Theory of Computing Systems (2020) 64:199–226
https://doi.org/10.1007/s00224-019-09927-x

# New Bounds for Truthful Scheduling on Two Unrelated Selfish Machines

**Olga Kuryatnikova**[1] · **Juan C. Vera**[1]

Published online: 30 May 2019


**Abstract**
We consider the minimum makespan problem for $n$ tasks and two unrelated parallel selfish machines. Let $R_n$ be the best approximation ratio of randomized monotone scale-free algorithms. This class contains the most efficient algorithms known for truthful scheduling on two machines. We propose a new $Min - Max$ formulation for $R_n$, as well as upper and lower bounds on $R_n$ based on this formulation. For the lower bound, we exploit pointwise approximations of cumulative distribution functions (CDFs). For the upper bound, we construct randomized algorithms using distributions with piecewise rational CDFs. Our method improves upon the existing bounds on $R_n$ for small $n$. In particular, we obtain almost tight bounds for $n = 2$ showing that $|R_2 - 1.505996| < 10^{-6}$.

**Keywords** Minimax optimization · Truthful scheduling · Approximation · Piecewise functions · Algorithmic mechanism design

## 1 Introduction and Main Results

Scheduling on unrelated parallel machines is a classical problem in discrete optimization. In this problem one has to allocate $n$ independent, indivisible tasks to $m$ simultaneously working unrelated machines. The goal is to minimize the time to complete all the tasks. This time is called the makespan, and the scheduling problem is called the minimum makespan problem. Lenstra et al. [16] proved that the problem is NP-complete and that a polynomial-time algorithm cannot achieve an approximation ratio less than $\frac{3}{2}$ unless $P = NP$.

✉ Olga Kuryatnikova
o.kuryatnikova@uvt.nl

Juan C. Vera
j.c.veralizcano@uvt.nl

[1] Department of Econometrics and Operations Research, Tilburg University, 5037, AB Tilburg, Netherlands

We restrict ourselves to the case of $m = 2$ machines. For this case there is a linear-time algorithm by Potts [24] and a polynomial-time algorithm by Shchepin and Vakhania [26] which provide $\frac{3}{2}$-approximations. Both algorithms use linear programming (LP) relaxations of integer programs and rounding techniques.

We are interested in the minimum makespan problem in the setting of algorithmic mechanism design. In this setting, every machine belongs to a rational agent who requires payments for performing tasks and aims to maximize his or her utility. Nisan and Ronen [23] introduced this approach to model interactions on the Internet, such as routing and information load balancing. The minimum makespan problem is one of many optimization problems considered in algorithmic mechanism design. These include, among others, combinatorial auctions (see, e.g., [2, 6] and references therein) and graph theoretic problems, such as the shortest paths tree [11] and the maximum matching problem [27].

To solve the minimum makespan problem in algorithmic mechanism design, one can use an *allocation mechanism*. An allocation mechanism consists of two algorithms: one allocates tasks to machines, and the other allocates payments to agents (the machines' owners). The goal of the mechanism is to choose a task allocation algorithm that minimizes the makespan and a payment allocation algorithm that motivates the agents to act according to the wishes of the algorithm designer. For instance, it is desirable that the agents reveal their correct task processing times to the designer. We consider direct revelation mechanisms. These mechanisms collect the information about the processing times from each agent and allocate the tasks and payments based on this information according to a policy known to the agents in advance. To maximize their utilities, the agents can lie about processing times of tasks on their machines. As a result, direct revelation mechanisms may be hard to implement correctly.

To motivate the agents to tell the right processing times, one can use a *truthful* direct revelation mechanism. With such mechanisms, telling the truth becomes a dominant strategy for each agent regardless of what the other agents do. This property guarantees that the processing times used to construct the mechanism are correct. There is a vast literature on truthful mechanisms [4, 21, 23, 25]. Not all task allocation algorithms can be used in truthful mechanisms. For instance, there is no known truthful mechanism for the polynomial-time algorithms by Potts [24] and Shchepin and Vakhania [26]. Finding the best approximation ratio for truthful scheduling on unrelated machines is one of the hardest fundamental questions in mechanism design.

Saks and Yu [25] showed that a task allocation algorithm can be used in a truthful mechanism if and only if the algorithm is *monotone*. Intuitively, a task allocation algorithm is monotone if it assigns a higher load to a machine as long as the processing times on this machine decrease (see Section 2 for the formal definition of monotonicity). In this paper we focus on monotone task allocation algorithms and do not consider the allocation of payments.

Nisan and Ronen [23] show that no deterministic monotone algorithm can achieve an approximation ratio smaller than two, but randomized algorithms can do better in expectation. From here on we say that a randomized allocation algorithm has a given property, e.g., monotonicity, if this property holds with probability one according to

the distribution of the random bits of the algorithm. Randomized task allocation algorithms that are monotone in this sense give rise to universally truthful mechanisms considered in this paper.

Next, a deterministic algorithm is *task-independent* when the allocation of any task does not change as long as the processing times of this task stay fixed. Every deterministic monotone allocation algorithm on two machines with a finite approximation ratio is task-independent (Dobzinski and Sundararajan [6]). Therefore, if a given randomized algorithm has a finite expected approximation ratio, this algorithm is task-independent with probability one. Hence we can restrict ourselves without loss of generality to monotone and task-independent randomized algorithms to find the best truthful approximation ratio for two machines.

Finally, we restrict our attention to *scale-free* algorithms. An algorithm is scale-free if scaling all processing times by some positive number does not influence the output. Following Lu [17], we note that for $m = 2$, scale-freeness and allocation independence imply that the allocation of each task depends only on the ratio of this task's processing times, which simplifies the analysis. Scale-free algorithms are widely used in the literature, and the latest most efficient algorithms for truthful scheduling on two machines by Chen et al. [3], Lu [17] or Lu and Yu [18] are scale-free. In the sequel we work with monotone, task-independent, scale-free (denoted by MIS) task allocation algorithms. These algorithms provide good upper bounds on approximation ratios in scheduling [3, 17–19, 23]. Lu and Yu [17–19] present a way to construct a payment allocation procedure for MIS algorithms which results in truthful allocation mechanisms.

Denote by $R_n$ the best worst-case expected approximation ratio of randomized MIS algorithms for the makespan minimization on two machines with $n$ tasks. For simplicity, in the rest of the paper we call $R_n$ the best approximation ratio. Our approach is to formulate a mathematical optimization problem for $R_n$. This approach was not common until recently when several successful truthful or truthful in expectation mechanisms have been constructed using linear or nonlinear programs [2, 3, 7, 15]. This paper continues the trend to combine optimization with mechanism design and has the following contributions:

1. A $Min - Max$ formulation for $R_n$, see (13) in Corollary 3.
2. A unified approach to construct upper and lower bounds on $R_n$, see Section 4. In formulation (13), the outer minimization is over multivariate cumulative distribution functions (CDFs) and the inner maximization is over the positive orthant in two dimensions. This problem is in general not tractable, so we build bounds on the optimal value. The lower bounds are the result of restricting the inner maximization to a finite subset of the positive orthant. To obtain the upper bounds, we restrict the outer minimization to the set of piecewise constant CDFs. This is a general approach which could work for any $Min - Max$ problem that requires optimizing over a set of functions, not necessarily CDFs.
3. New upper and lower bounds on $R_n$ for $n \in \{2, 3, 4\}$ and the task allocation algorithms corresponding to the given upper bounds (see Table 1). The resulting upper bounds are currently the best for all monotone algorithms (not only MIS) on two machines.

**Table 1** Bounds on $R_n$

| $n$ | Lower bound | | Upper bound | |
|---|---|---|---|---|
| | Existing | New | Existing | New |
| 2 | 1.505949 [17] | 1.5059953 | 1.5068 [3] | 1.5059964 |
| 3 | 1.505949 [17] | 1.5076 | 1.5861 [3] | 1.5238 |
| 4 | 1.505949 [17] | 1.5195 | 1.5861 [3] | 1.5628 |

4. Almost tight bounds on $R_2$ (see Table 1).
   For $n = 2$ tasks the initial problem (13) simplifies to problem (30), where the outer minimization is over univariate CDFs. We use piecewise rational CDFs to obtain the upper and lower bounds with a gap not larger than $10^{-6}$.

Notice that another, less restrictive way to define randomized algorithms would be to say that the properties hold in expectation over the random bits of the algorithm. Monotone in expectation task allocation algorithms can be used in truthful in expectation mechanisms. For certain classes of problems (e.g., for combinatorial auctions), one can convert LP relaxations with rounding into truthful in expectation mechanisms, see Azar et al. [2], Elbassioni et al. [7] or Lavi and Swamy [15]. Truthful in expectation mechanisms could perform better in expectation than the universally truthful ones. In this paper we do not analyze the former type of mechanisms. We refer the reader to Auletta et al. [1] and Lu and Yu [19] for more information on truthfulness in expectation.

The outline of the paper is as follows. In Section 2 we provide more details about randomized MIS algorithms for two machines, describe results from earlier research, and formulate a mathematical optimization problem for $R_n$. In Section 3 we exploit the symmetry of this problem to analyze the performance of MIS algorithms and to obtain our $Min - Max$ formulation (13) for $R_n$. In Section 4 we construct and compute bounds on the optimal value of the $Min - Max$ problem for several small $n$. In Section 5 we analyze the case with two tasks in more detail to improve the bounds for this case. Section 6 concludes the paper. In Section 7 we provide the omitted proofs. All computations are done in MATLAB R2017a on a computer with the processor Intel® Core™ i5-3210M CPU @ 2.5 GHz and 7.7 GiB of RAM. To solve linear programs, we use IBM ILOG CPLEX 12.6.0 solver.

## 2 Preliminaries

Unless otherwise specified, lower-case letters denote numbers, bold lower-case letters denote vectors, and capital letters denote matrices. For a given positive number $n$, let $\mathbb{R}^n$ be the set of real vectors with $n$ entries. The notations $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ refer to nonnegative and strictly positive real-valued vectors, respectively. For a given positive number $m$, we use $[m]$ to denote the set $\{1, \ldots, m\}$. Unless otherwise stated, we

use parentheses to denote vectors and brackets to denote intervals, e.g., $(x_1, x_2)$ is a vector while $[x_1, x_2]$ and $[x_1, x_2)$ are intervals.

The input into the minimum makespan problem with $m$ machines and $n$ tasks is a matrix of processing times $T = (T_{ij}),\ i \in [m],\ j \in [n]$. We describe the solution to the problem by a task allocation matrix $X \in \{0, 1\}^{m\times n}$, such that $X_{ij} = 1$ if task $j$ is processed on machine $i$ and $X_{ij} = 0$ otherwise. Now, for given $X$ and $T$, we define the makespan $M_i$ of machine $i$ and the overall makespan $M$.

$$M_i(X, T) := \sum_{j=1}^{n} X_{ij}T_{ij}, \quad M(X, T) := \max_{i\in[m]} M_i(X, T), \tag{1}$$

The optimal makespan for $T$ is

$$M^*(T) := \min_{X\in\{0,1\}^{m\times n}} M(X, T). \tag{2}$$

For an allocation algorithm $\mathcal{A}$ and an input matrix $T$, $X^{\mathcal{A},T} \in \{0, 1\}^{m\times n}$ denotes the output of $\mathcal{A}$ on $T$. If $\mathcal{A}$ is randomized, $X^{\mathcal{A},T}$ is a random matrix, and we use $M(X^{\mathcal{A},T}, T)$ to denote the expected makespan. The worst-case (expected) approximation ratio of $\mathcal{A}$ equals the supremum of the ratio $\frac{M(X^{\mathcal{A},T},T)}{M^*(T)}$ over all time matrices $T$. We refer the reader to Motwani and Raghavan [20] for a comprehensive discussion on randomized algorithms.

### 2.1 The Best Approximation Ratio of Randomized MIS Algorithms

According to the definition in Section 1, randomized MIS algorithms are monotone, task-independent and scale-free (MIS) with probability one. Therefore, by fixing the random bits of a randomized MIS algorithm, we obtain a deterministic MIS algorithm with probability one. We provide next a formal description of deterministic MIS algorithms.

A task allocation algorithm is monotone if for every two processing time matrices $T$ and $T'$ which differ only on machine $i$, $\sum_{j=1}^{n}(X_{ij}^{\mathcal{A},T} - X_{ij}^{\mathcal{A},T'})(T_{ij} - T'_{ij}) \le 0$ (see [4]). That is, the load of a machine increases as long as the processing times on this machine decrease. An algorithm is task-independent if the allocation of a task depends only on its processing times. To be precise, for any two time matrices $T$ and $T'$ such that $T_{ij} = T'_{ij}$ for task $j$ and all $i \in [m]$, the allocation of task $j$ is identical, i.e., $X_{ij}^{\mathcal{A},T} = X_{ij}^{\mathcal{A},T'}$, for all $i \in [m]$. An algorithm is scale-free if the multiplication of all processing times by the same positive number does not change the allocation. That is, for any $T \in \mathbb{R}_{++}^{m\times n}$ and $\lambda > 0$, the output of the algorithm on the inputs $T$ and $\lambda T$ is identical.

Deterministic MIS algorithms for $m = 2$ have been characterized by Lu [17]:

**Theorem 1** *[Lu [17]] All deterministic MIS algorithms for scheduling on two unrelated machines are of the following form. For every task $j \in [n]$, assign a threshold $z_j \in \mathbb{R}_{++}$ and one of the following two conditions: $T_{1j} < z_j T_{2j}$ or $T_{1j} \le z_j T_{2j}$. The task goes to the first machine if and only if the corresponding condition is satisfied.*

Let $\mathcal{C}$ be the class of (randomized) algorithms which randomly assign a threshold $z_j$ and a condition $T_{1j} < z_j T_{2j}$ or $T_{1j} \leq z_j T_{2j}$ to each task $j$ and then proceed as given in Theorem 1 for the deterministic case. With probability one a randomized MIS algorithm is a MIS algorithm, and therefore of the form given by Theorem 1. Hence, to find the best approximation ratio, it is enough to consider only algorithms in $\mathcal{C}$. Next, we show that to find the best approximation ratio, we can restrict ourselves to a subclass of $\mathcal{C}$.

Let $\mathcal{P}_n$ be the family of Borel probability measures supported on the positive orthant, i.e. $\text{supp}(\mathbb{P}) \subseteq \mathbb{R}^n_{++}$ for all $\mathbb{P} \in \mathcal{P}_n$, where $\text{supp}(\mathbb{P})$ is the support of $\mathbb{P}$. We use $\mathbf{E}_{\mathbb{P}}[\,]$ and $\mathbf{P}_{\mathbb{P}}[\,]$ to denote the expectation and the probability over the measure $\mathbb{P}$, respectively. In the sequel we use the notions of a probability measure and the corresponding probability distribution interchangeably. For a $\mathbb{P} \in \mathcal{P}_n$ we define algorithm $\mathcal{A}^{\mathbb{P}}$ as follows:

**Algorithm 1** A monotone, task-independent, scale-free task allocation algorithm $\mathcal{A}^{\mathbb{P}}$ for two machines.

**Input**: processing time matrix $T = (T_{ij}) \in \mathbb{R}^{2\times n}_{++}$
**Output**: allocation $X \in \{0, 1\}^{2\times n}$
1. Draw a vector of thresholds $(z_1, z_2, \ldots, z_n)$ from $\mathbb{P}$
2. For each task $j = 1, 2, \ldots, n$ do
3. If $\frac{T_{1j}}{T_{2j}} < z_j$: $X_{1j} \leftarrow 1,\ X_{2j} \leftarrow 0$
4. Else: $X_{1j} \leftarrow 0,\ X_{2j} \leftarrow 1$
5. Output $X$

Denote the family of all algorithms of the form above by $\mathcal{A}^{\mathcal{P}_n}$:

$$\mathcal{A}^{\mathcal{P}_n} := \{\mathcal{A}^{\mathbb{P}} \,:\, \mathbb{P} \in \mathcal{P}_n\}.$$

Consider a measure $\mathbb{P} \in \mathcal{P}_n$ and the corresponding algorithm $\mathcal{A}^{\mathbb{P}} \in \mathcal{A}^{\mathcal{P}_n}$. Let $X^{\mathbb{P},T} \in \{0, 1\}^{2\times n}$ be the randomized allocation produced by $\mathcal{A}^{\mathbb{P}}$ on time matrix $T$. The expected makespan of $\mathcal{A}^{\mathbb{P}}$ on $T$ is

$$M(\mathbb{P}, T) = \mathbf{E}_{\mathbb{P}} \max \left\{ \sum_{j\in[n]} T_{2j} X^{\mathbb{P},T}_{2j},\ \sum_{j\in[n]} T_{1j} X^{\mathbb{P},T}_{1j} \right\}. \tag{3}$$

Recall that $M^*(T)$, defined in (2), denotes the optimal makespan for $T$. Let $R_n(\mathbb{P}, T)$ be the expected approximation ratio of $\mathcal{A}^{\mathbb{P}}$ on $T$ and $R_n(\mathbb{P})$ be the worst-case approximation ratio:

$$R_n(\mathbb{P}, T) = \frac{M(\mathbb{P}, T)}{M^*(T)}, \tag{4}$$

$$R_n(\mathbb{P}) = \sup_{T\in\mathbb{R}^{2\times n}_{++}} R_n(\mathbb{P}, T) \tag{5}$$

It could happen that for some $\mathbb{P}$ the ratio $R_n(\mathbb{P}, T)$ is unbounded in $T$. We do not consider these cases as we know that $R_n \leq 1.5861$ (see Section 2.2). For ease of

presentation, we work on $\overline{\mathbb{R}}_+ = \mathbb{R}_+ \cup \{\infty\}$ so that the supremum $\sup_{T \in \mathbb{R}_{++}^{2\times n}} R_n(\mathbb{P}, T)$ is always defined.

When a tie $\frac{T_{1j}}{T_{2j}} = z_j$ occurs for some $j \in [n]$, algorithms from the family $\mathcal{A}^{\mathcal{P}_n}$ send task $j$ to the second machine. In general, an algorithm in $\mathcal{C}$ could send the task to the first or the second machine. Next, we show that this behavior at the ties does not affect the worst-case performance:

**Theorem 2** *For a given number of tasks n, let $\mathbb{P} \in \mathcal{P}_n$ and define*

$$\mathcal{T} := \left\{ T \in \mathbb{R}_{++}^{2\times n} : \mathbf{P}_{z\sim\mathbb{P}} \left[ z_j = T_{1j}/T_{2j} \right] = 0 \text{ for all } j \in [n] \right\}. \tag{6}$$

*Let $R_n(\mathbb{P})$ be defined as in (5), then*

$$R_n(\mathbb{P}) = \sup_{T \in \mathcal{T}} R_n(\mathbb{P}, T).$$

Theorem 2 is proven in Section 7.1 and has the following implication:

**Corollary 1** *The best approximation ratio over all randomized MIS algorithms is the best approximation ratio over all algorithms in $\mathcal{A}^{\mathcal{P}_n}$.*

By Corollary 1, the best approximation ratio over all randomized MIS algorithms is

$$R_n = \inf_{\mathbb{P} \in \mathcal{P}_n} R_n(\mathbb{P}). \tag{7}$$

Later in the paper we show that $R_n(\mathbb{P})$ is invariant under permutations of the tasks for every $\mathbb{P} \in \mathcal{P}_n$ (see Theorem 3). Therefore, to compute $R_n$ using (7), we can restrict the optimization to the distributions $\mathbb{P} \in \mathcal{P}_n$ invariant under permutations of the random variables corresponding to the thresholds $(z_1, \ldots, z_n)$ (see Theorem 4). For such distributions, $R_n(\mathbb{P})$ is determined by the worst-case performance of $\mathcal{A}^{\mathbb{P}}$ on each pair of the tasks. See Section 3.2, and Proposition 2 in particular, for more detail. This means that for every $T \in \mathbb{R}_{++}^{2\times n}$, one can find the expected approximation ratio of $\mathcal{A}^{\mathbb{P}}$ by applying to all pairs of tasks the algorithm $\mathcal{A}^{\mathbb{P}_2}$, where $\mathbb{P}_2$ is the bivariate marginal distribution of $\mathbb{P}$ (by invariance, this distribution is the same for all pairs of thresholds). We use this property of family $\mathcal{A}^{\mathcal{P}_n}$ to construct problem (13) for $R_n$, which is one of the main results of this paper.

### 2.2 Connection to the Current Knowledge on Monotone Algorithms

The best approximation ratio for all monotone task allocation algorithms is unknown. For deterministic algorithms with $n$ tasks and $m$ machines, when $n$ and $m$ tend to infinity, the ratio lies in the interval $[1+\phi,\ m]$, where $\phi$ is the golden ratio. The upper bound is due to Nisan and Ronen [23], and the lower bound is due to Koutsoupias and Vidali [14]. To compute this lower bound, the authors use a matrix of processing times where the numbers of rows and columns tend to infinity. If $n$ or $m$ is finite, the lower bound may be different. Koutsoupias and Vidali [14] present bounds for several

finite time matrices as well. For randomized algorithms, the best approximation ratio lies in the interval $[2 - \frac{1}{m}, 0.83685m]$. The lower bound is due to Mu'alem and Schapira [21] who use Yao's minimax principle (for some details on this principle see, e.g., Motwani and Raghavan [20]). The upper bound is due to Lu and Yu [18]. The gap between the bounds grows with $m$, and the case with the smallest number of machines, $m = 2$, has gained much attention in the literature.

For $m = 2$, Nisan and Ronen [23] have shown that the best approximation ratio of deterministic monotone algorithms is equal to 2, for any finite $n$. The ratio for randomized monotone algorithms lies in the interval [1.5, 1.5861]. Chen et al. [3] compute the upper bound using an algorithm from the family $\mathcal{A}^{\mathcal{P}_n}$ with independently distributed thresholds. The lower bound is the earlier mentioned bound by Mu'alem and Schapira [21]. There exist tighter lower bounds for certain cases. Lu [17] shows that algorithms from the family $\mathcal{A}^{\mathcal{P}_n}$ (and thus, by Corollary 1, all randomized MIS algorithms) cannot achieve a ratio better than $\frac{25}{16}$ $(= 1.5625)$ for sufficiently large $n$. Chen et al. [3] prove that an algorithm $\mathcal{A}^{\mathbb{P}}$ cannot do better than 1.5852 when $\mathbb{P}$ is a product measure, i.e., when the thresholds are independent random variables.

The cases with $m = 2$ and small $n > 2$ are not well studied. Chen et al. [3] theoretically establish the upper bound 1.5861 for any finite $n$. They also present numerical computations of upper bounds smaller than 1.5861 for some $n$. Finding these smaller bounds requires solving non-convex optimization problems. However, the numerical method used by Chen et al. [3] does not guarantee global optimality.

The case with $m = 2$ and $n = 2$ is the simplest one, but even for this case the best approximation ratio is unknown. The ratio for algorithms from the family $\mathcal{A}^{\mathcal{P}_2}$ lies in the interval [1.505949, 1.5068]. The upper bound is due to Chen et al. [3], the lower bound is computed by Lu [17] using Yao's minimax principle. Notice that Lu [17] states that the lower bound is 1.506, but we repeated the calculations from this paper and obtained the number 1.505949. Thus, when reporting results, we use this number as the currently best lower bound. We improve this bound and show that $|R_2 - 1.505996| < 10^{-6}$, in particular, $R_2 < 1.506$.

From the description above, one can see that the existing bounds for truthful scheduling on unrelated machines are obtained using ad hoc procedures or (for the lower bounds) Yao's minimax principle. This paper develops a unified approach to construct upper and lower bounds on $R_n$ for any fixed $n$ and provides an alternative to Yao's minimax principle for the construction of lower bounds. As a result, we improve the bounds for truthful scheduling for $m = 2$ and $n \in \{2, 3, 4\}$.

Next, we compare our approach to the existing methods for upper bounds in [2, 3, 7, 15] that use optimization. Our method for upper bounds generalizes the approach by Chen et al. [3]. The generalization considers a broader class of algorithms and provides stronger upper bounds for $n \leq 4$. Our approach is fundamentally different from the methods in [2, 7, 15]. To begin with, our method is suitable for the minimum makespan problem on unrelated machines, while the methods in [2, 7, 15] are not guaranteed to work for this problem. Next, [2, 7, 15] use LP relaxations of the corresponding integer programs while we use the tools from continuous optimization to obtain possibly non-linear, but tractable approximations. Moreover, [2, 7, 15]

consider truthful in expectation mechanisms only while we work with universally truthful ones.

We obtain new bounds only for small $n \leq 4$ because of the growing size of the lower bound optimization problems. One can try to solve these problems in a more computationally efficient way using, for instance, the column generation technique (see, e.g., Gilmore and Gomory [10]). As a result, one could expect to obtain new bounds for $n > 4$ with our approach since the solution to the upper bound problem is a relatively simple construction based on the solution to the lower bound problem, as described in Section 4. However, improving the efficiency of the lower bound computation is out of the scope of the current paper.

## 3 Using the Symmetry of the Problem to Obtain a New Formulation for the Best Approximation Ratio

In this section we exploit the fact that problem (7) is invariant under permuting the tasks and the machines to simplify formulation (7) and obtain formulation (13) in Section 3.2.

### 3.1 Using the Symmetry of the Problem

For a given number of tasks $n$, let $S_n$ be the symmetric, or permutation, group on $n$ elements. Given a vector $\mathbf{z} \in \mathbb{R}^n$ and $\pi \in S_n$, the corresponding permutation of the elements of $\mathbf{z}$ by $\pi$ is denoted by $\mathbf{z}\pi$. The group $S_n$ corresponds to column permutations in a given time matrix $T$. We define another group, $S_{inv}$, which corresponds to row permutations in $T$. $S_{inv}$ consists of the identity action ${}^{id}$. and the action ${}^{inv}$. which takes element-wise reciprocals of any vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n_{++}$:

$$ {}^{id}\mathbf{x} = \mathbf{x}, \quad {}^{inv}\mathbf{x} = \left(\frac{1}{x_1}, \frac{1}{x_2}, \ldots, \frac{1}{x_n}\right). $$

Now, we define the action of $S_{inv} \times S_n$ on $\mathcal{P}_n$. Given $\mathbb{P} \in \mathcal{P}_n$, $\gamma \in S_{inv}$, $\pi \in S_n$ and a random variable $\mathbf{z} \sim \mathbb{P}$, we consider the transformation $\mathbf{z} \to {}^{\gamma}\mathbf{z}\pi$. We define ${}^{\gamma}\mathbb{P}\pi \in \mathcal{P}_n$ as the distribution of ${}^{\gamma}\mathbf{z}\pi$. Next, we prove that problem (7) is convex and invariant under the action of $S_{inv} \times S_n$ on $\mathbb{P}$. As a result, to find the infimum in (7), it is enough to optimize over the distributions $\mathbb{P}$ invariant under the action of $S_{inv} \times S_n$. This approach is regularly used in convex programming, see Dobre and Vera [5], Gatermann and Parrilo [9] or de Klerk et al. [13].

Given distributions $\mathbb{P}_1, \ldots, \mathbb{P}_k \in \mathcal{P}_n$, and weights $\alpha_i \geq 0$ for all $i \in [k]$ such that $\sum_{i=1}^k \alpha_i = 1$, we define the convex combination $\sum_{i=1}^k \alpha_i \mathbb{P}_i \in \mathcal{P}_n$ as the distribution where we draw from $P_i$, $i \in [k]$ with probability $\alpha_i$. The construction of $\sum_{i=1}^k \alpha_i \mathbb{P}_i$ and definitions (3), (4) imply that

$$ R_n\left(\sum_{i=1}^k \alpha_i \mathbb{P}_i, T\right) = \sum_{i=1}^k \alpha_i R_n(\mathbb{P}_i, T). $$

Therefore, using (5), we have

$$R_n\left(\sum_{i=1}^{k}\alpha_i\mathbb{P}_i\right) \le \sum_{i=1}^{k}\alpha_i R_n(\mathbb{P}_i), \tag{8}$$

that is, $R_n(\mathbb{P})$ is convex in $\mathbb{P}$. Now, we show the invariance of $R_n(\mathbb{P})$ under the action of $S_{inv}\times S_n$.

**Theorem 3** *For any given number of tasks n, $\mathbb{P}\in\mathcal{P}_n$, $\gamma\in S_{inv}$ and $\pi\in S_n$,*

$$R_n(\mathbb{P}) = R_n({}^{\gamma}\mathbb{P}\pi).$$

The proof of Theorem 3 is presented in Section 7.2.

**Theorem 4** *For any given number of tasks n,*

$$R_n = \inf_{\mathbb{P}\in\mathcal{P}_n} R_n(\mathbb{P}) \text{ such that } \mathbb{P} \text{ is invariant under the action of } S_{inv}\times S_n. \tag{9}$$

*Proof* As problem (9) has a smaller feasibility set than problem (7), the optimal value of problem (9) is not smaller than $R_n$. To prove the opposite inequality, we show that for any distribution $\mathbb{P}\in\mathcal{P}_n$ there is a distribution $\mathbb{Q}\in\mathcal{P}_n$ invariant under the action of $S_{inv}\times S_n$ such that $R_n(\mathbb{Q})\le R_n(\mathbb{P})$. Given $\mathbb{P}\in\mathcal{P}_n$, take $\alpha_i=\frac{1}{2(n!)}$ for $i\in[2(n!)]$ and consider the convex combination

$$\mathbb{Q} := \frac{1}{2(n!)}\sum_{(\gamma,\pi)\in S_{inv}\times S_n} {}^{\gamma}\mathbb{P}\pi.$$

By construction, $\mathbb{Q}$ has the required invariance property and

$$R_n(\mathbb{Q}) \overset{(8)}{\le} \tfrac{1}{2(n!)}\sum_{(\gamma,\pi)\in S_{inv}\times S_n} R_n({}^{\gamma}\mathbb{P}\pi) \overset{\text{Theorem 3}}{=} \tfrac{1}{2(n!)}\sum_{(\gamma,\pi)\in S_{inv}\times S_n} R_n(\mathbb{P}) = R_n(\mathbb{P}).$$

□

### 3.2 New Formulation for the Best Approximation Ratio

From Theorem 4, problem (7) is invariant under permuting the tasks and the machines. In the sequel we exploit the invariance under permuting the tasks only. First, this simplifies the presentation. Second, in our numerical computations using the invariance under the two types of permutations produced the same bounds as using invariance under task permutations only.

Let $\mathcal{C}_n\subset\mathcal{P}_n$ be the family of probability measures invariant under the actions of $S_n$:

$$\mathcal{C}_n = \{\mathbb{P}\in\mathcal{P}_n \mid \mathbb{P}=\mathbb{P}\pi, \text{ for all } \pi\in S_n\}. \tag{10}$$

In the rest of the paper we restrict the optimization to the distributions from $\mathcal{C}_n$.

**Corollary 2** *For any given number of tasks n,*

$$R_n = \inf_{\mathbb{P}\in\mathcal{C}_n} R_n(\mathbb{P}). \tag{11}$$

*Proof* The Corollary follows from Theorem 4 and

$$\{\mathbb{P} \in \mathcal{P}_n \mid \mathbb{P} = {}^{\gamma}\mathbb{P}\pi, \ \text{ for all } \ (\gamma, \pi) \in S_{inv} \times S_n\} \subset \mathcal{C}_n \subset \mathbb{P}_n. \qquad \square$$

Proposition 1 next is straightforward but crucial for our analysis.

**Proposition 1** *Let $\mathbb{P} \in \mathcal{C}_n$. Then $\mathbb{P}$ has a cumulative distribution function (CDF) invariant under permutations of the variables. Moreover, for $0 < k < n$, all $k$-variate marginal distributions are identical. In particular, $\mathbb{P}$ is a joint distribution of $n$ identically distributed random variables.*

By Proposition 1, if $\mathbb{P} \in \mathcal{C}_n$, then all univariate marginal distributions of $\mathbb{P}$ are identical and all bivariate marginal distributions of $\mathbb{P}$ are identical. Denote the corresponding univariate and bivariate CDFs by $F_{\mathbb{P}}$ and $H_{\mathbb{P}}$, respectively, and define

$$\begin{aligned}\phi^{\mathbb{P}}(x, y) \ = \ & 1 + y - \min\{1, \ 1 - 1/x + y\}\, F_{\mathbb{P}}(x) - y F_{\mathbb{P}}(y) \\ & + \min\{1 + 1/x, 1 + y\}\, H_{\mathbb{P}}(x, y)\end{aligned} \tag{12}$$

First, we present a result by Chen et al. [3], which follows from Lu and Yu [19]

**Proposition 2** (Chen et al. [3]) *For any given number of tasks n, $\mathbb{P} \in \mathcal{C}_n$, and $T \in \mathbb{R}_{++}^{2\times n}$,*

$$R_n(\mathbb{P}, T) \le \max_{j,k\in[n]} \phi^{\mathbb{P}}\left(\frac{T_{1j}}{T_{2j}}, \frac{T_{1k}}{T_{2k}}\right).$$

Notice that this upper bound is defined by only two tasks out of $n$. Using Proposition 2, we obtain the following formulation for $R_n(\mathbb{P})$:

**Theorem 5** *For any given number of tasks n, and $\mathbb{P} \in \mathcal{C}_n$,*

$$R_n(\mathbb{P}) = \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}}(x, y).$$

The proof of Theorem 5 is provided in Section 7.3.

*Remark 1* Theorem 5 implies that the worst-case approximation ratio for $n$ tasks and $\mathbb{P} \in \mathcal{P}_n$ is the worst-case approximation ratio for two tasks and the bivariate marginal distribution of $\mathbb{P}$.

The next corollary is the main result of this section, and we use it throughout the rest of the paper.

**Corollary 3** *For any given number of tasks n,*

$$R_n = \inf_{\mathbb{P}\in\mathcal{C}_n} \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}}(x, y). \tag{13}$$

*Proof* The result follows from Corollary 2 and Theorem 5. □

**Corollary 4** $R_{n+1} \geq R_n$ *for all* $n \geq 2$.

*Proof* The result follows from Corollary 3. □

## 4 Upper and Lower Bounds on the Best Approximation Ratio

To find $R_n$ using (13), one needs to optimize over a family of distributions, which is computationally intractable. Therefore we construct upper and lower bounds on the optimal value of the problem. The idea is to restrict the attention to some subset of feasible distributions or some subset of $\mathbb{R}^2_{++}$, over which it is easier to solve problem (13).

1. For the lower bound, we take a finite set $\mathcal{S} \subset \mathbb{R}_{++}$ and find the supremum in (13) for $x, y \in \mathcal{S}$ only. A conventional approach to lower bounds is to propose several good-guess time matrices $T$, use these matrices to build a randomized instance of the minimum makespan problem, and apply Yao's minimax principle. Our approach is different as we evaluate randomized algorithms on deterministic instances.
2. For the upper bound, we find a good-guess distribution $\mathbb{P}$ and solve the inner maximization problem for this distribution. The distribution is built using the solution to the lower bound problem for $n \in \{2, 3, 4\}$. For $n = 2$ we propose a more efficient approach in Section 5.

### 4.1 Characterizing CDFs

To implement the ideas above, we have to optimize over distributions. For this purpose we represent a distribution via its CDF. To characterize CDFs, we follow Nelsen [22]. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ such that $x_i \leq y_i$ for all $i \in [n]$, we define the $n$-box $\mathcal{B}_{xy} := [x_1, y_1] \times [x_2, y_2] \times \cdots \times [x_n, y_n]$. The set of vertices of $\mathcal{B}_{xy}$ is $V_{xy} = \{x_1, y_1\} \times \{x_2, y_2\} \times \cdots \times \{x_n, y_n\}$. The sign of vertex $\mathbf{b} \in V_{xy}$ is defined by

$$\operatorname{sgn}(\mathbf{b}) := \begin{cases} 1, & \text{if } b_i = x_i \text{ for an even number of entries } i \\ -1, & \text{if } b_i = x_i \text{ for an odd number of entries } i. \end{cases}$$

Given a set $D \subseteq \mathbb{R}$, define $\overline{D} := (D \cup \{0\} \cup \{\infty\})$. A function $G : \overline{\mathbb{R}}^n \to \mathbb{R}$ is called $n$-increasing on $\overline{D}^n$ when

$$\sum_{\mathbf{b}\in V_{xy}} \operatorname{sgn}(\mathbf{b}) G(\mathbf{b}) \geq 0, \text{ for all } \mathbf{x} \leq \mathbf{y}, \ \mathbf{x}, \mathbf{y} \in \overline{D}^n \tag{14}$$

*Remark 2* (Chapter 2.1 in Nelsen [22]) For $n > 1$, the fact that $G$ is $n$-increasing does not necessarily imply that $G$ is non-decreasing in each argument, and the other way round.

The following family of functions captures the concept of CDF.

**Definition 1** Let $\mathcal{S} \subseteq \mathbb{R}_{++}$. $\mathcal{G}_n(\mathcal{S})$ is the family of functions $G : \overline{\mathcal{S}}^n \to [0, 1]$ satisfying the conditions below.

1. $G$ is right continuous on $\overline{\mathcal{S}}^n$
2. $G$ is $n$-increasing on $\overline{\mathcal{S}}^n$
3. $G(\mathbf{z}) = 0$ for all $\mathbf{z}$ in $\overline{\mathcal{S}}^n$ such that at least one of $z_i = 0$
4. $G(\infty, \ldots, \infty) = 1$

**Proposition 3** (Definition 2.10.8. in Nelsen [22]) *A function $G : \overline{\mathbb{R}}^n_{++} \to [0, 1]$ is a CDF of some $\mathbb{P} \in \mathcal{P}_n$ if and only if $G \in \mathcal{G}_n(\mathbb{R}_{++})$.*

### 4.2 Formulation of Upper and Lower Bounds

To construct the upper bound, we restrict the inner maximization in problem (13) to a subset of $\mathbb{R}_{++}$. We do this using the next lemma.

**Lemma 1** *Let $\mathcal{S} \subseteq \mathbb{R}_{++}$ be a **finite** set. Then $g \in \mathcal{G}_n(\mathcal{S})$ if and only if there exists $G \in \mathcal{G}_n(\mathbb{R}_{++})$ such that $g = G|_{\overline{\mathcal{S}}^n}$. That is, $g$ is a restriction of $G$ to $\overline{\mathcal{S}}^n$.*

*Proof* If there is $G \in \mathcal{G}_n(\mathbb{R}_{++})$ such that $g = G|_{\overline{\mathcal{S}}^n}$, then $g \in \mathcal{G}_n(\mathcal{S})$ by definition of $\mathcal{G}_n(\mathbb{R}_{++})$. On the other hand, let $g \in \mathcal{G}_n(\mathcal{S})$ and consider a number $a > \max\{s : s \in \mathcal{S}\}$. Let $\mathcal{S}_a = \mathcal{S} \cup \{a\}$ and define a new function $\hat{g} : \overline{\mathcal{S}_a}^n \to [0, 1]$ such that $\hat{g}(\mathbf{z}) = g(\mathbf{z})$ for $\mathbf{z} \in \overline{\mathcal{S}}^n$. For $\mathbf{z} \notin \overline{\mathcal{S}}^n$, construct a new vector $\mathbf{y}$ by replacing all occurrences of $a$ in $\mathbf{z}$ with $\infty$ and define $\hat{g}(\mathbf{z}) = g(\mathbf{y})$. Consider the following piecewise constant function:

$$G(z_1, \ldots, z_n) := \hat{g}\left(\max_{x \in \overline{\mathcal{S}_a}}\{x : x \le z_1\}, \ldots, \max_{x \in \overline{\mathcal{S}_a}}\{x : x \le z_n\}\right). \quad (15)$$

It is straightforward to show that $G \in \mathcal{G}_n(\mathbb{R}_{++})$ and $g = G|_{\overline{\mathcal{S}}^n}$. See Fig. 1 for an illustration of the case $n = 1$. □

*Remark 3* The choice of $a$ in the proof of Lemma 1 is free and might influence our upper bound computations in Section 4.3.

For a finite $\mathcal{S} \subset \mathbb{R}_{++}$ and $g \in \mathcal{G}_n(\mathcal{S})$, we define the restriction of the objective in problem (13) to $\mathcal{S}$:

$$\begin{aligned} \phi^g(x, y) = \; & 1 + y - \min\{1,\ 1 - 1/x + y\}\, g(x, \infty, \ldots, \infty) \\ & - y g(y, \infty, \ldots, \infty) \\ & + \min\{1 + 1/x,\ 1 + y\}\, g(x, y, \infty, \ldots, \infty) \quad \text{for all } x, y \in \overline{\mathcal{S}}. \end{aligned} \quad (16)$$

By Lemma 1, $\phi^g = \phi^{\mathbb{P}}|_{\overline{\mathcal{S}}^2}$ for some $\mathbb{P} \in \mathcal{P}_n$.

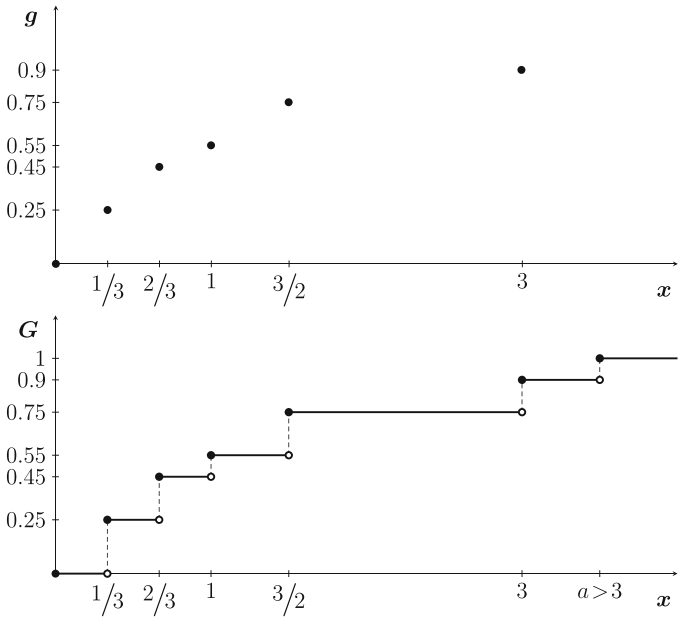


**Fig. 1** $\mathcal{S} = \{\frac{1}{3}, \frac{2}{3}, 1, \frac{3}{2}, 3\}$, $n = 1$. The left plot: a function $g \in \mathcal{G}_n(\mathcal{S})$. The right plot: a function $G \in \mathcal{G}_n(\mathbb{R}_{++})$. Notice that $G$ is a CDF and $g$ is the restriction of $G$ to $\overline{\mathcal{S}}^n$

**Theorem 6** *Given a number of tasks n, for any $\mathbb{P} \in \mathcal{C}_n$ and finite $\mathcal{S} \subset \mathbb{R}_{++}$, we have*

$$R_n(\mathbb{P}) \geq R_n \geq R_n(\mathcal{S}) := \inf_{g \in \mathcal{G}_n(\mathcal{S})} \sup_{x,y \in \mathcal{S}} \left\{ \phi^g(x, y) : g(\mathbf{z}) = g(\mathbf{z}\pi) \text{ for all } \pi \in S_n, \mathbf{z} \in \overline{\mathcal{S}}^n \right\} \tag{17}$$

*Proof* The first inequality follows immediately from Corollary 3. Now, we prove the second inequality. Every $\mathbb{P} \in \mathcal{C}_n$ has a CDF $G_{\mathbb{P}} \in \mathcal{G}_n(\mathbb{R}_{++})$ invariant under permutations of the variables by Proposition 1. At the same time, every such invariant $G \in \mathcal{G}_n(\mathbb{R}_{++})$ corresponds to some $\mathbb{P}_G \in \mathcal{C}_n$. Combining this with Corollary 3, we obtain

$$\begin{aligned} R_n &= \inf_{G \in \mathcal{G}_n(\mathbb{R}_{++})} \sup_{x,y \in \mathbb{R}_{++}} \{\phi^{\mathbb{P}_G}(x, y) : G(\mathbf{z}) = G(\mathbf{z}\pi) \text{ for all } \pi \in S_n,\ \mathbf{z} \in \overline{\mathbb{R}}^n_{++}\} \\ &\geq \inf_{G \in \mathcal{G}_n(\mathbb{R}_{++})} \sup_{x,y \in \mathcal{S}} \{\phi^{\mathbb{P}_G}(x, y) : G(\mathbf{z}) = G(\mathbf{z}\pi) \text{ for all } \pi \in S_n,\ \mathbf{z} \in \overline{\mathbb{R}}^n_{++}\} \\ &= \inf_{g \in \mathcal{G}_n(\mathcal{S})} \sup_{x,y \in \mathcal{S}} \{\phi^g(x, y) : g(\mathbf{z}) = g(\mathbf{z}\pi) \text{ for all } \pi \in S_n,\ \mathbf{z} \in \overline{\mathcal{S}}^n\}. \end{aligned}$$

The last equality holds by Lemma 1. Notice that if $G \in \mathcal{G}_n(\mathbb{R}_{++})$ is invariant under permutations of the variables, then so is the $g := G|_{\overline{\mathcal{S}}^n}$. On the other hand, if $g \in \mathcal{G}_n(\mathcal{S})$ is invariant under permutations of the variables, then so is the $G$ defined in (15). □

### 4.3 Implementation and Numerical Results

Let $S \subset \mathbb{R}_{++}$. To compute the lower bound $R_n(\mathcal{S})$ from (17), we use the epigraph form of the optimization problem for $R_n(\mathcal{S})$:

$$\begin{aligned} R_n(\mathcal{S}) = \inf_{g \in \mathcal{G}_n(\mathcal{S}),\ t \in \mathbb{R}} \quad & t \\ \text{s.t.} \quad & \phi^g(x, y) \le t && \text{for all } x, y \in \mathcal{S} \\ & g(\mathbf{z}) = g(\mathbf{z}\pi) && \text{for all } \pi \in S_n,\ \mathbf{z} \in \overline{\mathcal{S}}^n \end{aligned} \tag{18}$$

The optimization variable in the problem above is $g$. This variable is a vector in $\mathbb{R}^{(|\mathcal{S}|+2)^n}$ which represents a function $g \in \mathcal{G}_n(\mathcal{S})$. We slightly abuse the notation and do not use a bold symbol for $g$ to underline that $g$ corresponds to a function with a finite support. Family $\mathcal{G}_n(\mathcal{S})$ is an infinite family of functions $g$, and each of these functions is defined on a finite set $\overline{\mathcal{S}}^n$ with cardinality $(|\mathcal{S}| + 2)^n$. For the purpose of optimization, this means that we consider all vectors $g \in \mathbb{R}^{(|\mathcal{S}|+2)^n}$ which satisfy the four conditions in Definition 1 and the invariance property. All mentioned conditions are linear, and there are finitely many of them. Therefore the optimization over the infinite family of functions $\mathcal{G}_n(\mathcal{S})$ can be written as a finite LP. We use the invariance of $g$ (the second constraint) and Conditions 3-4 in Definition 1 to reduce the number of variables in problem (18) (the size of $g$ as a vector). To ensure that $g$ corresponds to an $n$-increasing function as specified in (14), it is enough to consider $\mathbf{x}, \mathbf{y} \in \overline{\mathcal{S}}^n$ such that $x_i, y_i$ are consecutive points in $\mathcal{S}$ for all $i \in [n]$. This reduces the number of constraints in problem (18).

To compute the upper bound $R_n(\mathbb{P})$ using formulation (17), we first construct a good-guess distribution $\mathbb{P}$. Given a set $\mathcal{S}$ and the solution $g$ to the lower bound problem (17) on $\mathcal{S}$, we use the distribution $\mathbb{P}_g$ which corresponds to the CDF (15) based on $g$. To construct this CDF, we choose a number $a > \max\{s : s \in \mathcal{S}\}$, as explained in the proof of Lemma 1. In the rest of this section we work with $\mathcal{S}_a = \mathcal{S} \cup \{a\}$. To solve (17) for $\mathbb{P}_g$, we define the following set of intervals:

$$\mathcal{I}_{\mathcal{S}} = \left\{I_1, \dots, I_{|\mathcal{S}|+2}\right\} = \left\{[0, s_1), \dots, [s_{|\mathcal{S}|}, a), [a, \infty)\right\},$$

This set of intervals covers $\mathbb{R}_+$, therefore by (17)

$$R_n(\mathbb{P}_g) = \sup_{x,y \in \mathbb{R}_{++}} \phi^{\mathbb{P}_g}(x, y) = \max_{I_i, I_j \in \mathcal{I}_{\mathcal{S}}} \left\{ \sup_{x \in I_i, y \in I_j} \phi^{\mathbb{P}_g}(x, y) \right\}. \tag{19}$$

We solve the inner maximization problem in (19) for each pair $i, j \in [|\mathcal{S}| + 2]$. The expression for $\phi^{\mathbb{P}_g}$ (12) for the case $xy \ge 1$ is different from the case $xy < 1$. To simplify the computations when the line $xy = 1$ crosses the rectangle $I_i \times I_j$, we restrict our attention to $\mathcal{S}$ of a particular type. Consider a collection of $k - 1$ positive real numbers $r_1 < r_2 < \cdots < r_{k-1} < 1$, let

$$\mathcal{S}_k = \{s_1,\ s_2, \dots, s_{2k-1}\} = \left\{r_1,\ r_2, \dots, r_{k-1},\ 1,\ \tfrac{1}{r_{k-1}}, \dots, \tfrac{1}{r_2},\ \tfrac{1}{r_1}\right\}. \tag{20}$$

For any $a > \frac{1}{r_1}$, $\mathcal{S}_{k,a} = \mathcal{S}_k \cup \{a\}$ subdivides $\mathbb{R}_+$ into $2k+1$ intervals $\mathcal{I}_{\mathcal{S}_k}$. First, consider a pair of intervals $I_i, I_j \in \mathcal{I}_{\mathcal{S}_k}$ such that $i \notin \{1, 2k+1\}$ and $j \notin \{1, 2k+1\}$ (i.e., neither of them are the first or the last interval). Due to the choice of $\mathcal{S}_k$, the line

$xy = 1$ crosses the rectangle $I_i \times I_j$ if and only if $i + j = 2k+1$. Denote the bivariate marginal CDF and the univariate marginal CDF of $\mathbb{P}_g$ by $H_g$ and $F_g$, respectively. Then

$$\phi^{\mathbb{P}_g}(x, y) = \begin{cases} 1+y-F_g(x)-yF_g(y)+(1+1/x)\,H_g(x, y) & i+j \geq 2k+1,\ xy \geq 1 \\ 1+y\left(1-F_g(x)-F_g(y)+H_g(x, y)\right) & \\ -(1-1/x)\,F_g(x)+H_g(x, y) & i+j \leq 2k+1,\ xy < 1. \end{cases} \tag{21}$$

We construct the CDF of $\mathbb{P}_g$ using (15), therefore

$$\begin{aligned} H_g(x, y) &= g(s_{i-1}, s_{j-1}, \infty, \ldots, \infty) \ \text{ for all } (x, y) \in I_i \times I_j \\ F_g(x) &= g(s_{i-1}, \infty, \ldots, \infty) \ \text{ for all } x \in I_i. \end{aligned} \tag{22}$$

That is, the marginal CDFs are constant on $I_i \times I_j$ and $I_i$, respectively. As the range of a CDF is [0, 1], we conclude that for $x \in I_i$, $y \in I_j$, $\phi^{\mathbb{P}_g}(x, y)$ is non-increasing in $x$ and non-decreasing in $y$. The latter holds since for any $\mathbb{P} \in \mathcal{P}_2$ invariant under $S_2$ and for any $x, y \in \mathbb{R}$,

$$\begin{aligned} H_{\mathbb{P}}(x, y) &= F_{\mathbb{P}}(x) + F_{\mathbb{P}}(y) - 1 + \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}[z_1 > x, z_2 > y] \\ &\geq \max\{0, F_{\mathbb{P}}(x) + F_{\mathbb{P}}(y) - 1\}. \end{aligned} \tag{23}$$

Hence the optimal value of the inner maximization problem in (19) can be obtained by first substituting the CDFs (22) into the function (21) and then substituting $x = s_{i-1}$, $y = s_j$. Note that this optimum is not attained. For the case $i + j = 2k + 1$, i.e., when the line $xy = 1$ crosses the rectangle $I_i \times I_j$, the result holds due to the choice of $\mathcal{S}_k$.

When $i \in \{1, 2k+1\}$ or $j \in \{1, 2k+1\}$, the function $\phi^{\mathbb{P}_g}(x, y)$ simplifies, and we solve such cases separately. The solution approach resembles the one from the previous paragraph.

In numerical experiments we use uniform finite sets

$$\mathcal{S}_k^u = \left\{\tfrac{1}{k}, \tfrac{2}{k}, \ldots, \tfrac{k-1}{k},\ 1,\ \tfrac{k}{k-1}, \ldots, k\right\}. \tag{24}$$

Table 2 shows the best obtained bounds and the $k$ we use to compute these bounds.

We round the lower bounds $R_n(\mathcal{S}_k^u)$ down and the upper bounds $R_n(\mathbb{P}_g)$ up. We verify all upper bounds with exact arithmetics using the MATLAB symbolic package and the following procedure. First, we obtain the optimal solution $g$ to problem (18) and round the elements of the set $\mathcal{S}_k^u$ and the number $a$ to the $8^{th}$ digit. Next, we

**Table 2** Bounds for the case of two machines and two, three, and four tasks based on Theorem 6

| $n$ | Lower bound | | Upper bound | | $k$ |
|---|---|---|---|---|---|
| | Existing | New | Existing | New | |
| 2 | 1.505949 [17] | 1.505980 | 1.5068 [3] | 1.5093 | 250 |
| 3 | 1.505949 [17] | 1.5076 | 1.5861 [3] | 1.5238 | 50 |
| 4 | 1.505949 [17] | 1.5195 | 1.5861 [3] | 1.5628 | 20 |

transform the rounded values into rational numbers and compute $R_n(\mathbb{P}_g)$ as a rational number. By Lemma 1 and Theorem 6, the rounded $g$ provides the algorithm $\mathcal{A}^{\mathbb{P}_g}$ with the worst-case approximation ratio $R_n(\mathbb{P}_g)$.

The upper bound for $n = 2$ in Table 2 is worse than the best existing upper bound. We improve our result and obtain a new best exiting upper bound for $n = 2$ in the next section.

## 5 More Precise Bounds for Two Tasks

In this section we analyze the case with $n = 2$ tasks and $m = 2$ machines in more detail. Now, to obtain an upper bound, we do not simply use some good-guess distribution as we did before, but we optimize over a subset of $\mathcal{C}_n$ (10). Moreover, as a side result of this optimization, we obtain a non-uniform set $\mathcal{S}_k$ which produces a better lower bound than the one from Table 2 in the previous section.

Problem (13) simplifies for $n = 2$. Given $F \in \mathcal{G}_1(\mathbb{R}_{++})$, define

$$H(x, y) := \max\{0,\ F(x) + F(y) - 1\}. \tag{25}$$

$H$ is a copula, i.e., there is $\mathbb{P}_{H,F} \in \mathcal{P}_2$ for which $H$ is its CDF and $F$ is its marginal CDF. See Nelsen [22] for the detailed description of copulas and their properties. Moreover, by construction $\mathbb{P}_{H,F} \in \mathcal{C}_2$. Therefore we can rewrite problem (13) using univariate marginal CDF's.

**Theorem 7** *Consider family $\mathcal{G}_1(\mathbb{R}_{++})$ from Definition 1.*

$$\begin{aligned} R_2 = \inf_{F\in\mathcal{G}_1(\mathbb{R}_{++})} \sup_{x,y\in\mathbb{R}_{++}} & \ 1 + y - \min\{1,\ 1 - 1/x + y\}\, F(x) - yF(y) \\ & + \min\{1 + 1/x,\ 1 + y\} \max\{0,\ F(x) + F(y) - 1\}. \end{aligned} \tag{26}$$

*Proof* Given $F \in \mathcal{G}_1(\mathbb{R}_{++})$, let

$$\begin{aligned} \phi^F(x, y) = & \ 1 + y - \min\{1,\ 1 - 1/x + y\}\, F(x) - yF(y) \\ & + \min\{1 + 1/x,\ 1 + y\} \max\{0,\ F(x) + F(y) - 1\}. \end{aligned}$$

Consider any $\mathbb{P} \in \mathcal{C}_2$ with the univariate CDF $F_{\mathbb{P}}$. Define $\phi^{\mathbb{P}}(x, y)$ as in (12). From (23) for all $x, y \in \mathbb{R}_{++}$,

$$\phi^{\mathbb{P}}(x, y) \geq \phi^{F_{\mathbb{P}}}(x, y).$$

Thus

$$\begin{aligned} R_2 \overset{(13)}{=} \inf_{\mathbb{P}\in\mathcal{C}_2} \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}}(x, y) &\geq \inf_{\mathbb{P}\in\mathcal{C}_2} \sup_{x,y\in\mathbb{R}_{++}} \phi^{F_{\mathbb{P}}}(x, y) \\ &\geq \inf_{F\in\mathcal{G}_1(\mathbb{R}_{++})} \sup_{x,y\in\mathbb{R}_{++}} \phi^{F}(x, y). \end{aligned}$$

On the other hand, for all $F \in \mathcal{G}_1(\mathbb{R}_{++})$ there is copula $H$ from (25) with the corresponding distribution $\mathbb{P}_{H,F} \in \mathcal{C}_2$. Hence

$$R_2 = \inf_{\mathbb{P}\in\mathcal{C}_2} \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}}(x,y) \le \inf_{F\in\mathcal{G}_1(\mathbb{R}_{++})} \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}_{H,F}}(x,y)$$
$$= \inf_{F\in\mathcal{G}_1(\mathbb{R}_{++})} \sup_{x,y\in\mathbb{R}_{++}} \phi^{F}(x,y). \qquad \square$$

*Remark 4* Nelsen [22] shows that for $n > 2$ the function

$$G(x_1,\dots,x_n) = \max\left\{0,\ \sum_{i=1}^{n} F(x_i) - n + 1\right\}$$

is not a CDF. We do not see other suitable $n$-variate CDFs which would have a bivariate margin $H$ from (25). As a result, the proof of Theorem 7 fails for $n > 2$.

### 5.1 New Upper Bound for Two Tasks

To compute a new upper bound on $R_2$, we restrict the set of functions in problem (26) to the family of piecewise rational univariate CDFs. We say that a function is piecewise rational if it can be written as a fraction where both the numerator and the denominator are polynomials. The domain of each CDF is subdivided into pieces by $\mathcal{S}_k$ defined in (20). Given $\mathcal{S}_k$, we introduce a collection of intervals:

$$\begin{aligned}\mathcal{I}_{\mathcal{S}_k} &= \{I_1, I_2, \dots, I_k, I_{k+1}, \dots, I_{2k-1}, I_{2k}\}\\ &= \left\{[0, r_1), [r_1, r_2), \dots, [r_{k-1}, 1), \left[1, 1/r_{k-1}\right), \dots, [1/r_2, 1/r_1), [1/r_1, +\infty)\right\}. \quad (27)\end{aligned}$$

*Remark 5* We build the intervals using the points from $\mathcal{S}_k$ only. This is different from Section 4.3 where we use an additional number $a > \max \mathcal{S}_k$ to construct the intervals.

Given $\mathcal{I}_{\mathcal{S}_k}$ and a family of continuous functions $\mathcal{F}$, we consider CDFs which "piecewisely" belong to $\mathcal{F}$.

**Definition 2** $\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)$ is a family of functions $F : \mathbb{R}_{++} \to [0,1]$ such that

$$F(x) = \begin{cases} f_1(1/x) & x \in I_1\\ f_2(1/x) & x \in I_2\\ \dots & \\ f_k(1/x) & x \in I_k\\ 1 - f_k(x) & x \in I_{k+1}\\ \dots & \\ 1 - f_2(x) & x \in I_{2k-1}\\ 1 - f_1(x) & x \in I_{2k}, \end{cases} \quad (28)$$

$f_1(x) = 0$, $f_i(x) \in \mathcal{F}$, $f_i(x)$ is non-decreasing, $f_k(1) \le 0.5$, $0 \le f_i(x_i) \le f_{i+1}(x_i)$ for all $i < k$. By construction, $F$ is a CDF and thus $\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k) \subset \mathcal{G}_1(\mathbb{R}_{++})$. Restricting the minimization in problem (26) to $\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)$ provides an upper bound on $R_2$. We use the symmetry of $F$ to simplify the expression for this bound.

**Proposition 4** *Define $\mathcal{S}_k$ as in (20) and consider family $\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)$ from Definition 2. $R_2$ is not larger than*

$$\begin{aligned} R_2\left(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)\right) &= \inf_{F\in\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)} \sup_{x,y\in\mathbb{R}_{++}} 1+y-\min\{1,\ 1-1/x+y\}\,F(x)-yF(y) \qquad (29)\\ &\quad + \min\{1+1/x,\ 1+y\}\max\{0,\ F(x)+F(y)-1\}\\ &= \inf_{F\in\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)} \sup_{x,y\in\mathbb{R}_{++},\ xy\geq 1} y-1/x+(1+1/x-y)\,F(y)+1/x\,F(x). \qquad (30)\end{aligned}$$

*Proof* Let $F\in\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)$. Problem (29) is a restriction of problem (26) to a smaller set of functions. Therefore the former problem defines an upper bound on $R_2$.

Next, we show the equality between (29) and (30). Denote by $\phi^F$ the objective of problem (29). We claim that for every $x, y\in\mathbb{R}_{++}$,

$$\phi^F(x,y)\leq \sup_{x,y\in\mathbb{R}_{++},\ xy\geq 1}\phi^F(x,y).$$

Let $\hat{x}>0, \hat{y}>0$. First, consider the case $\hat{x}\hat{y}\geq 1$. By construction of (28), $\hat{x}\hat{y}\geq 1$ implies $F(\hat{x})+F(\hat{y})\geq F(\hat{x})+F\left(1/\hat{x}\right)\geq 1$. Then

$$\begin{aligned}\phi^F(\hat{x},\hat{y}) &= 1+\hat{y}-F(\hat{x})-\hat{y}F(\hat{y})+\left(1+1/\hat{x}\right)\left(F(\hat{x})+F(\hat{y})-1\right)\\ &= \hat{y}-1/\hat{x}+\left(1+1/\hat{x}-\hat{y}\right)F(\hat{y})+1/\hat{x}F(\hat{x}). \qquad (31)\end{aligned}$$

Now, let $\hat{x}\hat{y}<1$. By construction of $\mathcal{I}_{\mathcal{S}_k}$ in (27), there are $I_i, I_j\in\mathcal{I}_{\mathcal{S}_k}$ such that $\hat{x}\in I_i,\ \hat{y}\in I_j$. The set $\mathcal{S}_k$ is finite, therefore there is a sequence $\{(x_t,y_t)\}_{t=1}^{\infty}$ such that for all $t$ the following holds: $x_t\in I_i,\ y_t\in I_j, x_t\to\hat{x}^+,\ y_t\to\hat{y}^+, x_ty_t<1$ and $x_t, y_t\notin\mathcal{S}_k$. For all $x\in\mathbb{R}_{++}\setminus\mathcal{S}_k$ we have $F(x)+F(1/x)=1$. Hence for all $t$

$$\begin{aligned}\phi^F(x_t,y_t) &= 1+y_t-(1-1/x_t+y_t)\,F(x_t)-y_tF(y_t)\\ &= y_t(1-F(y_t))+(1-1/x_t+y_t)(1-F(x_t))+1/x_t-y_t\\ &= 1/x_t-y_t+(1-1/x_t+y_t)\,F(1/x_t)+y_tF(1/y_t)\\ &\overset{(31)}{=} \phi^F(1/y_t,1/x_t). \qquad (32)\end{aligned}$$

Finally, $F$ is right continuous in $(\hat{x},\hat{y})$, and so is $\phi^F(\hat{x},\hat{y})$. Since $x_t\to\hat{x}^+, y_t\to\hat{y}^+$,

$$\phi^F(\hat{x},\hat{y})=\lim_{t\to\infty}\phi^F(x_t,y_t)\overset{(32)}{=}\lim_{t\to\infty}\phi^F(1/y_t,1/x_t)\leq\sup_{x,y\in\mathbb{R}_{++},\ xy>1}\phi^F(x,y).$$

The last inequality follows from $(1/y_t)(1/x_t)>1$. □

### 5.2 Implementing the New Upper Bound for Two Tasks

In this subsection, for a given $\mathcal{S}_k$, we choose $\mathcal{F}$ in Definition 2 to be the family of linear univariate functions

$$\mathcal{F}:=\{c^0+c^1x:\ c^0,c^1\in\mathbb{R}\}. \qquad (33)$$

Then $\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)$ includes, in particular, the CDFs from [3, 17, 18, 23] where the authors use piecewise functions with domains subdivided into two, four, or six intervals. We observe that the upper bounds are better when the domains are subdivided more times or when each piece has a more complex form than a constant function, i.e., when

$c^1$ can be non-zero. We improve upon the existing upper bounds by using a larger number of pieces and letting $c^1$ be non-zero in each piece. Define

$$\phi^F(x, y) := y - 1/x + (1 + 1/x - y)\, F(y) + 1/x F(x). \tag{34}$$

Let $\mathcal{X} := \{(x, y) \in \mathbb{R}^2_{++} : xy \geq 1\}$. Consider two formulations for $R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k))$ which are equivalent to problem (30)

$$\begin{aligned} R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)) = & \inf_{F \in \mathcal{C}_{\mathcal{F}}(\mathcal{S}_k),\ t} \quad t & (35)\\ & \text{s.t.} \quad \phi^F(x, y) \leq t, \ \text{for all } (x, y) \in \mathcal{X} \\ = & \inf_{F \in \mathcal{C}_{\mathcal{F}}(\mathcal{S}_k),\ t} \quad t & (36)\\ & \text{s.t.} \quad \sup_{x \in I_i, y \in I_j} \phi^F(x, y) \leq t, \ \text{for all } i + j \geq 2k + 1. \end{aligned}$$

The second equality follows from the equivalence of problems (30) and (29) since for $\mathcal{S}_k$ defined in (20) $xy \geq 1$ implies $x \in I_i,\ y \in I_j$ with $i + j \geq 2k + 1$. We use problems (35) and (36) to approximate $R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k))$ with high precision. Namely, we use relaxations of problem (35) to compute lower bounds on $R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k))$, and we use feasible solutions to problem (36) to compute upper bounds on $R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k))$.

For $F \in \mathcal{C}_{\mathcal{F}}(\mathcal{S}_k)$ satisfying (28) and (33), the variables in problem (35) are $\left(t, \{c_i^0\}_{i=1}^k, \{c_i^1\}_{i=1}^k\right)$. This problem is LP with infinitely many constraints: each $(x, y) \in \mathcal{X}$ induces a linear constraint. Such problems can be well approximated using the cutting-plane approach introduced by Kelley [12]. Namely, we start with a finite set $\mathcal{Y} \subset \mathcal{X}$ and restrict the set of constraints in (35) to its finite subset generated by $(x, y) \in \mathcal{Y}$. As a result, we obtain a finite linear problem. Denote its optimal solution by $(\underline{F},\ \underline{t})$. Then $\underline{t}$ is a lower bound on $R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k))$.

Next, we substitute $\underline{F}$ in (36) and find a feasible $t$. We compute the supremum for each pair $i, j \in [2k]$ with $i + j \geq 2k + 1$ using the Karush-Kuhn-Tucker (KKT) conditions. For each pair of intervals the inner maximization problem in (36) either is linearly constrained or satisfies the Mangasarian-Fromovitz constraint qualification. Therefore the optimum is among the KKT points, see, e.g., Section 3 in Eustaquio et al. [8] for more details. All problems are simple and have similar structure. Therefore we do not write the KKT conditions explicitly, but consider all possible critical points from the first order conditions and from the boundary conditions. This set contains all the KKT points, and thus the optimal one. At the end we choose the point $(x, y)$ with the maximal value of $\phi^{\underline{F}}(x, y)$ among the critical points. See Section 7.4 for more details. Let $\bar{t}$ be the maximum of $\phi^{\underline{F}}(x, y)$ over all pairs of intervals. The solution $(\underline{F}, \bar{t})$ is feasible for problem (36). Thus $\bar{t}$ is an upper bound on $R_2(\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k))$. Let $(x^*, y^*)$ be a point such that $\phi^{\underline{F}}(x^*, y^*) = \bar{t}$. If $|\bar{t} - \underline{t}| > 10^{-8}$, we proceed from the beginning by restricting problem (35) to the updated set $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{(x^*, y^*)\}$. Otherwise we stop.

To obtain numerical results, we use uniform sets $\mathcal{S}_k^u$ of the form (24). We work with family $\mathcal{C}_{\mathcal{F}}(\mathcal{S}_k^u)$ from Definition 2, where the underlying family of functions $\mathcal{F}$ is defined in (33). We initialize the cutting-plane procedure with $\mathcal{Y} = \{(x, y) : x, y \in \mathcal{S}_k^u,\ xy \geq 1\}$. The best obtained upper bound is indicated in bold in Table 3, it is stronger than the currently best upper bound 1.5068.

**Table 3** Upper bounds on the best approximation ratio for two tasks

| $k$ | 5 | 10 | 16 | 50 | 100 |
|---|---|---|---|---|---|
| Upper bound on $R_2$ | 1.5174 | 1.5096 | 1.5066 | 1.5060 | **1.5059964** |

We verify the upper bound 1.5059964 using exact arithmetics in a similar way as we do it for the upper bounds in Table 2 of Section 4.3.

### 5.3 New Lower Bound for Two Tasks

The cutting-plane approach from Section 5.1 generates not only the upper bound with the corresponding CDF, but also the set of points $\mathcal{Y}$. Using $\mathcal{Y}$, we build a new set $\mathcal{S}_k^*$ of the form (20), which is not uniform as in (24). We consider all $(x, y) \in \mathcal{Y}$ involved in the binding constraints of problem (35) at the last cutting-plane iteration. Next, we take the corresponding $x$, $y$ and their reciprocals, order ascending, round to the $8^{th}$ digit and obtain $\mathcal{S}_k^*$ with $k = 82$. For this set the lower bound $R_n(\mathcal{S}_k^*)$ from problem (18) is 1.5059953, which is stronger than our lower bound from Table 2. As a result, the lower and upper bounds become very close to each other: $|R_2 - 1.505996| \leq 10^{-6}$.

## 6 Conclusion

We consider randomized MIS algorithms to the minimum makespan problem on two unrelated parallel selfish machines. We propose a new $Min - Max$ formulation (13) to find $R_n$, the best approximation ratio over randomized MIS algorithms. The minimization in (13) goes over distributions and the maximization goes over $\mathbb{R}^2_{++}$. The problem is generally intractable, so we build upper and lower bounds on the optimal value. To obtain the lower bound, we solve the initial problem on a finite subset of $\mathbb{R}^2_{++}$. Using the resulting solution, we construct a piecewise constant cumulative distribution function (CDF) for which the worst-case performance is easy to estimate. In this way, we obtain the upper bound on $R_n$. We implement this approach and find new bounds for $n \in \{2, 3, 4\}$ tasks.

For $n = 2$ the best CDF is a known function of univariate margins (copula). We parametrize these margins as piecewise rational functions of degree at most one. The resulting upper bound problem (30) is a linear semi-infinite problem. We solve it by the cutting-plane approach. This approach provides the upper bound 1.5059964 and the CDF for which the algorithm achieves this bound.

As a side result of the cutting-plane approach, we obtain the lower bound 1.5059953, so $|R_2 - 1.505996| \leq 10^{-6}$. This work leaves open several questions for future research. First, our approach could be made more numerically efficient to provide better bounds for $m = 2$ machines. For example, column generation could solve lower bound problem (17) on denser grids and for the larger number of tasks $n$. Parametrizing distributions of more than two variables could improve the results

for the upper bound problem (17). Second, we work with $m = 2$, machines but there are algorithms for $m > 2$ machines with similar properties, e.g., by Lu and Yu [19]. It would be interesting to see how our approach works in the case of more than two machines. Finally, the suggested piecewise and pointwise constructions could be suitable for other problems with optimization over low dimensional functions, including problems from algorithmic mechanism design.

## 7 Proofs

In this section we use some additional notation. First, $\rho(T)$ denotes the vector of the processing time ratios

$$\rho(T) := \left(\frac{T_{11}}{T_{21}}, \frac{T_{12}}{T_{22}}, \ldots, \frac{T_{1n}}{T_{2n}}\right). \tag{37}$$

Second, for $\mathbf{z} \in \mathbb{R}^n_{++}$ we denote by $\mathcal{A}^{\mathbf{z}}$ the algorithm in $\mathcal{A}^{\mathcal{P}_n}$ with the thresholds fixed at $\mathbf{z}$. Finally, for $T \in \mathbb{R}^{2\times n}_{++}$ we let $X^{\mathbf{z},T}$ and $M(\mathbf{z}, T)$ be the output and the makespan of $\mathcal{A}^{\mathbf{z}}$ on $T$, respectively.

### 7.1 Proof of Theorem 2

We begin with a lemma.

**Lemma 2** *For a given number of tasks $n$, let $\mathbb{P} \in \mathcal{P}_n$. For every time matrix $T \in \mathbb{R}^{2\times n}_{++}$, there exists a sequence of time matrices $\{T_k\}_{k>0} \subset \mathbb{R}^{2\times n}_{++}$ such that $\mathbf{P}_{z\sim\mathbb{P}}\left[z_j = T_{k,1j}/T_{k,2j}\right] = 0$ for all $j \in [n]$ and $k$, and*

$$R_n(\mathbb{P}, T) = \lim_{k\to\infty} R_n(\mathbb{P}, T_k).$$

*Proof* Consider a sequence of nonnegative numbers $\{\epsilon_k\}$ such that

1. $\mathbf{P}_{z\sim\mathbb{P}}\left[z_j = \rho(T)_j + \epsilon_k\right] = 0,\ \forall\, k \in \mathbb{N},\ j \in [n]$
2. $\lim_{k\to\infty} \epsilon_k = 0$

A sequence with these properties exists since for every $j \in [n]$ the case $\mathbf{P}_{z\sim\mathbb{P}}\left[z_j = a\right] > 0$ is possible for countably many $a \in \mathbb{R}_{++}$ only. Next, we build a sequence of time matrices $\{T_k\}_{k>0}$, $T_k = (T_{k,ij})$:

$$T_{k,1j} = T_{1j} + \epsilon_k T_{2j} \ \text{ and } \ T_{k,2j} = T_{2j} \ \text{ for all } j \in [n].$$

Notice that $T = \lim_{k\to\infty} T_k$. By adding $\epsilon_k T_{2j}$ to every task on the first machine, we ensure that $\mathbf{P}_{z\sim\mathbb{P}}\left[z_j = T_{k,1j}/T_{k,2j}\right] = 0$ for all $j \in [n]$ and $k$. For each $k$ and all $j \in [n]$, $i \in \{1, 2\}$, we have $T_{ij} \le T_{k,ij}$. So $M^*(T) \le M^*(T_k) \le M(X^*, T_k)$, where $X^*$ is the optimal allocation for $T$. $X^*$ is finite (binary, in particular) and $T = \lim_{k\to\infty} T_k$. Combining this with (1), we see that $M(X^*, T_k)$ tends to $M^*(T)$ when $k$ tends to infinity. Therefore

$$\lim_{k\to\infty} M^*(T_k) = M^*(T). \tag{38}$$

For every time matrix $T_k$ and task $j$, consider the event $B_{k,j}$ : "$\rho(T)_j < z_j \leq \rho(T_k)_j$". Let $B_k = \bigcup_{j=1}^n B_{k,j}$ and let $B_k^c$ be the complement of $B_k$. When $B_k$ happens, outcomes of $\mathcal{A}^{\mathbf{z}}$ on $T$ and $T_k$ are different, otherwise they are the same. By construction of $\mathcal{A}^{\mathbf{z}}$, $M(\mathbf{z}, T)$ is finite for any $T$. Hence

$$\mathbf{E}_{\mathbf{z}\sim\mathbb{P}}\left[M(\mathbf{z}, T_k) - M(\mathbf{z}, T) \mid B_k^c\right] = 0, \quad \text{for all } k \in \mathbb{N}. \tag{39}$$

For any $j \in [n]$ we have $\rho(T_k)_j \to \rho(T)_j^+$. Since the CDF of $\mathbb{P}$ is continuous from the right,

$$\lim_{k\to\infty} \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}\left[z_j \leq \rho(T_k)_j\right] = \mathbf{P}_{z\sim\mathbb{P}}\left[z_j \leq \rho(T)_j\right],$$

which implies

$$\begin{aligned}\lim_{k\to\infty} \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}[B_k] &= \lim_{k\to\infty} \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}\left[\bigcup_{j=1}^n B_{k,j}\right] \leq \sum_{j=1}^n \lim_{k\to\infty} \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}\left[B_{k,j}\right] \\ &= \sum_{j=1}^n \lim_{k\to\infty} \left(\mathbf{P}_{\mathbf{z}\sim\mathbb{P}}\left[z_j \leq \rho(T_k)_j\right] - \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}\left[z_j \leq \rho(T)_j\right]\right) = 0. \end{aligned} \tag{40}$$

Finally, for any $k \in \mathbb{N}$

$$\begin{aligned} M(\mathbb{P}, T_k) - M(\mathbb{P}, T) &= \mathbf{E}_{\mathbf{z}\sim\mathbb{P}}\left[M(\mathbf{z}, T_k) - M(\mathbf{z}, T) \mid B_k\right]\mathbf{P}_{\mathbf{z}\sim\mathbb{P}}[B_k] \\ &\quad + \mathbf{E}_{\mathbf{z}\sim\mathbb{P}}\left[M(\mathbf{z}, T_k) - M(\mathbf{z}, T) \mid B_k^c\right](1 - \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}[B_k]) \\ &\overset{(39)}{=} \mathbf{E}_{\mathbf{z}\sim\mathbb{P}}\left[M(\mathbf{z}, T_k) - M(\mathbf{z}, T) \mid B_k\right]\mathbf{P}_{\mathbf{z}\sim\mathbb{P}}[B_k] \\ &\leq |T|_1 \mathbf{P}_{\mathbf{z}\sim\mathbb{P}}[B_k] \end{aligned}$$

Thus by (40),

$$\lim_{k\to\infty} M(\mathbb{P}, T_k) = M(\mathbb{P}, T), \tag{41}$$

and

$$R_n(\mathbb{P}, T) = \frac{M(\mathbb{P}, T)}{M^*(T)} \overset{(38),(41)}{=} \lim_{k\to\infty} \frac{M(\mathbb{P}, T_k)}{M^*(T_k)} = \lim_{k\to\infty} R_n(\mathbb{P}, T_k). \qquad \square$$

***Proof of Theorem 2*** Recall that $\mathcal{T}$ is defined in (6). By Lemma 2, for every $T \in \mathbb{R}_{++}^{2\times n}$ there exists a sequence of time matrices $\{T_k\}_{k>0} \subset \mathcal{T}$ such that

$$R_n(\mathbb{P}, T) = \lim_{k\to\infty} R_n(\mathbb{P}, T_k) \leq \sup_{T\in\mathcal{T}} R_n(\mathbb{P}, T).$$

Hence

$$R_n(\mathbb{P}) = \sup_{T\in\mathbb{R}_{++}^{2\times n}} R_n(\mathbb{P}, T) \leq \sup_{T\in\mathcal{T}} R_n(\mathbb{P}, T).$$

The opposite inequality holds since $\mathcal{T} \subset \mathbb{R}_{++}^{2\times n}$. $\square$

### 7.2 Proof of Theorem 3

Recall the definitions of $S_{inv}$ and $S_n$ from Section 3. Let $\sigma_{id}$ be the identity action of $S_2$ and $\sigma_{swap}$ be the non-identity action of $S_2$. We say that that the group $S_2 \times S_n$ acts on a matrix in $\mathbb{R}^{2\times n}$ by permuting the two rows and the $n$ columns of this matrix. Namely, for $A \in \mathbb{R}^{2\times n}$ and $(\sigma, \pi) \in S_2 \times S_n$, we define the action of $(\sigma, \pi) \in S_2 \times S_n$ on $A$ by

$$\sigma A \pi := (A_{\sigma i, \pi j}).$$

We show that for any $T \in \mathbb{R}^{2\times n}_{++}$ the optimal makespan $M^*(T)$ is invariant under the actions of $S_2 \times S_n$ on $T$. Moreover, the expected makespan $M(\mathbb{P}, T)$ is invariant under the actions of $S_2 \times S_n$ on $T$ and $S_{inv} \times S_n$ on $\mathbb{P}$. These two results together imply Theorem 3.

**Lemma 3** *$M^*(T) = M^*(\sigma T \pi)$ for all $(\sigma, \pi) \in S_2 \times S_n,\ T \in \mathbb{R}^{2\times n}_{++}$*

*Proof* Consider a time matrix $T$ and actions $\pi \in S_n,\ \sigma \in S_2$. Let $X^* = (X^*_{ij})$ be an optimal allocation matrix for $T$. Then $\sigma T \pi = (T_{\sigma i, \pi j}),\ \sigma X^* \pi = (X^*_{\sigma i, \pi j})$. This implies

$$M^*(\sigma T \pi) \leq \max_i \left\{ \sum_{j\in[n]} T_{\sigma i, \pi j} X^*_{\sigma i, \pi j} \right\} = \max_i \left\{ \sum_{j\in[n]} T_{ij} X^*_{ij} \right\} = M^*(T).$$

Analogously, for the time matrix $\sigma T \pi$ and actions $\pi^{-1} \in S_n,\ \sigma^{-1} \in S_2$, we obtain

$$M^*(T) \leq M^*(\sigma T \pi). \qquad \square$$

***Proof of Theorem 3*** Let $\mathbb{P} \in \mathcal{P}_n$, $\mathbf{z} \in \mathbb{R}^n_{++}$ and $T \in \mathbb{R}^{2\times n}_{++}$. Since we are interested in $R_n(\mathbb{P})$, by Lemma 2 we can assume without loss of generality that $T$ is such that $\mathbf{P}_{z\sim\mathbb{P}}\left[z_j = T_{1j}/T_{2j}\right] = 0$ for all $j \in [n]$. Consider $(\gamma, \pi) \in S_{inv} \times S_n$ and $\mathbf{y} = {}^{\gamma}\mathbf{z}\pi$. Let $\sigma = \sigma_{id}$ if $\gamma = {}^{id}$. and $\sigma = \sigma_{swap}$ if $\gamma = {}^{inv}$. Then $\mathcal{A}^{\mathbf{z}}$ sends task $j$ to machine $i$ on $T$ if and only if $\mathcal{A}^{\mathbf{y}}$ sends task $\pi j$ to machine $\sigma i$ on $\sigma T \pi$. As a result, $T_{ij} X^{\mathbf{z},T}_{ij} = T_{\sigma i \pi j} X^{\mathbf{y},\sigma T\pi}_{\sigma i \pi j}$ for all $i$, $j$ and

$$\begin{aligned} M(\mathbf{z}, T) &= \max_i \left\{ \sum_{j\in[n]} T_{ij} X^{\mathbf{z},T}_{ij} \right\} = \max_i \left\{ \sum_{j\in[n]} T_{\sigma i \pi j} X^{\mathbf{y},\sigma T\pi}_{\sigma i \pi j} \right\} \\ &= \max_i \left\{ \sum_{j\in[n]} T_{ij} X^{\mathbf{y},\sigma T\pi}_{ij} \right\} = M(\mathbf{y}, \sigma T \pi). \end{aligned}$$

Therefore

$$M(\mathbb{P}, T) = \mathbf{E}_{\mathbf{z}\sim\mathbb{P}} M(\mathbf{z}, T) = \mathbf{E}_{\mathbf{y}\sim{}^{\gamma}\mathbb{P}\pi} M(\mathbf{y}, \sigma T \pi) = M({}^{\gamma}\mathbb{P}\pi, \sigma T \pi), \tag{42}$$

Combining this with Lemma 3, obtain

$$R_n(\mathbb{P}, T) = R_n({}^{\gamma}\mathbb{P}\pi, \sigma T \pi).$$

By Theorem 2,

$$R_n(\mathbb{P}) = \sup_{T\in\mathcal{T}} R_n(\mathbb{P}, T) = \sup_{T\in\mathcal{T}} R_n({}^{\gamma}\mathbb{P}\pi, \sigma T\pi) = R_n({}^{\gamma}\mathbb{P}\pi),$$

where $\mathcal{T}$ is defined in (6). □

### 7.3 Proof of Theorem 5

By Proposition 2, $R_n(\mathbb{P}) \le \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}}(x, y)$. Next, we prove the opposite inequality. Consider $\mathbb{P} \in \mathcal{C}_n$. We start with the case $n = 2$ and proceed similarly to Lu and Yu [19]. Denote the bivariate marginal distribution of $\mathbb{P}$ by $\mathbb{P}_2$. Consider a time matrix $T \in \mathbb{R}_{++}^{2\times 2}$ and denote $\frac{T_{11}}{T_{21}}$ by $x$, $\frac{T_{12}}{T_{22}}$ by $y$. Construct the following matrix $T_0$:

| | Task 1 | Task 2 |
|---|---|---|
| Machine 1 | 1 | $y$ |
| Machine 2 | $1/x$ | 1 |

The expected makespan of $\mathcal{A}^{\mathbb{P}_2}$ on this instance is $M(\mathbb{P}_2, T_0)$:

$$\begin{aligned}
M(\mathbb{P}_2, T_0) &= \mathbf{P}_{\mathbf{z}\sim\mathbb{P}_2}[z_1 > x,\ z_2 > y]\,(1+y) + \mathbf{P}_{\mathbf{z}\sim\mathbb{P}_2}[z_1 > x,\ z_2 \le y] \\
&\quad + \mathbf{P}_{\mathbf{z}\sim\mathbb{P}_2}[z_1 \le x,\ z_2 > y]\max\{1/x,\ y\} \\
&\quad + \mathbf{P}_{\mathbf{z}\sim\mathbb{P}_2}[z_1 \le x,\ z_2 \le y]\,(1+1/x) \\
&= [1 - H(x,y) - (F(y) - H(x,y)) - (F(x) - H(x,y))]\,(1+y) \\
&\quad + (F(y) - H(x,y)) + (F(x) - H(x,y))\max\{1/x, y\} \\
&\quad + H(x,y)\,(1+1/x) \\
&= 1 + y - F(x)\,(1 + y - \max\{1/x,\ y\}) - yF(y) \\
&\quad + H(x,y)\,(y - \max\{1/x,\ y\} + 1 + 1/x) \\
&= 1 + y - \min\{1,\ 1 - 1/x + y\}\,F(x) - yF(y) \\
&\quad + \min\{1 + 1/x, 1 + y\}\,H(x,y) \\
&= \phi^{\mathbb{P}}(x,y).
\end{aligned}$$

Denote the minimum makespan on $T_0$ by $M^*$. By construction $M^* \le 1$, hence

$$R_2(\mathbb{P}_2) \ge R_2(\mathbb{P}_2, T_0) = \frac{M(\mathbb{P}_2, T_0)}{M^*} \ge M(\mathbb{P}_2, T_0) = \phi^{\mathbb{P}}(x,y).$$

This holds for all $x, y \in \mathbb{R}_{++}$, therefore

$$R_2(\mathbb{P}_2) \ge \sup_{x,y\in\mathbb{R}_{++}} \phi^{\mathbb{P}}(x,y).$$

Now, fix $n > 2$. Choose a small $\epsilon > 0$ and consider the following time matrix $T_\epsilon$:

| | Task 1 | Task 2 | Task 3 | ... | Task n |
|---|---|---|---|---|---|
| Machine 1 | 1 | $y$ | $\epsilon$ | ... | $\epsilon$ |
| Machine 2 | $1/x$ | 1 | $\epsilon$ | ... | $\epsilon$ |

The expected makespan of $\mathcal{A}^{\mathbb{P}}$ on this instance, $M(\mathbb{P}, T_\epsilon)$, satisfies

$$M(\mathbb{P}_2, T_0) \le M(\mathbb{P}, T_\epsilon) \le M(\mathbb{P}_2, T_0) + (n-2)\epsilon,$$

and the optimal makespan, $M^*(T_\epsilon)$, satisfies

$$M^* \le M^*(T_\epsilon) \le M^* + (n-2)\epsilon.$$

Using the result for $n = 2$,

$$R_n(\mathbb{P}) \ge \lim_{\epsilon\to 0} R_n(\mathbb{P}, T_\epsilon) = \lim_{\epsilon\to 0} \frac{M(\mathbb{P}, T_\epsilon)}{M^*(T_\epsilon)} = \frac{M(\mathbb{P}_2, T_0)}{M^*} \ge \phi^{\mathbb{P}}(x, y),$$

which holds for all $x, y \in \mathbb{R}_{++}$.

### 7.4 Possible Critical Points Computation for the Function (34)

In this section we find possible critical points for the function

$$\phi^F(x, y) = y - 1/x + (1 + 1/x - y)\, F(y) + 1/x F(x),$$

with $F$ defined in (28) using (33) on $I_i \times I_j$ such that $i, j \in [2k],\ i + j \ge 2k+1$. By construction, in each interval $F$ is differentiable, and $\phi^F$ is differentiable on $I_i \times I_j$ (the expression simplifies for $i = 1$ and $j = 2k$). The first derivatives of the function $\phi^F(x, y)$ are:

$$\frac{\partial \phi^F(x,y)}{\partial x} = \frac{1}{x^2}\left(1 - F(x) - F(y) + \frac{\partial F(x)}{\partial x} x\right) \text{ and}$$
$$\frac{\partial \phi^F(x,y)}{\partial y} = 1 - F(y) + (1 + 1/x - y)\frac{\partial F(y)}{\partial y}.$$

Next, we consider the three possible cases for $i, j$. For each of these cases we substitute $F(x),\ F(y)$ from (28) and find the analytical solution to the system $\frac{\partial \phi^F(x,y)}{\partial x} = 0,\ \frac{\partial \phi^F(x,y)}{\partial y} = 0$. For this purpose we use Wolfram|Alpha [28]. The obtained solution is denoted by $(x^*,\ y^*)$.

***Case 1.*** $k < i \le 2k,\ 1 \le j \le k$.

In this case $F(x) = 1 - c_i^0 - c_i^1 x,\ F(y) = c_j^0 + c_j^1/y$ and $y \in (0, 1)$. Hence

$$\frac{\partial \phi^F(x,y)}{\partial x} = \frac{1}{x^2}\left(c_i^0 - c_j^0 - \frac{c_j^1}{y}\right),\ \frac{\partial \phi^F(x,y)}{\partial y} \ge 0.$$

The latter holds since $F(y) \le 1$, $F(y)$ is non-decreasing by construction, and $y \in (0, 1)$. The sign of the derivative with respect to $x$ does not depend on $x$. The function $\phi^F(x, y)$ is non-decreasing in $y$ and is either non-increasing or non-decreasing in $x$. We not know this in advance, so we use the set $\{(s_{i-1}, s_j), (s_i, s_j)\}$ as possible critical points.

***Case 2.*** $k < i \le 2k,\ k < j \le 2k$.

In this case $F(x) = 1 - c_i^0 - c_i^1 x,\ F(y) = 1 - c_j^0 - c_j^1 y$ and $y \ge 1$. Hence

$$\frac{\partial \phi^F(x,y)}{\partial x} = 1/x^2\left(c_i^0 - 1 + c_j^0 + c_j^1 y\right),\ \frac{\partial \phi^F(x,y)}{\partial y}$$
$$= c_j^0 - (1 + 1/x - 2y)\, c_j^1.$$

As in ***Case 1***, the sign of the derivative with respect to $x$ does not depend on $x$, hence $\phi^F(x, y)$ is non-increasing or non-decreasing in $x$. We do not know this in advance, so we start with the set $\{s_{i-1}, s_i\} \times \{s_{j-1}, s_j, y^*\}$ as possible critical points. If $i+j = 2k+1$ (that is, the line $xy = 1$ crosses the rectangle $I_i \times I_j$), we additionally consider the point $(\frac{1}{y^*}, y^*)$. We check all pairs for feasibility and exclude the infeasible ones.

***Case 3.*** $1 \le i \le k,\ k < j \le 2k$.

In this case $F(x) = c_i^0 + c_i^1/x,\ F(y) = 1 - c_j^0 - c_j^1 y$, and

$$\frac{\partial \phi^F(x, y)}{\partial x} = \frac{1}{x^2}\left(-c_i^0 + c_j^0 + c_j^1 y - \frac{2c_i^1}{x}\right),$$

$$\frac{\partial \phi^F(x, y)}{\partial y} = c_j^0 - (1 + 1/x - 2y)\, c_j^1.$$

The sign of the derivatives is unknown, so we start with the set $\{s_{i-1}, s_i, x^*\} \times \{s_{j-1}, s_j, y^*\}$ as possible critical points. If $i + j = 2k + 1$ (that is, the line $xy = 1$ crosses the rectangle $I_i \times I_j$), we additionally consider the set $\left\{(x^*, \frac{1}{x^*}), (\frac{1}{y^*}, y^*)\right\}$. We check all resulting pairs for feasibility and exclude the infeasible ones.

When $x \in I_1$ or $y \in I_{2k}$, $\phi^F(x, y)$ simplifies by construction of (28). In our computations we analyze these situations separately.

**Acknowledgements** We would like to thank the two anonymous reviewers for pointing out an inconsistency in the first proof of Corollary 1 and suggesting numerous improvements in the presentation of the results in this paper.



## References

1. Auletta, V., Christodoulou, G., Penna, P.: Mechanisms for scheduling with single-bit private values. Theory Comput. Syst. **57**(3), 523–548 (2015)
2. Azar, Y., Hoefer, M., Maor, I., Reiffenhäuser, R., Vöcking, B.: Truthful mechanism design via correlated tree rounding. Math. Program. **163**(1), 445–469 (2017)
3. Chen, X., Du, D., Zuluaga, L.: Copula-based randomized mechanisms for truthful scheduling on two unrelated machines. Theory Comput. Syst. **57**(3), 753–781 (2015)
4. Christodoulou, G., Koutsoupias, E., Vidali, A.: A lower bound for scheduling mechanisms. Algorithmica **55**(4), 729–740 (2009)
5. Dobre, C., Vera, J.: Exploiting symmetry in copositive programs via semidefinite hierarchies. Math. Program. **151**(2), 659–680 (2015)
6. Dobzinski, S., Sundararajan, M.: On characterizations of truthful mechanisms for combinatorial auctions and scheduling. In: Proceedings of the 9th ACM Conference on Electronic Commerce EC '08, pp. 38–47 (2008)
7. Elbassioni, K., Mehlhorn, K., Ramezani, F.: Towards more practical linear programming-based techniques for algorithmic mechanism design. Theory Comput. Syst. **59**(4), 641–663 (2016)

8. Eustaquio, R.G., Karas, E.W., Ribeiro, A.A.: Constraint qualifications for nonlinear programming. In: Conference Proceedings. http://paginapessoal.utfpr.edu.br/eustaquio/my-research-interests/kkterabio.pdf (2008)
9. Gatermann, K., Parrilo, P.A.: Symmetry groups, semidefinite programs, and sums of squares. J. Pure Appl. Algebra **192**(1), 95–128 (2004)
10. Gilmore, P.C., Gomory, R.E.: A linear programming approach to the cutting-stock problem. Oper. Res. **9**, 849–859 (1961)
11. Gualà, L., Proietti, G.: Exact and approximate truthful mechanisms for the shortest paths tree problem. Algorithmica **49**(3), 171–191 (2007)
12. Jr., J.E.K.: The cutting-plane method for solving convex programs. SIAM J. Appl. Math. **8**, 703–712 (1960)
13. de Klerk, E., Pasechnik, D., Schrijver, A.: Reduction of symmetric semidefinite programs using the regular $*$-representation. Math. Program. **109**(2), 613–624 (2007)
14. Koutsoupias, E., Vidali, A.: A lower bound of $1+\phi$ for truthful scheduling mechanisms. Algorithmica **66**, 211–223 (2013)
15. Lavi, R., Swamy, C.: Truthful and near-optimal mechanism design via linear programming. J. ACM **58**(6), 1–25 (2011)
16. Lenstra, J., Shmoys, D., Tardos, É.: Approximation algorithms for scheduling unrelated parallel machines. Math. Program. **46**, 259–271 (1990)
17. Lu, P.: On 2-player randomized mechanisms for scheduling. In: Internet and Network Economics: 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings, pp. 30–41 (2009)
18. Lu, P., Yu, C.: An improved randomized truthful mechanism for scheduling unrelated machines. In: Dans Proceedings of the 25th Annual Symposium on the Theoretical Aspects of Computer Science - STACS 2008 Bordeaux : France (2008)
19. Lu, P., Yu, C.: Randomized truthful mechanisms for scheduling unrelated machines. In: Internet and Network Economics: 4th International Workshop, WINE 2008, Shanghai, China, December 17-20, 2008. Proceedings pp. 402–413 (2008)
20. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
21. Mu'alem, A., Schapira, M.: Setting lower bounds on truthfulness: Extended abstract. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms SODA '07, pp. 1143–1152 (2007)
22. Nelsen, R.: An introduction to Copulas. Springer, New York (2006)
23. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games Econom. Behav. **35**(1), 166–196 (2001)
24. Potts, C.: Analysis of a linear programming heuristic for scheduling unrelated parallel machines. Discrete Appl. Math **10**(2), 155–164 (1985)
25. Saks, M.E., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings 6th ACM Conference on Electronic Commerce (EC), pp. 286–293 (2005)
26. Shchepin, E.V., Vakhania, N.: An optimal rounding gives a better approximation for scheduling unrelated machines. Oper. Res. Lett. **33**(2), 127–133 (2005)
27. Vaze, R., Coupechoux, M.: Online budgeted truthful matching. SIGMETRICS Perform. Eval. Rev. **44**(3), 3–6 (2017)
28. Wolfram|Alpha: http://www.wolframalpha.com (2009)