The Price of Envy-Freeness in Machine Scheduling[☆]

Vittorio Bilò[a], Angelo Fanelli[b], Michele Flammini[c,d], Gianpiero Monaco[c], Luca Moscardelli[e,*]

[a] *Department of Mathematics and Physics "Ennio De Giorgi", University of Salento, Italy*
[b] *CNRS, (UMR-6211), France.*
[c] *Department of Information Engineering Computer Science and Mathematics, University of L'Aquila, Italy*
[d] *Gran Sasso Science Institute, L'Aquila, Italy*
[e] *Department of Economic Studies, University of Chieti-Pescara, Italy*

Abstract

We consider k-envy-free assignments for scheduling problems in which the completion time of each machine is not k times larger than the one she could achieve by getting the jobs of another machine, for a given factor $k \geq 1$. We introduce and investigate the notion of price of k-envy-freeness, defined as the ratio between the makespan of the best k-envy-free assignment and that of an optimal allocation achievable without envy-freeness constraints. We provide exact or asymptotically tight bounds on the price of k-envy-freeness for all the basic scheduling models, that is unrelated, related and identical machines. Moreover, we show how to efficiently compute such allocations with a worsening multiplicative factor being at most the best approximation ratio for the minimum makespan problem guaranteed by a polynomial time algorithm for each specific model. Finally, we extend our results to the case of restricted assignments and to the objective of minimizing the sum of the completion times of all the machines.

Keywords: Envy–freeness, Machine Scheduling

[☆]This research was partially supported by the PRIN 2010–2011 research project ARS TechnoMedia (Algorithmics for Social Technological Networks), funded by the Italian Ministry of University and Research and by COST Action IC1205 on Computational Social Choice. A preliminary version [2] of this work appeared in Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS), 2014.

[*]Corresponding author

Email addresses: vittorio.bilo@unisalento.it (Vittorio Bilò), angelo.fanelli@unicaen.fr (Angelo Fanelli), michele.flammini@univaq.it (Michele Flammini), gianpiero.monaco@univaq.it (Gianpiero Monaco), luca.moscardelli@unich.it (Luca Moscardelli)

1. Introduction

The evolution of scheduling closely tracked the development of computers. Given m machines that have to process n jobs, minimizing the makespan of an assignment of the jobs to the machines is one of the most well-studied problem in the Theory of Algorithms [13, 17, 18, 20]. In more details, assuming that the processing of job i on machine j requires time $p_{ij} > 0$, the completion time of machine j (under a certain assignment) is given by the sum of the processing times of all the jobs allocated to j. The makespan of an assignment is the maximum completion time among all the machines (we stress that an assignment is not forced to use all the available machines) and the objective of the scheduling problem is to find an assignment of minimum makespan.

In the literature, three different models of machines have been adopted. The general setting illustrated above is called scheduling problem with *unrelated machines* [20]. An interesting particular scenario is the case with *related machines* [18], where each job i has a load $l_i > 0$ and each machine j has a speed of processing $s_j > 0$, and thus the processing time of job i on machine j is given by $p_{ij} = l_i/s_j$. Finally, the even more specific setting in which the speed of each machine is 1 is referred to as the scheduling problem with *identical machines* [13, 17]. Even this latter problem is NP-hard [17].

The approximability of the scheduling problem has been well understood for all the three models described above. However, all the proposed solutions do not envisage fair allocations in which no machine prefers (or envies) the set of the tasks assigned to another machine, i.e., for which her completion time would be strictly smaller. In the literature, such fairness property is referred to as "envy-freeness" [9, 10]. Specifically, consider a scenario in which a set of tasks (jobs) has to be allocated among employees (machines) in such a way that the last task finishes as soon as possible. It is natural to consider fair allocations, that is allocations where no employee prefers (or envies) the set of tasks assigned to some other employee, i.e., a set of tasks for which her completion time would be strictly smaller than her actual one.

It is possible to consider two different variants of this model, depending on the fact that an employee (i) can envy the set of tasks assigned to any other employee or (ii) can only envy the set of tasks of other employees getting at least one job: in the latter case, employees not getting any job do not create envy. In the following, we provide some scenarios motivating both variants.

For the first variant, consider a company that receives an order of tasks that must be assigned among its m employees. For equity reasons, in order to make the workers satisfied with their task assignment so that they are as productive as they can, the tasks should be assigned in such a way that no envy is induced among the employees.

For the second variant, consider a scenario in which a company, in order to fulfill a complex job composed by several tasks, has to engage a set of employees that, for law or trade union reasons have to be all paid out the same wage. Again, for making the workers as productive as they can, it is required that no envy is induced, but in this case we are interested only in the envy among the *engaged*

employees, i.e. the ones receiving at least a task to perform.

We notice that the existence of envy-free schedules is not guaranteed in the first variant of the model. For instance, consider a scenario where the number of machines is strictly greater than the number of jobs. Clearly at least one machine would not get any job and all the machines getting at least one job would be envious. Therefore, in the following, we shall focus on the second variant of the model, in which envy-freeness is required only among machines getting at least one job.

We adopt a more general definition of envy-free allocations, namely the *k-envy-freeness* (for any $k \geq 1$): Given an assignment and two machines j, j' (where both j and j' get jobs), we say that j k-envies j' if the completion time of j is at least k times the completion time she would have when getting the set of jobs assigned to j'. In other words, an assignment is k-envy-free if no machine would decrease her completion time by a factor at least k by being assigned all the jobs allocated to another machine. Notice that a k-envy-free assignment always exists: a trivial one can be obtained by allocating all the jobs to a single machine, even if it might have a dramatically high makespan.

We are interested in analyzing the loss of performance due to the adoption of envy-free allocations. Our study has an optimistic nature and, then, aims at quantifying the efficiency loss in the best k-envy-free assignment. Therefore, we introduce the **price of k-envy-freeness**, defined as the ratio between the makespan of the best k-envy-free assignment and that of an optimal assignment. In the literature, other papers performed similar optimistic studies, see, for instance, [1, 7]. The price of k-envy-freeness represents an ideal limitation to the efficiency achievable by any k-envy-free assignment. In our work, we also show how to efficiently compute k-envy-free assignments which nicely compare with the performance of the best possible ones. We point out that the computation of non-trivial k-envy-free assignments is necessary to achieve good quality solutions, since the ratio between the makespan of the worst k-envy-free assignment and that of an optimal assignment can be very high. In particular, it is unbounded for unrelated machines, $n\frac{s_{max}}{s_{min}}$ for related ones, where s_{max} (resp. s_{min}) is the maximum (resp. minimum) speed among all the machines, and n for identical machines.

Related Work. The scheduling problem with unrelated machines has been studied in [20]. The authors provide a 2-approximation polynomial time algorithm and show that the problem cannot be approximated in polynomial time within a factor less than 3/2. Polynomial time approximation schemes for related and identical machines have been presented in [18] and [17], respectively.

The problem of fair allocation is a longstanding issue, thus, the literature on this topic includes hundreds of references. For a nice review, we refer the reader to the book [5]. One common notion of fairness, recurring in many papers and therefore adopted for central problems, is that of envy-freeness. For instance, the classical Vickrey auctions [24], as well as some optimal Bayesian auctions [3, 21], generate envy-free outcomes. An interesting paper explicitly dealing with envy-free auctions is [14]. Studies on envy-free divisions, typically

referred to as envy-free cake cutting, can be found in [4, 9, 10]. Furthermore, [11, 15] consider algorithmic issues related to the envy-free pricing problem, that is a scenario in which a seller has to set (envy-free) prices and allocations of items to buyers in order to maximize the total revenue.

Concerning scheduling problems, an important stream of research is the one focusing on envy-free algorithmic mechanism design. Roughly speaking, algorithmic mechanism design is the attempt of motivating the machines, through payments or incentives, to follow desired behaviors (*truthful mechanisms*). Upper and lower bounds on the approximation ratio achieved by truthful mechanisms have been given in [8, 19, 23]. However, such papers are not concerned with fair allocations. To the best of our knowledge, envy-free mechanisms for the scheduling problem with unrelated machines have been first considered in [16]: the authors prove a lower bound of $2 - 1/m$ and an upper bound of $(m + 1)/2$ on the performance guarantee of envy-free truthful mechanisms. Such upper and lower bounds have been improved in [6] to $O(\log m)$ and $\Omega\left(\frac{\log m}{\log \log m}\right)$, respectively. Recently, [12] shows that no truthful mechanism can guarantee an envy-free allocation with a makespan less than a factor of $O(\log m)$ the optimal one, thus closing the gap. It is worth noticing that, for $k = 1$, our model can be seen as a special case of the one considered in [6, 12, 16] when the same payment is provided to all the machines receiving at least a job, while no payment is given to the other machines.

The work most closely related to our study is [7]. The authors consider the envy-free scheduling problem with unrelated machines with some substantial differences with respect to our setting. Specifically, *i)* they only consider 1-envy-free assignments (while we consider k-envy-free assignments, for any $k \geq 1$); *ii)* the objective in their work is that of minimizing the sum of the completion times of all jobs (while we mainly consider the makespan); *iii)* in their setting all the machines contribute to create envy (while in our setting only machines getting at least one job are considered for the envy-freeness). Not surprisingly, the authors prove that, in their setting, the price of envy-freeness is unbounded.

Our Results. We consider the price of k-envy-freeness in the scheduling problem, that is, the ratio between the makespan of the best k-envy-free assignment and that of an optimal assignment. We investigate the cases of unrelated, related and identical machines and provide exact or asymptotically tight bounds on the price of k-envy-freeness. We stress that low values of k implies a greater attitude to envy, which tremendously reduces the set of k-envy-free assignments. A natural threshold that arose in our analysis of the cases with related and identical machines is the value $k = 2$, as it can be appreciated in the following table where we summarize our main results. They are fully described in Sections 3, 4 and 5 for identical, related and unrelated machines, respectively.

		Identical	Related	Unrelated
$k = 1$	UB and LB	$\min\{n, m\}$	$\min\{n, m\}$	$2^{\min\{n,m\}-1}$
$k \in (1, 2)$	UB	$\frac{2k}{k-1}$	$2k\sqrt{\frac{m}{k-1}}$	$\left(1 + \frac{1}{k}\right)^{\min\{n,m\}-1}$
	LB	$\Omega\left(\frac{2k}{k-1}\right)$	$\Omega\left(\sqrt{\frac{m}{k-1}}\right)$	$\left(1 + \frac{1}{k}\right)^{\min\{n,m\}-1}$
$k \geq 2$	UB	$1 + \frac{1}{k}$	$2 + \max\left\{1, \sqrt{\frac{m}{k}}\right\}$	$\left(1 + \frac{1}{k}\right)^{\min\{n,m\}-1}$
	LB	$1 + \frac{1}{k}$	$\max\left\{1, \sqrt{\frac{m}{k}}\right\}$	$\left(1 + \frac{1}{k}\right)^{\min\{n,m\}-1}$

A further result derives from the fact that our upper bound proofs are constructive and, therefore, they de facto provide polynomial time algorithms able to calculate good k-envy-free assignments. Such an extension is discussed in Section 6.

Furthermore, in Section 7.1 we also consider the *restricted* scheduling problem, where each job can be assigned only to a subset of machines. While the case with unrelated machines naturally envisages such a setting (in fact for each job i, it is enough to set to infinite the processing time p_{ij} for each machine j not able to schedule i), for related and identical machines, we again provide exact or asymptotically tight bounds on the price of k-envy-freeness.

Finally, besides considering the problem of minimizing the makespan, we consider in Section 7.2 the problem of minimizing the sum of the completion times of all the machines. Specifically, for such a problem, we show that the price of k-envy-freeness is 1 (for any $k \geq 1$) for the related and identical settings, while, for the unrelated setting, we extend the upper and lower bounds holding for the problem of minimizing the makespan.

2. Preliminaries

In the scheduling problem, there are $m \geq 2$ *machines* and n indivisible *jobs* to be assigned to the machines. In the *unrelated* case, the time of running job i on machine j is given by $p_{ij} > 0$. In the *related* setting, each job i has a load $l_i > 0$, each machine has a speed of processing $s_j > 0$, and the processing time of job i on machine j is given by $p_{ij} = l_i/s_j$. We refer to the specific setting in which the speed of each machine is 1 as *identical*, where $p_{ij} = l_i$.

For an integer $h > 0$, define $[h] := \{1, \ldots, h\}$. In the related and identical setting, we denote with $L = \sum_{i\in[n]} l_i$ the total load of all the jobs and with $l_{max} = \max_{i\in[n]} l_i$ the maximum load of a job.

An *assignment* or *solution* $\mathbf{N}$ is specified by a partition of the set of jobs into m components, i.e., $(N_j)_{j\in[m]}$, where N_j denotes the set of jobs assigned to machine j. Let Q be a set of jobs, we use the notation $C_j(Q)$ to denote the completion time of machine j on the set Q, i.e., $C_j(Q) = \sum_{i\in Q} p_{ij}$. Thus $C_j(N_j)$ denotes the completion time of machine j under the assignment $\mathbf{N}$. For the related and identical settings, let $L_j(\mathbf{N})$ be the total load of the jobs assigned by $\mathbf{N}$ to machine $j \in [m]$, i.e., $L_j(\mathbf{N}) = \sum_{i\in N_j} l_i$ and $L_{min}(\mathbf{N}) = \min\{L_j(\mathbf{N}) : j \in [m] \wedge N_j \neq \emptyset\}$ (resp. $L_{max}(\mathbf{N}) = \max\{L_j(\mathbf{N}) : j \in [m] \wedge N_j \neq \emptyset\}$)

the minimum (resp. maximum) load of the non-empty machines in $\mathbf{N}$. Notice that, in the related setting, we have $C_j(N_j) = L_j(\mathbf{N})/s_j$ and, for the identical one, $C_j(N_j) = L_j(\mathbf{N})$. The *makespan* of assignment $\mathbf{N}$ is defined as $\mathcal{M}(\mathbf{N}) = \max_{j\in[m]} C_j(N_j)$, that is the maximum processing time among all the machines. An *optimal* assignment is one minimizing the makespan. We denote by $\mathbf{O}$ an optimal assignment.

Given an assignment $\mathbf{N}$, a real value $k \geq 1$, and two machines j, j' such that $N_j \neq \emptyset$ and $N_{j'} \neq \emptyset$, we say that j **k-envies** j' if $C_j(N_j) > kC_j(N_{j'})$. An assignment $\mathbf{N}$ is **k-envy-free** if $C_j(N_j) \leq kC_j(N_{j'})$ for every pair of machines (j, j') such that $N_j \neq \emptyset$ and $N_{j'} \neq \emptyset$. Notice that a k-envy-free assignment can always be obtained by assigning all jobs to a single machine. The **price of k-envy-freeness** (PoEF_k) is defined as the ratio between the makespan of the best k-envy-free assignment and the makespan of an optimal assignment. More formally, let $\mathcal{F}_k$ be the set of the k-envy-free assignments, then $\text{PoEF}_k = \min_{\mathbf{N}\in\mathcal{F}_k} \frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{O})}$.

We conclude this section with some preliminary general results.

Proposition 1. *For the scheduling problem with related machines,* $\text{PoEF}_k \leq \min\{n, m\}$ *for any* $k \geq 1$.

Proof: Assume that machine 1 is the fastest one, i.e., $s_1 \geq s_j$ for each $j \in [m]$. Clearly, the solution $\mathbf{N}$ assigning all jobs to machine 1 is k-envy-free for any $k \geq 1$ and has $\mathcal{M}(\mathbf{N}) = \frac{L}{s_1} \leq \frac{nl_{max}}{s_1}$. By $\mathcal{M}(\mathbf{O}) \geq \frac{l_{max}}{s_1}$ and $\mathcal{M}(\mathbf{O}) \geq \frac{L}{ms_1}$, we obtain the claim. □

Such a simple upper bound on the price of k-envy-freeness proves to be tight when $k = 1$ even for the setting of identical machines.

Proposition 2. *For the scheduling problem with identical machines, there exists an instance for which* $\text{PoEF}_k = \min\{n, m\}$ *when* $k = 1$.

Proof: For any $\epsilon \in (0, 1)$, consider an instance defined by m machines and by $n = m$ jobs, such that $l_1 = 1 - \epsilon$ and $l_i = 1$ for $i = 2, \ldots, m$. It is easy to check that the optimal solution $\mathbf{O}$, having makespan 1, assigns job i to machine i for each $i \in [m]$. Since $k = 1$, in any envy-free assignment, all the loads of the non-empty machines must be equal. As it is easy to check, it is not possible to equally balance the loads of all the jobs on more than 1 machine; therefore, the only envy-free assignment $\mathbf{N}$ is the one in which all the jobs are assigned to a unique machine. Hence, $\mathcal{M}(\mathbf{N}) = m - \epsilon$, and the claim follows by $n = m$ and the arbitrariness of ϵ. □

We now show that, for finite values of k, a price of k-envy-freeness equal to 1 cannot be achieved even in the setting of identical machines.

Proposition 3. *For the scheduling problem with identical machines, no value of k (possibly depending on n and m) can guarantee* $\text{PoEF}_k = 1$.

Proof: For any $\epsilon \in (0, 1)$, consider an instance defined by 2 machines and 2 jobs, such that $l_1 = 1$ and $l_2 = \epsilon$. It is easy to check that the optimal solution

O, having makespan 1, assigns job i to machine i for $i = 1, 2$. We obtain a price of k-envy-freeness equal to 1, if and only if assignment **O** is k-envy-free; in fact, assigning both jobs to a unique machine would result in a makespan strictly grater than 1. In order for **O** to be k-envy-free, it must hold that $k \geq \frac{1}{\epsilon}$. The claim follows by the arbitrariness of ϵ. □

In the next lemma, we give an important result which helps to characterize the performance of k-envy-free solutions in the case of related machines.

Lemma 1. *For a value $k \geq 1$, an instance of the scheduling problem with related machines, and an integer $2 \leq h \leq \min\left\{m, \left\lfloor \frac{L(k-1)}{kl_{max}} \right\rfloor\right\}$, there always exists a k-envy-free solution $\mathbf{N}$ using exactly h machines and such that $\mathcal{M}(\mathbf{N}) \leq \frac{L/h+l_{max}}{s_h}$, where s_h is the speed of the h-th fastest machine.*

Proof: Fix a value $k \geq 1$ and an instance of the scheduling problem with related machines. First of all, we show that, for each $2 \leq h \leq \min\left\{m, \left\lfloor \frac{L(k-1)}{kl_{max}} \right\rfloor\right\}$, Algorithm 1 described below, by exploiting a simple greedy heuristic, returns an assignment $\mathbf{N}$ using exactly h machines and such that

$$L_{min}(\mathbf{N}) \geq L_{max}(\mathbf{N}) - l_{max}, \tag{1}$$

$$L_{min}(\mathbf{N}) \geq \frac{L}{h} - l_{max}, \tag{2}$$

$$L_{min}(\mathbf{N}) \leq \frac{L}{h}. \tag{3}$$

Algorithm 1

1: **Input:** h ▷ h is the number of machines to be used
2: Renumber the machines in non-increasing order of speed
3: **for** each machine $j \in [m]$ **do**
4: $\quad N_j \leftarrow \emptyset$
5: **end for**
6: **for** each job $i \in [n]$ **do**
7: $\quad j' \leftarrow \text{argmin}_{j \in [h]} \{L(N_j)\}$
8: $\quad N_{j'} \leftarrow N_{j'} \cup \{i\}$
9: **end for**
10: **return N**

In order to show the first inequality, assume, by contradiction, that there exist two machines j, j' such that $L_j(\mathbf{N}) < L_{j'}(\mathbf{N}) - l_{max}$. Let i be the index of the last job assigned to machine j'. Since $l_i \leq l_{max}$, at the beginning of the ith iteration of the second for cycle of Algorithm 1, it must be $L(N_j) \leq L_j(\mathbf{N}) < L_{j'}(\mathbf{N}) - l_{max} \leq L_{j'}(\mathbf{N}) - l_i = L(N_{j'})$. Thus, because of line 7 of Algorithm 1, job i cannot be assigned to machine j'. This yields a contradiction, and so inequality (1) holds. Inequalities (2) and (3), in turn, come from a simple averaging argument. In fact, if there exists a machine j'

such that $L_{j'}(\mathbf{N}) < \frac{L}{h} - l_{max}$, then by (1), it holds that $L_{max}(\mathbf{N}) < \frac{L}{h}$ which implies $L = \sum_{j=1}^{m} L_j(\mathbf{N}) \leq hL_{max}(\mathbf{N}) < L$. Similarly, if for each machine j, $L_j(\mathbf{N}) > \frac{L}{h}$, then $L = \sum_{j=1}^{m} L_j(\mathbf{N}) > h\frac{L}{h} = L$. Thus, in both cases, we get a contradiction.

We now show that $\mathbf{N}$ is k-envy-free. To this aim, we just need to prove that $kL_{min}(\mathbf{N}) \geq L_{max}(\mathbf{N})$. It holds that

$$\begin{aligned} kL_{min}(\mathbf{N}) &= (k-1)L_{min}(\mathbf{N}) + L_{min}(\mathbf{N}) \\ &\geq (k-1)\left(\frac{L}{h} - l_{max}\right) + L_{min}(\mathbf{N}) \\ &= (k-1)\frac{L}{h} - (k-1)l_{max} + L_{min}(\mathbf{N}) \\ &\geq kl_{max} - (k-1)l_{max} + L_{min}(\mathbf{N}) \\ &= L_{min}(\mathbf{N}) + l_{max} \\ &\geq L_{max}(\mathbf{N}), \end{aligned}$$

where the first inequality comes from (2), the second one comes from $h \leq \frac{L(k-1)}{kl_{max}}$ and the last one comes from (1).

As a consequence, we have that, for each $2 \leq h \leq \min\left\{m, \left\lfloor \frac{L(k-1)}{kl_{max}} \right\rfloor\right\}$, there exists a k-envy-free solution $\mathbf{N}$ such that $L_{max}(\mathbf{N}) \leq \frac{L}{h} + l_{max}$, because of inequalities (1) and (3). Since $\mathbf{N}$ is such that the only machines with positive load are the h fastest ones, the claim follows. □

3. Identical Machines

In this section, we consider the scheduling problem with identical machines. For the case of $k \geq 2$, we can prove a constant upper bound on the price of k-envy freeness.

Theorem 1. *For the scheduling problem with identical machines,* $\text{PoEF}_k \leq 1 + 1/k$ *for any* $k \geq 2$.

Proof: We argue that applying Algorithm 2 to any initial assignment $\mathbf{S}$, we get a k-envy free assignment $\mathbf{N}$ with makespan at most $\mathcal{M}(\mathbf{S})(1 + 1/k)$. The claim follows by choosing as the starting assignment $\mathbf{S}$ an optimal solution $\mathbf{O}$.

Initially Algorithm 2 manipulates the starting assignment in such a way that it becomes an assignment with makespan 1 with the minimal number of non-empty machines, and such that the machines are numbered so that to a smaller index corresponds a larger or equal load. After the first phase we assume that the jobs are assigned to machines in $[\overline{m}]$.

Since machine $\overline{m}$ is the least loaded one, if $L_{\overline{m}}(\mathbf{S}) \geq 1/k$, then $\mathbf{S}$ is k-envy-free and the claim follows. On the other side, if $L_{\overline{m}}(\mathbf{S}) < 1/k$, we move all the jobs in $\mathbf{S}$ from machine $\overline{m}$ to machine $\overline{m} - 1$ obtaining a new assignment $\mathbf{N}$ which is k-envy-free. In fact, in the new assignment $\mathbf{N}$, machine $\overline{m} - 1$ gets a load larger than 1, thus becoming the most loaded machine, whereas any other

Algorithm 2

Input: assignment $\mathbf{S}$
Rescale the loads in such a way that $\mathcal{M}(\mathbf{S}) = 1$
while there exists a pair of machines (j, j') s.t. $L_j(\mathbf{S}) + L_{j'}(\mathbf{S}) \leq 1$ **do**
    $S_j \leftarrow S_j \cup S_{j'}$
    $S_{j'} \leftarrow \emptyset$
end while
Renumber the machines in non-increasing order of loads
    $L_j(\mathbf{S}) \geq L_{j+1}(\mathbf{S})$ for each $j \in [m-1]$
Let $[\overline{m}]$ be the set of machines with at least one job assigned
Create a new assignment $\mathbf{N}$ defined as follows
if $L_{\overline{m}}(\mathbf{S}) < 1/k$ **then**
    $N_j \leftarrow S_j$ for each $j < \overline{m} - 1$
    $N_{\overline{m}-1} \leftarrow S_{\overline{m}-1} \cup S_{\overline{m}}$
    $N_j \leftarrow \emptyset$ for each $j > \overline{m} - 1$
else
    $N_j \leftarrow S_j$ for each $j \in [m]$
end if
return N

machine has a load smaller than 1. Machine $\overline{m} - 1$ does not envy any other machine, since $L_{\overline{m}-1}(\mathbf{N}) = L_{\overline{m}-1}(\mathbf{S}) + L_{\overline{m}}(\mathbf{S}) \leq 2L_{\overline{m}-1}(\mathbf{S}) \leq kL_{\overline{m}-1}(\mathbf{S}) \leq kL_j(\mathbf{N})$, for each $j \leq \overline{m} - 1$ and $k \geq 2$. Thus, we can conclude that the new assignment is k-envy-free. Finally we see that the makespan of $\mathbf{N}$ is at most $L_{\overline{m}-1}(\mathbf{N}) = L_{\overline{m}-1}(\mathbf{S}) + L_{\overline{m}}(\mathbf{S}) \leq L_{\overline{m}-1}(\mathbf{S}) + 1/k \leq 1 + 1/k = \mathcal{M}(\mathbf{S})(1 + 1/k)$. The claim follows. □

The next result shows that the above upper bound is tight for any $k \geq 2$.

Proposition 4. *For the scheduling problem with identical machines, given any $k \geq 2$, there exists an instance for which* $\text{PoEF}_k \geq 1 + 1/k - \epsilon$, *for any $\epsilon > 0$.*

Proof: Let us consider the instance with m machines and m jobs, $m - 1$ of which have load 1 and the remaining one has load $1/k - \epsilon$, with $\epsilon > 0$. It is obvious that $\mathbf{O}$ assigns a single job to each machine and $\mathcal{M}(\mathbf{O}) = 1$. However, $\mathbf{O}$ is not k-envy-free since each machine of load 1 k-envies the machine of load $1/k - \epsilon$. Thus, any k-envy-free assignment is forced to schedule at least two jobs on a same machine for a makespan of at least $1 + 1/k - \epsilon$. □

For the remaining case of $k \in (1, 2)$, the following bounds hold.

Theorem 2. *For the scheduling problem with identical machines,* $\text{PoEF}_k \leq \min\left\{\frac{2k}{k-1}, n, m\right\}$ *for any $k \in (1, 2)$.*

Proof: Fix a value $k \in (1, 2)$. Because of Proposition 1, we already have $\text{PoEF}_k \leq \min\{n, m\}$. Hence, we only need to show $\text{PoEF}_k \leq \frac{2k}{k-1}$.

Consider the assignment $\mathbf{N}$ computed by Algorithm 3 described below. We show $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{O})} \leq \frac{2k}{k-1}$.

Algorithm 3

$t \leftarrow \min\left\{m, \left\lfloor \frac{L(k-1)}{kl_{max}} \right\rfloor\right\}$
if $L < \frac{2kl_{max}}{k-1}$ **then**
    $\mathbf{N} \leftarrow$ **Algorithm 1**(1)
else
    $\mathbf{N} \leftarrow$ **Algorithm 1**(t)
end if
return N

If $L < \frac{2kl_{max}}{k-1}$, then $\mathbf{N}$ is such that only one machine is loaded and so $\mathcal{M}(\mathbf{N}) = L < \frac{2kl_{max}}{k-1}$. Since $\mathcal{M}(\mathbf{O}) \geq l_{max}$, it follows $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{O})} < \frac{2k}{k-1}$.

If $L \geq \frac{2kl_{max}}{k-1}$, then, by Lemma 1, $\mathbf{N}$ is a k-envy-free solution such that $\mathcal{M}(\mathbf{N}) \leq \frac{L}{t} + l_{max}$. We distinguish between two cases:

- If $t = m$, since $\mathcal{M}(\mathbf{O}) \geq l_{max}$ and $\mathcal{M}(\mathbf{O}) \geq \frac{L}{m}$ (otherwise the sum of the loads over all machines would be less than L), we obtain $\mathcal{M}(\mathbf{N}) \leq \frac{L}{t} + l_{max} \leq 2\mathcal{M}(\mathbf{O})$. Thus, $\text{PoEF}_k \leq 2 \leq \frac{2k}{k-1}$.
- If $t < m$, by the definition of t, it holds that

$$L = \frac{\frac{L(k-1)}{kl_{max}} kl_{max}}{k-1} \leq \frac{\left(\left\lfloor \frac{L(k-1)}{kl_{max}} \right\rfloor + 1\right) kl_{max}}{k-1} = \frac{(t+1)kl_{max}}{k-1}.$$

  Hence, $\mathcal{M}(\mathbf{O}) \geq l_{max}$ implies $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{O})} \leq \frac{(t+1)k}{t(k-1)} + 1$, with $t \geq 2$ because $L \geq \frac{2kl_{max}}{k-1}$ and $m \geq 2$. The value $\frac{(t+1)k}{t(k-1)}$ is decreasing in t, so it is maximized for $t = 2$, for which we have $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{O})} \leq \frac{5k-2}{2(k-1)}$. Thus, we can conclude that $\text{PoEF}_k \leq \frac{5k-2}{2(k-1)} \leq \frac{2k}{k-1}$, because $2k \geq \frac{5k-2}{2}$ for each $k \leq 2$.

□

The following lemma helps us to construct a family of lower bounding instances for values of k falling into certain subintervals of $(1, 2)$.

Lemma 2. *For any $k \in (1,2)$, $\delta \in \left(0, \frac{2+k-k^2}{2k}\right)$ and integer p such that $p+1$ is prime and $p < \frac{2}{k-1}\left(\frac{1}{k} - \delta\right)$, there exists an instance $I_{k,p,\delta}$ for which* $\text{PoEF}_k = \min\left\{p + \frac{1}{k} - \delta, n, m\right\}$.

Proof: For fixed $k \in (1,2)$, $\delta \in \left(0, \frac{2+k-k^2}{2k}\right)$ and integer p such that $p+1$ is prime and $p < \frac{2}{k-1}\left(\frac{1}{k} - \delta\right)$, the instance $I_{k,p,\delta}$ is defined by $p+1$ machines and $p+1$ jobs, such that one job has load $\frac{1}{k} - \delta$ and all the remaining ones have load 1. Note that $k < 2$ and $\delta < \frac{2+k-k^2}{2k}$ implies $\frac{2}{k-1}\left(\frac{1}{k} - \delta\right) > 1$, so that, for any $k \in (1,2)$ and $\delta \in \left(0, \frac{2+k-k^2}{2k}\right)$, at least one instance $I_{k,p,\delta}$ always exists.

We show that scheduling all jobs to the same machine is the only k-envy-free assignment for $I_{k,p,\delta}$. This is trivially true for the case in which $p = 1$, so, in the sequel of the proof, we assume $p \geq 2$. We claim that each assignment $\mathbf{N}$ using exactly $h \geq 2$ machines satisfies

$$\mathcal{M}(\mathbf{N}) \geq L_{min}(\mathbf{N}) + \frac{1}{k} - \delta \tag{4}$$

and

$$L_{min}(\mathbf{N}) \leq \left\lfloor \frac{p}{h} \right\rfloor. \tag{5}$$

The first inequality comes from the fact that, being $p + 1 \geq 3$ a prime number, there must be two machines receiving a different number of jobs. For the second one, assume $L_{j'}(\mathbf{N}) := L_{min}(\mathbf{N}) > \left\lfloor \frac{p}{h} \right\rfloor$. If $N_{j'}$ contains only jobs of load 1, then $L_{min}(\mathbf{N}) > \frac{p}{h}$ which implies $\mathcal{M}(\mathbf{N}) > \frac{p}{h} + \frac{1}{k} - \delta$ because of (4) and we get $L = \sum_{j=1}^{m} L_j(\mathbf{N}) > h\frac{p}{h} + \frac{1}{k} - \delta = L$, yielding a contradiction. If $N_{j'}$ contains $x > 0$ jobs including the one of load $\frac{1}{k} - \delta$, we can only claim that $L_{min}(\mathbf{N}) > \frac{p}{h} - 1 + \frac{1}{k} - \delta$. Anyway, in this case, the machine of completion time $\mathcal{M}(\mathbf{N})$ is assigned at least $x+1$ jobs of load 1, which implies $\mathcal{M}(\mathbf{N}) \geq L_{min}(\mathbf{N}) - \frac{1}{k} + \delta + 2 > \frac{p}{h} + 1$, and each of the remaining $h-2$ machines is assigned at least x jobs of load 1 for a total load equal to $(h-2)\left(L_{min}(\mathbf{N}) - \frac{1}{k} + \delta + 1\right) > (h-2)\frac{p}{h}$. We get $L = \sum_{j=1}^{m} L_j(\mathbf{N}) > (h-2)\frac{p}{h} + \frac{p}{h} + 1 + \frac{p}{h} - 1 + \frac{1}{k} - \delta = L$, yielding a contradiction.

Observe now that, for each $h \geq 2$, $\left\lfloor \frac{p}{h} \right\rfloor \leq \frac{p}{h} \leq \frac{p}{2}$. Thus, for each $h \geq 2$, it holds that

$$\begin{aligned}
kL_{min}(\mathbf{N}) &= (k-1)L_{min}(\mathbf{N}) + L_{min}(\mathbf{N}) \\
&\leq (k-1)\left\lfloor \frac{p}{h} \right\rfloor + L_{min}(\mathbf{N}) \\
&\leq (k-1)\frac{p}{2} + L_{min}(\mathbf{N}) \\
&< L_{min}(\mathbf{N}) + \frac{1}{k} - \delta \\
&\leq \mathcal{M}(\mathbf{N}),
\end{aligned}$$

where the first inequality comes from (5), the second one comes from $\left\lfloor \frac{p}{h} \right\rfloor \leq \frac{p}{2}$ for each $h \geq 2$, the third one comes from $p < \frac{2}{k(k-1)} - \frac{2\delta}{k-1}$ and the last one comes from (4). This shows that scheduling all jobs to the same machine is the only k-envy-free assignment for $I_{k,p,\delta}$ and, since $\mathcal{M}(\mathbf{O}) = 1$, we get $\text{PoEF}_k = p + \frac{1}{k} - \delta$. The claim then follows as $n = m = p + 1 > p + \frac{1}{k} - \delta$. □

We can exploit the above lemma to construct lower bounds as follows. For each prime number $p + 1$, we determine the values of k such that $\frac{2}{k(k-1)} > p$. This tells us for which values of k there exists an instance $I_{k,p,\delta}$ yielding $\text{PoEF}_k = p + \frac{1}{k} - \delta$, for a sufficiently small $\delta > 0$. The results obtained for some of the first prime numbers, taking the limit for δ going to zero, are listed in Figure 1.

Lower Bound	Holds For	With Prime Number
$\frac{k+1}{k}$	$k \in \left[\frac{1+\sqrt{5}}{2}, 2\right)$	2
$\frac{2k+1}{k}$	$k \in \left[\frac{1+\sqrt{3}}{2}, \frac{1+\sqrt{5}}{2}\right)$	3
$\frac{4k+1}{k}$	$k \in \left[\frac{3+\sqrt{21}}{6}, \frac{1+\sqrt{3}}{2}\right)$	5
$\frac{6k+1}{k}$	$k \in \left[\frac{5+3\sqrt{5}}{10}, \frac{3+\sqrt{21}}{6}\right)$	7
$\frac{10k+1}{k}$	$k \in \left[\frac{3+\sqrt{15}}{6}, \frac{5+3\sqrt{5}}{10}\right)$	11
$\frac{12k+1}{k}$	$k \in \left[\frac{2+\sqrt{6}}{4}, \frac{3+\sqrt{15}}{6}\right)$	13
$\frac{16k+1}{k}$	$k \in \left[\frac{11+\sqrt{165}}{22}, \frac{2+\sqrt{6}}{4}\right)$	17
$\frac{22k+1}{k}$	$k \in \left[\frac{7+3\sqrt{7}}{14}, \frac{11+\sqrt{165}}{22}\right)$	23
$\frac{28k+1}{k}$	$k \in \left[\frac{15+\sqrt{285}}{30}, \frac{7+3\sqrt{7}}{14}\right)$	29
$\frac{30k+1}{k}$	$k \in \left[\frac{3+\sqrt{11}}{6}, \frac{15+\sqrt{285}}{30}\right)$	31
$\frac{4+7k-k^2}{6k(k-1)}$	$k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$	≥ 37

Figure 1: Lower bounds on PoEF_k for values of $k \in (1, 2)$.

In order to obtain a general lower bound, holding for $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$, we exploit the following theorem on prime numbers.

Theorem 3 (Nagura [22]). *For any integer $n \geq 25$, there exists at least one prime number p such that $n < p < \frac{6n}{5}$.*

Theorem 4. *For any $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$ and $\delta \in \left(0, (k-1)^2\right]$, there exists an instance $I_{k,\delta}$ such that* $\text{PoEF}_k \geq \min\left\{\frac{4+7k-k^2}{6k(k-1)} - \frac{(3k+2)\delta}{3(k-1)}, n, m\right\}$.

Proof: First of all, observe that the function $f(k) = \frac{10+k-k^2}{6k(k-1)} - \frac{5(k-1)}{3}$ is decreasing in the interval $\left(1, \frac{3+\sqrt{11}}{6}\right)$, so that its minimum in this interval is $f\left(\frac{3+\sqrt{11}}{6}\right) = \frac{92}{3} - \frac{5\sqrt{11}}{18} > 29$. For fixed $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$ and $\delta \in \left(0, (k-1)^2\right]$, let x be the highest integer such that $x < \frac{5(2-k+k^2)}{6k(k-1)} - \frac{5\delta}{3(k-1)}$. Note that $x \geq \frac{5(2-k+k^2)}{6k(k-1)} - \frac{5\delta}{3(k-1)} - 1 = \frac{10+k-k^2}{6k(k-1)} - \frac{5\delta}{3(k-1)}$. Let $p+1$ be a prime number such that $x < p+1 < \frac{6x}{5}$. One such a number always exists since $x \geq \frac{10+k-k^2}{6k(k-1)} - \frac{5\delta}{3(k-1)} \geq \frac{10+k-k^2}{6k(k-1)} - \frac{5(k-1)}{3} \geq 25$ assures the application of Nagura's Theorem. Note also that $p+1 < \frac{6x}{5} < \frac{2-k+k^2}{k(k-1)} - \frac{2\delta}{k-1}$ implies $p < \frac{2}{k(k-1)} - \frac{2\delta}{k-1}$, so that Lemma 2 can be applied since $(k-1)^2 < \frac{2+k-k^2}{2k}$ for each $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$. By Lemma 2, we obtain that $\text{PoEF}_k = \min\left\{p + \frac{1}{k} - \delta, n, m\right\}$, where $p \geq x$. Since $x \geq \frac{10+k-k^2}{6k(k-1)} - \frac{5\delta}{3(k-1)}$,

we get $p + \frac{1}{k} - \delta \geq \frac{10+k-k^2}{6k(k-1)} - \frac{5\delta}{3(k-1)} + \frac{1}{k} - \delta = \frac{4+7k-k^2}{6k(k-1)} - \frac{(3k+2)\delta}{3(k-1)}$, which yields $\text{PoEF}_k = \min\left\{\frac{4+7k-k^2}{6k(k-1)} - \frac{(3k+2)\delta}{3(k-1)}, n, m\right\}$. □

Theorem 5. *For the scheduling problem with identical machines, given any $k \in (1,2)$, there exists an instance for which* $\text{PoEF}_k = \Omega\left(\min\left\{\frac{2k}{k-1}, n, m\right\}\right)$.

Proof: By theorem 4, for any $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$, there exists an instance $I_{k,\delta}$ such that $\text{PoEF}_k = \Omega\left(\min\left\{\frac{2k}{k-1}, n, m\right\}\right)$. For any $k \in \left[\frac{3+\sqrt{11}}{6}, 2\right)$, the claim trivially follows because in this case $\frac{2k}{k-1} = \Theta(1)$. □

4. Related Machines

In this section, we consider the scheduling problem with related machines.

Theorem 6. *For the scheduling problem with related machines,* $\text{PoEF}_k \leq 2 + \max\left\{1, \sqrt{\frac{m}{k}}\right\}$ *for any $k \geq 2$.*

Proof: Given an instance of the scheduling problem with related machines, consider any assignment $\mathbf{S}$. Let us normalize the machine speeds and the loads of the jobs so that the fastest machine has speed 1 and the makespan of solution $\mathbf{S}$ is 1, i.e., $\mathcal{M}(\mathbf{S}) = 1$. Let us rename the machines in such a way that $s_j \geq s_{j+1}$ for any $j = 1, \ldots, m-1$; notice that $L_1(\mathbf{S}) \leq 1$ and we can assume that $L_j(\mathbf{S}) \geq L_{j+1}(\mathbf{S})$ for any $j = 1, \ldots, m-1$ (otherwise by swapping S_j and S_{j+1} a solution having equal or better makespan could be obtained).

Denote by $M_1 = \{1, \ldots, |M_1|\}$ the set of machines having load at least $1/k$ in $\mathbf{S}$, i.e., $L_j(\mathbf{S}) \geq 1/k$ for any $j \in M_1$, and by M_2 the set of the remaining machines. Note that it also holds that $s_j \geq 1/k$ for any $j \in M_1$. Moreover, it is easy to check that no pair (j, j') of machines in M_1 is such that j k-envies j'. In the following we build a new allocation $\mathbf{N}$ starting from allocation $\mathbf{S}$.

Let L_j^i be the load of each machine j at the moment in which the job i is considered for allocation by Algorithm 4. The new assignment $\mathbf{N}$ is obtained as described in Algorithm 4.

Assignment $\mathbf{N}$ is initially set equal to assignment $\mathbf{S}$. When lines 18–20 are executed, it means that $k \leq m$ and $\sum_{p \geq j} L_p(\mathbf{S}) > \sqrt{\frac{m}{k}}$. It follows that $L_j(\mathbf{S}) > \frac{1}{\sqrt{mk}}$ (and therefore also $s_j > \frac{1}{\sqrt{mk}}$). In this case, the only machine receiving some new jobs is machine j. Since the load of any machine in M_2 is less than $1/k$, we can gather all the jobs of machines $j+1, j+2, \ldots, j'$ of total load between $\frac{1}{k} - L_j(\mathbf{S})$ and $\frac{2}{k} - L_j(\mathbf{S})$, and add in $\mathbf{N}$ all such jobs to machine j. We obtain $C_j(N_j) = \frac{L_j(\mathbf{N})}{s_j} \leq \frac{\frac{2}{k}}{\frac{1}{\sqrt{m}\sqrt{k}}} = 2\sqrt{\frac{m}{k}}$. Notice that machine j cannot be k-envied by any other machine, and (since $k \geq 2$) cannot k-envy other machines with load at least $1/k$ (all the machines in $M_1 \cup M'$ have load at least $1/k$ in assignment $\mathbf{N}$).

Algorithm 4

1: **Input:** assignment $\mathbf{S}$
2: $\mathbf{N} \leftarrow \mathbf{S}$
3: $M' \leftarrow \emptyset$
4: $j' \leftarrow |M_1|$
5: **while** $j' < m$ **do**
6: $\quad j \leftarrow j' + 1$
7: $\quad$ **if** $k > m$ **or** $\sum_{p \geq j} L_p(\mathbf{S}) \leq \sqrt{\frac{m}{k}}$ **then**
8: $\quad\quad$ **for** each job in $i \in \bigcup_{p \geq j} S_p$ **do**
9: $\quad\quad\quad$ Let $j'' \in M_1 \cup M'$ be the machine with the current smallest load
10: $\quad\quad\quad$ **if** $L_1^i + l_i \leq kL_{j''}^i$ **then**
11: $\quad\quad\quad\quad N_1 \leftarrow N_1 \cup \{i\}$ ▷ Assign job i to machine 1
12: $\quad\quad\quad$ **else**
13: $\quad\quad\quad\quad N_{j''} \leftarrow N_{j''} \cup \{i\}$ ▷ Assign job i to machine j''
14: $\quad\quad\quad$ **end if**
15: $\quad\quad$ **end for**
16: $\quad\quad j' \leftarrow m$
17: $\quad$ **else**
18: $\quad\quad M' \leftarrow M' \cup \{j\}$
19: $\quad\quad$ Let j' such that $\frac{1}{k} - L_j(\mathbf{S}) \leq \sum_{p=j+1}^{j'} L_p(\mathbf{S}) \leq \frac{2}{k} - L_j(\mathbf{S})$
20: $\quad\quad N_j \leftarrow \bigcup_{p=j}^{j'} S_p$
21: $\quad$ **end if**
22: **end while**
23: **return N**

Note that lines 8–16 can be executed only once. When they are executed, it means that $k > m$ or $\sum_{p \geq j} L_p(\mathbf{S}) \leq \sqrt{\frac{m}{k}}$. When $k > m$, the load in $\mathbf{S}$ of each machine in M_2 is at most $1/m$ and the total load of all machines in M_2 is at most 1. Therefore, in any case, the total load of all machines $p \geq j$ is at most $\max\left\{1, \frac{\sqrt{m}}{\sqrt{k}}\right\}$. Notice that, since *(i)* $k \geq 2$, *(ii)* the load of each machine in M_1 is at least $1/k$ already in allocation $\mathbf{S}$ and *(iii)* the load of each job to be assigned is at most $1/k$, it is always possible to maintain k-envy-free an allocation by assigning each job either to machine 1 or (in case the assignment to machine 1 would result in a state non being k-envy-free) to the machine of $M_1 \cup M'$ having the smallest load at that moment. In fact, consider any job i belonging in $\mathbf{S}$ to some machine $p \geq j$, and, for any $j \in M_1 \cup M'$, let L_j^i be the load of machine j at the moment in which the job i is considered for assignation by Algorithm 4. Assigning job i to machine j'' results in a k-envy-free state because $L_{j''}^i + l_i \leq kL_{j''}^i$ as conditions *(i)*, *(ii)* and *(iii)* hold.

Let us now compute the makespan of assignment $\mathbf{N}$, by considering only the machines receiving some new jobs in lines 8–16 of Algorithm 4:
The total load added to machine 1 is at most $\max\left\{1, \frac{\sqrt{m}}{\sqrt{k}}\right\}$ and therefore the total load $C_1(N_1)$ of machine 1 at the end of the process is at most

$1 + \max\left\{1, \frac{\sqrt{m}}{\sqrt{k}}\right\}$.
For any machine $j \in M_1 \setminus \{1\}$, let $last(j)$ be the last job assigned to machine j and $\ell_j = l_{last(j)}$ its load. Since $last(j)$ has not been assigned to machine 1, it must hold that $L_1^{last(j)} + \ell_j > kL_{j'}^{last(j)}$ for some $j' \in M_1 \setminus \{1\}$. In particular, $L_1^{last(j)} + \ell_j > kL_j^{last(j)}$ because $last(j)$ has been assigned to the machine with minimum load at that moment. Since the total load that can be given to machine 1 is at most $1 + \max\left\{1, \sqrt{\frac{m}{k}}\right\}$, it follows that $L_j^{last(j)} < \frac{L_1^{last(j)} + \ell_j}{k} \leq \frac{1+\max\left\{1, \sqrt{\frac{m}{k}}\right\}}{k}$. Finally, since $last(j)$ is the last job assigned to machine j, $L_j(\mathbf{N}) = L_j^{last(j)} + \ell_j \leq L_j^{last(j)} + 1/k$ and the completion time of machine j is $C_j(N_j) = \frac{L_j(\mathbf{N})}{s_j} \leq \frac{L_j^{last(j)} + 1/k}{1/k} \leq k\left(\frac{1+\max\left\{1, \sqrt{\frac{m}{k}}\right\}}{k} + \frac{1}{k}\right) = 2 + \max\left\{1, \sqrt{\frac{m}{k}}\right\}$.

The claim follows by choosing $\mathbf{O} = \mathbf{S}$. □

For the case of $k \in (1, 2)$, the following upper bound holds.

Theorem 7. *For the scheduling problem with related machines,* $\text{PoEF}_k \leq \min\left\{n, m, 2k\sqrt{\frac{m}{k-1}}\right\}$ *for any* $k \in (1, 2)$.

Proof: Fix a value $k \in (1, 2)$ and an assignment $\mathbf{S}$. Because of Proposition 1, $\text{PoEF}_k \leq \min\{n, m\}$; hence, we only need to show $\text{PoEF}_k \leq 2k\sqrt{\frac{m}{k-1}}$ for all the cases in which $2k\sqrt{\frac{m}{k-1}} < \min\{n, m\}$. In particular, $2k\sqrt{\frac{m}{k-1}} < m$ implies that $m > \frac{4k^2}{k-1}$.

Consider the assignment $\mathbf{N}$ computed by Algorithm 5 described below.

Algorithm 5

1: **Input:** assignment $\mathbf{S}$
2: scale the machine speeds so that the fastest machine has speed 1
3: scale the loads of the jobs so that $\mathcal{M}(\mathbf{S}) = 1$
4: $t \leftarrow \min\left\{m, \left\lfloor \frac{L(k-1)}{k} \right\rfloor\right\}$
5: let s_t be the speed of the t-th fastest machine
6: **if** $L < \frac{2k}{k-1}$ **or** $L < \sqrt{m}$ **or** $\left(t < \sqrt{\frac{m}{k-1}}$ **and** $s_t < \frac{1}{\sqrt{m(k-1)}}\right)$ **then**
7: $\quad \mathbf{N} \leftarrow$ **Algorithm 1**(1)
8: **else**
9: $\quad \mathbf{N} \leftarrow$ **Algorithm 1**(t)
10: **end if**
11: **return N**

Let us order the machines in non-increasing speed so that $s_1 = 1$ is the speed of the fastest one. Note that, after the scalings performed at lines 1 and 2, it holds that $L_j(\mathbf{S}) \leq s_j$ for each $j \in [m]$ and $l_{max} \leq 1$.

Let us first consider the cases in which $\mathbf{N}$ = Algorithm 1(1), that is, only the fastest machine is loaded in $\mathbf{N}$ for a completion time equal to $\frac{L}{s_1} = L$. If $L < \frac{2k}{k-1} \leq \frac{2k}{k-1}$, since $\mathcal{M}(\mathbf{S}) = 1$, it follows $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{S})} < \frac{2k}{k-1}$. Note that $\frac{2k}{k-1} \leq 2k\sqrt{\frac{m}{k-1}}$ when $m > \frac{4k^2}{k-1}$. Similarly, if $L < \sqrt{m}$, it follows $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{S})} < \sqrt{m}$. Finally, if $t < \sqrt{\frac{m}{k-1}}$ and $s_t < \frac{1}{\sqrt{m(k-1)}}$, since $L_j(\mathbf{S}) \leq s_j$ for each $j \in [m]$, it holds that

$$\begin{aligned} L &= \sum_{j\in[m]} L_j(\mathbf{S}) \leq \sum_{j\in[m]} s_j \\ &\leq \sum_{j\in[t]} s_j + ms_t < ts_1 + \sqrt{\frac{m}{k-1}} \\ &= t + \sqrt{\frac{m}{k-1}} < 2\sqrt{\frac{m}{k-1}}. \end{aligned}$$

Hence, it follows $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{S})} < 2\sqrt{\frac{m}{k-1}}$.

For the cases in which $\mathbf{N}$ = Algorithm 1(t), note that $t \geq 2$ since $L \geq \frac{2k}{k-1}$ and $m \geq 2$; moreover, it holds that $L \geq \sqrt{m}$.

Let us first consider the case in which $s_t \geq \frac{1}{\sqrt{m(k-1)}}$. By Lemma 1, $\mathbf{N}$ is a k-envy-free solution such that $\mathcal{M}(\mathbf{N}) \leq \frac{L/t+l_{max}}{s_t} \leq \frac{L/t+1}{s_t}$. Note that, as already shown in the proof of Theorem 2, by the definition of t, it holds that $L < \frac{(t+1)kl_{max}}{k-1}$. Hence, it follows $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{S})} \leq \frac{1}{s_t}\left(\frac{(t+1)k}{t(k-1)} + 1\right) \leq \sqrt{m(k-1)}\left(\frac{(t+1)k}{t(k-1)} + 1\right)$. Again, the value $\frac{(t+1)k}{t(k-1)}$ is decreasing in t, so it is maximized for $t = 2$, for which we have $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{S})} \leq \frac{5k-2}{2}\sqrt{\frac{m}{k-1}} < 2k\sqrt{\frac{m}{k-1}}$ for each $k \leq 2$.

It remains to consider the case in which $t \geq \sqrt{\frac{m}{k-1}}$. It holds that $\sum_{j\in[t]} L_j(\mathbf{S}) \leq s_1 t = t \leq \left\lfloor \frac{L(k-1)}{k} \right\rfloor \leq \frac{L(k-1)}{k}$. This implies that

$$\sum_{j=t+1}^{m} L_j(\mathbf{S}) = L - \sum_{j\in[t]} L_j(\mathbf{S}) \geq L - \frac{L(k-1)}{k} = \frac{L}{k}. \tag{6}$$

Since the machines are ordered in non-increasing speed, it also holds that

$$\sum_{j=t+1}^{m} L_j(\mathbf{S}) \leq ms_t. \tag{7}$$

Inequalities 6 and 7 together imply $s_t \geq \frac{L}{mk}$. Hence, $\frac{\mathcal{M}(\mathbf{N})}{\mathcal{M}(\mathbf{S})} \leq \frac{(L/t+1)(mk)}{L} = \frac{mk}{t} + \frac{mk}{L} \leq k\sqrt{m(k-1)} + k\sqrt{m} \leq 2k\sqrt{m(k-1)}$. Note that $2k\sqrt{m(k-1)} \leq 2k\sqrt{\frac{m}{k-1}}$ for each $k \in (1, 2)$. The claim follows by choosing an optimal solution $\mathbf{O}$ as the input assignment $\mathbf{S}$. □

We now show that the two upper bounds proved in Theorems 6 and 7 are asymptotically tight.

Proposition 5. *For the scheduling problem with related machines, given any $k \geq 1$, there exists an instance for which* $\text{PoEF}_k \geq \max\left\{1, \sqrt{\frac{m}{k}}\right\}$. *Moreover, for any $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$, there exists an instance for which* $\text{PoEF}_k = \Omega\left(\sqrt{\frac{m}{k-1}}\right)$.

Proof: For $k \geq 1$, consider an instance defined by $m > k$ machines and $n = m$ jobs, such that $l_1 = s_1 = 1$ and $l_i = s_i = \frac{1}{\sqrt{mk}}$ for $i = 2, \ldots, m$. It is easy to check that the optimal solution $\mathbf{O}$, having makespan 1, assigns job i to machine i for each $i \in [m]$. First, notice that, by assigning job 1 to a machine different from machine 1, the makespan of the obtained solution would be at least $1/s_2 = \sqrt{mk} \geq \sqrt{\frac{m}{k}}$. It remains to show that any k-envy-free solution $\mathbf{N}$ assigning job 1 to machine 1 is such that $\mathcal{M}(\mathbf{N}) \geq \sqrt{\frac{m}{k}}$.

Two possible cases may occur: *(i)* all the jobs are assigned to machine 1, or *(ii)* there exists a machine other than machine 1 which receives at least one of the jobs of load $\frac{1}{\sqrt{mk}}$.

In case *(i)*, the total load of machine 1 in assignment $\mathbf{N}$ would be $L_1(\mathbf{N}) = 1 + \frac{m-1}{\sqrt{mk}} > \sqrt{\frac{m}{k}}$ and, since $C_1(N_1) = L_1(\mathbf{N})$, the claim follows. In case *(ii)*, the load of any machine $j > 1$ has to be at least $1/k$ (otherwise machine 1 would k-envy machine j) and therefore $C_j(N_j) = \frac{L_j(\mathbf{N})}{s_j} \geq \frac{1/k}{1/\sqrt{mk}} = \sqrt{\frac{m}{k}}$.

For the case of $k \in \left(1, \frac{3+\sqrt{11}}{6}\right)$, the instance with identical machines defined in the proof of Theorem 4 has $\Theta\left(\frac{1}{k-1}\right)$ machines and yields $\text{PoEF}_k = \Omega\left(\frac{1}{k-1}\right)$. Thus, since $\sqrt{\frac{m}{k-1}} = \Theta\left(\frac{1}{k-1}\right)$, the claim follows. □

Note that, for each $k \in \left[\frac{3+\sqrt{11}}{6}, 2\right)$, it holds that $\text{PoEF}_k \leq 2k\sqrt{\frac{m}{k-1}} = O(\sqrt{m})$ by Theorem 7, while, by Proposition 5, we have $\text{PoEF}_k \geq \max\left\{1, \sqrt{\frac{m}{k}}\right\} = \Omega(\sqrt{m})$. This shows that all the bounds on the PoEF_k presented in this section are asymptotically tight.

5. Unrelated Machines

In this section, we consider the scheduling problem with unrelated machines. In this case we are able to give an exact characterization of the price of k-envy-freeness as witnessed by the upper and lower bounds given in the following.

Theorem 8. *For the scheduling problem with unrelated machines,* $\text{PoEF}_k \leq \left(1 + \frac{1}{k}\right)^{\min\{n,m\}-1}$ *for any $k \geq 1$.*

Proof: Let $\mathbf{S}$ be the any assignment. We can construct a k-envy free assignment by applying Algorithm 6. Starting from any assignment $\mathbf{S}$, it iteratively moves the jobs from a machine j' to j whenever j k-envies j'. It is obvious that, since the number of non-empty machines in $\mathbf{S}$ is at most $\min\{n, m\}$, the

Algorithm 6

Input: assignment $\mathbf{S}$
$\mathbf{N} \leftarrow \mathbf{S}$
while there exists a pair of machines (j, j') s.t. j k-envies j' **do**
    $N_j \leftarrow N_j \cup N_{j'}$
    $N_{j'} \leftarrow \emptyset$
end while
return N

number of iterations is at most $\min\{n, m\} - 1$. Moreover, it is easy to see that at each iteration, the current processing time of machine j, as a consequence of the movement of the jobs from j' to j, increases by a factor of at most $(1+1/k)$. It results that the makespan $\mathcal{M}(\mathbf{N})$ of the final assignment obtained from this procedure is at most $\mathcal{M}(\mathbf{S})\left(1+\frac{1}{k}\right)^{\min\{n,m\}-1}$. The claim follows by choosing an optimal solution $\mathbf{O}$ as the starting assignment $\mathbf{S}$. □

Proposition 6. *For the scheduling problem with unrelated machines, given any $k \geq 1$ and $\epsilon > 0$, there exists an instance for which* $\text{PoEF}_k = \left(1+\frac{1}{k+\epsilon}\right)^{\min\{n,m\}-1}$.

Proof: Let us consider the instance with m machines and $n = m$ jobs whose processing times are defined as follows: $p_{ii} = 1$ for each $i \in [n]$, $p_{i1} = \frac{1}{k+\epsilon}(1 + \frac{1}{k+\epsilon})^{i-2}$ for every $i \in \{2, \ldots, n\}$. All the remaining processing times are set to infinity. Alternatively, we can express p_{i1} as $\frac{1}{k+\epsilon}\sum_{q=1}^{i-1} p_{q1}$ for every $i \in \{2, \ldots, n\}$. In fact, $\sum_{q=1}^{i-1} p_{q1} = 1 + \frac{1}{k+\epsilon}\sum_{q=2}^{i-1}(1+\frac{1}{k+\epsilon})^{q-2} = 1 + \frac{1}{k+\epsilon}\sum_{r=0}^{i-3}(1 + \frac{1}{k+\epsilon})^r = 1 + (\frac{1}{k+\epsilon})\frac{1-(1+\frac{1}{k+\epsilon})^{i-2}}{1-(1+\frac{1}{k+\epsilon})} = (1+\frac{1}{k+\epsilon})^{i-2}$.

The optimal assignment $\mathbf{O}$ is the one in which each job i is assigned to machine i; thus $\mathcal{M}(\mathbf{O}) = 1$. It is easy to see that the assignment $\mathbf{N}$ in which all jobs are scheduled on machine 1 is a k-envy-free assignment with $\mathcal{M}(\mathbf{N}) = \sum_{q=1}^{n} p_{q1} = (1 + \frac{1}{k+\epsilon})^{n-1}$, which is equal to $\left(1+\frac{1}{k+\epsilon}\right)^{\min\{n,m\}-1}$, given that $n = m$. It remains to show that N is the best k-envy-free assignment. First notice that every solution in which a job i is assigned to a machine j different from 1 and i has infinite makespan. Thus, let us consider all the solutions in which each job i is assigned either to machine 1 or i. Let Q be the proper subset of jobs allocated to machine 1 and q be the job with he smallest index not allocated to machine 1 but to machine q. The completion time of machine 1 is at least $\sum_{r=1}^{q-1} p_{r1} = (1 + \frac{1}{k+\epsilon})^{q-2}$. Since $p_{q1} = \frac{1}{k+\epsilon}(1 + \frac{1}{k+\epsilon})^{q-2}$, it turns out that machine 1 k-envies machine q and the claim follows. □

6. Complexity

An important feature of the proofs we used to upper bound the PoEF_k in the various cases is that they rely on polynomial time algorithms constructing

k-envy-free assignments of reasonable low makespan. In particular, for identical machines with $k \in (1, 2)$, the algorithm used in the proof of Theorem 2 does not require any information to be executed; hence, it indeed constructs a k-envy-free assignment whose performance guarantee coincides with the upper bound on the PoEF_k.

For all the other cases, given an input solution $\mathbf{S}$, all the designed algorithms rearrange the allocations defined by $\mathbf{S}$ so as to obtain in polynomial time a k-envy-free assignment $\mathbf{N}$ such that $\mathcal{M}(\mathbf{N}) \leq \text{PoEF}_k \cdot \mathcal{M}(\mathbf{S})$. This means that, when given as input a solution $\mathbf{S}$ such that $\mathcal{M}(\mathbf{S}) \leq \alpha \cdot \mathcal{M}(\mathbf{O})$, each algorithm computes in polynomial time a k-envy-free assignment $\mathbf{N}$ such that $\mathcal{M}(\mathbf{N}) \leq \alpha \cdot \text{PoEF}_k \cdot \mathcal{M}(\mathbf{O})$. By recalling that there exists a PTAS for the scheduling problem with related and identical machines and a 2-approximation algorithm for the case of unrelated ones and by the fact that our upper bounds of PoEF_k are tight or asymptotically tight, it follows that we are able to compute in polynomial time k-envy-free assignments of best possible quality, when dealing with related and identical machines, and of at least half the best possible quality, when dealing with unrelated ones.

7. Extensions

In this section, we focus on two possible extensions of the original model, namely the restricted scheduling case and the scenario in which the goal is that of minimizing the sum of completion times of all machines.

It is worth noticing that the remarks of Section 6 about the complexity of computing envy-free solutions also hold for these extensions.

7.1. Restricted Scheduling

In this subsection, we focus on the case in which a job cannot be assigned to every machine: for any job i there is a set $M_i \subseteq \{1, \ldots, m\}$ containing the machines being admissible for job i. We have to clarify the definition of k-envy-freeness in this setting: an assignment $\mathbf{N}$ is k-envy-free if no machine k-envies any other one, where machine j k-envies machine j' if $C_j(N_j) > kC_j(N_{j'})$ and for each job $i \in N_{j'}$, $j \in M_i$. It can be easily verified that also in the case of restricted scheduling an envy–free solution always exists. In fact, starting from any feasible assignment, an envy–free solution can be obtained as follows: while there exist two machines j and j' such that j k-envies j', assign to machine j also all the jobs of machine j'.

As already remarked in the introduction, the setting of unrelated machines studied in Section 5 includes the case of restricted (unrelated) machines, as it is possible to assign a very large value to p_{ij} whenever machine $j \notin M_i$, so that neither an optimal solution, nor a k-envy-free one minimizing the makespan can assign a job to a machine not being admissible for it. Therefore, for the restricted case, it remains to analyze the related and identical settings. In the related case, for which the upper bound provided in Theorem 8 clearly holds, it is possible to modify the instance exploited in Proposition 6 so that it becomes a restricted instance of the related setting, and the following theorem holds.

Proposition 7. *For the restricted scheduling problem with related machines, given any $k \geq 1$ and $\epsilon > 0$, there exists an instance for which* $\text{PoEF}_k = \left(1 + \frac{1}{k+\epsilon}\right)^{\min\{n,m\}-1}$.

Proof: Let us consider the instance with n jobs and $m = n$ machines defined as follows: for $l_1 = s_1 = 1$ and $l_i = s_i = \frac{1}{k+\epsilon}\left(1 + \frac{1}{k+\epsilon}\right)^{i-2}$ for each $i \in \{2, \ldots, n\}$. Moreover, $M_1 = \{1\}$ and $M_i = \{1, i\}$ for each $i \in \{2, \ldots, n\}$.

As in the proof of Proposition 6, we can express l_i as $\frac{1}{k+\epsilon}\sum_{q=1}^{i-1} l_q$ for every $i \in \{2, \ldots, n\}$. The optimal assignment $\mathbf{O}$ is the one in which each job i is assigned to machine i; thus $\mathcal{M}(\mathbf{O}) = 1$. It is easy to see that the assignment $\mathbf{N}$ in which all jobs are scheduled on machine 1 is a k-envy-free assignment with $\mathcal{M}(\mathbf{N}) = \sum_{q=1}^{n} l_q = (1 + \frac{1}{k+\epsilon})^{n-1}$, which is equal to $\left(1 + \frac{1}{k+\epsilon}\right)^{\min\{n,m\}-1}$, given that $n = m$. It remains to show that $\mathbf{N}$ is the best k-envy-free assignment. Let Q be the proper subset of jobs allocated to machine 1 and q be the job with he smallest index not allocated to machine 1 but to machine q. The total load of jobs on machine 1 is at least $\sum_{r=1}^{q-1} l_r = (1+\frac{1}{k+\epsilon})^{q-2}$. Since $l_q = \frac{1}{k+\epsilon}(1+\frac{1}{k+\epsilon})^{q-2}$, it turns out that machine 1 k-envies machine q and the claim follows. □

Finally, for the case of identical machines, a trivial upper bound equal to $\min\{n, m\}$ holds as any solution (k-envy–free or not) approximates an optimal one by at most $\min\{n, m\}$, and the following lower bound holds.

Proposition 8. *For the restricted scheduling problem with identical machines, given any $k \geq 1$, there exists an instance for which* $\text{PoEF}_k = \Omega(\min\{n, m\})$.

Proof: Fix an $\epsilon > 0$. Let α be the smallest integer such that $\frac{1}{k+\epsilon}\left(1 + \frac{1}{k+\epsilon}\right)^{\alpha-2} > 1$. Let us consider the instance with $n = 2\alpha$ jobs and $m = n$ machines defined as follows: for $l_1 = 1$ and $l_i = \min\left\{1, \frac{1}{k+\epsilon}\left(1 + \frac{1}{k+\epsilon}\right)^{i-2}\right\}$ for each $i \in \{2, \ldots, n\}$. Notice that, by the choice of n, at least $n/2$ jobs have load 1. Moreover, $M_1 = \{1\}$ and $M_i = \{1, i\}$ for each $i \in \{2, \ldots, n\}$.

The optimal assignment $\mathbf{O}$ is the one in which each job i is assigned to machine i; thus $\mathcal{M}(\mathbf{O}) = 1$. It is easy to see that the assignment $\mathbf{N}$ in which all jobs are scheduled on machine 1 is a k-envy-free assignment with $\mathcal{M}(\mathbf{N}) \geq n/2$. It remains to show that $\mathbf{N}$ is the best k-envy-free assignment. Let Q be the proper subset of jobs allocated to machine 1 and q be the job with he smallest index not allocated to machine 1 but to machine q.

If $q < \alpha$, the total load of jobs on machine 1 is at least $\sum_{r=1}^{q-1} l_r = (1 + \frac{1}{k+\epsilon})^{q-2}$. Since $l_q = \frac{1}{k+\epsilon}(1 + \frac{1}{k+\epsilon})^{q-2}$, it turns out that machine 1 k-envies machine q and the claim follows.

If $q \geq \alpha$, the total load of jobs on machine 1 is at least $\sum_{r=1}^{q-1} l_r = (1 + \frac{1}{k+\epsilon})^{q-2} \geq k + \epsilon$. Since $l_q = 1$, again it turns out that machine 1 k-envies machine q and the claim follows. □

7.2. Sum of Machines Completion Times

In this subsection, we extend our study to the case where the objective is that of minimizing the sum of the completion times of all the machines. We refer to such a case as scheduling SUM problem. Formally, given an assignment $\mathbf{N}$ where $C_j(N_j)$ denotes the completion time of machine j under the assignment $\mathbf{N}$, an optimal assignment minimizes the sum $\sum_{j=1}^{m} C_j(N_j)$. We notice that an optimal solution can be trivially determined by assigning each job to the machine providing it the minimum possible processing time.

We show tight upper and lower bounds on the price of k-envy-freeness for such a problem for all the studied settings, i.e. unrelated, related and identical machines, by exploiting ideas used for the minimum makespan case.

We start by observing that for the scheduling SUM problem with related (and then identical) machines, $\text{PoEF}_k = 1$ for any $k \geq 1$. In fact, it is easy to see that an optimal solution assigns all the jobs to the fastest machine and we notice that such assignment is k-envy-free for any $k \geq 1$.

We now turn to unrelated machines. First of all we observe that for the scheduling SUM problem, $\text{PoEF}_k \leq \left(1 + \frac{1}{k}\right)^{\min\{n,m\}-1}$ for any $k \geq 1$. We use the same arguments exploited in Theorem 8. Specifically, we start from an optimal solution and while there is a machine j which envies a machine j', then we move all the job from j' to j and continue until we reach a k-envy-free assignment. The result follows by the facts that the sum of the completion times of all the machines increases by a factor of at most $(1 + 1/k)$ at each iteration and that there are at most $\min\{n, m\} - 1$ iterations. Moreover we show that there exists an instance of the scheduling SUM problem for which $\text{PoEF}_k = \left(1 + \frac{1}{k+\epsilon}\right)^{\min\{n,m\}-1}$. Specifically, we consider the same instance used in Proposition 6 with the difference of setting $p_{i,i} = \epsilon$, for every $i \in \{2, \ldots, n\}$. In this way we have that the optimum, in which each job i is assigned to machine i, has cost $1 + (n-1)\epsilon$. Moreover the best k-envy-free allocation assigns all the jobs to machine 1 resulting in a cost equal to $\left(1 + \frac{1}{k+\epsilon}\right)^{\min\{n,m\}-1}$.

8. Conclusions

We have introduced and analyzed the notion of price of k-envy-freeness in scheduling problems and determined exact or asymptotically tight bounds for unrelated, related and identical machines. We have also showed how to efficiently compute nicely performing k-envy-free allocations and we have extended our results to the case of restricted assignments and to the objective of minimizing the sum of the machines completion times.

Considering the minimization of the sum of the jobs completion times is an issue that deserves future research attention.

Our framework is very general and can be considered as a forwarding step toward the investigation of envy-free solutions for other objectives not considered in this paper, like the dual ones of maximizing the number of jobs completed within a given time deadline (the so-called Max Throughput problem), or of

minimizing the number of machines needed to complete all the jobs within a certain time.

In general, an interesting worth investigating issue is the extension of our results to other scheduling problems and settings, as for instance the online case.

References

[1] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, E. Tardos, T. Wexler, T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. *SIAM Journal on Computing*, 38(4): 1602–1623, 2008.

[2] V. Bilò, A. Fanelli, M. Flammini, G. Monaco, L. Moscardelli. The Price of Envy-Freeness in Machine Scheduling. *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS), Part II*, pp. 106–117, 2014.

[3] J. Bulow, J. Roberts. The Simple Economics of Optimal Auctions. *The Journal of Political Economy*, 97(5):1060–1090, 1989.

[4] S. J. Brams and A. D. Taylor. An Envy-Free Cake Division Protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

[5] S. J. Brams, A. D. Taylor. Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press, 1996.

[6] E. Cohen, M. Feldman, A. Fiat, H. Kaplan, S. Olonetsky. Envy-Free Makespan Approximation. *SIAM Journal on Computing* 41(1):12–25, 2012.

[7] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, M. Kyropoulou. The Efficiency of Fair Division. *Theory of Computing Systems* 50(4): 589–610, 2012.

[8] G. Christodoulou, E. Koutsoupias, A. Vidali. A Lower Bound for Scheduling Mechanisms. *Algorithmica* 55(4):729–740, 2009.

[9] L.E. Dubins, E.H. Spanier. How to cut a cake fairly. *American Mathematical Monthly*, 68:1–17, 1961.

[10] D. Foley. Resource allocation and the public sector. *Yale Economics Essays*, 7:45–98, 1967.

[11] M. Feldman, A. Fiat, S. Leonardi, P. Sankowski. Revenue maximizing envy-free multi-unit auctions with budgets. *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 532–549, 2012.

[12] A. Fiat, A. Levavi. Tight Lower Bounds on Envy-Free Makespan Approximation. *Proceedings of the International Workshop on Internet and Network Economics (WINE)*, pp. 553–558, 2012.

[13] R. L. Graham. Bounds for Certain Multiprocessing Anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.

[14] A. V. Goldberg, J. D. Hartline. Envy-free auctions for digital goods. *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 29–35, 2003.

[15] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, F. McSherry: On profit-maximizing envy-free pricing. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pp 1164–1173, 2005.

[16] J. Hartline, S. Ieong, A. Mualem, M. Schapira, A. Zohar. Multi-dimensional envy-free scheduling mechanisms. *Technical Report 1144, The Hebrew Univ.*, 2008.

[17] D. S. Hochbaum, D. B. Shmoys. Using Dual Approximation Algorithms for Scheduling Problems: Theoretical and Practical Results. *Journal of ACM*, 34(1):144–162, 1987.

[18] D. S. Hochbaum, D. B. Shmoys. A Polynomial Approximation Scheme for Scheduling on Uniform Processors: Using the Dual Approximation Approach. *SIAM Journal on Computing*, 17(3): 539–551, 1988.

[19] E. Koutsoupias, A. Vidali. A Lower Bound of $1+\varphi$ for Truthful Scheduling Mechanisms. *Algorithmica*, 66(1):211–223, 2013.

[20] J. K. Lenstra, D. B. Shmoys, E Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programing*, 46:259–271, 1990.

[21] R. B. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6:58–73, 1981.

[22] J. Nagura. On the interval containing at least one prime number. *Proceedings of the Japan Academy*, Series A 28:177–181, 1952.

[23] N. Nisan, A. Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior* 35(1-2):166–196, 2001.

[24] W. Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, 16:8–37, 1961.