechanism is defined by an optimal solution $\alpha_{i,j}^{\text{LP}}(\mathbf{t})$ to this program.

We denote an *optimal* solution³ to the above LP by $\alpha^{\text{LP}}(\mathbf{t}), \mu_{\mathbf{t}}^{\text{LP}}$ (dropping the LP superscript whenever this is clear from the context). The vector $\alpha^{\text{LP}}(\mathbf{t})$ can be straightforwardly interpreted as allocation probabilities or allocation fractions giving rise to a randomized and a fractional mechanism, respectively. We refer to the corresponding mechanisms as the LP randomized and the LP fractional mechanism. In Theorem 3 we show that both mechanisms are truthful, hence, we can think of $\mu_{\mathbf{t}}^{\text{LP}}$ as corresponding to the maximum (expected) cost/workload perceived by any machine.

It is a simple observation that in an optimal solution the workload must be fully balanced among all machines and that μ^{LP} can only increase when all execution times increase, i.e. $\mu_{\mathbf{t}}^{\text{LP}} \leq \mu_{\mathbf{t}'}^{\text{LP}}$ for $\mathbf{t} \leq \mathbf{t}'$ (pointwise).

Note that the proof of Theorem 3 is identical in both cases where the α correspond to fractions or allocation probabilities. Hence, the result holds for both the LP randomized and the LP fractional mechanism.

Theorem 3. *Under the LP (fractional or randomized) mechanism, truthfully reporting the execution times is a (weakly) dominant strategy for every machine.*

Proof. Recall that \mathbf{t} and $\hat{\mathbf{t}}$ denote the true and (some) declared execution times for all the machines. Fix some machine i and define vector $\mathbf{t}^{\text{max}_i}$ as follows: row i , $\mathbf{t}_i^{\text{max}_i}$, is the vector of point-wise maxima between true and declared times for machine i , that is

$$\mathbf{t}_i^{\text{max}_i} = (\max\{\hat{t}_{i,1}, t_{i,1}\}, \max\{\hat{t}_{i,2}, t_{i,2}\}, \dots, \max\{\hat{t}_{i,m}, t_{i,m}\}),$$

³ Notice that although $\mu_{\mathbf{t}}^{\text{LP}}$ is unique, there might be various allocation fractions $\alpha_{i,j}$ that give rise to the optimal makespan $\mu_{\mathbf{t}}^{\text{LP}}$, in which case we can choose an arbitrary one for $\alpha_{\mathbf{t}}^{\text{LP}}$, e.g. take the lexicographically smaller.

while every other row $k \neq i$ is $\mathbf{t}_k^{\max_i} = \hat{\mathbf{t}}_k$, i.e. $\mathbf{t}^{\max_i} = (\mathbf{t}_i^{\max_i}, \hat{\mathbf{t}}_{-i})$. Seen as a vector of declarations, \mathbf{t}^{\max_i} corresponds to machine i 's deviation from $\hat{\mathbf{t}}$ to $\mathbf{t}_i^{\max_i}$. Then we can derive the following:

$$\sum_{j=1}^m \alpha_{k,j}(\hat{\mathbf{t}}) t_{k,j}^{\max_i} = \sum_{j=1}^m \alpha_{k,j}(\hat{\mathbf{t}}) \hat{t}_{k,j} = \mu_{\hat{\mathbf{t}}} = \sum_{j=1}^m \alpha_{i,j}(\hat{\mathbf{t}}) \hat{t}_{i,j} \leq \sum_{j=1}^m \alpha_{i,j}(\hat{\mathbf{t}}) t_{i,j}^{\max_i}$$

and thus from the optimality of the LP solutions it must be that

$$\mu_{\mathbf{t}^{\max_i}} \leq \max_{l=1, \dots, n} \left\{ \sum_{j=1}^m \alpha_{l,j}(\hat{\mathbf{t}}) t_{l,j}^{\max_i} \right\} = \sum_{j=1}^m \alpha_{i,j}(\hat{\mathbf{t}}) t_{i,j}^{\max_i}.$$

Bringing everything together and taking into consideration that $(\mathbf{t}_i, \hat{\mathbf{t}}_{-i}) \leq \mathbf{t}^{\max_i}$ we get

$$C_i(\hat{\mathbf{t}}) = \sum_{j=1}^m \alpha_{i,j}(\hat{\mathbf{t}}) \max \{ \hat{t}_{i,j}, t_{i,j} \} \geq \mu_{\mathbf{t}^{\max_i}} \geq \mu_{(\mathbf{t}_i, \hat{\mathbf{t}}_{-i})} = \sum_{j=1}^m \alpha_{i,j}(\mathbf{t}_i, \hat{\mathbf{t}}_{-i}) t_{i,j} = C_i(\mathbf{t}_i, \hat{\mathbf{t}}_{-i}),$$

which shows that indeed, whatever the declarations of the other machines $\hat{\mathbf{t}}_{-i}$, machine i is always (weakly) better off by truthfully reporting \mathbf{t}_i .

Theorem 3 gives rise to the following two results.

Theorem 4. *The LP fractional mechanism has approximation ratio 1 for the fractional scheduling problem without money, for any number of machines and tasks.*

As discussed in Sect. 2, by 2 we know that the above performance guarantee can deteriorate at most by a factor of n when we use the fractions as allocation probabilities for the integral case:

Theorem 5. *The LP randomized mechanism has approximation ratio n for (integrally) scheduling any number of tasks to n machines without money.*

4.2 The Proportional Mechanism

In this section we briefly consider the proportional mechanism which allocates to each machine i a $t_i^{-1} / \sum_{k=1}^n t_k^{-1}$ fraction of the task or probability of getting the task, respectively, depending on whether we consider the randomized or the fractional variant. In [11] it was shown that this algorithm is truthful and that its approximation ratio for randomized allocations of a single task is n . With the following theorem we wish to stress the difference between fractional and (randomized) integral allocations. The theorem is about the fractional case and proves the optimality of the proportional mechanism for scheduling one task without payments.

Theorem 6. *The proportional mechanism has an optimal approximation ratio of 1 for the fractional scheduling problem of a single task. For m tasks the approximation ratio increases to at least m .*

5 Price of Stability and Mixed Equilibria

In this section we attempt a more optimistic approach regarding the problem of scheduling without payments. We consider the benchmark of the *best* (mixed Nash) equilibrium and prove that the following, most natural greedy algorithm can achieve optimality: allocates each task independently to the machine declaring the minimum cost (breaking ties arbitrarily).

Theorem 7. *The Price of Stability of the Greedy algorithm is 1 for scheduling without money any number of tasks to any number of machines.*

References

1. Angel, E., Bampis, E., Pascual, F., Tchetgnia, A.-A.: On truthfulness and approximation for scheduling selfish tasks. *J. Sched.* **12**(5), 437–445 (2009)
2. Ashlagi, I., Dobzinski, S., Lavi, R.: Optimal lower bounds for anonymous scheduling mechanisms. *Math. Oper. Res.* **37**(2), 244–258 (2012)
3. Auletta, V., De Prisco, R., Penna, P., Persiano, G.: The power of verification for one-parameter agents. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 171–182. Springer, Heidelberg (2004)
4. Christodoulou, G., Gourvès, L., Pascual, F.: Scheduling selfish tasks: about the performance of truthful algorithms. In: Lin, G. (ed.) COCOON 2007. LNCS, vol. 4598, pp. 187–197. Springer, Heidelberg (2007)
5. Christodoulou, G., Koutsoupias, E., Kovács, A.: Mechanism design for fractional scheduling on unrelated machines. *ACM Trans. Algorithms* **6**(2), 38: 1–38: 18 (2010)
6. Dughmi, S., Ghosh, A.: Truthful assignment without money. In: EC, pp. 325–334 (2010)
7. Fotakis, D., Tzamos, C.: Winner-imposing strategyproof mechanisms for multiple facility location games. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 234–245. Springer, Heidelberg (2010)
8. Giannakopoulos, Y., Koutsoupias, E., Kyropoulou, M.: The anarchy of scheduling without money. CoRR, abs/1607.03688 (2016). <http://arxiv.org/abs/1607.03688>
9. Gibbard, A.: Manipulation of voting schemes: a general result. *Econometrica* **41**(4), 587–601 (1973)
10. Guo, M., Conitzer, V.: Strategy-proof allocation of multiple items between two agents without payments or priors. In: AAMAS, pp. 881–888 (2010)
11. Koutsoupias, E.: Scheduling without payments. *Theory Comput. Syst.* **54**(3), 375–387 (2014)
12. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. *Comput. Sci. Rev.* **3**(2), 65–69 (2009)
13. Koutsoupias, E., Vidali, A.: A lower bound of $1+\varphi$ for truthful scheduling mechanisms. *Algorithmica* **66**(1), 211–223 (2013)
14. Levin, H., Schapira, M., Zohar, A.: Interdomain routing and games. In: STOC, pp. 57–66 (2008)
15. Mu’alem, A., Schapira, M.: Setting lower bounds on truthfulness: extended abstract. In: SODA, pp. 1143–1152 (2007)
16. Nisan, N., Ronen, A.: Algorithmic mechanism design. *Games Econ. Behav.* **35**(1/2), 166–196 (2001)

17. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: *Algorithmic Game Theory*. Cambridge University Press, New York (2007)
18. Nissim, K., Smorodinsky, R., Tennenholtz, M.: Approximately optimal mechanism design via differential privacy. In: *ITCS*, pp. 203–213 (2012)
19. Penna, P., Ventre, C.: Optimal collusion-resistant mechanisms with verification. *Games Econ. Behav.* **86**, 491–509 (2014)
20. Procaccia, A.D., Tennenholtz, M.: Approximate mechanism design without money. In: *EC*, pp. 177–186 (2009)
21. Satterthwaite, M.A.: Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *J. Econ. Theory* **10**(2), 187–217 (1975)