The Price of Anarchy for Minsum Related Machine Scheduling

Ruben Hoeksma and Marc Uetz

University of Twente, Dept. Applied Mathematics, P.O. Box 217, 7500AE Enschede, The Netherlands, {r.p.hoeksma, m.uetz}@utwente.nl

Abstract. We address the classical uniformly related machine scheduling problem with minsum objective. The problem is solvable in polynomial time by the algorithm of Horowitz and Sahni. In that solution, each machine sequences its jobs shortest first. However when jobs may choose the machine on which they are processed, while keeping the same sequencing rule per machine, the resulting Nash equilibria are in general not optimal. The price of anarchy measures this optimality gap. By means of a new characterization of the optimal solution, we show that the price of anarchy in this setting is bounded from above by 2. We also give a lower bound of $e/(e-1) \approx 1.58$. This complements recent results on the price of anarchy for the more general unrelated machine scheduling problem, where the price of anarchy equals 4. Interestingly, as Nash equilibria coincide with shortest processing time first (SPT) schedules, the same bounds hold for SPT schedules. Thereby, our work also fills a gap in the literature.

1 Introduction

The minsum related machine scheduling problem is one of the classical models in the area of scheduling. It has been solved already in the 1960s [5]. Given are n jobs with non-preemptive processing requirements, a set of m parallel machines with different processing speeds, the goal is to find a schedule that minimizes the sum of job completion times. In the 3-field notation of Graham et al. [8] the problem is denoted $Q||\sum C_j$. The problem is a special case of the more general unrelated machine scheduling problem $R||\sum C_j$, where the processing times of jobs on machines are represented by an arbitrary $n \times m$ matrix. The related machine problem is solved in $O(n \log nm)$ computation time by the MFT algorithm of Horowitz and Sahni [10]. The MFT algorithm is a refinement of the simple matching solution presented earlier by Conway et al. [5, pp. 78-79]. The MFT algorithm computes the optimal assignment of jobs to machines by considering them in the order longest processing time first (LPT), and the jobs eventually assigned to a given machine are then sequenced in the order shortest processing time first (SPT).

In this paper we are interested in the same problem, but in a **decentralized setting** where there is no central authority that assigns jobs to machines. Instead, jobs themselves choose the machine on which they want to be processed.

Any job j seeks to minimize its own completion time C_j, and does not care about the central objective function $\sum C_j$. This results in an n-agent strategic game where the strategy space of any job-agent is the set of machines. This game is well-defined once we determine how jobs are locally sequenced on each machine. Here we only consider the local sequencing rule that is locally optimal for the global objective $\sum C_j$, that is, the jobs on each machine are processed in order of shortest processing time first (SPT). In spite of doing the optimal thing locally, Nash equilibria of the resulting game do not necessarily lead to globally optimal solutions for the objective $\sum C_j$. This optimality gap is what we are interested in. Notice that the problem that we have described so far is an example of a coordination mechanism as defined by Christodoulou et al. [3], who suggested to use local sequencing rules per machine in order to influence the dynamics of the game and thereby the quality of the corresponding equilibrium outcomes.

The **price of anarchy** is being used since about a decade to measure the deterioration of system performance caused by the lack of central coordination [13,16]. It is defined by relating the quality of the worst possible Nash equilibrium to the quality of the globally optimal solution. Here, the metric for the quality of a solution is in terms of the central objective function, in our case $\sum C_j$. In the economic literature the central objective function is rather called social choice function [15]. In our case it is utilitarian, which means that the social choice function $\sum C_j$ is simply the sum of the valuation functions of the agents C_j.

For games with utilitarian social choice function, Roughgarden [17] recently introduced the concept of *smoothness* of games and its consequences for **robust price of anarchy** bounds. He points out that many of the existing price of anarchy bounds can actually be deduced from smoothness of the underlying games, and he shows that the corresponding bounds not only hold for pure Nash equilibria, but extend to mixed Nash equilibria, correlated equilibria as defined by Aumann [1], and even beyond.

The **contribution of this paper** is an analysis of the price of anarchy for the minsum related machine scheduling game as described above. More specifically, our main result is a proof that the pice of anarchy is at most 2. This analysis also extends beyond pure Nash equilibria in the same way as in [17], even though it is not exactly a smoothness argument in the sense of Roughgarden's definition in [17]. We also give a parametric example to show that the price of anarchy cannot be less than $e/(e-1) \approx 1.58$.

An interesting aspect of our work is that also the pure Nash equilibria can easily be computed in polynomial time through **SPT schedules**. In fact, it is well known that Nash equilibria are obtained as solutions of the Ibarra and Kim algorithm [11] when machines sequence jobs in SPT order. This is even true for the more general unrelated machine scheduling problem [9,12]. When applied to the related machine scheduling problem considered here, this means scheduling the jobs in SPT order, and when a job is scheduled it is placed on the machine that minimizes its completion time C_j[1]. This results in a pure Nash equilibrium

[1] This is not the same as the "SPT schedules" as discussed by Horowitz and Sahni [10, p. 321], as they assign jobs in SPT order in a greedy list scheduling fashion,

of the game, as no job has the possibility to improve its completion time by changing to another machine. Hence, we only need to compare optimum and Nash equilibrium solutions, none of which is blemished by NP-completeness. In a first instant we therefore thought the problem was trivial. Yet we first needed a new characterization of the optimal solution to get the job done. In any case, our results also show that SPT schedules can miss the optimum by no more than a factor 2, and can be as bad as $e/(e-1)$ times the optimum.

It is also worth mentioning that the literature related to analyzing the price of anarchy for scheduling problems has almost exclusively concentrated on the egalitarian[2] **makespan objective** $C_{\max}(= \max_j C_j)$ as social choice function [2,3,7,12,13,18]. The fact that most of the literature focusses on the makspan has potentially two reasons. First, this is the model that has been originally proposed by Koutsoupias and Papadimitriou [13]. Second, makespan scheduling is akin to load balancing, with applications for example in internet routing protocols [16]. Yet it is surprising that utilitarian social choice functions have hardly received any attention from the algorithms community, given that the model is certainly not less attractive from an application perspective.

We are aware of only two references that are very closely **related to our work**, these are the recent papers by Correa and Queyranne [6] and Cole et al. [4]. Both papers address the same problem as we do, but with additional job weights w_j and in the more general context of unrelated machine scheduling, $\mathrm{R}||\sum w_j C_j$. Their objective is thus weighted utilitarian. One of the main results in both papers is the proof that the price of anarchy equals 4 when machines sequence their jobs locally optimal, that is, according to nonincreasing ratios of weight over processing time. Cole et al. [4] also give an instance which establishes a lower bound of 4 for the price of anarchy, even in the unweighted case, $\mathrm{R}||\sum C_j$. Our results nicely fit into that context.

The **organization of the paper** is as follows. In Section 2 we briefly recap the algorithm of Horowitz and Sahni [10]. We then present a new characterization of optimal solutions, which is crucial for the subsequent analysis. In Section 4 we show that the price of anarchy is not greater than 2. The basic proof idea is akin to the arguments for showing $(2,0)$-smoothness of the game, but we crucially need the characterization of optimal solutions. Hence it is at best a relaxed sort of smoothness. Section 5 describes a parametric instance, for which we show that it's a price of anarchy is equal to $e/(e-1) > 1.5819$. We conclude with some further remarks in Section 6.

---

that is, to the machine that minimizes the jobs *starting* time. When doing that, the resulting SPT schedule can be arbitrarily far away from the optimum. When we refer to SPT schedules we refer to greedy list scheduling in SPT order, but jobs are placed on the machine that minimizes the completion time C_j.

[2] See Myerson [14] for a discussion of utilitarian and egalitarian social choice functions. The interpretation of $C_{\max}$ as egalitarian indeed makes sense in models where the objectives of the job-agents is the total load of the machine they are processed on, as for example in [13].

2 Characterization of Optimal Solutions

In this section we briefly recap the MFT algorithm of Horowitz and Sahni [10] and establish a new characterization for optimal solutions for minsum related machine scheduling. This characterization is crucial to our analysis in Section 4.

Throughout this paper we denote by J the set of n jobs and by M the set of m machines. Each job j has a length p_j and each machine i has a speed s_i. The processing requirement of job j on machine i is equal to $p_{ij} = p_j/s_i$. W.l.og. assume that $p_1 \leq p_2 \leq \cdots \leq p_n$ and $s_1 \leq s_2 \leq \cdots \leq s_m$. We assume ties on the ordering are broken consistently and that this is done based on index.

For the single machine case it is clear that the contribution of a job can be measured by its position in the schedule and its processing time. This follows from rewriting the objective function as follows. Let φ be an ordering of the jobs and let $\varphi(k)$ denote the k-th job in this ordering, then $\sum_{k=1}^{n} C_{\varphi(k)} = \sum_{k=1}^{n}\sum_{l=1}^{k} p_{\varphi(l)} = \sum_{k=1}^{n}(n-k+1)p_{\varphi(k)}$. Hence the only optimal schedules are schedules that schedule the jobs in order of nondecreasing processing time, as these match large p_j to small values $(n-k+1)$. The same idea can be extended to the case of parallel machines, even with speeds, resulting in the following *Minimum Mean Flow Time* (MFT) algorithm [10].

Algorithm 1: MFT Algorithm for problem $Q||\sum C_j$

For each machine i set $h_i = 0$
while *Not all jobs are placed* **do**
  Take from the unscheduled jobs the longest job j
  Assign job j to the machine with the smallest value of $(h_i + 1)/s_i$
  For that machine update $h_i = h_i + 1$
Sort the jobs on each machine in SPT order

Similar to the single machine case, the different values $(h_i + 1)/s_i$ are the values for a job's possible positions in the schedule, as in general, the x-th last job on a machine contributes to the objective value x times its processing time divided by the machine speed. The algorithm assigns the currently longest unscheduled job to the machine with the currently smallest position value.

Theorem 1 ([10]). *Any optimal schedule for* $Q||\sum C_j$ *can be computed by the MFT algorithm with the proper tie breaking rule.*

Since any optimal solution has the jobs on each machine sequenced in SPT order, we can identify a schedule by denoting for each job on which machine it is scheduled. Therefore we identify a schedule with an n-vector σ where σ_j is the machine on which job j is scheduled.

Next, let $h^\sigma(j)$ be the vector such that $h_i^\sigma(j) = |\{k > j | \sigma_k = i\}|$, indicating the number of jobs on machine i in schedule σ that have higher index than j.

Now any schedule σ is optimal if and only if

$$\frac{h^\sigma_{\sigma_j}(j)+1}{s_{\sigma_j}} \leq \frac{h^\sigma_i(j)+1}{s_i} \text{ for all jobs } j \text{ and all machines } i \ . \tag{1}$$

This because, for all machines i, $(h^\sigma_{\sigma_j}(j)+1)/s_i$ is the position value of i upon placement of job j in the MFT algorithm. This needs to be minimized for all j by any optimal schedule σ. The following lemma provides our new characterization of optimal solutions.

Lemma 1. *A schedule σ is optimal for $\mathrm{Q}||\sum C_j$ if and only if*

$$\frac{h^\sigma_i(j)+1}{s_i} \geq \frac{h^\sigma_\ell(j)}{s_\ell} \textit{ for all machines } i \textit{ and } \ell \ . \tag{2}$$

Proof. We show that (2) is true if and only if (1) is true. Let σ be an optimal schedule. Note that $h^\sigma_i(j) \geq h^\sigma_i(k)$ for all machines i and all jobs $k \geq j$. We therefore get from (1) that

$$\frac{h^\sigma_i(j)+1}{s_i} \geq \frac{h^\sigma_i(k)+1}{s_i} \geq \frac{h^\sigma_{\sigma_k}(k)+1}{s_{\sigma_k}}$$

for all machines i and all jobs $k \geq j$. Since for any machine ℓ either $h^\sigma_\ell(j) = 0$, or there is a job $k > j$ such that $\sigma_k = \ell$ and $h^\sigma_\ell(j) = h^\sigma_{\sigma_k}(j) = h^\sigma_{\sigma_k}(k)+1$, it follows that

$$\frac{h^\sigma_i(j)+1}{s_i} \geq \frac{h^\sigma_\ell(j)}{s_\ell} \text{ for all machines } i \text{ and } \ell \ .$$

Now let σ be a schedule that satisfies (2) and suppose it does not satisfy (1). Then there exist $j \in J$ and $i \in M$ such that

$$\frac{h^\sigma_{\sigma_j}(j)+1}{s_{\sigma_j}} > \frac{h^\sigma_i(j)+1}{s_i} \ ,$$

but then we get for job $j-1$ that

$$\frac{h^\sigma_{\sigma_j}(j-1)}{s_{\sigma_j}} = \frac{h^\sigma_{\sigma_j}(j)+1}{s_{\sigma_j}} > \frac{h^\sigma_i(j)+1}{s_i} = \frac{h^\sigma_i(j-1)+1}{s_i} \ ,$$

which contradicts (2). □

A intuitive interpretation for (2) is that, when applying the MFT algorithm, a job that is placed on a machine can not get a better position than the jobs already placed on a machine. While it is intuitive that this is indeed a necessary condition for the optimal solution, the intuition that it is also sufficient is not that clear. In that sense, it is indeed a nontrivial reformulation of (1).

3 Coordination Mechanism and Nash Equilibria

For the remainder of this paper we compare the optimal solution from Section 2 to outcomes of the scheduling game for $Q||\sum C_j$ where each job can individually choose on which machine it will be scheduled and machines sequence jobs in SPT order. The jobs act selfishly, each trying to minimize its own completion time. Nash equilibria are considered the natural outcomes of the resulting strategic game. The price of anarchy, defined in [13], compares the objective value of an optimal schedule to the objective value of a worst possible Nash equilibrium schedule. The resulting game for $Q||\sum C_j$ is a coordination mechanism in the sense of Christodolou et al. [3], where using SPT locally per machine proposes itself because it is locally optimal.

We denote schedules in the same way as in Section 2, but with respect to Nash equilibria, σ represents the *strategy profile* of the job-agents such that σ_j is the machine chosen by job j. Furthermore, σ_{-j} denotes the $(n-1)$-vector obtained from σ by deleting σ_j, so that $\sigma = (\sigma_j, \sigma_{-j})$. For the problem $Q||\sum C_j$ with SPT as local scheduling rule, Nash equilibria are defined as follows.

Definition 1 (Nash equilibrium). *A strategy profile $\sigma = (\sigma_j, \sigma_{-j})$ is a Nash equilibrium if and only if for all jobs j,*

$$\sum_{\substack{k \leq j \\ \sigma_k = \sigma_j}} \frac{p_k}{s_{\sigma_j}} \leq \sum_{\substack{k < j \\ \sigma_k = i}} \frac{p_k}{s_i} + \frac{p_j}{s_i} \quad \textit{for all machines } i \ . \tag{3}$$

It is well known [9] that the *Ibarra-Kim* algorithm [11] constructs all Nash equilibria depending on the way ties are broken. For uniformly related machines the algorithm is described as follows.

Algorithm 2: Ibarra-Kim Algorithm for problem $Q||\sum C_j$

while *Not all jobs are placed* **do**
- Take from the unscheduled jobs the shortest job k
- Let machine l be the machine where job k has minimal completion time
- Schedule job k directly after the jobs already scheduled on machine l

The Ibarra-Kim algorithm was originally designed as an approximation algorithm for unrelated machine scheduling [11]. To the best of our knowledge the performance of the resulting schedules for the related machine problem $Q||\sum C_j$ has not yet been analyzed, most probably because the problem to find optimal solutions was settled long before in [5].

4 Upper Bound on the Price of Anarchy

In this Section we establish an upper bound on the price of anarchy for minsum related machine scheduling. Our proof is (in retrospect) akin to a smoothness

argument for cost-minimization (=utilitarian) games, as introduced by Roughgarden [17].

Definition 2 ([17] Smooth Games). *A cost-minimization game is* (λ, μ)-smooth *if for every two outcomes* ν *and* σ*,*

$$\sum_{j=1}^{n} C_j(\sigma_j, \nu_{-j}) \leq \lambda \cdot \sum_{j=1}^{n} C_j(\sigma) + \mu \cdot \sum_{j=1}^{n} C_j(\nu) \ . \tag{4}$$

If a utilitarian game is (λ, μ)-smooth with $\lambda \geq 0$ and $\mu < 1$, it follows that for any Nash equilibrium ν and optimal solution σ

$$\sum_{j=1}^{n} C_j(\nu) \leq \sum_{j=1}^{n} C_j(\sigma_j, \nu_{-j}) \leq \lambda \cdot \sum_{j=1}^{n} C_j(\sigma) + \mu \cdot \sum_{j=1}^{n} C_j(\nu) \ . \tag{5}$$

From (5) it follows directly that $\frac{\lambda}{1-\mu}$ is an upper bound on the price of anarchy for any (λ, μ)-smooth game. Roughgarden [17] defines the *robust price of anarchy* as the least upper bound on the price of anarchy that is provable through a smoothness argument.

Definition 3 ([17] Robust PoA). *The robust price of anarchy of a cost-minimization game is*

$$\inf \left\{ \frac{\lambda}{1-\mu} \middle| \text{the game is } (\lambda, \mu)\text{-smooth} \right\} \ .$$

Instead of proving (4) for *any* two outcomes ν and σ, we crucially need the characterization of the optimal solution from Lemma 1 and therefore will prove (4) with σ restricted to be an optimal solution. However, note that the resulting bound on the price of anarchy also extends to (mixed) Nash equilibria, correlated equilibria or no-regret sequences (see [17]) when (4) only holds for arbitrary strategy profiles ν and an optimal solution σ.

In the following, let therefore σ be an optimal schedule resulting from the MFT algorithm, and recall that for the objective value in the optimal solution σ we have

$$\sum_{j=1}^{n} C_j(\sigma) = \sum_{j=1}^{n} \left(h^{\sigma}_{\sigma_j}(j) + 1 \right) \frac{p_j}{s_{\sigma_j}} \ .$$

The next Theorem is the main result of this paper.

Theorem 2. *The price of anarchy for the minsum related machine scheduling problem* $\mathrm{Q}||\sum C_j$ *with SPT as local sequencing rule is no greater than 2.*

Proof. We show that the game is "$(2, 0)$-smooth", by showing that

$$\sum_{j=1}^{n} C_j(\sigma_j, \nu_{-j}) \leq 2 \sum_{j=1}^{n} C_j(\sigma) \tag{6}$$

for an optimal schedule σ and any strategy profile ν.

Let $J_i(\sigma) = \{j|\sigma_j = i\}$ be the set of jobs scheduled on machine i in the optimal solution σ, likewise let $J_i(\nu) = \{j|\nu_j = i\}$ be the set of jobs scheduled on machine i in schedule ν. For any job j in $J_i(\sigma)$, its completion time $C_j(\sigma_j, \nu_{-j})$ consists of the processing times of all jobs that are on machine i in ν and that have smaller index than j, plus its own processing time on machine i. Summing the completion times of all jobs that are on machine i in the optimal solution gives us

$$\begin{aligned}\sum_{j\in J_i(\sigma)} C_j(\sigma_j, \nu_{-j}) &= \sum_{j\in J_i(\sigma)} \left(\frac{p_j}{s_i} + \sum_{\substack{k\in J_i(\nu)\\ k<j}} \frac{p_k}{s_i} \right) \\ &= \sum_{j\in J_i(\sigma)} \frac{p_j}{s_i} + \sum_{j\in J_i(\sigma)} \sum_{\substack{k\in J_i(\nu)\\ k<j}} \frac{p_k}{s_i} \ . \end{aligned} \tag{7}$$

Note that the number of times that a job k is counted on the right hand side of (7) equals the number of jobs with higher index than j on machine i in the optimal solution, times $\frac{1}{s_i}$. In other words, the second part of (7) can be rewritten as

$$\sum_{j\in J_i(\sigma)} \sum_{\substack{k\in J_i(\nu)\\ k<j}} \frac{p_k}{s_i} = \sum_{k\in J_i(\nu)} h_i^\sigma(k) \cdot \frac{p_k}{s_i} \ .$$

This gives us

$$\sum_{j\in J_i(\sigma)} C_j(\sigma_j, \nu_{-j}) = \sum_{j\in J_i(\sigma)} \frac{p_j}{s_i} + \sum_{k\in J_i(\nu)} h_i^\sigma(k) \cdot \frac{p_k}{s_i} \ .$$

Now, note that by definition $\sigma_j = \nu_k = i$, so

$$\sum_{j\in J_i(\sigma)} C_j(\sigma_j, \nu_{-j}) = \sum_{j\in J_i(\sigma)} \frac{p_j}{s_{\sigma_j}} + \sum_{k\in J_i(\nu)} h_{\nu_k}^\sigma(k) \cdot \frac{p_k}{s_{\nu_k}} \ .$$

Summing over all i leads to

$$\begin{aligned}\sum_{j=1}^{n} C_j(\sigma_j, \nu_{-j}) &= \sum_{i=1}^{m} \sum_{j\in J_i(\sigma)} C_j(\sigma_j, \nu_{-j}) \\ &= \sum_{i=1}^{m} \sum_{j\in J_i(\sigma)} \frac{p_j}{s_{\sigma_j}} + \sum_{i=1}^{m} \sum_{k\in J_i(\nu)} h_{\nu_k}^\sigma(k) \cdot \frac{p_k}{s_{\nu_k}} \\ &= \sum_{j=1}^{n} \frac{p_j}{s_{\sigma_j}} + \sum_{j=1}^{n} h_{\nu_j}^\sigma(j) \cdot \frac{p_j}{s_{\nu_j}} \ . \end{aligned}$$

From Lemma 1 we know

$$\sum_{j=1}^{n} h_{\nu_j}^\sigma(j) \cdot \frac{p_j}{s_{\nu_j}} \leq \sum_{j=1}^{n} \left(h_{\sigma_j}^\sigma(j) + 1 \right) \cdot \frac{p_j}{s_{\sigma_j}} = \sum_{j=1}^{n} C_j(\sigma) \ . \tag{8}$$

Also, the completion time of any job is at least its processing time on the machine it is scheduled on, so

$$\sum_{j=1}^{n} \frac{p_j}{s_{\sigma_j}} \leq \sum_{j=1}^{n} C_j(\sigma) \ . \tag{9}$$

Combining the above, we get

$$\sum_{j=1}^{n} C_j(\sigma_j, \nu_{-j}) \leq 2 \sum_{j=1}^{n} C_j(\sigma) \qquad \text{for all strategy profiles } \nu \ .$$

□

5 Lower Bound on the Price of Anarchy

In this Section we describe a parametric instance which has price of anarchy equal to $e/(e-1)$. The Nash equilibrium is the schedule with all jobs on the fastest machine (which is easily shown to be an upper bound on the quality of Nash equilibria in general, so in that sense, this is a worst case scenario).

Instance 1 *Let $\mathcal{I}$ be the parametric group of instances $I(s)$ that satisfy the following. $I(s)$ has m machines, one of which has speed $s > 1$ and all the other machines have speed 1. All speeds are integer. Furthermore, $I(s)$ has $n = m+s-1$ jobs, with length equal to*

$$p_j = \begin{cases} 1 & \text{if } 1 \leq j \leq s \\ x^{j-s} & \text{if } s+1 \leq j \leq n \end{cases} \ ,$$

where $x = s/(s-1)$.

Lemma 2. *Instances from $\mathcal{I}$ have a Nash equilibrium with all jobs on the fastest machine.*

Proof. In the schedule with all jobs in SPT order on the fastest machine, the completion time of a job $j < s$ is equal to

$$C_j = \sum_{k=1}^{j} \frac{p_k}{s} = \sum_{k=1}^{j} \frac{1}{s} = \frac{j}{s} \leq 1 \ . \tag{10}$$

For a job $j \geq s$, the completion time is equal to

$$C_j = \sum_{k=1}^{j} \frac{p_k}{s} = \frac{s-1}{s} + \sum_{k=s}^{j} \frac{\left(\frac{s}{s-1}\right)^{k-s}}{s}$$

$$= \frac{1}{s}\left(s-1+\sum_{k=0}^{j-s}\left(\frac{s}{s-1}\right)^k\right) = \frac{1}{s}\left(s-1+\frac{\left(\frac{s}{s-1}\right)^{j-s+1}-1}{\left(\frac{s}{s-1}\right)-1}\right)$$
$$= \frac{1}{s}\left(s-1+(s-1)\left(\frac{s}{s-1}\right)^{j-s+1}-(s-1)\right)$$
$$= \left(\frac{s}{s-1}\right)^{j-s} = p_j \ . \tag{11}$$

So the Nash equilibrium condition (3) holds, as all other machines have speed 1.

We use this to compute a lower bound on the price of anarchy.

Theorem 3. *The price of anarchy for the minsum related machine scheduling problem* $Q||\sum C_j$ *with SPT local scheduling rule is no less than* $e/(e-1) \approx 1.58$.

Proof. Consider instances $I(s)$ from $\mathcal{I}$ as defined above. In the optimal solution the s longest jobs are on the fastest machine. All other jobs are on a slow machine. So the objective value in the optimal solution is equal to

$$\begin{aligned}
\mathrm{OPT}(I(s)) &= \sum_{j=1}^{s-1} p_j + \sum_{j=s}^{n-s} p_j + \sum_{j=n-s+1}^{n}\sum_{k=n-s+1}^{j}\frac{p_k}{s} \\
&= \sum_{j=1}^{s-1} p_j + \sum_{j=s}^{n-s} x^{j-s} + \sum_{j=n-s+1}^{n}\sum_{k=n-s+1}^{j}\frac{x^{k-s}}{s} \\
&= \sum_{j=1}^{s-1} 1 + \sum_{j=0}^{n-2s} x^j + \sum_{j=n-s+1}^{n}\frac{1}{s}\left(\sum_{k=0}^{j-s} x^k - \sum_{k=0}^{n-2s} x^k\right) \\
&= s-1+(s-1)x^{n-2s+1} - (s-1) + \sum_{j=n-s+1}^{n}\left(x^{j-s}-x^{n-2s}\right) \\
&= (s-1)x^{n-2s+1} + \sum_{j=n-2s+1}^{n-s} x^j - \sum_{j=n-s+1}^{n} x^{n-2s} \\
&= (s-1)x^{n-2s+1} + (s-1)x^{n-s+1} - (s-1)x^{n-2s+1} - sx^{n-2s} \\
&= (s-1)x^{n-s+1} - (s-1)x^{n-2s+1} \ .
\end{aligned} \tag{12}$$

From Lemma 2 we know that the schedule with all jobs on the fastest machine is a Nash equilibrium. From (10) and (11) we know that the completion time of the jobs in this schedule is equal to

$$C_j = \begin{cases} \frac{j}{s} & \text{if } j \le s-1 \\ \left(\frac{s}{s-1}\right)^{j-s} & \text{otherwise} \end{cases} \ .$$

From this we compute the objective value in the Nash equilibrium

$$\begin{aligned}
\mathrm{NE}(I(s)) &= \sum_{j=1}^{s-1} \frac{j}{s} + \sum_{j=s}^{n} x^{j-s} \\
&= \frac{s(s-1)}{2s} + \sum_{j=0}^{n-s} x^{j} \\
&= \frac{(s-1)}{2} + (s-1)x^{n-s+1} - (s-1) \\
&= (s-1)x^{n-s+1} - \frac{(s-1)}{2} \ . \qquad (13)
\end{aligned}$$

Combining (12) and (13) gives us the price of anarchy:

$$\begin{aligned}
\mathrm{PoA}(I(s)) &= \frac{(s-1)x^{n-s+1} - \frac{(s-1)}{2}}{(s-1)x^{n-s+1} - (s-1)x^{n-2s+1}} \\
&= \frac{x^{n-s+1} - \frac{1}{2}}{x^{n-s+1} - x^{n-2s+1}} \\
&= \frac{x^{s} - \frac{1}{2}x^{-(n-2s+1)}}{x^{s} - 1} \\
&= \frac{\left(\frac{s}{s-1}\right)^{s} - \frac{1}{2}\left(\frac{s}{s-1}\right)^{-(n-2s+1)}}{\left(\frac{s}{s-1}\right)^{s} - 1} \ . \qquad (14)
\end{aligned}$$

Now, if we let n go to infinity, (14) becomes:

$$\lim_{n\to\infty} \mathrm{PoA}(I(s)) = \frac{\left(\frac{s}{s-1}\right)^{s}}{\left(\frac{s}{s-1}\right)^{s} - 1}\,, \qquad (15)$$

and letting also s go to infinity, (15) goes to $e/(e-1) \approx 1.58$. □

6 Concluding Remarks

Of course, the question remains what the truth is concerning the price of anarchy for the considered problem, which we could bound in the interval $[1.58, 2]$. This gap may be due to the fact that the upper bound holds for more general equilibria than only pure Nash equilibria. While for the parametric instances from Theorem 3, scheduling all jobs on the fastest machine is even a dominant strategy equilibrium.

Note that it is indeed possible for mixed Nash equilibria to induce (significantly) worse price of anarchy than pure Nash equilibria. This can be seen by the simple example of two identical machines with two identical jobs. For such an instance pure Nash equilibria are optimal solutions. However, the randomized

schedule where each job choses each machine with equal probability of 1/2 is a mixed Nash equilibrium, and yields an expected objective value 5/4 times the optimal value.

All this leaves open the possibility that indeed 2 would be the true value of the robust price of anarchy, while the true value for the (pure) price of anarchy is $e/(e-1)$. We believe however that an improvement on the upper bound of 2 is possible, because either of the two terms that appears in our analysis in (8) and (9) can be equal to the optimum value, but we have not been able to construct instances where both inequalities are tight. Neither have we been able (so far) to offset the two terms against each other, which might be a feasible approach for improving our analysis for the upper bound.

References

1. R. J. Aumann. Subjectivity and correlation in randomized strategies. *J. Math. Econom.*, 1(1):67–96, 1974.
2. Y. Azar, K. Jain, and V. Mirrokni. (Almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proceedings 19th SODA*, pages 323–332. ACM/SIAM, 2008.
3. G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. *Theoret. Comput. Sci.*, 410(36):3327–3336, 2009.
4. R. Cole, J. R. Correa, V. Gkatzelis, V. Mirrokni, and N. Olver. Inner Product Spaces for MinSum Coordination Mechanisms. In *Proceedings 43rd STOC*, pages 539-548, ACM, 2011.
5. R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling.* Addison-Wesley Publishing Co., Reading, 1967.
6. J. Correa and M. Queyranne. Efficiency of Equilibria in Restricted Uniform Machine Scheduling with MINSUM Social Cost. *Manuscript*, 2010.
7. A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Trans. Algorithms*, 3(1):Art. 4, 17, 2007.
8. R. Graham, E. Lawler, J. Lenstra, and A. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5(2):287–326, 1979.
9. B. Heydenreich, R. Müller, and M. Uetz. Games and mechanism design in machine scheduling - An introduction. *Production and Operations Management*, 16(4):437–454, 2007.
10. E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM*, 23(2):317–327, 1976.
11. O. Ibarra and C. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM*, 24(2):280–289, 1977.
12. N. Immorlica, L. Li, V. S. Mirrokni, and A. S. Schulz. Coordination mechanisms for selfish scheduling. *Theoret. Comput. Sci.*, 410(17):1589–1598, 2009.
13. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings 16th STACS*, LNCS 1563, pages 404–413. Springer-Verlag, 1999.
14. R. B. Myerson. Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica*, 49(4):pp. 883–897, 1981.
15. R. B. Myerson. *Game theory - Analysis of conflict.* Harvard University Press, Cambridge, MA, 1991.

16. C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings 33rd STOC*, pages 749–753. ACM, 2001.
17. T. Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings 41st STOC*, pages 513–522, ACM, 2009.
18. L. Yu, K. She, H. Gong, and C. Yu. Price of anarchy in parallel processing. *Inform. Process. Lett.*, 110(8-9):288–293, 2010.