Algorithmica (2021) 83:2895–2913
https://doi.org/10.1007/s00453-021-00847-2

A New Lower Bound for Deterministic Truthful Scheduling

Yiannis Giannakopoulos[1] · **Alexander Hammerl**[2] · **Diogo Poças**[3]

Received: 14 August 2020 / Accepted: 14 June 2021 / Published online: 23 June 2021

Abstract

We study the problem of truthfully scheduling m tasks to n selfish unrelated machines, under the objective of makespan minimization, as was introduced in the seminal work of Nisan and Ronen (in: The 31st Annual ACM symposium on Theory of Computing (STOC), 1999). Closing the current gap of $[2.618, n]$ on the approximation ratio of deterministic truthful mechanisms is a notorious open problem in the field of algorithmic mechanism design. We provide the first such improvement in more than a decade, since the lower bounds of 2.414 (for $n = 3$) and 2.618 (for $n \to \infty$) by Christodoulou et al. (in: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007) and Koutsoupias and Vidali (in: Proceedings of Mathematical Foundations of Computer Science (MFCS), 2007), respectively. More specifically, we show that the currently best lower bound of 2.618 can be achieved even for just $n = 4$ machines; for $n = 5$ we already get the first improvement, namely 2.711; and allowing the number of machines to grow arbitrarily large we can get a lower bound of 2.755.

A significant part of this work was done while Y. Giannakopoulos and D. Poças were members of the Operations Research group at TU Munich, supported by the Alexander von Humboldt Foundation with funds from the German Federal Ministry of Education and Research (BMBF). D. Poças was also supported by FCT via LASIGE Research Unit, ref. UIDB/00408/2020. A preliminary version of this paper appeared in SAGT 2020 [17]

✉ Diogo Poças
dmpocas@fc.ul.pt

Yiannis Giannakopoulos
yiannis.giannakopoulos@fau.de

Alexander Hammerl
alexander.hammerl@tum.de

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany

[2] TU Munich, Munich, Germany

[3] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Lisbon, Portugal

1 Introduction

Truthful scheduling of unrelated parallel machines is a prototypical problem in *algorithmic mechanism design*, introduced in the seminal paper of Nisan and Ronen [32] that essentially initiated this field of research. It is an extension of the classical combinatorial problem for the makespan minimization objective (see, e.g., [36, Ch. 17] or [20, Sect. 1.4]), with the added twist that now machines are rational, *strategic* agents that would not hesitate to *lie* about their actual processing times for each job, if this can reduce their personal cost, i.e., their own completion time. The goal is to design a scheduling mechanism, using payments as incentives for the machines to *truthfully* report their true processing costs, that allocates all jobs in order to minimize the makespan, i.e., the maximum completion time across machines.

Nisan and Ronen [33] showed right away that no such truthful deterministic mechanism can achieve an approximation better than 2 to the optimum makespan; this is true even for just $n = 2$ machines. It is worth emphasizing that this lower bound is not conditioned on *any* computational complexity assumptions; it is purely a direct consequence of the added truthfulness requirement and holds even for mechanisms that have unbounded computational capabilities. It is interesting to compare this with the classical (i.e., non-strategic) algorithmic setting where we do know [26] that a 2-approximate polynomial-time algorithm does exist and that it is NP-hard to approximate the minimum makespan within a factor smaller than $\frac{3}{2}$. On the positive side, it is also shown in [33] that the mechanism that myopically allocates each job to the machine with the fastest reported time for it, and compensates her with a payment equal to the report of the second-fastest machine, achieves an approximation ratio of n (where n is the number of machines); this mechanism is truthful and corresponds to the paradigmatic VCG mechanism (see, e.g., [31]).

Based on these, Nisan and Ronen [33, Conjecture 4.9] made the bold conjecture that their upper bound of n is actually the *tight* answer to the approximation ratio of deterministic scheduling; more than 20 years after the first conference version of their paper [32] though, very little progress has been made in closing their gap of $[2, n]$. Thus, the *Nisan-Ronen conjecture* remains up to this day one of the most important open questions in algorithmic mechanism design. Christodoulou et al. [10] improved the lower bound to $1 + \sqrt{2} \approx 2.414$, even for instances with only $n = 3$ machines and, soon after, Koutsoupias and Vidali [22] showed that by allowing $n \to \infty$ the lower bound can be increased to $1 + \phi \approx 2.618$. The journal versions of these papers can be found at [11] and [23], respectively. In our paper we provide the first improvement on this lower bound in well over a decade.[1]

Another line of work tries to provide better lower bounds by imposing further assumptions on the mechanism, in addition to truthfulness. Most notably, Ashlagi et al. [1] were actually able to resolve the Nisan-Ronen conjecture for the important special case of *anonymous* mechanisms, by providing a lower bound of n. The

[1] After the conference version of our paper [17], Dobzinski and Shaulker [13] and Christodoulou et al. [9] uploaded manuscripts improving this lower bound to 3 and $\sqrt{n-1}+1$, respectively. The latter result is remarkably the first superconstant lower bound for this problem.

same can be shown for mechanisms with strongly-monotone allocation rules [30, Sect. 3.2] and for mechanisms with additive or local payment rules [33, Sect. 4.3.3].

Better bounds have also been achieved by modifying the scheduling model itself. For example, Lavi and Swamy [25] showed that if the processing times of all jobs can take only two values ("high" and "low") then there exists a 2-approximate truthful mechanism; they also give a lower bound of $\frac{11}{10}$. Very recently, Christodoulou et al. [8] showed a lower bound of $\Omega(\sqrt{n})$ for a slightly generalized model where the completion times of machines are allowed to be submodular functions (of the costs of the jobs assigned to them) instead of additive in the standard setting.

Although in this paper we focus exclusively on deterministic mechanisms, randomization is also of great interest and has attracted a significant amount of attention [30, 33, 38], in particular the two-machine case [4, 24, 27–29]. The currently best general lower bound on the approximation ratio of randomized (universally) truthful mechanisms is $2 - \frac{1}{n}$ [30], while the upper one is $0.837n$ [28]. For the more relaxed notion of *truthfulness in expectation*, the upper bound is $\frac{n+5}{2}$ [29]. Related to the randomized case is also the fractional model, where mechanisms (but also the optimum makespan itself) are allowed to split jobs among machines. For this case, [7] prove lower and upper bounds of $2 - \frac{1}{n}$ and $\frac{n+1}{2}$, respectively; the latter is also shown to be tight for task-independent mechanisms.

Other variants of the strategic unrelated machine scheduling problem that have been studied include the Bayesian model [5, 14, 18] (where job costs are drawn from probability distributions), scheduling without payments [19, 21] or with verification [33, 34, 37], and strategic behaviour beyond (dominant-strategy) truthfulness [16]. The *related* machines model, which is essentially a single-dimensional mechanism design variant of our problem, has of course also been well-studied (see, e.g., [2, 3, 12]) and a deterministic PTAS exists [6, 15].

1.1 Our Results and Techniques

We present new lower bounds on the approximation ratio of deterministic truthful mechanisms for the prototypical problem of scheduling unrelated parallel machines, under the makespan minimization objective, introduced in the seminal work of Nisan and Ronen [33]. Our main result (Theorem 2) is a bound of $\rho \approx 2.755$, where ρ is the solution of the cubic equation (6). This improves upon the lower bound of $1 + \phi \approx 2.618$ by Koutsoupias and Vidali [23] which appeared well over a decade ago [22]. Similar to [23], we use a family of instances with the number of machines growing arbitrarily large ($n \to \infty$).

Furthermore, our construction (see Sect. 3.4) provides improved lower bounds also *pointwise*, as a function of the number of machines n that we are allowed to use. More specifically, for $n = 3$ we recover the bound of $1 + \sqrt{2} \approx 2.414$ by [11]. For $n = 4$ we can already match the 2.618 bound that [23] could achieve only in the limit as $n \to \infty$. The first strict improvement, namely 2.711, comes for $n = 5$. As the number of machines grows, our bound converges to 2.755. Our results are summarized in Table 1.

Table 1 Lower bounds on the approximation ratio of deterministic truthful scheduling, as a function of the number of machines n, given by our Theorem 2 (bottom line)

n	3	4	5	6	7	8	…	∞
Previous work	2.414	2.465	2.534	2.570	2.590	2.601	…	2.618
This paper	2.414	2.618	2.711	2.739	2.746	2.750	…	2.755

The previous tate-of-the-art is given in the line above and first appeared in [10] ($n = 3$) and [22] ($n \geq 4$). The case with $n = 2$ machines was completely resolved in [32], with an approximation ratio of 2

A central feature of our approach is the formulation of our lower bound as the solution to a (non-linear) optimization programme (NLP); we then provide optimal, analytic solutions to it for all values of $n \geq 3$ (Lemma 3). It is important to clarify here that, in principle, just giving *feasible* solutions to this programme would still suffice to provide valid lower bounds for our problem. However, the fact that we pin down and use the actual *optimal* ones gives rise to an interesting implication: our lower bounds are provably the best ones that can be derived using our construction.

There are two key elements that allow us to derive our improved bounds, compared to the approach in previous related works [11, 23]. First, we deploy the weak-monotonicity (Theorem 1) characterization of truthfulness in a slightly more delicate way; see Lemma 1. This gives us better control and flexibility in considering deviating strategies for the machines (see our case-analysis in Sect. 3). Secondly, we consider more involved instances, with two auxiliary parameters (namely r and a; see, e.g., (3) and (4)) instead of just one. On the one hand, this increases the complexity of the solution, which now has to be expressed in an implicit way via the aforementioned optimization programme (NLP). But at the same time, fine-tuning the optimal choice of the variables allows us to (provably) push our technique to its limits. Finally, let us mention that, for a small number of machines ($n = 3, 4, 5$) we get $r = 1/a$ in an optimal choice of parameters. Under $r = 1/a$, we end up with a as the only free parameter, and our construction becomes closer to that of [11, 23]; in fact, for 3 machines it is essentially the same construction as in [11] (which explains why we recover the same lower bound). However, for $n \geq 6$ machines we need a more delicate choice of r.

2 Notation and Preliminaries

Before we go into the construction of our lower bound (Sect. 3), we use this section to introduce basic notation and recall the notions of mechanism, truthfulness, monotonicity, and approximation ratio. We also provide a technical tool (Lemma 1) that is a consequence of weak monotonicity (Theorem 1); this lemma will be used several times in the proof of our main result.

2.1 Unrelated Machine Scheduling

In the unrelated machine scheduling setting, we have a number n of machines and a number m of tasks to allocate to these machines. These tasks can be performed in any order, and each task has to be assigned to exactly one machine; machine i requires t_{ij} units of time to process task j. Hence, the complete description of a problem instance can be given by a $n \times m$ *cost matrix* of the values t_{ij}, which we denote by $\boldsymbol{t}$. In this matrix, row i, denoted by $\boldsymbol{t}_i$, represents the processing times for machine i (on the different tasks) and column j represents the processing times for task j (on the different machines). These values t_{ij} are assumed to be nonnegative real quantities, $t_{ij} \in \mathbb{R}_+$.

Applying the methodology of mechanism design, we assume that the processing times for machine i are known only by machine i herself. Moreover, machines are selfish agents; in particular, they are not interested in running a task unless they receive some compensation for doing so. They may also lie about their processing times if this would benefit them. This leads us to consider the central notion of (direct-revelation) *mechanisms*: each machine reports her values, and a mechanism decides on an allocation of tasks to machines, as well as corresponding payments, based on the reported values.

Definition 1 (Allocation rule, payment rule, mechanism) Given n machines and m tasks,

- a (deterministic) *allocation rule* is a function that describes the allocation of tasks to machines for each problem instance. Formally, it is represented as a function $\boldsymbol{a} : \mathbb{R}_+^{n\times m} \to \{0,1\}^{n\times m}$ such that, for every $\boldsymbol{t} = (t_{ij}) \in \mathbb{R}_+^{n\times m}$ and every task $j = 1, \ldots, m$, there is exactly one machine i with $a_{ij}(\boldsymbol{t}) = 1$, that is,

$$\sum_{i=1}^{n} a_{ij}(\boldsymbol{t}) = 1; \tag{1}$$

- a *payment rule* is a function that describes the payments to machines for each problem instance. Formally, it is represented as a function $p : \mathbb{R}_+^{n\times m} \to \mathbb{R}^n$;
- a (direct-revelation, deterministic) *mechanism* is a pair $(\boldsymbol{a}, p)$ consisting of an allocation and payment rules.

By a feasible allocation, we mean a matrix $\boldsymbol{a} = (a_{ij}) \in \{0,1\}^{n\times m}$ satisfying (1). Given a feasible allocation $\boldsymbol{a}$, we let $\boldsymbol{a}_i$ denote its row i, that is, the allocation to machine i. Similarly, given a payment vector $\boldsymbol{p} \in \mathbb{R}^n$, we let p_i denote the payment to machine i; note that the payments represent an amount of money given to the machine, which is somewhat the opposite situation compared to other mechanism design frameworks (such as auctions, where payments are done by the agents to the mechanism designer).

2.2 Truthfulness and Monotonicity

Whenever a mechanism assigns an allocation $\boldsymbol{a}_i$ and a payment p_i to machine i, this machine incurs a quasi-linear *utility* equal to her payment minus the sum of processing times of the tasks allocated to her,

$$p_i - \boldsymbol{a}_i \cdot \boldsymbol{t}_i = p_i - \sum_{j=1}^{m} a_{ij} t_{ij}.$$

Note that the above quantity depends on the machine's both *true* and *reported* processing times, which in principle might differ. As already explained, machines behave selfishly. Thus, from the point of view of a mechanism designer, we wish to ensure a predictable behaviour of all parties involved. In particular, we are only interested in mechanisms that encourage agents to report their true valuations.

Definition 2 A mechanism $(\boldsymbol{a}, p)$ is *truthful* if every machine maximizes their utility by reporting truthfully, regardless of the reports by the other machines. Formally, for every machine i, every $\boldsymbol{t}_i, \boldsymbol{t}'_i \in \mathbb{R}^m_+$, $\boldsymbol{t}_{-i} \in R_+^{(n-1)\times m}$, we have that

$$p_i(\boldsymbol{t}_i, \boldsymbol{t}_{-i}) - \boldsymbol{a}_i(\boldsymbol{t}_i, \boldsymbol{t}_{-i}) \cdot \boldsymbol{t}_i \geq p_i(\boldsymbol{t}'_i, \boldsymbol{t}_{-i}) - \boldsymbol{a}_i(\boldsymbol{t}'_i, \boldsymbol{t}_{-i}) \cdot \boldsymbol{t}_i. \tag{TR}$$

In (TR), we "freeze" the reports of all machines other than i. The left hand side corresponds to the utility achieved by machine i when her processing times correspond to $\boldsymbol{t}_i$ and she truthfully reports $\boldsymbol{t}_i$. The right hand side corresponds to the utility achieved if machine i lies and reports $\boldsymbol{t}'_i$.

The most important example of a truthful mechanism in this setting is the VCG mechanism that assigns each task independently to the machine that can perform it fastest, and paying that machine (for that task) a value equal to the second-lowest processing time. Note that this is somewhat the equivalent of second-price auctions (that sell each item independently) for the scheduling setting.

A fundamental result in the theory of mechanism design is a very useful property of truthful mechanisms, in terms of "local" monotonicity of the allocation function with respect to single-machine deviations.

Theorem 1 (Weak monotonicity [25, 33]) *Let $\boldsymbol{t}$ be a cost matrix, i be a machine, and $\boldsymbol{t}'_i$ another report from machine i. Let $\boldsymbol{a}_i$ be the allocation of i for cost matrix $\boldsymbol{t}$ and $\boldsymbol{a}'_i$ be the allocation of i for cost matrix $(\boldsymbol{t}'_i, \boldsymbol{t}_{-i})$. Then, if the mechanism is truthful, it must be that*

$$(\boldsymbol{a}_i - \boldsymbol{a}'_i) \cdot (\boldsymbol{t}_i - \boldsymbol{t}'_i) \leq 0. \tag{WMON}$$

As a matter of fact, (WMON) is also a *sufficient* condition for truthfulness, thus providing an exact *characterization* of truthfulness [35]. However, for our purposes in this paper we will only need the direction in the statement of Theorem 1 as stated above. We will make use of the following lemma, which exploits the notion of weak monotonicity in a straightforward way. The second part of

this lemma can be understood as a refinement of a technical lemma that appeared before in [11, Lemma 2] (see also [23, Lemma 1]).

Lemma 1 *Suppose that machine i changes her report from $\boldsymbol{t}$ to $\boldsymbol{t}'$, and that a truthful mechanism correspondingly changes her allocation from $\boldsymbol{a}_i$ to $\boldsymbol{a}'_i$. Let $\{1, \dots, m\} = S \cup T \cup V$ be a partition of the tasks into three disjoint sets.*

1. *Suppose that (a) the costs of i on V do not change, that is, $\boldsymbol{t}_{i,V} = \boldsymbol{t}'_{i,V}$ and (b) the allocation of i on S does not change, that is, $\boldsymbol{a}_{i,S} = \boldsymbol{a}'_{i,S}$. Then*

$$(\boldsymbol{a}_{i,T} - \boldsymbol{a}'_{i,T}) \cdot (\boldsymbol{t}_{i,T} - \boldsymbol{t}'_{i,T}) \leq 0.$$

2. *Suppose additionally that (c) the costs of i strictly decrease on her allocated tasks in T and strictly increase on her unallocated tasks in T. Then her allocation on T does not change, that is, $\boldsymbol{a}_{i,T} = \boldsymbol{a}'_{i,T}$.*

Proof To prove the first point, simply apply (WMON) and split the sum into the three sets of tasks,

$$\begin{aligned} 0 &\geq (\boldsymbol{a}_i - \boldsymbol{a}'_i) \cdot (\boldsymbol{t}_i - \boldsymbol{t}'_i) \\ &= (\boldsymbol{a}_{i,S} - \boldsymbol{a}'_{i,S}) \cdot (\boldsymbol{t}_{i,S} - \boldsymbol{t}'_{i,S}) + (\boldsymbol{a}_{i,T} - \boldsymbol{a}'_{i,T}) \cdot (\boldsymbol{t}_{i,T} - \boldsymbol{t}'_{i,T}) + (\boldsymbol{a}_{i,V} - \boldsymbol{a}'_{i,V}) \cdot (\boldsymbol{t}_{i,V} - \boldsymbol{t}'_{i,V}); \end{aligned}$$

since $\boldsymbol{t}_{i,V} = \boldsymbol{t}'_{i,V}$ and $\boldsymbol{a}_{i,S} = \boldsymbol{a}'_{i,S}$, the result follows.

To prove the second point, we look at each term appearing in the inner product $(\boldsymbol{a}_{i,T} - \boldsymbol{a}'_{i,T}) \cdot (\boldsymbol{t}_{i,T} - \boldsymbol{t}'_{i,T})$. Let $j \in T$ be a task which was originally allocated to machine i; then, $a_{i,j} = 1$ and, by assumption, $t_{i,j} > t'_{i,j}$. Since $a'_{i,j}$ is either 1 or 0, it follows that $(a_{i,j} - a'_{i,j})(t_{i,j} - t'_{i,j})$ is either 0 (if the allocation does not change) or $t_{i,j} - t'_{i,j} > 0$ (if the allocation changes). Similarly, assume now that $j \in T$ was originally not allocated to machine i; then, $a_{i,j} = 0$ and, by assumption, $t_{i,j} < t'_{i,j}$. Since $a'_{i,j}$ is either 0 or 1, it follows that $(a_{i,j} - a'_{i,j})(t_{i,j} - t'_{i,j})$ is either 0 (if the allocation does not change) or $(-1) \cdot (t_{i,j} - t'_{i,j}) > 0$ (if the allocation changes). By the first point, the sum over all these terms must be non-positive. We conclude that all these terms must be zero, and hence, the allocation of machine i for tasks on T must not change. □

2.3 Approximation ratio

One of the main open questions in the theory of algorithmic mechanism design is to figure out what is the "best" possible truthful mechanism, with respect to the objective of makespan minimization. This can be quantified in terms of the approximation ratio of a mechanism.

Definition 3 Given n machines and m tasks:

- Let $\boldsymbol{a}$ be a feasible allocation and $\boldsymbol{t}$ a problem instance. The *makespan* of $\boldsymbol{a}$ on $\boldsymbol{t}$ is defined as the quantity

$$\text{makespan}\,(\boldsymbol{a},\boldsymbol{t}) = \max_{i=1,\ldots,n} \sum_{j=1}^{m} a_{ij}t_{ij}.$$

- Let $\boldsymbol{t}$ be a problem instance. The *optimal makespan* is defined as the quantity

$$\text{OPT}\,(\boldsymbol{t}) = \min_{\boldsymbol{a}} \text{makespan}\,(\boldsymbol{a},\boldsymbol{t}),$$

  where the minimum ranges over all feasible allocations.
- Let $\boldsymbol{a}$ be an allocation rule. We say that $\boldsymbol{a}$ has *approximation ratio* $\rho \geq 1$ if, for any problem instance $\boldsymbol{t}$, we have that

$$\text{makespan}\,(\boldsymbol{a}(\boldsymbol{t}),\boldsymbol{t}) \leq \rho\,\text{OPT}\,(\boldsymbol{t});$$

  if no such quantity ρ exists, we say that $\boldsymbol{a}$ has *infinite approximation ratio*.

As shown in [33], the VCG mechanism has an approximation ratio of n, the number of machines. The long-standing conjecture by Nisan and Ronen states that this mechanism is essentially the best one; any truthful mechanism is believed to attain a worst-case approximation ratio of at least n (for sufficiently many tasks). In this paper, we prove lower bounds on the approximation ratio of any truthful mechanism (Table 1 and Theorem 2); our bounds converge to 2.755 as $n \to \infty$.

3 Lower Bound

To prove our lower bound, from here on we assume $n \geq 3$ machines, since the case $n = 1$ is trivial and the case $n = 2$ is resolved by [32] (with an approximation ratio of 2). Our construction will be made with the choice of two parameters r, a, such that $a > 1 > r > 0$. Later we will optimize the choices of r and a in order to achieve the best lower bound possible by our construction.

We will use L_n to denote the $n \times n$ matrix with 0 in its diagonal and ∞ elsewhere,

$$L_n = \begin{bmatrix} 0 & \infty & \cdots & \infty \\ \infty & 0 & \cdots & \infty \\ \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \cdots & 0 \end{bmatrix}.$$

We should mention here that allowing $t_{ij} = \infty$ is a technical convenience. If only finite values are allowed, we can replace ∞ by an arbitrarily high value. We also follow the usual convention, and use an asterisk * to denote a full or partial allocation. Our lower bound begins with the following cost matrix for n machines and $2n - 1$ tasks:

$$A_0 = \left[\begin{array}{c|cccccc} & {}^{*}1 & 1 & a^{-1} & a^{-2} & \cdots & a^{-n+3} \\ & 1 & 1 & a^{-1} & a^{-2} & \cdots & a^{-n+3} \\ & \infty & 1 & \infty & \infty & \cdots & \infty \\ L_n & \infty & \infty & a^{-1} & \infty & \cdots & \infty \\ & \infty & \infty & \infty & a^{-2} & \cdots & \infty \\ & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & \infty & \infty & \infty & \infty & \cdots & a^{-n+3} \end{array}\right]. \tag{2}$$

The tasks of cost matrix A_0 can be partitioned in two groups. The first n tasks (i.e., the ones corresponding to the L_n submatrix) will be called *dummy* tasks. Machine i has a cost of 0 for dummy task i and a cost of ∞ for all other dummy tasks. The second group of tasks, numbered $n+1, \ldots, 2n-1$, will be called *proper* tasks. Notice that machines 1 and 2 have the same costs for proper tasks; they both need time 1 to execute task $n+1$ and time a^{-j+2} to execute task $n+j$, for all $j = 2, \ldots n-1$. Finally for $i \geq 3$, machine i has a cost of a^{-i+3} on proper task $n+i-1$ and ∞ cost for all other proper tasks.

In order for a mechanism to have a finite approximation ratio, it must not assign any tasks with unbounded costs. In particular, each dummy task must be assigned to the unique machine that completes it in time 0; and proper task $n+1$ must be assigned to either machine 1 or 2. Since the costs of machines 1 and 2 are the same on all proper tasks, we can without loss assume that machine 1 receives proper task $n+1$. Hence, the allocation on A_0 should be as (designated by an asterisk) in (2).

Next, we reduce the costs of all proper tasks for machine 1, and get the cost matrix

$$A_1 = \left[\begin{array}{c|cccccc} & r & a^{-1} & a^{-2} & a^{-3} & \cdots & a^{-n+2} \\ & 1 & 1 & a^{-1} & a^{-2} & \cdots & a^{-n+3} \\ & \infty & 1 & \infty & \infty & \cdots & \infty \\ L_n & \infty & \infty & a^{-1} & \infty & \cdots & \infty \\ & \infty & \infty & \infty & a^{-2} & \cdots & \infty \\ & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & \infty & \infty & \infty & \infty & \cdots & a^{-n+3} \end{array}\right]. \tag{3}$$

Under the new matrix A_1, the cost of machine 1 for proper task $n+1$ is reduced from 1 to r; and her cost for any other proper task $n+j$, $j = 2, \ldots, n-1$, is reduced by a factor of a, that is, from a^{-j+2} to a^{-j+1}. The key idea in this step is the following: we want to impose a constraint on r and a that ensures that *at least one* of the proper tasks $n+1, n+2$ is still allocated to machine 1. Using the properties of truthfulness, this can be achieved via the following lemma:

Lemma 2 *Consider a truthful scheduling mechanism that, on cost matrix A_0, assigns proper task $n+1$ to machine 1. Suppose also that*

$$1 - r > a^{-1} - a^{-n+2}. \tag{4}$$

Then, on cost matrix A_1, machine 1 must receive at least one of the proper tasks $n+1, n+2$.

Proof We apply part 1 of Lemma 1, taking $S = \emptyset$, V as the set of dummy tasks, and T as the set of proper tasks. If $\boldsymbol{a}_1$, $\boldsymbol{a}'_1$ denote the allocations of machine 1 for cost matrices A_0, A_1 respectively, we get that

$$(\boldsymbol{a}_{1,T} - \boldsymbol{a}'_{1,T}) \cdot (\boldsymbol{t}_{1,T} - \boldsymbol{t}'_{1,T}) \leq 0.$$

Assume further, for the sake of obtaining a contradiction, that on cost matrix A_1, machine 1 does not get either task $n+1$ or $n+2$; that is, $a'_{1,n+1} = a'_{1,n+2} = 0$. Notice that $a_{1,n+1} = 1$ (since machine 1 gets task $n+1$ on cost matrix A_0) and we have the lower bounds $a_{1,n+2} \geq 0$ as well as $a_{1,n+j} - a'_{1,n+j} \geq -1$ for $j = 3, \ldots, n-1$. Combining all these, we get

$$\begin{aligned}
0 &\geq (\boldsymbol{a}_{1,T} - \boldsymbol{a}'_{1,T}) \cdot (\boldsymbol{t}_{1,T} - \boldsymbol{t}'_{1,T}) \\
&= (a_{1,n+1} - a'_{1,n+1})(t_{1,n+1} - t'_{1,n+1}) + (a_{1,n+2} - a'_{1,n+2})(t_{1,n+2} - t'_{1,n+2}) \\
&\quad + (a_{1,n+3} - a'_{1,n+3})(t_{1,n+3} - t'_{1,n+3}) + \cdots + (a_{1,2n-1} - a'_{1,2n-1})(t_{1,2n-1} - t'_{1,2n-1}) \\
&\geq 1 \cdot (1 - r) + 0 \cdot (1 - a^{-1}) + (-1) \cdot (a^{-1} - a^{-2}) + \cdots + (-1) \cdot (a^{-n+3} - a^{-n+2}) \\
&= 1 - r - a^{-1} + a^{-n+2},
\end{aligned}$$

where in the last step we observe that the terms for tasks $n+3, \ldots, 2n-1$ form a telescoping sum. Thus, we obtain that $1 - r \leq a^{-1} - a^{-n+2}$, which contradicts our original assumption (4). □

For the remainder of our construction, we assume that r and a are such that (4) is satisfied. Next, we split the analysis depending on the allocation of the proper tasks $n+1, \ldots 2n-1$ to machine 1 on cost matrix A_1, as restricted by Lemma 2.

3.1 Case 1: Machine 1 gets *all* proper tasks

In this case, we perform the following changes in machine 1's tasks, obtaining a new cost matrix B_1. We increase the cost of dummy task 1, from 0 to 1, and we decrease the costs of all her proper tasks by an arbitrarily small amount. Notice that

- for the mechanism to achieve a finite approximation ratio, it must still allocate the dummy task 1 to machine 1;
- given that the mechanism does not change the allocation on dummy task 1, and that machine 1 only decreases the completion times of her proper tasks, part 2 of Lemma 1 implies that machine 1 still gets all proper tasks.

Thus, the allocation must be as shown below (for ease of exposition, in the cost matrices that follow we omit the "arbitrarily small" amounts by which we change allocated / unallocated tasks):

$$B_1 = \left[\begin{array}{cccccc|cccccc} {}^*1 & \infty & \infty & \infty & \cdots & \infty & {}^*r & {}^*a^{-1} & {}^*a^{-2} & \cdots & {}^*a^{-n+2} \\ \infty & {}^*0 & \infty & \infty & \cdots & \infty & 1 & 1 & a^{-1} & \cdots & a^{-n+3} \\ \infty & \infty & {}^*0 & \infty & \cdots & \infty & \infty & 1 & \infty & \cdots & \infty \\ \infty & \infty & \infty & {}^*0 & \cdots & \infty & \infty & \infty & a^{-1} & \cdots & \infty \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \infty & \cdots & {}^*0 & \infty & \infty & \infty & \cdots & a^{-n+3} \end{array}\right].$$

This allocation achieves a makespan of $1 + r + a^{-1} + \cdots + a^{-n+2}$, while a makespan of 1 can be achieved by assigning each proper task $n + j$ to machine $j + 1$. Hence, this case yields an approximation ratio of at least $1 + r + a^{-1} + \cdots + a^{-n+2}$.

3.2 Case 2: Machine 1 gets task $n + 1$, but does *not* get all proper tasks

That is, at least one of tasks $n + 2, \ldots 2n - 1$ is not assigned to machine 1. Suppose that task $n + j$ is the lowest indexed proper task that is not allocated to her. We decrease the costs of her *allocated* proper tasks $n + 1, \ldots, n + j - 1$ to 0, while increasing the cost a^{-j+1} of her (unallocated) proper task $n + j$ by an arbitrarily small amount. By Lemma 1, the allocation of machine 1 on the proper tasks $n + 1, \ldots, n + j$ does not change. Hence we get a cost matrix of the form

$$B_2 = \left[\begin{array}{c|cccccc} & {}^*0 & {}^*0 & \cdots & a^{-j+1} & \cdots & a^{-n+2} \\ & 1 & 1 & \cdots & a^{-j+2} & \cdots & a^{-n+3} \\ & \infty & 1 & \cdots & \infty & \cdots & \infty \\ L_n & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ & \infty & \infty & \cdots & a^{-j+2} & \cdots & \infty \\ & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ & \infty & \infty & \cdots & \infty & \cdots & a^{-n+3} \end{array}\right].$$

Since task $n + j$ is not allocated to machine 1, and the mechanism has finite approximation ratio, it must be allocated to either machine 2 or machine $j + 1$. In either case, we increase the cost of the dummy task of this machine from 0 to a^{-j+1}, while decreasing the cost of her proper task $n + j$ by an arbitrarily small amount. For example, if machine 2 got task $n + j$, we would end up with

$$C_2 = \left[\begin{array}{ccccccc|ccccc} {}^*0 & \infty & \infty & \cdots & \infty & \cdots & \infty & 0 & 0 & \cdots & a^{-j+1} & \cdots & a^{-n+2} \\ \infty & {}^*a^{-j+1} & \infty & \cdots & \infty & \cdots & \infty & 1 & 1 & \cdots & {}^*a^{-j+2} & \cdots & a^{-n+3} \\ \infty & \infty & {}^*0 & \cdots & \infty & \cdots & \infty & \infty & 1 & \cdots & \infty & \cdots & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \cdots & {}^*0 & \cdots & \infty & \infty & \infty & \cdots & a^{-j+2} & \cdots & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \cdots & \infty & \cdots & {}^*0 & \infty & \infty & \cdots & \infty & \cdots & a^{-n+3} \end{array}\right].$$

Similarly to the previous Case 1, the mechanism must still allocate the dummy task to this machine, and given that the allocation does not change on the dummy task, Lemma 1 implies that the allocation must also remain unchanged on the proper task $n + j$. Finally, observe that the present allocation achieves a makespan of at least

$a^{-j+1} + a^{-j+2}$, while a makespan of a^{-j+1} can be achieved by assigning proper task $n+j$ to machine 1 and proper task $n+j'$ to machine $j'+1$, for $j' > j$. Hence, this case yields an approximation ratio of at least

$$\frac{a^{-j+1} + a^{-j+2}}{a^{-j+1}} = 1 + a.$$

3.3 Case 3: Machine 1 does *not* get task $n+1$

By Lemma 2, machine 1 must receive proper task $n+2$. In this case, we decrease the cost of her task $n+2$, from a^{-1} to 0, while increasing the cost r of her (unallocated) task $n+1$ by an arbitrarily small amount. Since by truthfulness, the allocation of machine 1 for these two tasks does not change, the allocation must be as below:

$$B_3 = \left[\begin{array}{c|ccccc} & r & {}^*0 & a^{-2} & \cdots & a^{-n+2} \\ & {}^*1 & 1 & a^{-1} & \cdots & a^{-n+3} \\ & \infty & 1 & \infty & \cdots & \infty \\ L_n & \infty & \infty & a^{-1} & \cdots & \infty \\ & \vdots & \vdots & \vdots & \ddots & \vdots \\ & \infty & \infty & \infty & \cdots & a^{-n+3} \end{array}\right].$$

Since task $n+1$ is not allocated to machine 1, and the mechanism has finite approximation ratio, it must be allocated to machine 2. We now increase the cost of the dummy task of machine 2 from 0 to $\max\{r, a^{-1}\}$, while decreasing the cost of her proper task $n+1$ by an arbitrarily small amount. Similarly to Cases 1 and 2, the mechanism must still allocate the dummy task to machine 2, and preserve the allocation of machine 2 on the proper task $n+1$. Thus, we get the allocation shown below:

$$C_3 = \left[\begin{array}{ccccc|ccccc} {}^*0 & \infty & \infty & \infty & \cdots & \infty & r & 0 & a^{-2} & \cdots & a^{-n+2} \\ \infty^* & \max\{r, a^{-1}\} & \infty & \infty & \cdots & \infty & {}^*1 & 1 & a^{-1} & \cdots & a^{-n+3} \\ \infty & \infty & {}^*0 & \infty & \cdots & \vdots & \infty & 1 & \infty & \cdots & \infty \\ \infty & \infty & \infty & {}^*0 & \cdots & \infty & \infty & \infty & a^{-1} & \cdots & \infty \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \infty & \cdots & {}^*0 & \infty & \infty & \infty & \cdots & a^{-n+3} \end{array}\right].$$

This allocation achieves a makespan of at least $1 + \max\{r, a^{-1}\}$, while a makespan of $\max\{r, a^{-1}\}$ can be achieved by assigning proper tasks $n+1, n+2$ to machine 1 and proper task $n+j'$ to machine $j'+1$, for all $j' > 2$. Hence, this case yields an approximation ratio of at least

$$\frac{1 + \max\{r, a^{-1}\}}{\max\{r, a^{-1}\}} = 1 + \min\{r^{-1}, a\}.$$

3.4 Main Result

The three cases considered above give rise to possibly different approximation ratios; our construction will then yield a lower bound equal to the *smallest* of these ratios. First notice that Case 3 always gives a worse bound than Case 2: the approximation ratio for the former is $1 + \min\{r^{-1}, a\}$, whereas for the latter it is $1 + a$. Thus we only have to consider the minimum between Cases 1 and 3.

Our goal then is to find a choice of r and a that achieves the largest possible such value. We can formulate this as a nonlinear optimization problem on the variables r and a. To simplify the exposition, we also consider an auxiliary variable ρ, which will be set to the minimum of the approximation ratios:

$$\begin{aligned}\rho &= \min\left\{1 + r + a^{-1} + \cdots + a^{-n+2}, 1 + \min\{r^{-1}, a\}\right\} \\ &= \min\left\{1 + r + a^{-1} + \cdots + a^{-n+2}, 1 + r^{-1}, 1 + a\right\}.\end{aligned}$$

This can be enforced by the constraints $\rho \le 1 + r + a^{-1} + \cdots + a^{-n+2}$, $\rho \le 1 + r^{-1}$ and $\rho \le 1 + a$. Thus, our optimization problem becomes

$$\begin{aligned}\sup \quad & \rho \\ \text{s.t.} \quad & \rho \le 1 + r + a^{-1} + \cdots + a^{-n+2} \\ & \rho \le 1 + r^{-1} \\ & \rho \le 1 + a \\ & 0 < r < 1 < a \\ & 1 - r > a^{-1} - a^{-n+2}\end{aligned} \tag{NLP}$$

Notice that *any* feasible solution of (NLP) gives rise to a lower bound on the approximation ratio of truthful machine scheduling. In our next lemma, we characterize the limiting optimal solution of the above optimization problem. Thus, the lower bound achieved corresponds to the best possible lower bound using the general construction in this paper.

Lemma 3 *An optimal solution to the optimization problem given by (NLP) is as follows.*

1. *For $n = 3, 4, 5$, choose $\rho = 1 + a$, $r = \frac{1}{a}$, and a as the positive solution of the equation*

$$\begin{aligned}\frac{2}{a} &= a, \quad \text{for } n = 3; \\ \frac{2}{a} + \frac{1}{a^2} &= a, \quad \text{for } n = 4; \\ \frac{2}{a} + \frac{1}{a^2} + \frac{1}{a^3} &= a, \quad \text{for } n = 5.\end{aligned}$$

2. *For $n \geq 6$, choose $\rho = 1 + a$, $r = 1 - \frac{1}{a} + \frac{1}{a^{n-2}}$, and a as the positive solution of the equation*

$$1 + \frac{1}{a^2} + \cdots + \frac{1}{a^{n-3}} + \frac{2}{a^{n-2}} = a. \tag{5}$$

We defer the (admittedly technical) proof of Lemma 3 to Sect. 3.5 below; for the time being, we show how this lemma allows us to prove our main result.

Theorem 2 *No deterministic truthful mechanism for unrelated machine scheduling can have an approximation ratio better than $\rho \approx 2.755$, where ρ is the (unique real) solution of equation*

$$(\rho - 1)(\rho - 2)^2 = 1. \tag{6}$$

For a restricted number of machines the lower bounds can be seen in Table 1.

Proof For n large enough we can use Case 2 of Lemma 3. In particular, taking the limit of (5) as $n \to \infty$, we can ensure a lower bound of $\rho = a + 1$, where a is the (unique) real solution of equation

$$1 + \sum_{i=2}^{\infty} \frac{1}{a^i} = 1 + \frac{1}{a(a-1)} = a.$$

Performing the transformation $a = \rho - 1$, and multiplying throughout by $(\rho - 1)(\rho - 2)$, we get exactly (6).

For a fixed number of machines n, we can directly solve the equations given by either Case 1 ($n = 3, 4, 5$) or Case 2 of Lemma 3 to derive the corresponding value of a, for a lower bound of $\rho = a + 1$. In particular, for $n = 3, 4, 5$ one gets $a = \sqrt{2} \approx 1.414$, $a = \phi \approx 1.618$ (i.e., the *golden ratio*) and $a \approx 1.711$, respectively. The values of ρ for up to $n = 8$ machines are given in Table 1. □

3.5 Proof of Lemma 3

For the remainder of the paper we focus on proving Lemma 3, that is, we characterize the limiting optimal solution of (NLP). We begin by introducing a new variable $z = a^{-1}$, and restate the problem in terms of r, z, ρ.

$$\begin{aligned} \sup \quad & \rho \\ \text{s.t.} \quad & \rho \le 1 + r + z + \cdots + z^{n-2} \\ & \rho \le 1 + r^{-1} \\ & \rho \le 1 + z^{-1} \\ & 0 < r, z < 1 \\ & r < 1 - z + z^{n-2} \end{aligned} \tag{7}$$

Notice that the function $(r, z) \mapsto \min\{1 + r + z + \cdots + z^{n-2}, 1 + r^{-1}, 1 + z^{-1}\}$, defined in the feasibility domain $D = \{(r, z) \ : \ 0 < r, z < 1 \text{and} r < 1 - z + z^{n-2}\}$, has a continuous extension to the closure $\bar{D} = \{(r, z) \ : \ 0 \le r, z \le 1 \text{ and} r \le 1 - z + z^{n-2}\}$, which is a compact set. By the extreme value theorem, the continuous extension must achieve its supremum at some point in $\bar{D}$; that is to say, the supremum of (7) corresponds to the maximum of the relaxed problem,

$$\begin{aligned} \max \quad & \rho \\ \text{s.t.} \quad & \rho \le 1 + r + z + \cdots + z^{n-2} \\ & \rho \le 1 + r^{-1} \\ & \rho \le 1 + z^{-1} \\ & 0 \le r, z \le 1 \\ & r \le 1 - z + z^{n-2} \end{aligned} \tag{8}$$

which always exists.

Let (r, z, ρ) be an optimal solution. Our next step is to prove that $\rho = 1 + z^{-1}$. Suppose otherwise; then, since $\rho = \min\{1 + r + z + \cdots + z^{n-2}, 1 + r^{-1}, 1 + z^{-1}\}$, one must have that either

$$1 + r + z + \cdots + z^{n-2} < 1 + z^{-1} \qquad \text{or} \qquad 1 + r^{-1} < 1 + z^{-1}. \tag{9}$$

We will show that, under such circumstances, we could find a perturbed $(\tilde{r}, \tilde{z}, \tilde{\rho})$ with a strictly better objective value, thus yielding a contradiction. Our analysis proceeds in three cases.

Case 1: $r = 0$. This implies that $1 + r^{-1} = \infty$, and thus $\rho = 1 + r + z + \cdots + z^{n-2} < 1 + z^{-1} \le 1 + r^{-1}$ Also, since $1 - z + z^{n-2} > 0$ for $0 \le z \le 1$, (8) is not tight, that is to say, $r < 1 - z + z^{n-2}$. Thus, we can increase r by an arbitrarily small $\varepsilon > 0$, thus yielding a feasible solution $(r + \varepsilon, z, \rho + \varepsilon)$ with a strictly better objective value.

Case 2: $r > 0$ and $z = 1$. This cannot occur, since it would imply both

$$1 + r + z + \cdots + z^{n-2} \ge 1 + z^{-1} \qquad \text{and} \qquad 1 + r^{-1} \ge 1 + z^{-1},$$

which would contradict (9).

Case 3: $r > 0$ and $z < 1$. Take $\varepsilon > 0$ sufficiently small and perturb (r, z) to a new pair $(r - \varepsilon, z + \varepsilon)$, so that $r - \varepsilon > 0$, $z + \varepsilon < 1$, and (9) remains valid. Notice that, under this perturbation, r decreases, z increases, and $r + z$ remains constant. Hence, we do not leave the feasibility region; in particular, (8) can be written

as $r + z \leq 1 + z^{n-2}$, and this inequality can only remain valid after the perturbation. Finally, the perturbation increases both left-hand sides and decreases both right-hand sides of (9). Therefore, the perturbed $(\tilde{r}, \tilde{z})$ gives rise to a strictly better objective value.

We have thus deduced that $\rho = 1 + z^{-1}$ in an optimal solution. This allows us to restate the optimization problem,

$$\begin{aligned} \max \quad & 1 + z^{-1} \\ \text{s.t.} \quad & 1 + z^{-1} \leq 1 + r + z + \cdots + z^{n-2} \\ & 1 + z^{-1} \leq 1 + r^{-1} \\ & 0 \leq r, z \leq 1 \\ & r \leq 1 - z + z^{n-2} \end{aligned}$$

Further rearranging, and removing unnecessary inequalities, yields

$$\begin{aligned} \max \quad & 1 + z^{-1} \\ \text{s.t.} \quad & r \geq z^{-1} - z - \cdots - z^{n-2} \\ & r \geq 0 \\ & r \leq z \\ & r \leq 1 - z + z^{n-2} \\ & 0 \leq z \leq 1 \end{aligned}$$

Next observe that we can remove the dependency on r by setting $r = \min\{z, 1 - z + z^{n-2}\}$, as long as a feasible choice of r exists. Thus, we end up with

$$\max \quad 1 + z^{-1}$$

$$\text{s.t.} \quad z^{-1} - z - \cdots - z^{n-2} \leq z \tag{10}$$

$$z^{-1} - z - \cdots - z^{n-2} \leq 1 - z + z^{n-2} \tag{11}$$

$$0 \leq z \leq 1$$

$$0 \leq 1 - z + z^{n-2} \tag{12}$$

Notice that (12) is redundant from $0 \leq z \leq 1$, and can be removed. Also, we can rewrite (10) and (11) as

$$z^{-1} \leq 2z + z^2 + \cdots + z^{n-3} + z^{n-2}, \qquad z^{-1} \leq 1 + z^2 + \cdots + z^{n-3} + 2z^{n-2}. \tag{13}$$

In both of the above inequalities, the left hand side is decreasing in z, from ∞ as $z \to 0$ to 1 at $z = 1$, whereas the right hand side is increasing in z, from either 0 or 1

at $z = 0$ to $n - 1$ at $z = 1$. Hence, there are unique positive solutions $z_{n,1}$, $z_{n,2}$ to the equations

$$z^{-1} = 2z + z^2 + \cdots + z^{n-3} + z^{n-2};$$
$$z^{-1} = 1 + z^2 + \cdots + z^{n-3} + 2z^{n-2};$$

and moreover, (10) and (11) are equivalent to $z \geq z_{n,1}$ and $z \geq z_{n,2}$, respectively. By substituting $z = 1$ in the right hand sides of (13), we get $n - 1$, which is strictly larger than $z^{-1} = 1$ for $n \geq 3$. Therefore, we deduce that $z_{n,1}, z_{n,2} < 1$. Our goal is to maximize $1 + z^{-1}$, which is the same as minimizing z. Since (10) and (11) are equivalent to $z \geq z_{n,1}$ and $z \geq z_{n,2}$, the minimum z is obtained by taking the maximum of $z_{n,1}$, $z_{n,2}$.

We can finally convert back to $a = z^{-1}$. Since $0 < z < 1$, $1 < a < \infty$. We recover r via $r = \min\{z, 1 - z + z^{n-2}\} = \min\{a^{-1}, 1 - a^{-1} + a^{-n+2}\}$. Also, the reciprocals of $z_{n,1}$, $z_{n,2}$ correspond to the unique positive solutions $a_{n,1}$, $a_{n,2}$ to the equations

$$a = 2a^{-1} + a^{-2} + \cdots + a^{-n+2}; \tag{14}$$

$$a = 1 + a^{-2} + \cdots + a^{-n+3} + 2a^{-n+2}; \tag{15}$$

and the maximum of $z_{n,1}$, $z_{n,2}$ corresponds to the minimum of $a_{n,1}$, $a_{n,2}$.

Now, for $n = 3, 4, 5$, one can numerically check that $a_{n,1} < a_{n,2}$: we have

$$a_{3,1} \approx 1.414 \quad a_{4,1} \approx 1.618 \quad a_{5,1} \approx 1.711$$
$$a_{3,2} \approx 1.618 \quad a_{4,2} \approx 1.696 \quad a_{5,2} \approx 1.725$$

Thus, for $n = 3, 4, 5$, the optimal solution corresponds to taking a such that $a = 2a^{-1} + a^{-2} + \cdots + a^{-n+2}$; and therefore, (10) is tight, so that $r = a^{-1}$. On the other hand, for $n = 6$, we have $a_{n,1} \approx 1.755 > 1.739 \approx a_{n,2}$; and moreover, as we increment n, the right hand side of (14) increases by an extra term a^{-n+2} whereas the right hand side of (15) increases by $2a^{-n+2} - a^{-n+3} = a^{-n+2}(2 - a)$, which is nonnegative: by plugging $a = 2$ in (15) we see that $a_{n,2} < 2$. Hence, the sequences $a_{n,1}$ and $a_{n,2}$ are both increasing, and in particular $a_{n,2}$ converges to some value $a_{\infty,2}$ which is the solution of

$$a = 1 + \sum_{i=2}^{\infty} \frac{1}{a^i} = 1 + \frac{1}{a(a-1)}.$$

We can directly check that $a_{\infty,2} = a_{6,1} \approx 1.755$, by comparing the respective equations:

$$a_{\infty,2} = 1 + \frac{1}{a_{\infty,2}(a_{\infty,2}-1)} \Rightarrow a_{\infty,2}(a_{\infty,2}-1)^2 = 1$$
$$\Rightarrow a_{\infty,2}^3 - 2a_{\infty,2}^2 + a_{\infty,2} - 1 = 0;$$
$$a_{6,1} = 2a_{6,1}^{-1} + a_{6,1}^{-2} + a_{6,1}^{-3} + a_{6,1}^{-4} \Rightarrow a_{6,1}^5 - 2a_{6,1}^3 - a_{6,1}^2 - a_{6,1} = 1$$
$$\Rightarrow (a_{6,1}^3 - 2a_{6,1}^2 + a_{6,1} - 1)(a_{6,1}+1)^2 = 0.$$

Thus, for $n \geq 6$, we have that $a_{n,2} < a_{\infty,2} = a_{6,1} \leq a_{n,1}$. We conclude that the optimal solution corresponds to taking a such that $a = 1 + a^{-2} + \cdots + a^{-n+3} + 2a^{-n+2}$; this means that (11) is tight, so that $r = 1 - a^{-1} + a^{-n+2}$. This finishes the proof.

References

1. Ashlagi, I., Dobzinski, S., Lavi, R.: Optimal lower bounds for anonymous scheduling mechanisms. Math. Oper. Res. **37**(2), 244–258 (2012). https://doi.org/10.1287/moor.1110.0534
2. Auletta, V., De Prisco, R., Penna, P., Persiano, G.: The power of verification for one-parameter agents. J. Comput. Syst. Sci. **75**(3), 190–211 (2009). https://doi.org/10.1016/j.jcss.2008.10.001
3. Archer, A., Tardos, É.: Truthful mechanisms for one-parameter agents. In: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science (FOCS), pp. 482–491, (2001). https://doi.org/10.1109/sfcs.2001.959924
4. Chen, X., Donglei, D., Zuluaga, L.F.: Copula-based randomized mechanisms for truthful scheduling on two unrelated machines. Theory Comput. Syst. **57**(3), 753–781 (2015). https://doi.org/10.1007/s00224-014-9601-5
5. Chawla, S., Hartline, J.D., Malec, D., Sivan B.: Prior-independent mechanisms for scheduling. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC), pp. 51–60, (2013). https://doi.org/10.1145/2488608.2488616
6. Christodoulou, G., Kovács, A.: A deterministic truthful PTAS for scheduling related machines. SIAM J. Comput. **42**(4), 1572–1595 (2013). https://doi.org/10.1137/120866038
7. Christodoulou, G., Koutsoupias, E., Kovács, A.: Mechanism design for fractional scheduling on unrelated machines. ACM Trans. Algorithms **6**(2), 1–18 (2010). https://doi.org/10.1145/1721837.1721854
8. Christodoulou, G., Koutsoupias, E., Kovács A.: On the Nisan-Ronen conjecture for submodular valuations. In: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC), pp. 1086–1096, (2020). https://doi.org/10.1145/3357713.3384299
9. Christodoulou, G., Koutsoupias, E., Kovács, A.: On the Nisan-Ronen conjecture. *CoRR*, abs/2011.14434, 2020. [arXiv:2011.14434v3]
10. Christodoulou, G., Koutsoupias, E., Vidali, A.: A lower bound for scheduling mechanisms. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1163–1170, (2007). https://doi.org/10.5555/1283383.1283508
11. Christodoulou, G., Koutsoupias, E., Vidali, A.: A lower bound for scheduling mechanisms. Algorithmica **55**(4), 729–740 (2009). https://doi.org/10.1007/s00453-008-9165-3
12. Dhangwatnotai, P., Dobzinski, S., Dughmi, S., Roughgarden, T.: Truthful approximation schemes for single-parameter agents. SIAM J. Comput. **40**(3), 915–933 (2011). https://doi.org/10.1137/080744992
13. Dobzinski, S., Shaulker, A.: Improved lower bounds for truthful scheduling. abs/2007.04362, 2020. [arXiv:2007.04362v2]
14. Daskalakis, C., Weinberg, S.M.: Bayesian truthful mechanisms for job scheduling from bi-criterion approximation algorithms. In: Proceedings of the 26th annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1934–1952, (2015). https://doi.org/10.1137/1.9781611973730.130
15. Epstein, L., Levin, A., Van Stee, R.: A unified approach to truthful scheduling on related machines. Math. Oper. Res. **41**(1), 332–351 (2016). https://doi.org/10.1287/moor.2015.0730
16. Filos-Ratsikas, A., Giannakopoulos, Y., Lazos, P.: The Pareto frontier of inefficiency in mechanism design. In: Proceedings of the 15th Conference on Web and Internet Economics (WINE). pp. 186–199, (2019). https://doi.org/10.1007/978-3-030-35389-6_14

17. Giannakopoulos, Y., Hammerl, A., Poças, D.: A new lower bound for deterministic truthful scheduling. In: Proceedings of the 13th symposium on algorithmic game theory (SAGT), (2020). https://doi.org/10.1007/978-3-030-57980-7_15
18. Giannakopoulos, Y., Kyropoulou, M.: The VCG mechanism for Bayesian scheduling. ACM Trans. Econ. Comput. **5**(4), 19:1-19:16 (2017). https://doi.org/10.1145/3105968
19. Giannakopoulos, Y., Koutsoupias, E., Kyropoulou, M.: The anarchy of scheduling without money. Theor. Comput. Sci **778**, 19–32 (2019). https://doi.org/10.1016/j.tcs.2019.01.022
20. Hall, L.A.: Approximation algorithms for scheduling. In: Hochbaum, D.S. (ed.) Approximation Algorithms for NP-hard Problems, pp. 1–45. PWS Publishing Company, Boston (1997)
21. Koutsoupias, E.: Scheduling without payments. Theory Comput. Syst. **54**(3), 375–387 (2014). https://doi.org/10.1007/s00224-013-9473-0
22. Koutsoupias, E., Vidali, A.: A lower bound of $1 + \phi$ for truthful scheduling mechanisms. In: Proceedings of Mathematical Foundations of Computer Science (MFCS), pp. 454–464, (2007). https://doi.org/10.1007/978-3-540-74456-6_41
23. Koutsoupias, E., Vidali, A.: A lower bound of $1+\phi$ for truthful scheduling mechanisms. Algorithmica **66**(1), 211–223 (2013). https://doi.org/10.1007/s00453-012-9634-6
24. Kuryatnikova, O., Vera, J.C.: New bounds for truthful scheduling on two unrelated selfish machines. Theory Comput. Syst. **64**(2), 199–226 (2019). https://doi.org/10.1007/s00224-019-09927-x
25. Lavi, R., Swamy, C.: Truthful mechanism design for multidimensional scheduling via cycle monotonicity. Games Econ. Behav. **67**(1), 99–124 (2009). https://doi.org/10.1016/j.geb.2008.08.001
26. Lenstra, J.K., Shmoys, D.B., Tardos, É.: Approximation algorithms for scheduling unrelated parallel machines. Math. Program. **46**(1), 259–271 (1990). https://doi.org/10.1007/bf01585745
27. Lu P.: On 2-player randomized mechanisms for scheduling. In: Proceedings of the 5th international workshop on internet and network economics (WINE), pp. 30–41, (2009). https://doi.org/10.1007/978-3-642-10841-9_5
28. Lu, P., Yu, C.: An improved randomized truthful mechanism for scheduling unrelated machines. In: Proceedings of the 25th international symposium on theoretical aspects of computer science (STACS), pp. 527–538, (2008). https://doi.org/10.4230/LIPIcs.STACS.2008.1314
29. Lu, P., Yu, C.: Randomized truthful mechanisms for scheduling unrelated machines. In: Proceedings of the 4th International Workshop on Internet and Network Economics (WINE), pp. 402–413, (2008). https://doi.org/10.1007/978-3-540-92185-1_46
30. Mu'alem, A., Schapira, M.: Setting lower bounds on truthfulness. Games Econ. Behav. **110**, 174–193 (2018). https://doi.org/10.1016/j.geb.2018.02.001
31. Nisan, N.: Introduction to mechanism design (for computer scientists). In: Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V. (eds.) Algorithmic Game Theory, chapter 9. Cambridge University Press, Cambridge (2007). https://doi.org/10.1017/cbo9780511800481.011
32. Nisan, N., Ronen, A.: Algorithmic mechanism design (extended abstract). In: The 31st Annual ACM symposium on Theory of Computing (STOC), pp. 129–140, (1999)
33. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games Econ. Behav. **35**(1/2), 166–196 (2001). https://doi.org/10.1006/game.1999.0790
34. Penna, P., Ventre, C.: Optimal collusion-resistant mechanisms with verification. Games Econ. Behav. **86**, 491–509 (2014). https://doi.org/10.1016/j.geb.2012.09.002
35. Saks M., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings of the 6th ACM Conference on Electronic Commerce (EC), pp. 286–293, (2005). https://doi.org/10.1145/1064009.1064040
36. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2003). https://doi.org/10.1007/978-3-662-04565-7
37. Ventre, C.: Truthful optimization using mechanisms with verification. Theor. Comput. Sci. **518**, 64–79 (2014). https://doi.org/10.1016/j.tcs.2013.07.034
38. Yu, C.: Truthful mechanisms for two-range-values variant of unrelated scheduling. Theor. Comput. Sci. **410**(21–23), 2196–2206 (2009). https://doi.org/10.1016/j.tcs.2009.02.001