Smooth Inequalities and Equilibrium Inefficiency in Scheduling Games

Johanne Cohen[1], Christoph Dürr[2], and Nguyen Kim Thang[3]

[1] PRiSM, Université de Versailles St-Quentin-en-Yvelines, France
[2] LIP6, Université Pierre et Marie Curie, France
[3] IBISC, Université Evry Val d'Essonne, France

Abstract. We study coordination mechanisms for Scheduling Games (with unrelated machines). In these games, each job represents a player, who needs to choose a machine for its execution, and intends to complete earliest possible. In the paper, we focus on a general class of ℓ_k-norm (for parameter k) on job completion times as social cost, that permits to balance overall quality of service and fairness. Our goal is to design scheduling policies that always admit a pure Nash equilibrium and guarantee a small price of anarchy for the ℓ_k-norm social cost. We consider strongly-local and local policies (the policies with different amount of knowledge about jobs). First, we study the inefficiency in ℓ_k-norm social costs of a strongly-local policy SPT that schedules the jobs non-preemptively in order of increasing processing times. We show that the price of anarchy of policy SPT is $O(k^{\frac{k+1}{k}})$ and this bound is optimal (up to a constant) for all deterministic, non-preemptive, strongly-local and non-waiting policies (non-waiting policies produce schedules without idle times). Second, we consider the makespan (ℓ_∞-norm) social cost by making connection within the ℓ_k-norm functions. We present a local policy Balance. This policy guarantees a price of anarchy of $O(\log m)$, which makes it the currently best known policy among the anonymous local policies that always admit a pure Nash equilibrium.

1 Introduction

With the development of the Internet, large-scale systems consisting of autonomous decision-makers (players) become more and more important. The rational behavior of players who compete for the usage of shared resources generally leads to an unstable and inefficient outcome. This creates a need for *resource usage policies* that guarantee stable and near-optimal outcomes.

From a game theoretical point of view, stable outcomes are captured by the concept of *Nash equilibria*. Formally, in a game with n players, each player j chooses a strategy x_j from a set S_j and this induces a cost $c_j(\mathbf{x})$ for player j depending all chosen strategies $\mathbf{x}$. A strategy profile $\mathbf{x} = (x_1, \ldots, x_n)$ is a *pure Nash equilibrium* if no player can decrease its cost by an unilateral deviation, i.e., $c_j(x'_j, x_{-j}) \geq c_j(\mathbf{x})$ for every player j and $x'_j \in S_j$, where x_{-j} denotes the strategies selected by players different from j.

The *better-response dynamic* is the process of repeatedly choosing an arbitrary player that can improve its cost and let it take a better strategy while other player strategies remain unchanged. It is desirable that in a game the better-response dynamic converges to a Nash equilibrium as it is a natural way that selfish behavior leads the game to a stable outcome. A *potential game* is a game in which for any instance, the better-response dynamic always converges [10].

A standard measure of inefficiency is the *price of anarchy (PoA)*. Given a game with an objective function and a notion of equilibrium (e.g pure Nash equilibrium), the PoA of the game is defined as the ratio between the largest cost of an equilibrium and the cost of an optimal profile, which is not necessarily an equilibrium. The PoA captures the worst-case paradigm and it guarantees the efficiency of every equilibrium.

The social cost of a game is an objective function measuring the quality of strategy profiles. In the literature there are two main extensively-studied objective functions: (i) the *utilitarian social cost* is the total individual costs; while (ii) the *egalitarian social cost* is the maximum individual cost. The two objective functions are included in a general class of social costs: the class of ℓ_k norms of the individual costs, with utilitarian and the egalitarian social costs corresponding to the cases $k = 1$ and $k = \infty$, respectively. There is a need to design policies that guarantee the efficiency (e.g the PoA) of games under some specific objective function. Moreover, it would be interesting to come up with a policy, that would be efficient for every social costs from this class.

1.1 Coordination Mechanisms in Scheduling Games

In a scheduling game, there are n jobs and m unrelated machines. Each job needs to be scheduled on exactly one machine. We consider the unrelated parallel machine model, where each machine could be specialized for a different type of jobs. In this general setting, the processing time of job j on machine i is some given arbitrary value $p_{ij} > 0$. A strategy profile $\mathbf{x} = (x_1, \ldots, x_n)$ is an assignment of jobs to machines, where x_j denotes the machine (strategy) of job j in the profile. The *cost* c_j of a job j is its completion time and every job strategically chooses a machine to minimize the cost. In the game, we consider the social cost as the ℓ_k-norm of the individual costs. The *social cost* of profile $\mathbf{x}$ is $C(\mathbf{x}) = \left(\sum_j c_j^k\right)^{1/k}$.

The traditional ℓ_1, ℓ_∞-norms represent the total completion time and the makespan, respectively. Both objectives are natural. Minimizing the total completion time guarantees a quality of service while minimizing the makespan ensures the fairness of schedule. Unfortunately, in practice schedules which optimize the total completion time are not implemented due to a lack of fairness and vice versa. Implementing a fair schedule is one of the highest priorities in most systems [16]. A popular and practical method to enforce the fairness of a schedule is to optimize the ℓ_k-norm of completion times for some fixed k. By optimizing the ℓ_k-norm of completion time, one balances overall quality of service and fairness, which is generally desirable. So the system takes into account a trade-off between quality of service and fairness by optimizing the ℓ_k-norm of completion time [14, 16].

A *coordination mechanism* is a set of *scheduling policies*, one for each machine, that determine how to schedule the jobs assigned to a machine. The idea is to connect the individual cost to the social cost, in such a way that the selfishness of the agents will lead to equilibria with small social cost. We distinguish between *local* and *strongly-local* policies. These policies are classified in the decreasing order of the amount of information that ones could use for their decisions. Formally, let $\mathbf{x} = (x_1, \ldots, x_n)$ be a profile.

- A policy is *local* if the scheduling of jobs on machine i depends only on the processing times of jobs assigned to the machine, i.e., $\{p_{i'j} : x_j = i, 1 \leq i' \leq m\}$.
- A policy is *strongly-local* if the policy of machine i depends only on the processing times for this machine i for all jobs assigned to i, i.e., $\{p_{ij} : x_j = i\}$.

In addition, a policy is *anonymous* if it does not use any *global* ordering of jobs or any *global* job identities. Note that for any deterministic policy, *local* job identities are necessary as a machine may need such information in order to break ties (a job may have different identities on different machines). Moreover, we call a policy *non-waiting* if the schedule contains no idle time between job executions.

Instead of specifying the actual schedule, we rather describe a scheduling policy as a function, mapping every job j to some completion time $c_j(\mathbf{x})$. Such a policy is said *feasible* if for any profile $\mathbf{x}$, there exists a schedule where job j completes at time $c_j(\mathbf{x})$. Formally, for any job j, we must have $c_j(\mathbf{x}) \geq \sum_{j'} p_{ij'}$ where the sum is take over all jobs j' with $x_j = x_{j'}$ and $c_{j'}(\mathbf{x}) \leq c_j(\mathbf{x})$. Certainly, any designed deterministic policy needs to be feasible.

1.2 Overview & Contributions

Recently, Roughgarden [12] developed the *smoothness argument*, a unifying method to show upper bounds of the PoA for utilitarian games. This canonical method is elegant in its simplicity and its power. Here we give a brief description of this argument.

A cost-minimization game with the total cost objective $C(\mathbf{x}) = \sum_j c_j(\mathbf{x})$ is (λ, μ)-*smooth* if for every profile $\mathbf{x}$ and $\mathbf{x}^*$,

$$\sum_j c_j(x_j^*, x_{-j}) \leq \mu \sum_j c_j(\mathbf{x}) + \lambda \sum_j c_j(\mathbf{x}^*)$$

The smooth argument [12] states that the robust price of anarchy (including the PoA of pure, mixed, correlated equilibria, etc) of a cost-minimization game is bounded by

$$\inf \left\{ \frac{\lambda}{1-\mu} : \lambda \geq 0, \mu < 1, \text{ the game is } (\lambda, \mu)\text{-smooth} \right\}.$$

We will make use of this argument to settle the equilibrium inefficiency in scheduling games. We will prove the robust PoA by applying the smooth argument to the game with $C^k(\mathbf{x}) = \sum_j c_j^k(\mathbf{x})$ where $C(\mathbf{x})$ is the ℓ_k-norm social cost of Scheduling Games. The main difficulty in applying the smooth argument to Scheduling Games has arisen from the fact that jobs on the same machine have different costs, which is in contrast to Congestion Games [11] where players incurs the same cost at the same resource. The key technique in this paper is a system of inequalities, called *smooth inequalities*, that are useful to prove the smoothness of the game.

Our contributions are the following:

1. We study the equilibrium inefficiency for the ℓ_k-norm objective function. We consider a strongly-local policy SPT that schedules the jobs non-preemptively in order of increasing processing times (with a deterministic tie-breaking rule for each machine)[4]. We prove that the PoA of the game under the deterministic strongly-local policy SPT is at most $O(k^{\frac{k+1}{k}})$. Moreover, we show that any deterministic non-preemptive, non-waiting and strongly-local policy has a PoA at least $\Omega(k^{\frac{k+1}{k}})$, which matches to the performance of SPT policy. Hence, for any ℓ_k-norm social cost, SPT is optimal among deterministic non-preemptive, non-waiting, strongly-local policy. (The cases $k = 1$ and $k = \infty$ are confirmed in [6] and [2, 9], respectively.) If one considers theoretical evidence to classify algorithms for practical use then SPT is a good candidate due to its simplicity and theoretically guaranteed performance on any combination of the quality and the fairness of schedules.
2. We study the equilibrium inefficiency for the makespan objective function (e.g., ℓ_∞-norm) for local policies by making connection between ℓ_k-norm functions. We present a policy Balance (definition is given is Section 4). The game under that policy always admits Nash equilibrium and induces the PoA of $O(\log m)$ — the currently best performance among anonymous local policies that always possess pure Nash equilibria.

Our results naturally extend to the case when jobs have weights and the objective is the ℓ_k-norm of weighted completion times, i.e., $(\sum_j (w_j c_j(\mathbf{x}))^k)^{1/k}$.

1.3 Related results

The smooth argument has been formalized in [12]. It has been used to establish tight PoA of congestion games [11], a fundamental class of games. The argument is also applied to prove bounds on the PoA of weighted congestion games [3]. Subsequently, Roughgarden and Schoppman [13] have extended the argument to prove tight bounds on the PoA of atomic splittable congestion games for a large class of latencies.

Coordination mechanisms for scheduling games was introduced in [5] where the makespan (ℓ_∞-norm) objective was considered. For strongly-local policies,

[4] Formal definition of SPT is given in Section 3

Immorlica et al. [9] gave a survey on the existence and inefficiency of different policies such as SPT, LPT, RANDOM. Some tight bounds on the PoA under different policies were given. Azar et al. [2] initiated the study on local policies. They designed a non-preemptive policy with PoA of $O(\log m)$. However, the game under that policy does not necessarily guarantee a Nash equilibrium. The authors modified the policy and gave a preemptive one that always admits an equilibrium with a larger PoA as $O(\log^2 m)$. Subsequently, Caragiannis [4] derived a non-anonymous local policy ACOORD and anonymous local policies BCOORD and CCOORD with PoA of $O(\log m)$, $O(\log m/\log\log m)$ and $O(\log^2 m)$, respectively where the first and the last ones always admit a Nash equilibrium. Fleischer and Svitkina [7] showed a lower bound of $\Omega(\log m)$ for all deterministic non-preemptive, non-waiting local policies. Recently, Abed and Huang [1] proved that every deterministic (even preemptive) local policy, that satisfies natural properties, has price of anarchy at least $\Omega(\log m/\log\log m)$.

Cole et al. [6] studied the game with total completion time (ℓ_1-norm) objective. They considered strongly-local policies with weighted jobs, and derived a non-preemptive policy inspired by the Smith's rule which has PoA = 4. This bound is tight for deterministic non-preemptive non-waiting strongly-local policies. Moreover, some preemptive policies are also designed with better performance guarantee.

1.4 Organization

In Section 2, we state some smooth inequalities that will be used in settling the PoA for different policies. In Section 3, we study the scheduling game with the ℓ_k-norm social cost. We define and prove the inefficiency of the policiy SPT. We also provide an lower bound on the PoA for any deterministic non-preemptive non-waiting strongly-local policy. In Section 4, we consider the makespan (ℓ_∞-norm) social cost for local policies. We define and analyze the performance of policy Balance. Due to the space constraint, some proofs are given in the appendix.

2 Smooth Inequalities

In the section we show various inequalities that are useful for the analysis.

Lemma 1. *Let k be a positive integer. Let $0 < a(k) \leq 1$ be a function on k. Then, for any $x, y > 0$, it holds that*

$$y(x+y)^k \leq \frac{k}{k+1} a(k) x^{k+1} + b(k) y^{k+1}$$

where α is some constant and

$$b(k) = \begin{cases} \Theta\left(\alpha^k \cdot \left(\frac{k}{\log k a(k)}\right)^{k-1}\right) & \text{if } \lim_{k\to\infty}(k-1)a(k) = \infty, & \text{(1a)} \\ \Theta\left(\alpha^k \cdot k^{k-1}\right) & \text{if } (k-1)a(k) \text{ are bounded } \forall k, & \text{(1b)} \\ \Theta\left(\alpha^k \cdot \frac{1}{k a(k)^k}\right) & \text{if } \lim_{k\to\infty}(k-1)a(k) = 0. & \text{(1c)} \end{cases}$$

Note that the case (1a) of Lemma 1 could be used to settle the tight bound on the PoA of Congestion Games in which delay functions are polynomials with positive coefficients. [15] proved this case for $a(k) = 1$ and $b(k) = \Theta(\frac{1}{k}(k/\log k)^k)$ in order to upper bound of the PoA in Selfish Load Balancing Games.

Lemma 2. *It holds that* $(k+1)z \geq 1 - (1-z)^{k+1}$ *for all* $0 \leq z \leq 1$ *and for all* $k \geq 0$.

Proof. Consider $f(z) = (k+1)z - 1 + (1-z)^{k+1}$ for $0 \leq z \leq 1$. We have $f'(z) = (k+1) - (k+1)(1-z)^k \geq 0 \ \forall 0 \leq z \leq 1$. So f is non-decreasing function, thus $f(z) \geq f(0) = 0$. Therefore, $(k+1)z \geq 1 - (1-z)^{k+1}$ for all $0 \leq z \leq 1$. □

In the following, we prove inequalities to bound the PoA of the scheduling game. Remark that until the end of the section, we use i, j as the indices. The following is the main lemma to show the upper bound $O(k^{(k+1)/k})$ of the PoA under policy SPT in the next section.

Lemma 3. *For any non-negative sequences* $(n_i)_{i=1}^P$, $(m_i)_{i=1}^P$, *and for any positive increasing sequence* $(q_i)_{i=1}^P$, *define* $A_{i,j} := n_1 q_1 + \ldots + n_{i-1} q_{i-1} + j \cdot q_i$ *for* $1 \leq i \leq P, 1 \leq j \leq n_i$ *and* $B_{i,j} := m_1 q_1 + \ldots + m_{i-1} q_{i-1} + j \cdot q_i$ *for* $1 \leq i \leq P, 1 \leq j \leq m_i$. *Then, it holds that*

$$\sum_{i=1}^{P} \sum_{j=1}^{m_i} (A_{i,n_i} + j \cdot q_i)^k \leq \mu_k \sum_{i=1}^{P} \sum_{j=1}^{n_i} A_{i,j}^k + \lambda_k \sum_{i=1}^{P} \sum_{j=1}^{m_i} B_{i,j}^k,$$

where $\mu_k = \frac{k+1}{k+2}$ *and* $\lambda_k = \Theta(\alpha^k (k+1)^k)$ *for some constant* α.

3 ℓ_k-norms of Completion Times under Strongly-Local Policies

We consider the coordination mechanism under the strongly-local policy SPT that schedules jobs in the order of non-decreasing processing times. The formal definition of that policy is the following.

Policy SPT Let $\mathbf{x}$ be a strategy profile. Let $\prec_i$ be an order of jobs on machine i, where $j' \prec_i j$ iff $p_{ij'} < p_{ij}$ or $p_{ij'} = p_{ij}$ and j is priority over j' (machine i chooses a local preference over jobs based on their local identities to break ties). The cost of job j under the SPT [9] policy is

$$c_j(\mathbf{x}) = \sum_{\substack{j':\ x_{j'}=i \\ j' \preceq j}} p_{ij'}.$$

Note that the policy SPT is feasible. Since all p_{ij} could be written as a multiple of ϵ (a small precision) without loss of generality, assume that all jobs processing times (scaling by ϵ^{-1}) are integers and upper-bounded by P.

Lemma 4. *Let* $\mathbf{x}$ *be an assignment of jobs to machines. Then, among all feasible schedules,* SPT *policy minimizes the* ℓ_k*-norm of job completion times with respect to this assignment.*

Theorem 1. *The PoA of* SPT *with respect to the* ℓ_k*-norm of job completion times is* $O(k^{\frac{k+1}{k}})$.

Proof. Let $\mathbf{x}$ and $\mathbf{x}^*$ be two arbitrary profiles. We focus on a machine i. Let $n_1, \ldots, n_P$ be the numbers of jobs in $\mathbf{x}$ which are assigned to machine i and have processing times $1, \ldots, P$, respectively. Similarly, $m_1, \ldots, m_P$ are defined for profile $\mathbf{x}^*$. Note that n_a and m_a are non-negative for $1 \leq a \leq P$. Applying Lemma 3 for non-negative sequences $(n_a)_{a=1}^P$, $(m_a)_{a=1}^P$ and the positive increasing sequence $(a)_{a=1}^P$, we have:

$$\sum_{a=1}^{P} \left[\left(\sum_{b=1}^{a} bn_b + a \right)^k + \left(\sum_{b=1}^{a} bn_b + 2a \right)^k + \ldots + \left(\sum_{b=1}^{a} bn_b + m_a \cdot a \right)^k \right]$$

$$\leq \frac{k+1}{k+2} \cdot \sum_{a=1}^{P} \left[\left(\sum_{b=1}^{a-1} bn_b + a \right)^k + \left(\sum_{b=1}^{a-1} bn_b + 2a \right)^k + \ldots + \left(\sum_{b=1}^{a-1} bn_b + n_a \cdot a \right)^k \right]$$

$$+ \Theta\left(\alpha^k (k+1)^k\right) \cdot \left[\left(\sum_{b=1}^{a-1} bm_b + a \right)^k + \left(\sum_{b=1}^{a-1} bm_b + 2a \right)^k + \ldots + \left(\sum_{b=1}^{a-1} bm_b + m_a \cdot a \right)^k \right]$$

where α is a constant.

Observe that, by definition of the cost under the SPT policy, the left-hand side (of the inequality above) is an upper bound for $\sum_{j:x_j^*=i} c_j^k(x_{-j}, x_j^*)$, while the right-hand side is exactly $\frac{k+1}{k+2} \cdot \sum_{j:x_j=i} c_j^k(\mathbf{x}) + \Theta\left(\alpha^k (k+1)^k\right) \cdot \sum_{j:x_j^*=i} c_j^k(\mathbf{x}^*)$. Thus,

$$\sum_{j:x_j^*=i} c_j^k(x_{-j}, x_j^*) \leq \frac{k+1}{k+2} \cdot \sum_{j:x_j=i} c_j^k(\mathbf{x}) + \Theta\left(\alpha^k (k+1)^k\right) \cdot \sum_{j:x_j^*=i} c_j^k(\mathbf{x}^*)$$

As the inequality above holds for every machine i, summing over all machines we have:

$$\sum_{j} c_j^k(x_{-j}, x_j^*) \leq \frac{k+1}{k+2} \cdot \sum_{j} c_j^k(\mathbf{x}) + \Theta\left(\alpha^k (k+1)^k\right) \cdot \sum_{j} c_j^k(\mathbf{x}^*)$$

By the smooth argument, $C^k(\mathbf{x}) \leq \left(\alpha^k (k+1)^{k+1}\right) C^k(\mathbf{x}^*)$. Therefore, $C(\mathbf{x}) \leq O(k^{\frac{k+1}{k}}) C(\mathbf{x}^*)$.

Choosing $\mathbf{x}^*$ as an optimal assignment. By Lemma 4, the optimal schedule for this assignment could be done using the SPT policy, i.e., the optimal social cost is $C(\mathbf{x}^*)$. Therefore, the PoA is $O(k^{\frac{k+1}{k}})$. □

The following theorem proved that the bound on the PoA is tight. The construction is a generalization of the one in [6] where the authors showed a tight bound for the ℓ_1-norm.

Theorem 2. *The PoA of any deterministic non-preemptive non-waiting strongly-local policy is $\Omega(k^{\frac{k+1}{k}})$ with respect to the ℓ_k-norm of job completion times.*

Proof. Using the technique described in [6], it is sufficient to prove that the PoA of SPT is $\Omega(k^{\frac{k+1}{k}})$.

Let t and m be integers such that $m = \prod_{u=1}^{t} u^k$. (In fact, for the proof it is enough to choose m such that m/u^k is integer for every $1 \leq u \leq t$.) Consider an instance in which there are m machines and the jobs are $\{j_{u,v} : 1 \leq u \leq t, 1 \leq v \leq m/u^k\}$. A job $j_{u,v}$ has unit processing time on every machine $1 \leq i \leq v$ and has processing time infinity on other machines. In other words, job $j_{u,v}$ is allowed to be scheduled only on machine with index at most v. We say that a job $j_{u,v}$ has more *priority* than job $j_{u',v'}$ if $v > v'$; or if $v = v'$ and $u < u'$. If two jobs $j_{u,v}$ and $j_{u',v'}$ are both assigned to the same (allowed) machine then the job with higher priority will be scheduled before the other (note that those jobs have the same unit processing times in the machine).

We first give an assignment of jobs to machines with a small social cost. Consider an assignment $\mathbf{x}^*$ in which job $j_{u,v}$ for $1 \leq u \leq t, 1 \leq v \leq m/u^k$ is scheduled in machine v. An illustration is given in the left of Figure 1. By the priority order, the completion time of job $j_{u,v}$ is u. By the construction, the number of jobs with completion time u for $1 \leq u \leq t$ is m/u^k. Hence, the social cost of the assignment satisfies $C^k(\mathbf{x}^*) = \sum_{u=1}^{t} u^k m/u^k = mt$.

Now we construct a Nash equilibrium with high social cost. Roughly speaking, for each $1 \leq s \leq t$, we will assign the set of jobs $\mathcal{J}_s = \{j_{u,v} : 1 \leq u \leq s, m/(s+1)^k < v \leq m/s^k\}$ to a subset of machines i for $1 \leq i \leq m/s^{k+1}$ in such a way that their completion times are between $k(s-1)+1$ and ks. Moreover, in the assignment apart of those jobs, no other has completion time in $[k(s-1)+1, ks]$. As there are t such sets $\mathcal{J}_s$ and each set gives rise to k units in the completion times, the desired lower bound follows.

Formally, fix $1 \leq s \leq t$ and consider the set $\mathcal{J}_s = \{j_{u,v} : 1 \leq u \leq s, m/(s+1)^k < v \leq m/s^k\}$. Partition $\mathcal{J}_s = \mathcal{J}_{s,1} \cup \ldots \cup \mathcal{J}_{s,k}$ where

$$\mathcal{J}_{s,a} := \left\{ j_{u,v} : 1 \leq u \leq s, \frac{m}{s^{k-a}(s+1)^a} < v \leq \frac{m}{s^{k+1-a}(s+1)^{a-1}} \right\}.$$

for $1 \leq a \leq k$. The cardinal of $\mathcal{J}_{s,a}$ is

$$|\mathcal{J}_{s,a}| = s\left(\frac{m}{s^{k+1-a}(s+1)^{a-1}} - \frac{m}{s^{k-a}(s+1)^a}\right) = \frac{m}{s^{k-a}(s+1)^a}.$$

Note that by definition, jobs in $\mathcal{J}_{s',a'}$ have higher priority then the ones in $\mathcal{J}_{s,a}$ in case $s > s'$ or in case $s = s'$ and $a > a'$. In total, there are $k \cdot t$ sets $\mathcal{J}_{s,a}$ since $1 \leq s \leq t$ and $1 \leq a \leq k$.

Consider a profile $\mathbf{x}$ in which jobs in $\mathcal{J}_{s,a}$ for $1 \leq s \leq t$ and $1 \leq a \leq k$ are assigned arbitrarily one-to-one to machines $1, 2, \ldots, |\mathcal{J}_{s,a}|$. It is feasible since a

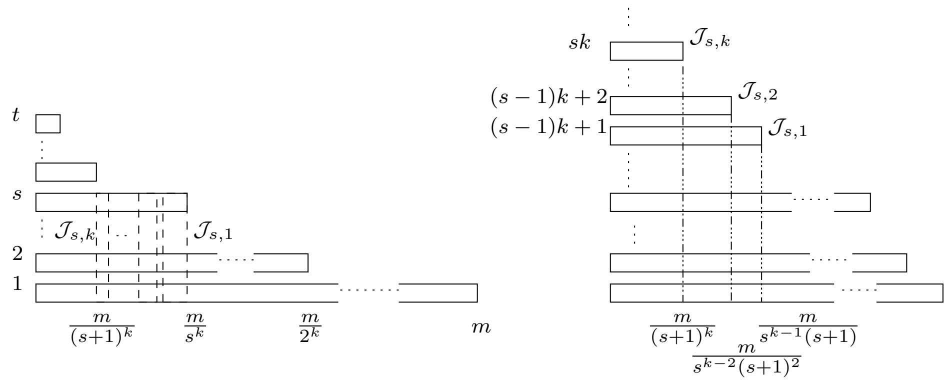

Fig. 1. Illustration of profiles $\mathbf{x}^*$ (in the left) and $\mathbf{x}$ (in the right). The horizontal and vertical axes represent machines and completion times, respectively.

job $j_{u,v} \in \mathcal{J}_{s,a}$ has index $v > \frac{m}{s^{k-a}(s+1)^a} = |\mathcal{J}_{s,a}|$, meaning that the job could be scheduled on any machine in $1, 2, \ldots, |\mathcal{J}_{s,a}|$. In this assignment, jobs in the same set $\mathcal{J}_{s,a}$ have the same cost, which is $(s-1)k + a$. An illustration is given in the right of Figure 1. We show that profile $\mathbf{x}$ is indeed a Nash equilibrium. Let $j_{u,v}$ be a job in $\mathcal{J}_{s,a}$. This job has cost $(s-1)k + a$ and cannot be scheduled on any machine with index larger then $\frac{m}{s^{k+1-a}(s+1)^{a-1}}$. Recall that if $a > 1$, $\frac{m}{s^{k+1-a}(s+1)^{a-1}} = |\mathcal{J}_{s,a-1}|$; and if $a = 1$ and $s > 1$, $\frac{m}{s^{k+1-a}(s+1)^{a-1}} = |\mathcal{J}_{s-1,k}|$. In profile $\mathbf{x}$, the jobs assigned to machines $1, 2, \ldots, \frac{m}{s^{k+1-a}(s+1)^{a-1}}$ with cost strictly smaller then $(s-1)k + a$ are jobs in $\mathcal{J}_{s',a'}$ where either $s' < s$ or $s' = s$ and $a' < a$. The jobs have higher priority then $j_{u,v}$. Therefore, job $j_{u,v} \in \mathcal{J}_{s,a}$ for $(s, a) \neq (1, 1)$ cannot unilaterally change machine to improve its cost. Besides, jobs in $\mathcal{J}_{1,1}$ have no incentive to change their machines as their cost are 1 and they cannot strictly decrease by doing so. Thus, $\mathbf{x}$ is a Nash equilibrium.

In profile $\mathbf{x}$, there are exactly $|\mathcal{J}_{s,a}|$ jobs with cost $(s-1)k + a$. Therefore, the social cost $C(\mathbf{x})$ satisfies:

$$C^k(\mathbf{x}) = \sum_{s=1}^{t}\sum_{a=1}^{k} \frac{m}{s^{k-a}(s+1)^a}[(s-1)k + a]^k \geq k^k m \sum_{s=1}^{t}\sum_{a=1}^{k} \frac{(s-1)^k}{s^{k-a}(s+1)^a}$$

$$\geq k^{k+1} m \sum_{s=1}^{t} \frac{(s-1)^k}{(s+1)^k} \geq k^{k+1} m (t-1) \frac{1}{3^k}$$

Hence, we deduce that $C(\mathbf{x})/C(\mathbf{x}^*) \geq \frac{1}{4} k^{\frac{k+1}{k}}$. □

4 ℓ_∞-norms of Completion Times under Local Policies

For any profile $\mathbf{x}$, the social cost $C(\mathbf{x}) = \max_j c_j$. Let $\mathbf{x}(i) = \{j : x_j = i\}$ be the set of jobs assigned to machine i. Define $L(\mathbf{x}(i)) := \sum_{j:x_j=i} p_{ij}$ as the *load* of of machine i for $1 \leq i \leq m$ in profile $\mathbf{x}$. The makespan of the profile is $L(\mathbf{x}) := \max_i L(\mathbf{x}(i))$. Observe that in an optimal assignment $\mathbf{x}^*$, $C(\mathbf{x}^*) = L(\mathbf{x}^*)$ since there is no idle-time in an optimal schedule. For each job j, denote $q_j := \min\{p_{ij} : 1 \leq i \leq m\}$ and define $\rho_{ij} := p_{ij}/q_j$ for all i, j. Note that a local policy can compute q_j for every job j while a strongly-local one cannot.

A profile $\mathbf{x}$ is *m-efficient* if $\rho_{x_j,j} \leq m$ for every job j. The following lemma guarantees that the restriction to the m-efficient profiles worsens the optimal social cost only by a constant factor.

Lemma 5 ([4]). *Let $\mathbf{y}^*$ be an optimal assignment. Then, there exits a m-efficient assignment $\mathbf{x}^*$ such that $L(\mathbf{x}^*) \leq 2L(\mathbf{y}^*)$.*

Policy Balance Let $\mathbf{x}$ be a strategy profile. Let $\prec_i$ be a total order on the jobs assigned to machine i, which is a SPT-like order. Formally, $j \prec_i j'$ if $p_{ij} < p_{ij'}$, or $p_{ij} = p_{ij'}$ and j is priority over j' (machine i chooses a local preference over jobs based on their local identities to break ties). Note that the policy does not need a global job identities (there is no communication cost between machines about job identities) and a job may have different priority on different machines. The policy is clearly anonymous.

The cost c_j of job j assigned to machine i is defined as follows where k is a constant to be chosen later.

$$c_j^k(\mathbf{x}) = \begin{cases} \frac{1}{q_j}\left[\Big(p_{ij} + \sum\limits_{\substack{j':j' \prec_i j \\ x_{j'}=i}} p_{ij'}\Big)^{k+1} - \Big(\sum\limits_{\substack{j':j' \prec_i j \\ x_{j'}=i}} p_{ij'}\Big)^{k+1}\right] & \text{if } \rho_{ij} \leq m, \\ \infty & \text{otherwise.} \end{cases}$$

Intuitively, the cost of a job scheduled on a machine is proportional to its *marginal contribution* to the load of the machine (up to some power). Moreover, by the definition, jobs are allowed to be scheduled only on machines with inefficiency smaller than m.

Observe that the cost $c_j(\mathbf{x})$ of job j satisfies

$$\begin{aligned} c_j^k(\mathbf{x}) &\geq \frac{1}{q_j}\left[\Big(p_{ij} + \sum_{j':j' \prec_i j,\ x_{j'}=i} p_{ij'}\Big)^{k+1} - \Big(\sum_{j':j' \prec_i j,\ x_{j'}=i} p_{ij'}\Big)^{k+1}\right] \\ &\geq \frac{p_{ij}}{q_j}\Big(p_{ij} + \sum_{j':j' \prec_i j,\ x_{j'}=i} p_{ij'}\Big)^{k} \geq \Big(p_{ij} + \sum_{j':j' \prec_i j,\ x_{j'}=i} p_{ij'}\Big)^{k} \end{aligned}$$

since $p_{ij}/q_j \geq 1$. As that holds for every job j assigned to machine i, policy Balance is feasible.

Lemma 6. *The best-response dynamic under the* Balance *policy converges to a Nash equilibrium.*

Proof. By the definition of the policy, any job j will choose a machine i such that $\rho_{ij} \leq m$. Moreover, since q_j is fixed for each job j, the behavior of jobs is similar to that in the following game. In the latter, the set of strategy of a player j is the same as in the former except the machines i with $\rho_{ij} > m$. Moreover, in the new game, player j in profile $\mathbf{x}$ has cost $c'_j(\mathbf{x})$ such that

$$\Big(c'_j(\mathbf{x})\Big)^k = \Big(p_{ij} + \sum_{j' \prec_i j} p_{ij'}\Big)^{k+1} - \Big(\sum_{j' \prec_i j} p_{ij'}\Big)^{k+1}$$

Hence, it is sufficient to prove that the better-response dynamic in the new game always converges. The argument is the same as the one to prove the existence of Nash equilibrium for policy SPT [9]. Here we present a proof based on a geometrical approach [8].

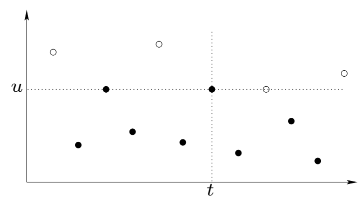

Fig. 2. An geometrical illustration of $|\mathbf{x}|_{u,t}$, every dot is a $(j, c_j(\mathbf{x}))$ pair, colored black if counted in $|\mathbf{x}|_{u,t}$.

First, define $\texttt{pos}_i(j) := 1 + |\{j' : j' \prec_i j, 1 \leq j' \neq j \leq n\}|$ which represents the priority of job j on machine i. For a value $u \in \mathbb{R}^+$ and a job index $1 \leq t \leq n$, we associate to every profile $\mathbf{x}$ the quantity

$$|\mathbf{x}|_{u,t} := |\{j : c'_j(\mathbf{x}) < u \text{ or } c'_j(\mathbf{x}) = u, \texttt{pos}_{x_j}(j) \leq t\}|.$$

We use it to define a partial order $\prec$ on profiles. Formally $\mathbf{x} \prec \mathbf{y}$ if for the lexicographically smallest pair (u, t) such that $|\mathbf{x}|_{u,t} \neq |\mathbf{y}|_{u,t}$ we have $|\mathbf{x}|_{u,t} < |\mathbf{y}|_{u,t}$.

We show that the profile strictly increases according to this order, whenever a job changes to another machine while decreasing its cost. Let j be such a job changing from machine a in profile $\mathbf{x}$ to machine b, resulting in a profile $\mathbf{y}$. We know that $c'_j(\mathbf{y}) < c'_j(\mathbf{x})$. Remark that only jobs j' with $x_{j'} = b$ might have the cost in $\mathbf{y}$ larger than that in $\mathbf{x}$ (by definition of the cost c'). Moreover, such job j' with $x_{j'} = b$ and j' has a different costs in $\mathbf{x}$ and $\mathbf{y}$, it must be $j \prec_b j'$, which also implies $c'_{j'}(\mathbf{x}) \geq c'_j(\mathbf{y})$. In the same spirit, some jobs j' with $x_{j'} = a$ might decrease their cost, but not below $c'_j(\mathbf{x})$.

Consider $u = c'_j(\mathbf{y})$ and $t = \texttt{pos}_b(j)$. We have that $|\mathbf{x}|_{u',t'} = |\mathbf{y}|_{u',t'}$ for all $u' < u$ and all t'. If job j is the only job with processing time p_{bj} among the ones $\{j' : x_{j'} = b\}$, then $|\mathbf{y}|_{u,t} = |\mathbf{x}|_{u,t} + 1$. Otherwise, $|\mathbf{y}|_{u,t'} = |\mathbf{x}|_{u,t'}$ for $t' < t$ and $|\mathbf{y}|_{u,t} = |\mathbf{x}|_{u,t} + 1$.

Therefore (u, t) is the first lexicographical pair where $|\mathbf{x}|_{u,t} \neq |\mathbf{y}|_{u,t}$ and $|\mathbf{y}|_{u,t} > |\mathbf{x}|_{u,t}$. Hence, since the set of strategy profiles is finite, the better-response dynamic must converge to a pure Nash equilibrium. This completes the proof. □

Remark that the game under Balance convergences fast to Nash equilibria in the best-response dynamic (the argument is the same as [9, Theorem 12]).

Lemma 7. *Let* $\mathbf{x}$ *and* $\mathbf{x}^*$ *be an equilibrium and an* m*-efficient arbitrary profile, respectively. Then,* $\sum_{i=1}^{m} L^{k+1}(\mathbf{x}(i)) \leq O(\alpha^k k^{k+1}) \sum_{i=1}^{m} L^{k+1}(\mathbf{x}^*(i))$ *where* α *is some constant.*

Proof. We focus on an arbitrary job j. Denote $i = x_j$ and $i^* = x_j^*$. As $\mathbf{x}$ is an equilibrium, we have $c_j^k(\mathbf{x}) \leq c_j^k(x_{-j}, x_j^*)$, i.e,

$$\begin{aligned}
&\Big(p_{ij} + \sum_{\substack{j': j' \prec_i j \\ x_{j'} = i}} p_{ij'}\Big)^{k+1} - \Big(\sum_{\substack{j': j' \prec_i j \\ x_{j'} = i}} p_{ij'}\Big)^{k+1} \\
&\quad \leq \Big(p_{i^*j} + \sum_{\substack{j': j' \prec_{i^*} j \\ x_{j'} = i^*}} p_{i^*j'}\Big)^{k+1} - \Big(\sum_{\substack{j': j' \prec_{i^*} j \\ x_{j'} = i^*}} p_{i^*j'}\Big)^{k+1} \\
&\quad \leq \Big(p_{i^*j} + L(\mathbf{x}(i^*))\Big)^{k+1} - \Big(L(\mathbf{x}(i^*))\Big)^{k+1} \\
&\quad \leq (k+1) p_{i^*j} \Big(p_{i^*j} + L(\mathbf{x}(i^*))\Big)^{k}
\end{aligned} \tag{2}$$

where the second inequality is due to the fact that $(z+a)^{k+1} - z^{k+1}$ is increasing in z (for $a > 0$) and $\sum_{\substack{j': j' \prec_i j \\ x_{j'} = i^*}} p_{i^*j'} \leq L(\mathbf{x}(i^*))$; the third inequality is due to Lemma 2 (by dividing both sides by $(p_{i^*j} + L(\mathbf{x}(i^*)))^{k+1}$ and applying $z = \frac{p_{i^*j}}{p_{i^*j} + L(\mathbf{x}(i^*))}$ in the statement of Lemma 2). Therefore,

$$\begin{aligned}
\sum_{i=1}^{m} L^{k+1}(\mathbf{x}(i)) &= \sum_{i=1}^{m} \sum_{j: x_j = i} q_j c_j^k(\mathbf{x}) \leq \sum_{i=1}^{m} \sum_{j: x_j = i} q_j c_j^k(x_{-j}, x_j^*) \\
&\leq \sum_{i=1}^{m} \sum_{j: x_j^* = i} (k+1) p_{ij} \Big(p_{ij} + L(\mathbf{x}(i))\Big)^{k} \\
&\leq (k+1) \sum_{i=1}^{m} L(\mathbf{x}^*(i)) \Big(L(\mathbf{x}(i)) + L(\mathbf{x}^*(i))\Big)^{k} \\
&\leq (k+1) \sum_{i=1}^{m} \frac{k}{(k+1)^2} L^{k+1}(\mathbf{x}(i)) + O\left(\alpha^k k^{k-1}\right) L^{k+1}(\mathbf{x}^*(i))
\end{aligned}$$

where the first inequality is because $\mathbf{x}$ is an equilibrium; the second inequality is due to the sum of Inequality (2) taken over all jobs j; and the fourth inequality follows by applying case (1b) of Lemma 1 for $a(k) = 1/(k+1)$. Arranging the terms, the lemma follows. □

Theorem 3. *The PoA of the game under policy* Balance *is at most* $O(\log m)$ *by choosing* $k = \log m$.

Proof. Let $\mathbf{y}^*$ be an optimal assignment and $\mathbf{x}^*$ be an m-efficient assignment with property of Lemma 5. Let $\mathbf{x}$ be an equilibrium. Remark that $\mathbf{x}$ is a m-efficient assignment since every job can always get a bounded cost. Consider a job j assigned to machine i in profile $\mathbf{x}$. As $\mathbf{x}$ is a m-efficient assignment, by the definition of policy Balance

$$c_j^k(\mathbf{x}) = \frac{1}{q_j}\left[\Big(p_{ij} + \sum_{\substack{j':j' \prec_i j \\ x_{j'}=i}} p_{ij'}\Big)^{k+1} - \Big(\sum_{\substack{j':j' \prec_i j \\ x_{j'}=i}} p_{ij'}\Big)^{k+1}\right]$$
$$\leq \frac{1}{q_j}\left[\Big(L(\mathbf{x}(i))\Big)^{k+1} - \Big(L(\mathbf{x}(i)) - p_{ij}\Big)^{k+1}\right] \leq (k+1)\rho_{ij} L^k(\mathbf{x}(i))$$

where the first inequality is because function $(a+x)^{k+1} - x^{k+1}$ is increasing; and the last inequality is due to Lemma 2 (by dividing both sides by $L^{k+1}(\mathbf{x}(i))$ and applying $z = \frac{p_{ij}}{L(\mathbf{x}(i))}$ in the statement of Lemma 2). Moreover, by Lemma 7, we have

$$L^{k+1}(\mathbf{x}) \leq \sum_{i=1}^{m} L^{k+1}(\mathbf{x}(i)) \leq O(\alpha^k k^{k+1}) \sum_{i=1}^{m} L^{k+1}(\mathbf{x}^*(i)) \leq O(\alpha^k k^{k+1} m) L^{k+1}(\mathbf{x}^*)$$

for some constant α. Therefore,

$$C(\mathbf{x}) = \max_j c_j(\mathbf{x}) \leq \max_{i,j} \Big((k+1)\rho_{ij}\Big)^{1/k} L(\mathbf{x}(i)) \leq \Big((k+1)m\Big)^{1/k} L(\mathbf{x})$$
$$\leq O\left(\Big(k^{k+2} m^2\Big)^{1/k}\right) L(\mathbf{x}^*) \leq O\left(\Big(k^{k+2} m^2\Big)^{1/k}\right) L(\mathbf{y}^*)$$

where the last inequality is due to Lemma 5. Choosing $k = \log m$, the theorem follows. ◻

Acknowledgments We thank José Correa for showing us a lower bound construction in [6] and several helpful discussions and useful comments.

References

1. Fidaa Abed and Chien-Chung Huang. Preemptive coordination mechanisms for unrelated machines. In *Proc. of the 20th European Symposium on Algorithms*, 2012.
2. Yossi Azar, Kamal Jain, and Vahab S. Mirrokni. (Almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 323–332, 2008.
3. Kshipra Bhawalkar, Martin Gairing, and Tim Roughgarden. Weighted congestion games: Price of anarchy, universal worst-case examples, and tightness. In *Proceedings of the 18th Annual European Symposium on Algorithms (ESA)*, pages 17–28, 2010.

4. Ioannis Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 815–824, 2009.
5. George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 345–357, 2004.
6. Richard Cole, José R. Correa, Vasilis Gkatzelis, Vahab S. Mirrokni, and Neil Olver. Inner product spaces for minsum coordination mechanisms. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 539–548, 2011.
7. Lisa Fleischer and Zoya Svitkina. Preference-constrained oriented matching. In *Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 56–65, 2010.
8. Laurent Gourvès, Jérôme Monnot, Stefano Moretti, and Nguyen Kim Thang. Congestion games with capacitated resources. In *Proc. of the 5th Symposium on Algorithmic Game Theory*, 2012.
9. Nicole Immorlica, Li (Erran) Li, Vahab S. Mirrokni, and Andreas S. Schulz. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410(17):1589–1598, 2009.
10. Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.
11. Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
12. Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 513–522, 2009.
13. Tim Roughgarden and Florian Schoppmann. Local smoothness and the price of anarchy in atomic splittable congestion games. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.
14. Abraham Silberschatz and Peter Galvin. *Operating System Concepts*. Addison-Wesley, 1994.
15. Subhash Suri, Csaba D. Tóth, and Yunhong Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 47(1):79–96, 2007.
16. Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice Hall Press, 2007.

Appendix

Lemma 1. *Let k be a positive integer. Let $0 < a(k) \leq 1$ be a function on k. Then, for any $x, y > 0$, it holds that*

$$y(x+y)^k \leq \frac{k}{k+1} a(k) x^{k+1} + b(k) y^{k+1}$$

where α is some constant and

$$b(k) = \begin{cases} \Theta\left(\alpha^k \cdot \left(\frac{k}{\log ka(k)}\right)^{k-1}\right) & \text{if } \lim_{k\to\infty}(k-1)a(k) = \infty, \quad (1a) \\ \Theta\left(\alpha^k \cdot k^{k-1}\right) & \text{if } (k-1)a(k) \text{ are bounded } \forall k, \quad (1b) \\ \Theta\left(\alpha^k \cdot \frac{1}{ka(k)^k}\right) & \text{if } \lim_{k\to\infty}(k-1)a(k) = 0. \quad (1c) \end{cases}$$

Proof. Let $f(z) := \frac{k}{k+1}a(k)z^{k+1} - (1+z)^k + b(k)$. To show the claim, it is equivalent to prove that $f(z) \geq 0$ for all $z > 0$.

We have $f'(z) = ka(k)z^k - k(1+z)^{k-1}$. We claim that the equation $f'(z) = 0$ has an unique positive root z_0. Consider the equation $f'(z) = 0$ for $z > 0$. It is equivalent to

$$\left(\frac{1}{z} + 1\right)^k \cdot \frac{1}{z} = a(k)$$

The left-hand side is a strictly decreasing function and the limits when z tends to 0 and ∞ are ∞ and 0, respectively. As $a(k)$ is a positive constant, there exists an unique root $z_0 > 0$.

Observe that function f is decreasing in $(0, z_0)$ and increasing in $(z_0, +\infty)$, so $f(z) \geq f(z_0)$ for all $z > 0$. Hence, by choosing

$$b(k) = \left|\frac{k}{k+1}a(k)z_0^{k+1} - (1+z_0)^k\right| = (1+z_0)^{k-1}\left(1 + \frac{z_0}{k+1}\right) \quad (3)$$

it follows that $f(z) \geq 0 \; \forall z > 0$.

We study the positive root z_0 of equation

$$a(k)z^k - (1+z)^{k-1} = 0 \quad (4)$$

Note that $f'(1) = k(a(k) - 2^{k-1}) < 0$ since $0 < a(k) \leq 1$. Thus, $z_0 > 1$. For the sake of simplicity, we define the function $g(k)$ such that $z_0 = \frac{k-1}{g(k)}$ where $0 < g(k) < k-1$. Equation (4) is equivalent to

$$\left(1 + \frac{g(k)}{k-1}\right)^{k-1} g(k) = (k-1)a(k)$$

Note that $e^{w/2} < 1 + w < e^w$ for $w \in (0,1)$. For $w := \frac{g(k)}{k-1}$, we obtain the following upper and lower bounds for the term $(k-1)a(k)$:

$$e^{g(k)/2}g(k) < (k-1)a(k) < e^{g(k)}g(k) \tag{5}$$

Recall the definition of *Lambert W function*. For each $y \in \mathbb{R}^+$, $W(y)$ is defined to be solution of the equation $xe^x = y$. Note that, xe^x is increasing with respect to x, hence $W(\cdot)$ is increasing.

By definition of the Lambert W function and Equation (5), we get that

$$W\left((k-1)a(k)\right) < g(k) < 2W\left(\frac{(k-1)a(k)}{2}\right) \tag{6}$$

First, consider the case where $\lim_{k\to\infty}(k-1)a(k) = \infty$. The asymptotic sequence for $W(x)$ as $x \to +\infty$ is the following: $W(x) = \ln x - \ln\ln x + \frac{\ln\ln x}{\ln x} + O\left(\left(\frac{\ln\ln x}{\ln x}\right)^2\right)$. So, for large enough k, $W((k-1)a(k)) = \Theta(\log((k-1)a(k)))$. Since $z_0 = \frac{k-1}{g(k)}$, from Equation (6), we get $z_0 = \Theta\left(\frac{k}{\log(ka(k))}\right)$. Therefore, by (3) we have $b(k) = \Theta\left(\alpha^k \cdot \left(\frac{k}{\log ka(k)}\right)^{k-1}\right)$ for some constant α.

Second, consider the case where $(k-1)a(k)$ is bounded by some constants. So by (6), we have $g(k) = \Theta(1)$. Therefore $z_0 = \Theta(k)$ which again implies $b(k) = \Theta\left(\alpha^k \cdot k^{k-1}\right)$ for some constant α.

Third, we consider the case where $\lim_{k\to\infty}(k-1)a(k) = 0$. We focus on the Taylor series W_0 of W around 0. It can be found using the Lagrange inversion and is given by

$$W_0(x) = \sum_{i=1}^{\infty} \frac{(-i)^{i-1}}{i!}x^i = x - x^2 + O(1)x^3.$$

Thus, for k large enough $g(k) = \Theta((k-1)a(k))$. Hence, $z_0 = \Theta(1/a(k))$. Once again that implies $b(k) = \Theta\left(\alpha^k \cdot \frac{1}{ka(k)^k}\right)$ for some constant α. □

Lemma 3. *For any non-negative sequences* $(n_i)_{i=1}^P$, $(m_i)_{i=1}^P$, *and for any positive increasing sequence* $(q_i)_{i=1}^P$, *define* $A_{i,j} := n_1q_1 + \ldots + n_{i-1}q_{i-1} + j \cdot q_i$ *for* $1 \le i \le P, 1 \le j \le n_i$ *and* $B_{i,j} := m_1q_1 + \ldots + m_{i-1}q_{i-1} + j \cdot q_i$ *for* $1 \le i \le P, 1 \le j \le m_i$. *Then, it holds that*

$$\sum_{i=1}^{P}\sum_{j=1}^{m_i}(A_{i,n_i} + j \cdot q_i)^k \le \mu_k \sum_{i=1}^{P}\sum_{j=1}^{n_i} A_{i,j}^k + \lambda_k \sum_{i=1}^{P}\sum_{j=1}^{m_i} B_{i,j}^k,$$

where $\mu_k = \frac{k+1}{k+2}$ *and* $\lambda_k = \Theta(\alpha^k(k+1)^k)$ *for some constant* α.

Proof. Denote $r_i = 1/q_i$ for $1 \le i \le P$. The inequality is equivalent to

$$\sum_{i=1}^{P} r_i \sum_{j=1}^{m_i} q_i(A_{i,n_i} + j \cdot q_i)^k \le \mu_k \sum_{i=1}^{P} r_i \sum_{j=1}^{n_i} q_i A_{i,j}^k + \lambda_k \sum_{i=1}^{P} r_i \sum_{j=1}^{m_i} q_i B_{i,j}^k$$

For convenience set $r_{P+1} = 0$. This inequality could be written as

$$\sum_{i=1}^{P}(r_i - r_{i+1})\left[\sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t(A_{t,n_t} + j \cdot q_t)^k\right]$$
$$\leq \sum_{i=1}^{P}(r_i - r_{i+1})\left[\mu_k \sum_{t=1}^{i}\sum_{j=1}^{n_t} q_t A_{t,j}^k + \lambda_k \sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t A_{t,j}^k\right]$$

As $(r_i)_{i=1}^P$ is decreasing a sequence (so $r_i - r_{i+1} \geq 0\ \forall 1 \leq i \leq P$), it is sufficient to prove that for all $1 \leq i \leq P$,

$$\sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t(A_{t,n_t} + j \cdot q_t)^k \leq \mu_k \sum_{t=1}^{i}\sum_{j=1}^{n_t} q_t A_{t,j}^k + \lambda_k \sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t A_{t,j}^k. \tag{7}$$

Fix an index i. For convenience set $A_{0,j} = 0$ for any j. By Lemma 2, we have

$$(k+1)q_t A_{t,j}^k \geq A_{t,j}^{k+1} - (A_{t,j} - q_t)^{k+1} = A_{t,j}^{k+1} - A_{t,j-1}^{k+1} \quad \forall 1 \leq t \leq P, 2 \leq j \leq n_t$$
$$(k+1)q_t A_{t,1}^k \geq A_{t,1}^{k+1} - (A_{t,1} - q_t)^{k+1} = A_{t,1}^{k+1} - A_{t-1,n_{t-1}}^{k+1} \quad \forall 1 \leq t \leq P.$$

Therefore,

$$(k+1)\sum_{t=1}^{i}\sum_{j=1}^{n_t} q_t A_{t,j}^k \geq \sum_{t=1}^{i}\left[\sum_{j=2}^{n_t}\left(A_{t,j}^{k+1} - A_{t,j-1}^{k+1}\right) + A_{t,1}^{k+1} - A_{t-1,n_{t-1}}^{k+1}\right] = A_{i,n_i}^{k+1},$$

since the sums telescope. Similarly, $(k+1)\sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t B_{t,j}^k \geq B_{i,m_i}^{k+1}$. Thus, to prove Inequality (7), it is sufficient to prove that for all $1 \leq i \leq P$,

$$\sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t(A_{t,n_t} + j \cdot q_t)^k \leq \left(\frac{\mu_k}{k+1}A_{i,n_i}^{k+1} + \frac{\lambda_k}{k+1}B_{i,m_i}^{k+1}\right)$$

Besides,

$$\sum_{t=1}^{i}\sum_{j=1}^{m_t} q_t(A_{t,n_t} + j \cdot q_t)^k \leq \sum_{t=1}^{i} m_t q_t(A_{t,n_t} + B_{i,m_i})^k \leq B_{i,m_i}(A_{i,n_i} + B_{i,m_i})^k$$

where the inequalities follow $B_{i,m_i} = \sum_{t=1}^{i} m_t q_t \geq j m_t$ for $j \leq m_t$ and $A_{i,n_i} \geq A_{t,n_t}$ for $t \leq i$.

Hence, we only need to argue that

$$B_{i,m_i}(A_{i,n_i} + B_{i,m_i})^k \leq \left(\frac{\mu_k}{k+1}A_{i,n_i}^{k+1} + \frac{\lambda_k}{k+1}B_{i,m_i}^{k+1}\right) \tag{8}$$

Choose $\mu_k = \frac{k+1}{k+2}$ and apply case (1b) of Lemma 1 (now $a(k) = \frac{(k+1)}{k(k+2)}$ and $(k-1)a(k)$ is bounded by a constant), we deduce that: for $\lambda_k = \Theta(\alpha^k(k+1)^k)$ where α is a constant, Inequality (8) holds. □

Lemma 4. *Let* $\mathbf{x}$ *be an assignment of jobs to machines. Then, the* SPT *policy minimizes the* ℓ_k*-norm of job completion times with respect to this assignment among all feasible policies.*

Proof. Consider a machine i and let N be the number of jobs assigned to i by the profile $\mathbf{x}$. These N jobs are renamed in order of non-decreasing processing times, and since we fixed machine i, for convenience we drop index i in the processing times. So we denote the N processing times as $p_1 \leq p_2 \leq \ldots \leq p_N$. In any schedule of those jobs, there exist distinct jobs with completion times at least $p_1, p_1 + p_2, \ldots, p_1 + \ldots + p_h$. Hence, the ℓ_k-norm on the completion times of such jobs is at least $\left(\sum_{j=1}^{h} (p_1 + \ldots + p_j)^k\right)^{1/k}$, which is attained by the SPT policy. □