The price of anarchy for utilitarian scheduling games on related machines

Ruben Hoeksma[a], Marc Uetz[b]

[a] *Universität Bremen, FB3: Mathematik/Informatik, Bibliotheksstr., 28359 Bremen, Germany*
[b] *Dept. Applied Mathematics, University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands*

Abstract

We present bounds on the efficiency of Nash equilibria in a scheduling game where jobs are players who choose a machine out of a set of machines to be processed on. Machines may have different speeds, and sequence the jobs in shortest processing time first order. When players selfishly choose a machine to minimize their own completion time, we analyze the price of anarchy for the sum of the completion times of the jobs. We show that it is bounded from below by $e/(e-1) \approx 1.58$ and from above by 2.

Keywords: price of anarchy, scheduling, utilitarian, related machines

1. Introduction

In settings with multiple noncooperative decision makers, it is well known that the lack of central coordination may cause a loss in overall efficiency. The systematic study of what is now known as the *price of anarchy* [21] was initiated in a 1999 paper by Koutsoupias and Papadimitriou [18], even though there was earlier work to address the inefficiency of equilibria, e.g., Dubey [10]. It was Koutsoupias and Papadimitriou, however, who suggested to systematically compare the worst case social cost of a solution that may arise as a Nash equilibrium to that of a solution with minimum social cost. Since then, the price of anarchy has been analyzed in many game theoretic versions of combinatorial optimization problems, most prominently auctions, network routing and scheduling. Specifically, the literature on analysis of the price of anarchy for scheduling problems is very extensive. Various models exist in which either machines or jobs are interpreted as selfish agents, and different objectives may be studied that depend on either machine loads L_i or job completion times C_j.

Following Koutsoupias and Papadimitriou [18], a lot of work was done on the makespan $C_{\max}(= \max_j C_j)$ as social cost function [2, 5, 9, 17, 18, 25]. In that model the assumption is that the objective of a job-agent is the total load L_i of the machine i it is processed on, hence minimizing the makespan models an egalitarian (*minimax*) social choice function. (See Myerson [20] for a discussion of egalitarian and utilitarian social choice functions.)

Email addresses: `hoeksma@uni-bremen.de` (Ruben Hoeksma), `m.uetz@utwente.nl` (Marc Uetz)

Later, publications have also addressed models in which job-agents j care about their own completion time C_j in a schedule, hence the order of jobs per machine plays a role [3, 6, 8, 22]. For this model, it is rather the utilitarian social cost function which makes sense, and it translates into minimizing $\sum_j C_j$, or $\sum_j w_j C_j$ when jobs have individual weights w_j.

This paper addresses one of the simplest, non-trivial utilitarian machine scheduling games: Job-agents select a machine to be processed on, with the goal to minimize their individual completion times C_j. The utilitarian social cost function is then $\sum_j C_j$. In order to define the game that is being played by the jobs, we make the natural assumption that each machine sequences its jobs in the locally optimal order for the objective $\sum_j C_j$, which is shortest processing time first (SPT).

The main result in this paper is an analysis of the price of anarchy when the machines may have different speeds. That model is denoted Q$||\sum C_j$ in the three-field scheduling notation of Graham et al. [12]. We also briefly comment on the case where all these machine speeds are identical, known as P$||\sum C_j$. More specifically, we show that the price of anarchy for the case where machines may have different speeds is at most 2 and at least $e/(e-1) \approx 1.58$. When all speeds are identical, the price of anarchy equals $3/2 - 1/(2m)$, where m is the number of machines. The latter result appears to be a bit of folklore in the community; it shows up as a special case, e.g. in [22]. We include it with a simple proof for the sake of completeness.

We believe that our results are interesting for mainly two reasons. First, an interesting feature of the problem we address is the fact that both, computation of a (pure) Nash equilibrium and an optimal solution can be done in polynomial time by simple, well known algorithms. In fact, for the problem we address it is well known that (pure) Nash equilibria are obtained as solutions of the Ibarra and Kim algorithm [16]. Moreover, an optimal solution is obtained by the *Minimum Mean Flow Time* (MFT) algorithm by Horowitz and Sahni [15]. Yet, despite this fact which seems to place the problem that we address here close to trivial, we require a new characterization of optimal solutions to get our analysis done. That characterization is interesting in its own right. Moreover, despite being a target in the community since several years, since the publication of the conference paper where our results have been announced [14], the gap for the price of anarchy has not been closed and seems to require new techniques.

As to related work, the papers by Correa and Queyranne [8] and Cole et al. [6] are very closely related to our work. Both address the same problem as we do, but with additional job weights and in the more general context of unrelated machine scheduling. One of the main results in both papers is a proof that the price of anarchy is 4 when machines sequence their jobs locally optimal, that is, according to non-increasing ratios of weight over processing time. Cole et al. [6] also give an instance which establishes a lower bound of 4 for the price of anarchy, even in the unweighted case. Our paper adds to this line of work by improved price of anarchy bounds for a special case, yet obtained by different techniques. In the meantime, our proof is known to fall into the general category of "smoothness" proofs for price of anarchy bounds; see Roughgarden [24]. We therefore present it in this context. Subsequent to our work a slight improvement of our upper bound proof has been published [26], however asymptotically it yields the same upper bound of 2. We briefly discuss that result at the end of the paper.

The organization of this paper is as follows. In Section 3 we briefly recap the *Minimum Mean Flow Time* (MFT) algorithm by Horowitz and Sahni [15]. We then present a new characterization of optimal solutions, which is crucial for the subsequent analysis. In Section 4.1 we show that the price of anarchy is not greater than 2. The proof goes by showing what is now known as $(2, 0)$-semi-smoothness of the game [1, 4]. To do that, our new characterization of optimal solutions is key. Section 4.2 then describes a parametric instance, for which we show that its a price of anarchy is (asymptotically) equal to $e/(e-1) > 1.58$. We briefly address the special case of identical machines in Section 5 and conclude with some remarks in Section 6.

2. Preliminaries

In this paper we are interested in the price of anarchy in the following, related machine scheduling game. Given are a set J of n jobs and a set M of m machines. Each job j has a processing requirement p_j and each machine i has a speed s_i. The processing time of job j on machine i is equal to p_j/s_i. Without loss of generality we assume that $p_1 \leq p_2 \leq \cdots \leq p_n$ and $s_1 \leq s_2 \leq \cdots \leq s_m$. In the case of ties on the ordering, we assume these are broken consistently and that this is done based on index. Each job j is owned by a different strategic player that tries to minimize the completion time C_j, while the social cost function is utilitarian, meaning that it is the sum of the players' utilities, which is the total completion time of the jobs $\sum C_j$. We discuss the setting where on each machine the jobs are processed in shortest processing time first (SPT) order. For this setting pure Nash equilibria always exist. While scheduling the jobs in SPT order is optimal on a local, per machine level, the resulting Nash equilibria are not globally optimal in general. We denote the strategies or choices of the players by a vector σ such that σ_j denotes the machine that job j chooses, or, in the case of mixed strategies, the probability distribution over the machines chosen by job j. Moreover, we let σ_{-j} denote the $(n-1)$-vector obtained from σ by deleting σ_j, such that $\sigma = (\sigma_j, \sigma_{-j})$. By $C_j(\sigma)$ we denote the (expected) completion time of job j given the strategy vector σ and by $C(\sigma) = \sum_{j \in J} C_j(\sigma)$ we denote the total (expected) completion time corresponding to that same strategy vector σ.

For games with utilitarian social choice function, Roughgarden [24] introduced the concept of *smoothness* of games and its consequences for *robust price of anarchy* bounds. Smoothness of a game directly implies upper bounds on the price of anarchy for pure Nash equilibria that also extends to mixed Nash equilibria, correlated equilibria, and coarse correlated equilibria [24]. In contrast to regular smooth games, our analysis for the upper bound crucially needs properties of optimal solutions. It turns out that the conditions for smoothness can be relaxed to allow the use of properties of particular solutions, without losing the generality of the bounds [1, 19, 24]. This relaxed version of smoothness has been called *semi-smoothness* by Lucier and Paes Leme [19]. Anshelevich et al. [1] and Caragiannis et al. [4] give an even more relaxed definition of semi-smoothness that is still lossless in the same sense.

Definition 1 ([1, 4] Semi-smooth games)**.** A cost-minimization game is (λ, μ)-*semi-smooth* if for every player j there exists a mixed strategy σ_j such that for every strategy vector ν,

$$\mathbb{E}_\sigma \left[\sum_{j=1}^{n} C_j(\sigma_j, \nu_{-j}) \right] \leq \lambda \cdot C(\sigma^*) + \mu \cdot C(\nu) \,, \tag{1}$$

where σ^* is an optimal solution.

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

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

where σ_j is the strategy from (1). It follows directly that $\frac{\lambda}{1-\mu}$ is an upper bound on the price of anarchy for any (λ, μ)-semi-smooth game.

3. Characterization of Optimal Solutions

In this section we briefly recap the MFT algorithm by Horowitz and Sahni [15] and establish a new characterization for optimal solutions for related machine scheduling. This characterization is crucial to our analysis in Section 4.1.

When considering only a single machine 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 the sum of completion times when processing the jobs in that order on a single machine with speed 1 is equal to

$$\sum_{k=1}^{n} C_{\varphi(k)} = \sum_{k=1}^{n} \sum_{l=1}^{k} p_{\varphi(l)} = \sum_{k=1}^{n} (n-k+1) p_{\varphi(k)}\,.$$

Hence, the only optimal schedules are schedules that schedule the jobs in order of non-decreasing processing time, as these match large p_j to small values $(n-k+1)$. This idea can be extended to the case of parallel machines, even with speeds, resulting in the MFT algorithm (Algorithm 1) [15, Algorithm 1]. In the MFT algorithm, similar to the single machine case, we make use of the values $(z_i + 1)/s_i$ which 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 requirement divided by the machines speed. The algorithm assigns the currently longest unscheduled job to the machine with the currently smallest position value.

Algorithm 1 MFT Algorithm for related machine scheduling

```
1: For each machine i set z_i = 0
2: while Not all jobs are placed do
3:     Take from the unscheduled jobs the longest job j
4:     Assign job j to the machine with the smallest value of (z_i + 1)/s_i
5:     For that machine update z_i = z_i + 1
6: end while
7: Sort the jobs on each machine in SPT order
```

Theorem 1 (Horowitz and Sahni [15]). *The MFT algorithm produces optimal schedules for* $\mathrm{Q}||\sum C_j$ *and any optimal schedule or* $\mathrm{Q}||\sum C_j$ *can be computed by the MFT algorithm with the proper tie breaking rule.*

From here on, for each machine we let the jobs assigned to it be scheduled in SPT order. That said, we identify a schedule with an n-vector σ, where σ_j is the machine on which job j is scheduled, similar to a strategy vector.

Next, let $z(\sigma, j)$ be the vector such that $z_i(\sigma, j) = |\{k > j | \sigma_k = i\}|$. It is the number of jobs on machine i in schedule σ that have higher index than j. Then, any schedule σ is optimal if and only if

$$\frac{z_{\sigma_j}(\sigma, j) + 1}{s_{\sigma_j}} \leq \frac{z_i(\sigma, j) + 1}{s_i} \tag{2}$$

for all jobs j and all machines i. This because, for all machines i, $(z_i(\sigma, j) + 1)/s_i$ is the value of the next position on machine i upon placement of job j by the MFT algorithm. Indeed, $p_j(z_{\sigma_j}(\sigma, j)+1)/s_{\sigma_j}$ is exactly the contribution of job j to the objective value in schedule σ. The sum of these contributions needs to be minimized by any optimal schedule. The following provides our new characterization of optimal solutions; we labelled it as theorem because we think it is interesting in its own right.

Theorem 2. *A schedule σ is optimal for $\mathrm{Q}||\sum C_j$ if and only if* [1]

$$\frac{z_i(\sigma, j) + 1}{s_i} \geq \frac{z_\ell(\sigma, j)}{s_\ell} \tag{3}$$

for all machines i and ℓ.

Proof. We show that (3) is true if and only if (2) is true. Let σ be an optimal schedule and let the ordering of the jobs be fixed and in SPT order. Note that $z_i(\sigma, j) \geq z_i(\sigma, k)$ for all machines i and all jobs $k \geq j$. Therefore, we have from (2) that

$$\frac{z_i(\sigma, j) + 1}{s_i} \geq \frac{z_i(\sigma, k) + 1}{s_i} \geq \frac{z_{\sigma_k}(\sigma, k) + 1}{s_{\sigma_k}},$$

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

$$\frac{z_i(\sigma, j) + 1}{s_i} \geq \frac{z_\ell(\sigma, j)}{s_\ell}$$

for all machines i and ℓ.

Now let σ be a schedule that satisfies (3) and suppose it does not satisfy (2). Then there exist a job $j \in N$ and a machine $i \in M$ such that

$$\frac{z_{\sigma_j}(\sigma, j) + 1}{s_{\sigma_j}} > \frac{z_i(\sigma, j) + 1}{s_i}.$$

However, then we have for job $j - 1$ that

$$\frac{z_{\sigma_j}(\sigma, j-1)}{s_{\sigma_j}} = \frac{z_{\sigma_j}(\sigma, j) + 1}{s_{\sigma_j}} > \frac{z_i(\sigma, j) + 1}{s_i} = \frac{z_i(\sigma, j-1) + 1}{s_i},$$

which contradicts (3). □

[1] In case of ties in the SPT ordering, there exist multiple optimal schedules, produced by interchanging symmetric jobs, jobs with equal processing times, or symmetric machines, machines with equal speed, in any optimal schedule. In this case, (2) and (3) describe optimal schedules that correspond to one particular SPT ordering and the rest can be obtained by interchanging symmetric jobs. Note that also machine symmetries are accounted for in (2) and (3).

An intuitive interpretation for (3) is that, when applying the MFT algorithm, a job that is placed on a machine can not get a better position value than the jobs already placed on any other 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 (2).

4. Price of Anarchy for the SPT Scheduling Rule

In this section we provide both an upper and a lower bound on the price of anarchy for the SPT scheduling rule for the related machine scheduling game. For the problem $\mathrm{Q}||\sum C_j$ with SPT as local scheduling rule, a strategy profile $\nu = (\nu_j, \nu_{-j})$ is a pure Nash equilibrium if and only if for all jobs j and all machines i,

$$\sum_{\substack{k \leq j \\ \nu_k = \nu_j}} \frac{p_k}{s_{\nu_j}} \leq \sum_{\substack{k < j \\ \nu_k = i}} \frac{p_k}{s_i} + \frac{p_j}{s_i}\,. \tag{4}$$

It is well known [13] that the *Ibarra-Kim* algorithm [16] constructs all Nash equilibria depending on the way ties are broken. This is even true for the more general unrelated machine scheduling problem [13, 17]. For related machines the algorithm is described in pseudo-code by Algorithm 2. The Ibarra-Kim algorithm was originally designed as

Algorithm 2 Ibarra-Kim Algorithm for problem $\mathrm{Q}||\sum C_j$

```
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
end while
```

an approximation algorithm for unrelated machine scheduling [16]. Therefore, the main result that we discuss in this paper is also an analysis of an upper and lower bound to the performance of this simple greedy algorithm, since the outcomes exactly coincide with Nash equilibria. To the best of our knowledge these performance bounds for the related machine scheduling problem $\mathrm{Q}||\sum C_j$ have not yet been analyzed. Most probably because the problem to find optimal solutions was settled long before [7].

4.1. Upper bound on the price of anarchy

Here we establish an upper bound on the price of anarchy for the SPT scheduling rule. Let σ be an optimal schedule resulting from the MFT algorithm, and recall that for the objective value in the optimal solution σ we have

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

The next theorem is the main result of this paper.

Theorem 3. *The price of anarchy for the related machine scheduling problem* $\mathrm{Q}||\sum C_j$ *with SPT as local sequencing rule is no greater than 2.*

Proof. By our previous discussion, it suffices to show that the game is $(2,0)$-semi-smooth, by showing that

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

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

$$\sum_{j \in J_i(\sigma)} C_j(\sigma_j, \nu_{-j}) = \sum_{j \in J_i(\sigma)} \left(\frac{p_j}{s_i} + \sum_{\substack{k \in J_i(\nu) \\ k<j}} \frac{p_k}{s_i} \right) = \sum_{j \in J_i(\sigma)} \frac{p_j}{s_i} + \sum_{j \in J_i(\sigma)} \sum_{\substack{k \in J_i(\nu) \\ k<j}} \frac{p_k}{s_i} . \quad (5)$$

Note that the number of times that a job k is counted on the right hand side of (5) equals the number of jobs with higher index than j on machine i in the optimal solution, times $1/s_i$. In other words, the second part of (5) can be rewritten as

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

This gives us

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

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

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

Summing over all i leads to

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

From Theorem 2 we know

$$\sum_{j=1}^{n} z_{\nu_j}(\sigma, j) \cdot \frac{p_j}{s_{\nu_j}} \leq \sum_{j=1}^{n} \left(z_{\sigma_j}(\sigma, j) + 1\right) \cdot \frac{p_j}{s_{\sigma_j}} = \sum_{j=1}^{n} C_j(\sigma)\,. \tag{6}$$

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

$$\sum_{j=1}^{n} \frac{p_j}{s_{\sigma_j}} \leq \sum_{j=1}^{n} C_j(\sigma)\,. \tag{7}$$

Combining the above, we get

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

for all strategy profiles ν. □

4.2. Lower bound on the price of anarchy

Here we describe a parametric instance which has price of anarchy approaching $e/(e-1)$ as the number of machines, jobs, and the maximum speed of the fastest machine grows. The Nash equilibrium will be the schedule with all jobs on the fastest machine.

Instance 1. Let $\mathcal{I}$ be the set of parametric instances $I(s,m)$ that satisfy the following. $I(s,m)$ has m machines, one with speed $s \in \mathbb{N}$, with $s > 1$, and all the other machines with speed 1. We assume that $m > s$. Furthermore, $I(s,m)$ has $n = m-1+s$ jobs, with processing requirements

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

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

Lemma 4. *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

$$C_j = \sum_{k=1}^{j} \frac{1}{s} = \frac{j}{s} < 1\,. \tag{8}$$

Observe that $1/(x-1) = s-1$. Therefore, for job $j \geq s$, the completion time is equal to

$$\begin{aligned} C_j = \sum_{k=1}^{j} \frac{p_k}{s} &= \frac{1}{s}\left((s-1) + \sum_{k=0}^{j-s} x^k\right) \\ &= \frac{1}{s}\left((s-1) + \frac{x^{j-s+1}-1}{x-1}\right) \\ &= \frac{1}{s}\left((s-1) + (s-1)x^{j-s+1} - (s-1)\right) \\ &= x^{j-s} = p_j\,. \end{aligned} \tag{9}$$

Thus, the Nash equilibrium condition (4) holds for all jobs, since all other machines have speed 1. $\square$

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

Theorem 5. *The (pure strategy) price of anarchy for the related machine scheduling problem* $\mathrm{Q}||\sum C_j$ *with SPT local scheduling rule is at least* $e/(e-1)$*.*

Proof. Consider instances $I(s,m)$ from $\mathcal{I}$ as defined above. Consider the (optimal) assignment where the s longest jobs $n-s+1=m,\dots,n$ are on the fast machine, while the first $n-s=m-1$ jobs are each on a slow machine. Note that, because $m>s$ also jobs with $p_j>1$ will be scheduled on slow machines (jobs $s,\dots,m-1$). The objective value in this solution is therefore equal to

$$\begin{aligned}
\sum_{j=1}^{s-1} 1 + \sum_{j=s}^{m-1} x^{j-s} + \frac{1}{s}\sum_{j=m}^{n}\sum_{k=m}^{j} x^{k-s} &= (s-1) + \sum_{j=0}^{m-s-1} x^j + \frac{1}{s}\sum_{j=m}^{n}\left(\sum_{k=0}^{j-s} x^k - \sum_{k=0}^{m-s-1} x^k\right)\\
&= (s-1) + (s-1)(x^{m-s}-1) + \sum_{j=m}^{n}\left(x^{j-s} - x^{m-s-1}\right)\\
&= (s-1)x^{m-s} + \sum_{j=m-s}^{n-s} x^j - \sum_{j=m}^{n} x^{m-s-1}\\
&= sx^{m-s-1} + (s-1)(x^{n-s+1}-x^{m-s}) - sx^{m-s-1}\\
&= (s-1)(x^{n-s+1}-x^{m-s})\,. \qquad (10)
\end{aligned}$$

To compute the objective value for the Nash equilibrium, recall Lemma 4 and the expressions for the job completion times in (8) and (9). From this we compute the objective value in the Nash equilibrium as

$$\begin{aligned}
\sum_{j=1}^{s-1}\frac{j}{s} + \sum_{j=s}^{n} x^{j-s} &= \frac{(s-1)}{2} + \sum_{j=0}^{n-s} x^j\\
&= \frac{(s-1)}{2} + (s-1)(x^{n-s+1}-1)\\
&= (s-1)\left(x^{n-s+1} - \frac{1}{2}\right)\,. \qquad (11)
\end{aligned}$$

Combining (10) and (11) gives us the following bound for the price of anarchy

$$\mathrm{PoA}(I(s,m)) \geq \frac{x^{n-s+1}-\frac{1}{2}}{x^{n-s+1}-x^{m-s}} = \frac{x^s - \frac{1}{2}x^{-(m-s)}}{x^s-1}\,.$$

Now if we let m (and n) go to infinity, and because $x>1$, we get that

$$\lim_{m\to\infty}\mathrm{PoA}(I(s,m)) \geq \frac{x^s}{x^s-1} = \frac{\left(\frac{s}{s-1}\right)^s}{\left(\frac{s}{s-1}\right)^s - 1}\,. \qquad (12)$$

Now, the right hand side of (12) converges to $e/(e-1)$ as s goes to infinity. $\square$

5. Identical Machines

Two special cases arise when either the machines or the jobs are all identical. In both cases, note that all pure Nash equilibria are optimal solutions, and the (pure) price of anarchy would equal 1. However, even if both the machines and the jobs are identical, i.e., $s_i = 1$ for all machines i and $p_j = 1$ for all jobs j, mixed Nash equilibria may have price of anarchy $\approx 3/2$.

Theorem 6 (Folklore)**.** *The (mixed strategy) price of anarchy for the parallel machine scheduling problem* $\mathrm{P}||\sum C_j$ *with SPT local scheduling rule is at least* $3/2 - 1/(2m)$*.*

Proof. Consider an instance with n jobs and $m = n$ machines. The optimal solution has one job on each machine and $\sum_j C_j = n$. Assume that ties are broken according to index. Now consider the mixed Nash equilibrium ν, where each job is scheduled on each machine with probability $1/n$. This is a Nash equilibrium, as the expected load of the jobs with index less than j is divided equally over all machines. Now, for any job j, the expected completion time is equal to

$$\mathbb{E}_\nu C_j(\nu) = 1 + \frac{j-1}{n}\,.$$

Summing over all jobs gives

$$\sum_{j=1}^{n} \mathbb{E}_\nu C_j(\nu) = \sum_{j=1}^{n} 1 + \frac{j-1}{n} = n + \frac{n(n-1)}{2}n = \frac{3n}{2} - \frac{1}{2}\,.$$

Dividing by $n = m$ gives $3/2 - 1/(2m)$. □

The identical machine model, where all machines have speed 1, has robust price of anarchy of exactly $3/2 - 1/(2m)$. This result was also found by Rivera [23] and Rahn and Schäfer [22]. Here we give a short and simple proof for a relaxed version of that claim that we discussed in private communication with J.R. Correa. The proof follows the framework of (λ, μ)-niceness as defined by Anshelevich et al. [1].

Definition 2 ((λ, μ)-nice games [1])**.** A cost-minimization game is (λ, μ)-*nice* if for every mixed strategy vector ν there is a mixed strategy vector ν' such that

$$\sum_{j=1}^{n} C_j(\nu'_j, \nu_{-j}) \le \lambda \cdot C(\sigma) + \mu \cdot C(\nu)\,, \tag{13}$$

where σ is an optimal solution.

Niceness implies bounds on the price of anarchy for pure Nash equilibria, mixed Nash equilibria and correlated equilibria, but not coarse correlated equilibria [1].

Theorem 7. *The mixed strategy price of anarchy for the parallel machine scheduling problem* $\mathrm{P}||\sum C_j$ *with SPT local scheduling rule is* $3/2 - 1/(2m)$*.*

Proof. We are only left to prove the upper bound. Again, let the jobs be indexed according to the order in which the machines process them. For any strategy profile ν and any job j, let ν'_j be a best response of j to ν_{-j}, so that

$$C_j(\nu'_j, \nu_{-j}) \leq C_j(\nu^*_j, \nu_{-j}).$$

As any job j has to care only about the jobs $k < j$, by a standard averaging argument over the machines, irrespective of ν_{-j} there must be a machine where job j can get a start time at most $\sum_{k<j} p_k/m$. Since this holds for all jobs $j \in N$, we have

$$\sum_{j\in J} C_j(\nu'_j, \nu_{-j}) \leq \sum_{j\in J}\sum_{k<j} \frac{p_k}{m} + \sum_{j\in J} p_j .$$

Now let σ be a strategy profile that results in an optimal solution. The lower bound on the optimal solution from Eastman et al. [11, Thm. 1] gives:

$$\sum_{j\in J}\sum_{k=1}^{j} \frac{p_k}{m} + \left(\frac{1}{2} - \frac{1}{2m}\right)\sum_{j\in J} p_j \leq \sum_{j\in J} C_j(\sigma).$$

Therefore

$$\begin{aligned}
\sum_{j\in J} C_j(\nu'_j, \nu_{-j}) &\leq \sum_{j\in J}\sum_{k=1}^{j-1} \frac{p_k}{m} + \sum_{j\in J} p_j \\
&\leq \sum_{j\in J}\sum_{k=1}^{j} \frac{p_k}{m} + \left(1 - \frac{1}{m}\right)\sum_{j\in J} p_j \\
&\leq \sum_{j\in J} C_j(\sigma) + \left(\frac{1}{2} - \frac{1}{2m}\right)\sum_{j\in J} p_j \\
&\leq \left(\frac{3}{2} - \frac{1}{2m}\right)\sum_{j\in J} C_j(\sigma).
\end{aligned}$$

This proves that the game is $(\frac{3}{2} - \frac{1}{2m}, 0)$-nice and therefore the price of anarchy is at most $3/2 - 1/(2m)$. □

6. Concluding Remarks

We leave open to determine the exact value for the price of anarchy in $[1.58, 2]$. The gap may be due to the fact that our proof for the upper bound yields a robust price of anarchy bound, while for the parametric instances from Theorem 5, scheduling any job on the fastest machine is even a pure dominant strategy (and hence also Nash) equilibrium. Specifically, the problem at hand may have separated price of anarchy bounds for pure and mixed Nash equilibria.

Our analysis to obtain the upper bound of 2 for the robust price of anarchy can be slightly improved, as was recently shown by Zhang et al. [26]. They work with a slightly better bound for (6) and, together with another upper bound on the optimal solution value, they obtain an upper bound on the price of anarchy equal to $2 - \frac{2}{(n+m)(n+1)}$. Yet, asymptotically this does not yield any improvement. Hence, this still leaves open the possibility that 2 is the exact asymptotic value for the robust price of anarchy, while the exact value for pure strategy Nash equilibria is $e/(e-1)$.

We do believe that an asymptotic improvement of the upper bound of 2 is possible, however: Either of the terms that appear in our analysis in (6) and (7) can be equal to the optimum value (asymptotically), but we have not been able to construct instances where both inequalities are tight simultaneously.

Acknowledgments

This work was partly done while the first author was with the University of Twente and the University of Chile.

References

[1] E. Anshelevich, J. Postl, and T. Wexler. Assignment games with conflicts: Robust price of anarchy and convergence results via semi-smoothness. *Theory of Computing Systems*, 59(3):440–475, 2016. doi: 10.1007/s00224-015-9646-0.

[2] Y. Azar, L. Fleischer, K. Jain, V. Mirrokni, and Z. Svitkina. Optimal coordination mechanisms for unrelated machine scheduling. *Operations Research*, 63(3):489–500, 2015. doi: 10.1287/opre.2015.1363.

[3] S. Bhattacharya, S. Im, J. Kulkarni, and K. Munagala. Coordination mechanisms from (almost) all scheduling policies. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, pages 121–134. ACM, 2014. doi: 10.1145/2554797.2554811.

[4] I. Caragiannis, C. Kaklamansis, P. Kanellopoulos, M. Kyropoulou, B. Lucier, R. Paes Leme, and É. Tardos. Bounding the inefficiency of outcomes in generalized second price auctions. *Journal of Economic Theory*, 156:343–388, 2015. doi: 10.1016/j.jet.2014.04.010.

[5] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. *Theoretical Computer Science*, 410(36):3327 – 3336, 2009. doi: 10.1016/j.tcs.2009.01.005.

[6] R. Cole, J. R. Correa, V. Gkatzelis, V. Mirrokni, and N. Olver. Decentralized utilitarian mechanisms for scheduling games. *Games and Economic Behavior*, 92:306–326, 2015. doi: 10.1016/j.geb.2013.03.011.

[7] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of scheduling*. Addison-Wesley Publishing Co., 1967.

[8] J. R. Correa and M. Queyranne. Efficiency of equilibria in restricted uniform machine scheduling with total weighted completion time as social cost. *Naval Research Logistics*, 59(5):384–395, 2012. doi: 10.1002/nav.21497.

[9] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Trans. Algorithms*, 3(1):4:1–4:17, Feb. 2007. doi: 10.1145/1186810.1186814.

[10] P. Dubey. Inefficiency of Nash equilibria. *Mathematics of Operations Research*, 11(1):1–8, 1986. doi: 10.1287/moor.11.1.1.

[11] W. L. Eastman, S. Even, and I. M. Isaacs. Bounds for the optimal scheduling of n jobs on m processors. *Management Science*, 11(2):268–279, 1964. doi: 10.1287/mnsc.11.2.268.

[12] R. Graham, E. Lawler, J. Lenstra, and A. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326, 1979. doi: 10.1016/S0167-5060(08)70356-X.

[13] 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. doi: 10.1111/j.1937-5956.2007.tb00271.x.

[14] R. Hoeksma and M. Uetz. The price of anarchy for minsum related machine scheduling. In R. Solis-Oba and G. Persiano, editors, *Approximation and Online Algorithms*, volume 7164 of *Lecture Notes in Computer Science*, pages 261–273. Springer, 2012. doi: 10.1007/978-3-642-29116-6_22.

[15] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM*, 23(2):317–327, Apr. 1976. doi: 10.1145/321941.321951.

[16] O. H. Ibarra and C. E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM*, 24(2):280–289, 1977. doi: 10.1145/322003.322011.

[17] N. Immorlica, L. E. Li, V. S. Mirrokni, and A. S. Schulz. Coordination mechanisms for selfish scheduling. *Theoretical Computer Science*, 410(17):1589–1598, 2009. doi: 10.1016/j.tcs.2008.12.032.

[18] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In C. Meinel and S. Tison, editors, *16th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 404–413. Springer, 1999. doi: 10.1007/3-540-49116-3_38.

[19] B. Lucier and R. Paes Leme. Gsp auctions with correlated types. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 71–80, New York, NY, USA, 2011. ACM. doi: 10.1145/1993574.1993587.

[20] R. B. Myerson. Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica: Journal of the Econometric Society*, pages 883–897, 1981. doi: 10.2307/1912508.

[21] C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, pages 749–753. ACM, 2001. doi: 10.1145/380752.380883.

[22] M. Rahn and G. Schäfer. Bounding the inefficiency of altruism through social contribution games. In Y. Chen and N. Immorlica, editors, *Web and Internet Economics*, volume 8289 of *Lecture Notes in Computer Science*, pages 391–404. Springer, 2013. doi: 10.1007/978-3-642-45046-4_32.

[23] O. L. Rivera. Cotas para el precio de la anarquía de juegos de scheduling. Master's thesis, Universidad de Chile, 2012. URL `http://www.repositorio.uchile.cl/handle/2250/111965`.

[24] T. Roughgarden. Intrinsic robustness of the price of anarchy. *J. ACM*, 62(5):32:1–32:42, Nov. 2015. doi: 10.1145/2806883.

[25] L. Yu, K. She, H. Gong, and C. Yu. Price of anarchy in parallel processing. *Information Processing Letters*, 110(8–9):288–293, 2010. doi: 10.1016/j.ipl.2010.02.003.

[26] L. Zhang, Y. Zhang, D. Du, and Q. Bai. Improved price of anarchy for machine scheduling games with coordination mechanisms. *Optimization Letters*, 2018. doi: 10.1007/s11590-018-1285-3.