Selfish Jobs with Favorite Machines: Price of Anarchy vs. Strong Price of Anarchy

Cong Chen[1,2(✉)], Paolo Penna[2], and Yinfeng Xu[1]

[1] School of Management, Xi'an Jiaotong University, Xi'an, China
chencong2779@stu.xjtu.edu.cn

[2] Department of Computer Science, ETH Zurich, Zurich, Switzerland

Abstract. We consider the well-studied game-theoretic version of machine scheduling in which jobs correspond to *self-interested* users and machines correspond to resources. Here each user chooses a machine trying to minimize *her own* cost, and such selfish behavior typically results in some *equilibrium* which is not globally *optimal*: An equilibrium is an allocation where no user can reduce her own cost by moving to another machine, which in general need not minimize the makespan, i.e., the maximum load over the machines.

We provide *tight* bounds on two well-studied notions in algorithmic game theory, namely, the *price of anarchy* and the *strong price of anarchy* on machine scheduling setting which lies in between the related and the unrelated machine case. Both notions study the social cost (makespan) of the *worst* equilibrium compared to the optimum, with the strong price of anarchy restricting to a stronger form of equilibria. Our results extend a prior study comparing the price of anarchy to the strong price of anarchy for two related machines (Epstein [13], Acta Informatica 2010), thus providing further insights on the relation between these concepts. Our exact bounds give a qualitative and quantitative comparison between the two models. The bounds also show that the setting is indeed easier than the two unrelated machines: In the latter, the strong price of anarchy is 2, while in ours it is strictly smaller.

1 Introduction

Scheduling jobs on *unrelated* machines is a classical optimization problem. In this problem, each job has a (possibly different) processing time on each of the m machines, and a schedule is simply an assignment of jobs to machines. For any such schedule, the load of a machine is the sum of all processing times of the jobs assigned to that machine. The objective is to find a schedule minimizing the *makespan*, that is, the maximum load among the machines.

In its *game-theoretic* version, jobs correspond to *self-interested* users who choose which machine to use accordingly without any centralized control, and naturally aim at minimizing their *own cost* (i.e. the load of the machine

For a full version of this work see [8].

X. Gao et al. (Eds.): COCOA 2017, Part II, LNCS 10628, pp. 226–240, 2017.
https://doi.org/10.1007/978-3-319-71147-8_16

they choose). This will result in some *equilibrium* in which no player has an incentive to deviate, though the resulting schedule is not necessarily the optimal in terms of makespan. Indeed, even for two unrelated machines it is quite easy to find equilibria whose makespan is arbitrarily larger than the optimum.

Example 1 (bad equilibrium for two unrelated machines). Consider two jobs and two unrelated machines, where the processing times are given by the following table:

	job 1	job 2
machine 1	1	s
machine 2	s	1

The allocation represented by the gray box is a (pure Nash) equilibrium: if a job moves to the other machine, its own cost increases from s to $s+1$. As the optimal makespan is 1 (swap the allocation), even for two machines the ratio between the cost of the worst equilibrium and the optimum is unbounded (at least s).

The inefficiency of equilibria in games is a central concept in algorithmic game theory, as it quantifies the *efficiency loss* resulting from a *selfish behavior* of the players. In particular, the following two notions received quite a lot of attention:

– *Price of Anarchy (PoA)* [23]. The price of anarchy is the ratio between cost of the *worst* Nash equilibrium (NE) and the *optimum*.
– *Strong Price of Anarchy (SPoA)* [1]. The strong price of anarchy is the ratio between cost of the *worst* strong Nash equilibrium (SE) with the *optimum*.

The only difference between the two notions is in the equilibrium concept: While in a Nash equilibrium no player can *unilaterally* improve by deviating, in *strong* Nash equilibrium no *group* of players can deviate and, in this way, all of them improve [4]. For instance, the allocation in Example 1 is *not* a SE because the two players could change strategy and both will improve.

Several works pointed out that the price of anarchy may be too pessimistic because, even for two unrelated machines, the price of anarchy is *unbounded* (see Example 1 above). Research thus focused on providing bounds for the strong price of anarchy and comparing the two bounds according to the problem restriction:

	$SPoA$	PoA
Unrelated	m	∞
$m=2$	2	∞
Related	$\Theta(\frac{\log m}{(\log\log m)^2})$	$\Theta(\frac{\log m}{\log\log m})$
$m=2$	$\frac{\sqrt{5}+1}{2} \simeq 1.618$	$\frac{\sqrt{5}+1}{2} \simeq 1.618$
$m=3$		2
Identical	$\frac{2m}{m+1}$	$\frac{2m}{m+1}$
$m=2$	4/3	4/3

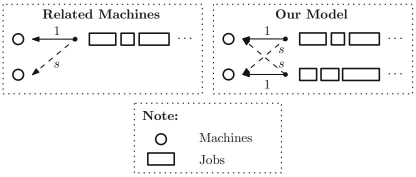


Fig. 1. Relationship between 2 related machines and our model.

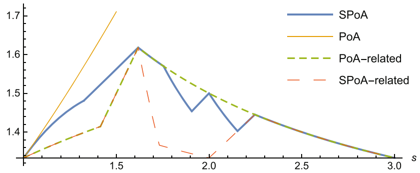


Fig. 2. Comparison between PoA and $SPoA$ in our model and in two related machines [13]. (Color figure online)

In *unrelated* machines, each job can have different processing times on different machines. In *related* machines, each job has a *size* and each machine a *speed*, and the processing time of a job on a machine is the size of the job divided by the speed of the machine. For *identical* machines, the processing time of a job is the same on all machines. The main difference between the identical machines and the other cases is obviously that in the latter the processing times are different. For two related machines, the worst bound of PoA and $SPoA$ is achieved only when the *speed ratio* s equals a specific value. Indeed, [13] characterize and compare the PoA and the $SPoA$ for all values of s, showing that $SPoA < PoA$ only in a specific interval of values (see next section for details). The lower bound on the PoA for two unrelated machines (Example 1) is unbounded when the ratio between different processing time s is unbounded.

1.1 Our Contribution

Following the approach by [13] on two related machines, we study the *price of anarchy* and the *strong price of anarchy* for the case of two machines though in a more general setting. Specifically, we consider the case of jobs with *favorite machines* [7] which is defined as follows (see Fig. 1). Each job has a certain *size* and a *favorite* machine; The processing time of a job on its favorite machine is just its *size*, while on any non-favorite machine it is s *times slower*, where $s \geq 1$ is common parameter across all jobs. This parameter is the speed ratio when considering the special case of two related machines (see Fig. 1). The model is also a restriction of unrelated machines and the bad NE in Example 1 corresponds to two jobs of size one in our model. That is, when s in unbounded, the price of anarchy is *unbounded* also in our model,

$$PoA \geq s. \tag{1}$$

This motivates the study of strong equilibria and $SPoA$ in our setting. We provide exact bounds on both the PoA and the $SPoA$ for all values of s.

We first give an intuitive bound on $SPoA$ which holds for all possible values of s.

Theorem 1. *$SPoA \leq 1 + 1/s \leq 2$.*

By a more detailed an involved analysis, we prove further *tight* bounds on $SPoA$.

Theorem 2. *$SPoA = \hat{\ell}_1$, where (see the blue line in Fig. 2)*

$$\hat{\ell}_1 = \begin{cases} \frac{s^3+s^2+s+1}{s^3+2}, & 1 \leq s \leq s_1 \approx 1.325 \\ \frac{s^2+2s+1}{2s+1}, & s_1 \leq s \leq s_2 = \frac{1+\sqrt{5}}{2} \approx 1.618 \\ \frac{s+1}{s}, & s_2 \leq s \leq s_3 \approx 1.755 \\ \frac{s^3-s^2+2s-1}{s^3-s^2+s-1}, & s_3 \leq s \leq s_4 \approx 1.907 \\ \frac{s+1}{2}, & s_4 \leq s \leq s_5 = 2 \\ \frac{s^2-s+1}{s^2-s}, & s_5 \leq s \leq s_6 \approx 2.154 \\ \frac{s^2}{2s-1}, & s_6 \leq s \leq s_7 \approx 2.247 \\ \frac{s+1}{s}, & s_7 \leq s. \end{cases}$$

At last we give *exact* bounds on the PoA, which show that the bound in (1) based on Example 1 is never the worst case.

Theorem 3. *$PoA = \frac{s^3+s^2+s+1}{s^2+s+1}$ (see the orange line in Fig. 2).*

These bounds express the dependency on the parameter s and suggest a natural comparison with the case of two related machines (with the same s).

1.2 Related Work

The bad Nash equilibrium in Example 1 appears in several works [1,11,19,25] to show that even for two machines the price of anarchy is unbounded, thus suggesting that the notion should be refined. Among these, the *strong price of anarchy*, which considers *strong* NE, is studied in [1,13,16]. The *sequential price of anarchy*, which considers *sequential* equilibria arising in extensive form games, is studied in [6,19,20,25]. In [11] the authors investigate *stochastically stable* equilibria and the resulting *price of stochastic anarchy*, while [21] focuses on the equilibria produced by the *multiplicative weights update algorithm.* A further distinction is between *mixed* (randomized) and *pure* (deterministic) equilibria: in the former, players choose a probability distribution over the strategies and regard their expected cost, in the latter they choose deterministically one strategy. In this work we focus on pure equilibria and in the remaining of this section we write *mixed PoA* to denote the bounds on the price of anarchy for mixed equilibria.

The following bounds have been obtained for scheduling games:

- *Unrelated machines.* The PoA is *unbounded* even for *two* machines, while the $SPoA$ is exactly m, for any number m of machines [1,16].

– *Related machines.* The price of anarchy is *bounded* for *constant number of machines*, and grows otherwise. Specifically, *mixed PoA* $= \Theta(\frac{\log m}{\log\log\log m})$ and $PoA = \Theta(\frac{\log m}{\log\log m})$ [12], while $SPoA = \Theta(\frac{\log m}{(\log\log m)^2})$ [16]. The case of a *small number of machines* is of particular interest. For *two* and *three* machines, $PoA = \frac{\sqrt{5}+1}{2}$ and $PoA = 2$ [15], respectively. For *two* machines, exact bounds as a function of the *speed ratio* s on both PoA and $SPoA$ are given in [13].
– *Restricted assignment.* The price of anarchy bounds are similar to related machines: *mixed* $PoA = \Theta(\frac{\log m}{\log\log\log m})$ and $PoA = \Theta(\frac{\log m}{\log\log m})$ [5], where the analysis of PoA is also in [18].
– *Identical machines.* The price of anarchy and the strong price of anarchy for pure equilibria are *identical* and bounded by a *constant*: $PoA = SPoA = \frac{2m}{m+1}$ [1], where the upper and lower bounds on PoA can be deduced from [17] and [27], respectively. Finally, *mixed* $PoA = \Theta(\frac{\log m}{\log\log m})$ [12,22,23].

For further results on other problems and variants of these equilibrium concepts we refer the reader to e.g. [2,9,10,26] and references therein.

2 Preliminaries

2.1 Model (Favorite Machines) and Basic Definitions

In unrelated machine scheduling, there are m machines and n jobs. Each job j has some processing time p_{ij} on machine i. A schedule is an assignment of each job to some machine. The *load* of a machine i is the sum of the processing times of the jobs assigned to machine i. The *makespan* is the maximum load over all machines. In this work, we consider the restriction of jobs with *favorite* machines: Each job j consists of a pair (s_j, f_j), where s_j is the *size* of job j and f_j is the *favorite machine* of this job. For a *common parameter* $s \geq 1$, the processing time of a job in a favorite machine is just its size ($p_{ij} = s_j$ if $i = f_j$), while on non-favorite machines is it s *times slower* ($p_{ij} = s \cdot s_j$ if $i \neq f_j$).

We consider jobs as *players* whose cost is the load of the machine they choose: For an allocation $x = (x_1, \ldots, x_n)$, where x_j denotes the machine chosen by job j, and $\ell_i(x) = \sum_{j:x_j=i} p_{ij}$ is the load of machine i. We say that x is a *Nash equilibrium (NE)* if no player j can unilaterally deviate and improve her own cost, i.e., move to a machine $\hat{x}_j$ such that $\ell_{\hat{x}_j}(\hat{x}) < \ell_{x_j}(x)$ where $\hat{x} = (x_1, \ldots, \hat{x}_j, \ldots, x_n)$ is the allocation resulting from j's move. In a *strong Nash equilibrium (SE)*, we require that in any group of deviating players, at least one of them does not improve: allocation x is a SE if, for any $\hat{x}$ which differ in exactly a subset J of players, there is one $j \in J$ such that $\ell_{\hat{x}_j}(\hat{x}) \geq \ell_{x_j}(x)$. The *price of anarchy (PoA)* is the worst-case ratio between the cost (i.e., makespan) of a NE and the optimum: $PoA = \max_{x \in \mathsf{NE}} \frac{C(x)}{opt}$ where $C(x) = \max_i \ell_i(x)$ and NE is the set of pure Nash equilibria. The *strong price of anarchy (SPoA)* is defined analogously w.r.t. the set SE of strong Nash equilibria: $SPoA = \max_{x \in \mathsf{SE}} \frac{C(x)}{opt}$.

2.2 First Step of the Analysis (Reducing to Eight Groups of Jobs)

To bound the strong price of anarchy we have to compare the *worst SE* with the optimum. The analysis consists of two main parts. We first consider the subset of jobs that needs to be reallocated in any allocation (or any SE) in order to obtain the optimum. It turns out that there are eight such subsets of jobs, and essentially the analysis reduced to the cases of eight jobs only. We then exploit the condition that possible reshuffling of these eight subsets must guarantee in order to be a SE.

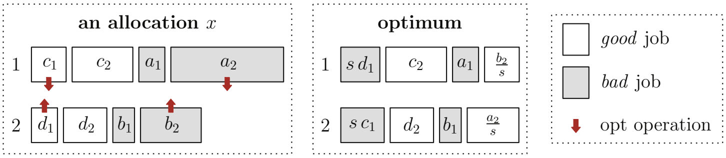


Fig. 3. Comparing SE with the optimum.

We say that a job is *good*, for an allocation under consideration, if it is allocated to its *favorite* machine. Otherwise the job is *bad*. Consider an allocation x and the optimum, respectively. As shown in Fig. 3, let

$$\ell_1 = a_1 + a_2 + c_1 + c_2 \quad \text{and} \quad \ell_2 = b_1 + b_2 + d_1 + d_2$$

be the load of machine 1 and machine 2 in allocation x, where $a_1 + a_2$ and $b_1 + b_2$ are load of *bad* jobs and $c_1 + c_2$ and $d_1 + d_2$ are load of *good* jobs. Note that these quantities correspond to (possibly empty) subsets of jobs. Without loss of generality suppose $\ell_1 \geq \ell_2$, that is

$$C(x) = \max(\ell_1, \ell_2) = \ell_1.$$

In the optimum, some of the jobs will be processed by different machine as in allocation x. Suppose the jobs associated with a_2, b_2, c_1 and d_1 are the difference. Thus, the load of the two machines in the optimum are

$$\ell_1^* = a_1 + b_2/s + c_2 + s \cdot d_1 \quad \text{and} \quad \ell_2^* = a_2/s + b_1 + s \cdot c_1 + d_2.$$

Remark 1. This setting can be also used to analyze *PoA* and *SPoA* for *two related machines* [13], which corresponds to two special cases $a_1, a_2, d_1, d_2 = 0$ and $b_1, b_2, c_1, c_2 = 0$.

2.3 Conditions for SE

Without loss of generality we suppose *opt* $= 1$, i.e.,

$$\ell_1^* \leq 1, \tag{2}$$

$$\ell_2^* \leq 1. \tag{3}$$

Since allocation x is SE, we have that the minimum load in allocation x must be at most *opt*,

$$\ell_2 \leq 1, \tag{4}$$

because otherwise we could swap some of the jobs to obtain *opt* and in this will improve the cost of all jobs.

Next, we shall provide several necessary conditions for allocation x to be a SE. Though these conditions are only a subset of those that SE must satisfy, they will lead to tight bounds on the $SPoA$.

1. No job in machine 1 will go to machine 2:

$$\ell_1 \leq \ell_2 + a_1/s, \qquad \text{for } a_1 > 0; \tag{5}$$
$$\ell_1 \leq \ell_2 + a_2/s, \qquad \text{for } a_2 > 0; \tag{6}$$
$$\ell_1 \leq \ell_2 + s \cdot c_1, \qquad \text{for } c_1 > 0; \tag{7}$$
$$\ell_1 \leq \ell_2 + s \cdot c_2, \qquad \text{for } c_2 > 0. \tag{8}$$

2. No a_2 - b_2 swap:

$$\ell_1 \leq \ell_2 - b_2 + a_2/s, \tag{9}$$
$$\text{or} \quad \ell_2 \leq \ell_1 - a_2 + b_2/s. \tag{10}$$

3. No a_2 - d_1 swap:

$$\ell_1 \leq \ell_2 - d_1 + a_2/s, \tag{11}$$
$$\text{or} \quad \ell_2 \leq \ell_1 - a_2 + s \cdot d_1. \tag{12}$$

4. No a_2 - $\{b_1, d_2\}$ swap:

$$\ell_1 \leq \ell_2 - b_1 - d_2 + a_2/s, \tag{13}$$
$$\text{or} \quad \ell_2 \leq \ell_1 - a_2 + b_1/s + s \cdot d_2. \tag{14}$$

5. No $\{a_2, c_2\}$ - $\{b_1, b_2, d_1, d_2\}$ swap:

$$\ell_1 \leq a_2/s + c_2 \cdot s\,, \tag{15}$$
$$\text{or} \quad \ell_2 \leq a_1 + c_1 + (b_1 + b_2)/s + s(d_1 + d_2). \tag{16}$$

6. No $\{a_1, a_2, c_1, c_2\}$ - $\{b_1, b_2, d_1, d_2\}$ swap:

$$\ell_1 \leq (a_1 + a_2)/s + s(c_1 + c_2), \tag{17}$$
$$\text{or} \quad \ell_2 \leq (b_1 + b_2)/s + s(d_1 + d_2). \tag{18}$$

3 Strong Price of Anarchy

We first prove a simpler general upper bound (Theorem 1) and then refine the result by giving tight bounds for all possible values of s (Theorem 2). The first result says that the strong price of anarchy is bounded and actually gets better as s increases.

Proof (of Theorem 1). We distinguish the following cases:

($a_2 = 0$.) By definition of ℓ_1 and ℓ_1^* we have $\ell_1 \leq c_1 + \ell_1^*$, and also by definition of ℓ_2^* we have $\ell_2^* \geq s \cdot c_1$. Therefore, along with (2) and (3) it holds that $\ell_1 \leq \frac{\ell_2^*}{s} + \ell_1^* \leq \frac{1}{s} + 1$.

($a_2 > 0$.) In this case we use that at least one between (9) or (10) must hold. If (9) holds then:

$$\ell_1 \leq \ell_2 - b_2 + \frac{a_2}{s} \leq d_1 + \ell_2^* \leq \frac{\ell_1^*}{s} + \ell_2^* \leq \frac{1}{s} + 1$$

where the last two inequalities follow by the fact that $\ell_1^* \geq s \cdot d_1$ and (2)-(3).

If (10) holds then we use that at least one between (17) or (18) must hold. If (17) holds then by definition of ℓ_1 this can be rewritten as

$$\frac{a_1 + a_2}{s} \leq c_1 + c_2.$$

By adding $a_1 + a_2$ on both sides, this is the same as

$$a_1 + a_2 + \frac{a_1 + a_2}{s} \leq \underbrace{a_1 + a_2 + c_1 + c_2}_{\ell_1} \quad \Rightarrow \quad a_2 \leq \frac{s}{s+1}\ell_1.$$

By (4) and (6) we also have

$$\ell_1 \leq 1 + \frac{a_2}{s}$$

and by putting the last two inequalities together we obtain

$$\ell_1 \leq 1 + \frac{\ell_1}{s+1} \quad \Leftrightarrow \quad \ell_1 \leq 1 + \frac{1}{s}.$$

If (18) holds then we first observe that, by definition, the following identity holds:

$$\ell_1 = \ell_1^* + \frac{\ell_2^*}{s} + \frac{s^2 - 1}{s^2}a_2 - \frac{b_1 + b_2}{s} - sd_1 - \frac{d_2}{s}.$$

We shall prove that

$$\frac{s^2 - 1}{s^2}a_2 - \frac{b_1 + b_2}{s} - sd_1 - \frac{d_2}{s} \leq 0 \tag{19}$$

and thus conclude from (2) and (3) that

$$\ell_1 \leq \ell_1^* + \frac{\ell_2^*}{s} \leq 1 + \frac{1}{s}.$$

From (6) and (10) we have

$$a_2 \leq \frac{b_2}{s-1} \leq \frac{b_1 + b_2}{s-1}$$

and plugging into left hand side of (19) we get

$$\frac{s^2-1}{s^2}a_2 - \frac{b_1+b_2}{s} - sd_1 - \frac{d_2}{s} \leq \frac{b_1+b_2}{s^2} - sd_1 - \frac{d_2}{s}.$$

Finally, by definition of ℓ_2, (18) can be rewritten as

$$\frac{b_1+b_2}{s^2} \leq \frac{d_1+d_2}{s}$$

which implies

$$\frac{b_1+b_2}{s^2} - sd_1 - \frac{d_2}{s} \leq \frac{d_1+d_2}{s} - sd_1 - \frac{d_2}{s} \leq 0$$

which proves (19) and concludes the proof of this last case. $\square$

3.1 Notation Used for the Improved Upper Bound

We shall break the proof into several subcases. First, we consider different intervals for s. Then, for each interval, we consider the quantities a_1, a_2, c_1, c_2 and break the proof into subcases, according to the fact that some of these quantities are zero or strictly positive (Lemmas 1–5 below). Finally, in each subcase, use a subset of the SE constraints to obtain the desired bound.

Table 2 shows the subcases and which constraints are used to prove a corresponding bound. Note that for the chosen constraints, we also specify some *weight* which essentially says how these constraints are combined together in the actual proof. We explain this with the following example.

An Illustrative Example (Weighted Combination of Constraints). Consider the case 1–2 in Table 2 (second row). In the third column, the four numbers show whether the four variable a_1, a_2, c_1, c_2 are zero or non-zero, where “1” represents non-zero and “$*$” represents non-negative. The last column is the bound we obtain for ℓ_1 and thus for the $SPoA$. Specifically, in this case we want to prove the following:

Claim. If $a_1 > 0$, $a_2 = 0$, and $c_1 > 0$, then $\ell_1 \leq \frac{s^3+s^2+s+1}{s^3+s^2+1}$.

Proof. First summing all the constraints with the corresponding weights given in columns 4 and 5 of Table 2 (second row):

$$(2)\frac{s^3+s}{s^3+s^2+1} + (3)\frac{s^2+1}{s^3+s^2+1} + (5)\frac{s^2}{s^3+s^2+1} + (7)\frac{1}{s^3+s^2+1}.$$

This simplifies as

$$a_1 + \frac{s^3+s^2+s+1}{s^4+s^3+s}a_2 + c_1 + \frac{s^3+s^2+s+1}{s^3+s^2+1}c_2 + \frac{s^4-1}{s^3+s^2+1}d_1 \leq \frac{s^3+s^2+s+1}{s^3+s^2+1}. \tag{20}$$

Note that all the weights (column 5) are positive when s in the given interval (column 2), so that the direction of the inequalities (column 4) remains.

According to columns 2 and 3, we have $a_2 = 0$, $\frac{s^3+s^2+s+1}{s^3+s^2+1} \geq 1$ and $\frac{s^4-1}{s^3+s^2+1} \geq 0$. This and (20) imply

$$\ell_1 = a_1 + a_2 + c_1 + c_2 \leq \frac{s^3+s^2+s+1}{s^3+s^2+1}.$$

We therefore get the bound in column 6.

3.2 The Actual Proof

We break the proof of Theorem 2 into several lemmas, and prove the upper bound in each of them. The lemmas are organized depending on the value of a_1, a_2, c_1, c_2. Finally, we show that the bounds of these lemmas are tight (Lemma 6).

Lemma 1. *If $a_1 > 0$, then $\ell_1 \leq \hat{\ell}_1$.*

Lemma 2. *If $a_1 = a_2 = 0$, then $\ell_1 \leq \hat{\ell}_1$.*

Lemma 3. *If $a_1 = 0$, $a_2, c_1 > 0$, then $\ell_1 \leq \hat{\ell}_1$.*

Lemma 4. *If $a_1 = c_1 = c_2 = 0$ and $a_2 > 0$, then $\ell_1 \leq \hat{\ell}_1$.*

Lemma 5. *If $a_1 = c_1 = 0$ and $a_2, c_2 > 0$, then $\ell_1 \leq \hat{\ell}_1$.*

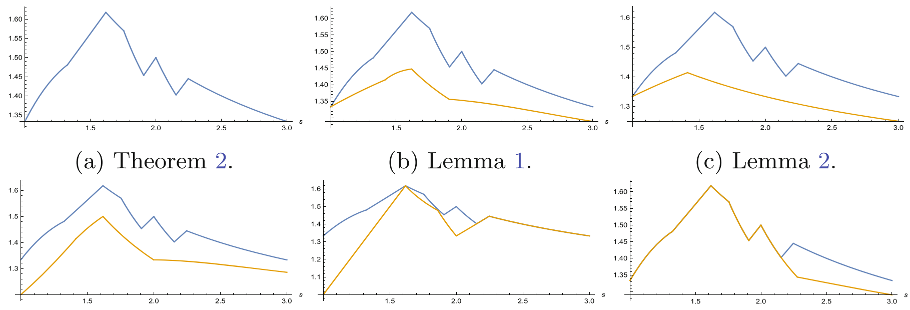
Lemma 6. *The lower bound of $SPoA \geq \hat{\ell}_1$ is given by the instances in* Table 1.

Table 1. Lower bound for $SPoA$.

	s	a_2	b_2	c_2	d_1	d_2	LB $(= \ell_1)$	ℓ_2
LB1	$[0, s_1]$	$\frac{s^3+s^2}{s^3+2}$	$\frac{s}{s^3+2}$	$\frac{s+1}{s^3+2}$	$\frac{s^2-1}{s^3+2}$	$\frac{s^3-s^2-s+2}{s^3+2}$	$\frac{s^3+s^2+s+1}{s^3+2}$	$\frac{s^3+1}{s^3+2}$
LB2	$[s_1, s_2]$	$\frac{s^2+s}{2s+1}$	$\frac{s^2}{2s+1}$	$\frac{s+1}{2s+1}$	0	$\frac{s}{2s+1}$	$\frac{s^2+2s+1}{2s+1}$	$\frac{s^2+s}{2s+1}$
LB3	$[s_2, s_3]$	1	$s-1$	$\frac{1}{s}$	0	$2-s$	$\frac{s+1}{s}$	1
LB4	$[s_3, s_4]$	$\frac{s^2/(s-1)}{s^2+1}$	$\frac{s^2}{s^2+1}$	$\frac{s^2-s+1}{s^2+1}$	0	$\frac{1}{s^2+1}$	$\frac{s^3-s^2+2s-1}{s^3-s^2+s-1}$	1
LB5	$[s_4, s_5]$	$\frac{s}{2}$	$\frac{s}{2}$	$\frac{1}{2}$	0	$\frac{2-s}{2}$	$\frac{s+1}{2}$	1
LB6	$[s_5, s_6]$	$\frac{1}{s-1}$	1	$\frac{s-1}{s}$	0	0	$\frac{s^2-s+1}{s^2-s}$	1
LB7	$[s_6, s_7]$	$\frac{s^2}{2s-1}$	0	0	$\frac{(s-1)^2}{2s-1}$	$\frac{s-1}{2s-1}$	$\frac{s^2}{2s-1}$	$\frac{s^2-s}{2s-1}$
LB8	$[s_7, \infty]$	$\frac{s+1}{s}$	0	0	$\frac{1}{s}$	$\frac{s^2-s-1}{s^2}$	$\frac{s+1}{s}$	$\frac{s^2-1}{s^2}$

*Note that $a_1 = b_1 = c_1 = 0$, and a_2, b_2, c_2, d_1, d_2 each represent a single job here.

Figure 4 illustrates the relation between the general bound and the bound proved in each of the these lemmas.



(a) Theorem 2. (b) Lemma 1. (c) Lemma 2.

(d) Lemma 3. (e) Lemma 4. (f) Lemma 5.

Fig. 4. Proof of Theorem 2. (Color figure online)

The proofs of these lemmas are based on Table 2. Note that in Table 2, each row has a bound for ℓ_1 in the last column. Since the above illustrative example has already explained how the bounds are generated, here we mainly focus on the relationship of these bounds with the lemmas.

Table 2. Subcases to prove Lemmas 1–5 and Lemmas 8 and 9

Lemma.subcases	s	a_1, a_2, c_1, c_2	Constrains needed	Weight coefficient	Bounds
1.1	$[1, \infty]$	1, 0, 0, *	(2)	$\{1\}$	1
1.2	$[1, \infty]$	1, 0, 1, *	(2), (3), (5), (7)	$\frac{\{s^3+s;s^2+1;s^2;1\}}{s^3+s^2+1}$	$\frac{s^3+s^2+s+1}{s^3+s^2+1}$
1.3-a	$[1, \sqrt{2}]$	1, 1, 0, *	(2), (3), (5), (6)	$\{2s;\ 2;\ s^2;\ 2-s^2\}\cdot\frac{1}{s+2}$	$\frac{2(s+1)}{s+2}$
1.3-b	$[\sqrt{2}, \infty]$	1, 1, 0, *	(2), (3), (4), (5)	$\frac{\{s^2;s;s(s^2-2);s(s^2-1)\}}{s^3-s+1}$	$\frac{s(s^2+s-1)}{s^3-s+1}$
1.3-c	$[1, \infty]$	1, 1, 0, *	(3), (5), (6)	$\{2s;\ s;\ s\}\cdot\frac{1}{2s-1}$	$\frac{2s}{2s-1}$
1.4-a	$[1, 1.272]$	1, 1, 1, *	(2), (3), (5), (6), (7)	$\frac{\{2s^3+s;2s^2+1;s^4;-s^4+s^2+1;s^2\}}{s^3+2s^2+s+1}$	$\frac{2s^3+2s^2+s+1}{s^3+2s^2+s+1}$
1.4-b	$[1.272, \infty]$	1, 1, 1, *	(2), (3), (4), (5), (7)	$\frac{\{s^3;s^2;s^4-s^2-1;s^2(s^2-1);s^2-1\}}{s^4+s-1}$	$\frac{s^4+s^3-1}{s^4+s-1}$
2.1	$[1, \infty]$	0, 0, 0, 1	(2)	$\{1\}$	1
2.2	$[1, \infty]$	0, 0, 1, 0	(3)	$\{1/s\}$	$1/s$
2.3-a	$[1, \infty]$	0, 0, 1, 1	(2), (3), (7), (8)	$\{2s;2;1;1\}\cdot\frac{1}{s+2}$	$\frac{2(s+1)}{s+2}$
2.3-b	$[1, \infty]$	0, 0, 1, 1	(2), (3), (7)	$\{s;2;1\}\cdot\frac{1}{s+1}$	$\frac{s+2}{s+1}$
3.1-a	$[1, \infty]$	0, 1, 1, 0	(3), (4), (7)	$\frac{\{s^2;s^2-1;s^2-1\}}{s^2+s-1}$	$\frac{2s^2-1}{s^2+s-1}$
3.1-b.(9)	$[1.272, \infty]$	0, 1, 1, 0	(2), (3), (4), (7), (9)	$\frac{\{s^3;s^4;s^4-s^2-1;s^4-1;s^4-s^2\}}{2s^4-s^2+s-1}$	$\frac{2s^4+s^3-s^2-1}{2s^4-s^2+s-1}$
3.1-b.(10)	$[1.272, \infty]$	0, 1, 1, 0	(3), (6), (10)	$\frac{\{s^2-s+1;(s-1)^2(s+1);s(s^2-1)\}}{s^3-2s^2+s+1}$	$\frac{s^2-s+1}{s^3-2s^2+s+1}$
3.2-a.(15)	$[1, \infty]$	0, 1, 1, 1	(4), (6), (7), (15)	$\{s^2;\ s^2-1;1;1\}\cdot\frac{1}{s^2-s+1}$	$\frac{s^2}{s^2-s+1}$
3.2-a.(16)	$[1, \infty]$	0, 1, 1, 1	(4), (6), (8), (16)	$\frac{\{s^2-s+2;s^2;1;s\}}{s^2-s+1}$	$\frac{s^2-s+2}{s^2-s+1}$

(*continued*)

Table 2. (*continued*)

Lemma.subcases	s	a_1, a_2, c_1, c_2	Constrains needed	Weight coefficient	Bounds
3.2-b	$[1, \sqrt{2}]$	0, 1, 1, 1	(2), (3), (6), (7), (8)	$\frac{\{s(s^2+2);s^2+2;2-s^2;s^2;s^2\}}{s^2+2s+2}$	$\frac{s^3+s^2+2s+2}{s^2+2s+2}$
3.2-c	$[\sqrt{2}, \infty]$	0, 1, 1, 1	(2), (3), (4), (7)	$\frac{\{s;s^2;s^2-2;s^2-1\}}{s^2+s-1}$	$\frac{2s^2+s-2}{s^2+s-1}$
4-a	$[1, \frac{1+\sqrt{5}}{2}]$	0, 1, 0, 0	(3)	$\{s\}$	s
4-b.(9)-a	$[1, \infty]$	0, 1, 0, 0	(2), (3), (6), (9)	$\{s; s^2; 1; s^2-1\} \cdot \frac{1}{s^2}$	$1+\frac{1}{s}$
4-b.(9)-b.(13)-a	$[1, \infty]$	0, 1, 0, 0	(2), (3), (9), (13)	$\frac{\{s^2;s(s^2-1);s(s^2-1);s\}}{s^3-1}$	$\frac{s(s^2+s-1)}{s^3-1}$
4-b.(9)-b.(13)-b.(11)	$[1, \infty]$	0, 1, 0, 0	(3), (4), (9), (11)	$\{\frac{s}{2s-1};\frac{s}{2s-1};\frac{s}{2s-1};\frac{s}{2s-1}\}$	$\frac{2s}{2s-1}$
4-b.(9)-b.(13)-b.(12)	$[1, \infty]$	0, 1, 0, 0	(2), (9), (12), (13)	$\frac{\{s^2;s^2-s;s^2-s;s^2\}}{2s^2-3s+1}$	$\frac{s^2}{2s^2-3s+1}$
4-b.(9)-b.(14)	$[1, \infty]$	0, 1, 0, 0	(3), (6), (14)	$\{s^2; s; s\} \cdot \frac{1}{2s-1}$	$\frac{s^2}{2s-1}$
4-b.(10)	$[1, \infty]$	0, 1, 0, 0	–	–	–
5.(17)-a	$[1, \infty]$	0, 1, 0, 1	(2), (3), (6), (17)	$\frac{\{s(s+1);s+1;s+1;s^2/(s-1)\}}{2s+1}$	$\frac{(s+1)^2}{2s+1}$
5.(17)-b	$[1, \infty]$	0, 1, 0, 1	(4), (6), (17)	$\{\frac{s+1}{s};\frac{s+1}{s};\frac{1}{(s-1)s}\}$	$1+\frac{1}{s}$
5.(17)-c.(9)	$[1, \infty]$	0, 1, 0, 1	(2), (3), (6), (8), (9)	$\frac{\{s^2+1;s^3+s;1/s;s;s^3-1/s\}}{s^3+s+1}$	$\frac{s^3+s^2+s+1}{s^3+s+1}$
5.(17)-c.(10)-a	$[1, \infty]$	0, 1, 0, 1	(4), (6), (10)	$\{\frac{s^2-s+1}{(s-1)s};\frac{s}{s-1};\frac{1}{s-1}\}$	$\frac{s^2-s+1}{(s-1)s}$
5.(17)-c.(10)-b.(13)	$[1, \infty]$	0, 1, 0, 1	(2), (13), (17)	$\{\frac{s+1}{2};\frac{s+1}{2s};\frac{s^2+1}{2(s-1)s}\}$	$\frac{s+1}{2}$
5.(17)-c.(10)-b.(14)	$[1, \infty]$	0, 1, 0, 1	(2), (4), (6), (10), (14)	$\frac{\{(s-1)(s^2+1);s;s(s^2+1);s^3+s-1;1\}}{s^3-s^2+s-1}$	$\frac{s^3-s^2+2s-1}{s^3-s^2+s-1}$
5.(18)-a.(11)	$[1, \infty]$	0, 1, 0, 1	(2), (3), (18), (11)	$\frac{\{s;s^2+s+1;s^2/(s-1);s+1\}}{2s+1}$	$\frac{(s+1)^2}{2s+1}$
5.(18)-a.(12)	$[1, \infty]$	0, 1, 0, 1	(2), (3), (6), (8), (12)	$\frac{\{s^3+s;s^2+1;s^3+s^2-s;s+1-\frac{1}{s};s^3-\frac{1}{s}\}}{s^3+2}$	$\frac{s^3+s^2+s+1}{s^3+2}$
5.(18)-b.(9)	$[1, \infty]$	0, 1, 0, 1	(2), (3), (6), (8), (9)	$\frac{\{s^2+1;s^3+s;1/s;s;s^3-1/s\}}{s^3+s+1}$	$\frac{s^3+s^2+s+1}{s^3+s+1}$
5.(18)-b.(10)	$[1, \infty]$	0, 1, 0, 1	(4), (6), (10), (18)	$\frac{\{s^2;s(s+1);s+1;1/(s-1)\}}{s^2-1}$	$\frac{s^2}{s^2-1}$
8.1-a	$[1, \sqrt{2}]$	1, 1, *, 0	(2), (3), (5), (6)	$\{2s; 2; s^2; 2-s^2\} \cdot \frac{1}{s+2}$	$\frac{2(s+1)}{s+2}$
8.1-b	$[\sqrt{2}, \infty]$	1, 1, *, 0	(2), (3), (5)	$\{s^2; s(s^2-1); s\} \cdot \frac{1}{s^2+s-1}$	s
8.2	$[1, \infty]$	0, 1, 1, 0	(2), (3), (7)	$\frac{\{s(s^2-1);s^2;s^2-1\}}{s^2+s-1}$	s
9	$[1, \infty]$	*, 1, *, 1	(2), (3), (6), (8)	$\frac{\{s^3+s;s^2+1;1;s^2\}}{s^2+s+1}$	$\frac{s^3+s^2+s+1}{s^2+s+1}$

Note: "*" means *either* 0 *or* 1 in the third column.

The index (column 1) of each row in Table 2 encodes the relationship between those bounds (last column) and how these bounds should be combined to obtain the corresponding lemma. Specifically, cases separated by "." are subcases that should take *maximum* of them due to we aim to measure the worst performance, while cases separated by "-" are a single case bounded by several different combinations of constrains that should take *minimum* of them due to these constrains should hold at the same time. For instance, we consider three subcases in Lemma 2, since $a_1 = a_2 = 0$: Case 2.1 ($c_1 = 0, c_2 > 0$), Case 2.2 ($c_1 > 0, c_2 = 0$) and Case 2.3 ($c_1 > 0, c_2 > 0$), which correspond to rows 2.1 to 2.3-b in Table 2. The bound for Case 2.3 is the minimum of the bounds of 2.3-a and 2.3-b. Finally, the maximum of the bounds of the three subcases give the bound for Lemma 2, i.e., $\max\left\{1, \frac{1}{s}, \min\{\frac{2(s+1)}{s+2}, \frac{s+2}{s+1}\}\right\}$ (orange line of Fig. 4c).

The proofs of Lemmas 1–6 are given in full detail in the full version [8].

4 Price of Anarchy

In this section we prove the bounds on the PoA in Theorem 3. Suppose the smallest jobs in a_1, a_2, c_1, c_2 are ${a_1}'$, ${a_2}'$, ${c_1}'$, ${c_2}'$ respectively. To guarantee NE, it must hold that no single job in machine 1 can improve by moving to machine 2, so that (5), (6), (7) and (8) are also true for NE since ${a_1}' \leq a_1$, ${a_2}' \leq a_2$, ${c_1}' \leq c_1$ and ${c_2}' \leq c_2$. Like in the analysis of the $SPoA$, we assume without loss of generality that $opt = 1$, and thus (2) and (3) hold. We use these six constraints to prove Theorem 3.

It is easy to see that if at most one of a_1, a_2, c_1, c_2 is nonzero, then $\ell_1 \leq s$, thus here we only discuss the cases where at least two of them are nonzero. Similar to the proof of $SPoA$ the proofs of these lemmas are based on last four rows of Table 2.

Lemma 7. *If* $a_2 = 0$*, then* $\ell_1 \leq \frac{s^3+s^2+s+1}{s^2+s+1}$.

Lemma 8. *If* $a_2 > 0$ *and* $c_2 = 0$*, then* $\ell_1 \leq \frac{s^3+s^2+s+1}{s^2+s+1}$.

Lemma 9. *If* $a_2 > 0$ *and* $c_2 > 0$*, then* $\ell_1 \leq \frac{s^3+s^2+s+1}{s^2+s+1}$.

Lemma 10. *The lower bound of PoA is achieved by the following case,*

$$a_2 = \frac{s^3+s^2}{s^2+s+1}, \quad b_1 = \frac{1}{s^2+s+1}, \quad b_2 = \frac{s^3}{s^2+s+1}, \quad c_2 = \frac{s+1}{s^2+s+1},$$

and $a_1 = c_1 = d_1 = d_2 = 0$.

Lemmas 7–10 complete the proof of Theorem 3.

5 Conclusion and Open Questions

In this work, we have analyzed both the *price of anarchy* and the *strong price of anarchy* on a simple though natural model of two machines in which each job has its own *favorite* machine, and the other machine is *s times slower* machine. The model and the results extend the case of two *related machines* with speed ratio s [13]. In particular, we provide *exact* bounds on PoA and $SPoA$ for *all values of* s. On the one hand, this allows us to compare with the same bounds for two related machines (see Fig. 2). On the other hand, to the best of our knowledge, this is one of the first studies which considers in the analysis the *processing time* *ratio* between different machines (with the exception of [13]). Prior work mainly focused on the asymptotic on the number of machines (resources) or/and number of jobs (users). Instead, the loss of efficiency due to selfish behavior is perhaps also caused by the presence of *different* resources, even when the latter are few.

Unlike for two related machines, in our setting the PoA grows with s and thus the influence of coalitions and the resulting $SPoA$ is more evident. Note for example that the $SPoA \leq \phi = \frac{\sqrt{5}+1}{2} \simeq 1.618$ and this bound is attained for $s = \phi$ exactly like for two related machines (see Fig. 1). Also, for sufficiently large s, the two problems have the exact same $SPoA$, though the PoA is very much different.

It is natural to study the PoA and $SPoA$ depending on the specific speed ratio, or processing time ratio. In that sense, it would be interesting to extend the analysis to more machines in the *favorite* machines setting [7]. There, an important parameter is also the minimum number k of favorite machines per job. The case $k = 1$ is perhaps interesting as, in the online setting, this gives a problem which is as difficult as the more general unrelated machines. Is it possible to characterize the PoA and the $SPoA$ in this setting for any s? Do these bounds improve for larger k? Another interesting restriction would be the case of *unit-size* jobs, which means that each job has processing time 1 or s. Such *two-values* restrictions have been studied in the *mechanism design* setting with *selfish machines* [3,24], where players are machines and they possibly speculate on their true cost. Considering other well studied solution concepts would also be interesting, including *sequential PoA* [6,19,20,25], *approximate SPoA* [14], and the *price of stochastic anarchy* [11].

References

1. Andelman, N., Feldman, M., Mansour, Y.: Strong price of anarchy. Games Econ. Behav. **2**(65), 289–317 (2009)
2. Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. SIAM J. Comput. **38**(4), 1602–1623 (2008)
3. Auletta, V., Christodoulou, G., Penna, P.: Mechanisms for scheduling with single-bit private values. Theor. Comput. Syst. **57**(3), 523–548 (2015)
4. Aumann, R.J.: Acceptable Points in General Cooperative n-Person Games, pp. 287–324. Princeton University Press, Princeton (1959)
5. Awerbuch, B., Azar, Y., Richter, Y., Tsur, D.: Tradeoffs in worst-case equilibria. Theor. Comput. Sci. **361**(2), 200–209 (2006)
6. Bilò, V., Flammini, M., Monaco, G., Moscardelli, L.: Some anomalies of farsighted strategic behavior. Theor. Comput. Syst. **56**(1), 156–180 (2015)
7. Chen, C., Penna, P., Xu, Y.: Online Scheduling of Jobs with Favorite Machines (2017, submitted)
8. Chen, C., Penna, P., Xu, Y.: Selfish jobs with favorite machines: price of anarchy vs. strong price of anarchy. arXiv e-prints, CoRR, abs/1709.06367 (2017)
9. Chien, S., Sinclair, A.: Strong and pareto price of anarchy in congestion games. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 279–291. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02927-1_24

10. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: Proceedings of the 37th annual ACM Symposium on Theory of Computing (STOC), pp. 67–73 (2005)
11. Chung, C., Ligett, K., Pruhs, K., Roth, A.: The price of stochastic anarchy. In: Monien, B., Schroeder, U.-P. (eds.) SAGT 2008. LNCS, vol. 4997, pp. 303–314. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79309-0_27
12. Czumaj, A., Vöcking, B.: Tight bounds for worst-case equilibria. ACM Trans. Algorithms (TALG) **3**(1), 4 (2007)
13. Epstein, L.: Equilibria for two parallel links: the strong price of anarchy versus the price of anarchy. Acta Informatica **47**(7), 375–389 (2010)
14. Feldman, M., Tamir, T.: Approximate strong equilibrium in job scheduling games. J. Artif. Intell. Res. **36**, 387–414 (2009)
15. Feldmann, R., Gairing, M., Lücking, T., Monien, B., Rode, M.: Nashification and the coordination ratio for a selfish routing game. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 514–526. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45061-0_42
16. Fiat, A., Kaplan, H., Levy, M., Olonetsky, S.: Strong price of anarchy for machine load balancing. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 583–594. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73420-8_51
17. Finn, G., Horowitz, E.: A linear time approximation algorithm for multiprocessor scheduling. BIT Numer. Math. **19**(3), 312–320 (1979)
18. Gairing, M., Lücking, T., Mavronicolas, M., Monien, B.: The price of anarchy for restricted parallel links. Parallel Process. Lett. **16**(01), 117–131 (2006)
19. Giessler, P., Mamageishvili, A., Mihalák, M., Penna, P.: Sequential solutions in machine scheduling games. CoRR abs/1611.04159 (2016)
20. de Jong, J., Uetz, M.: The sequential price of anarchy for atomic congestion games. In: Liu, T.-Y., Qi, Q., Ye, Y. (eds.) WINE 2014. LNCS, vol. 8877, pp. 429–434. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13129-0_35
21. Kleinberg, R., Piliouras, G., Tardos, É.: Multiplicative updates outperform generic no-regret learning in congestion games. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), pp. 533–542 (2009)
22. Koutsoupias, E., Mavronicolas, M., Spirakis, P.: Approximate equilibria and ball fusion. Theor. Comput. Syst. **36**(6), 683–693 (2003)
23. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 404–413. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49116-3_38
24. Lavi, R., Swamy, C.: Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. Games Econ. Beh. **67**(1), 99–124 (2009)
25. Leme, R.P., Syrgkanis, V., Tardos, É.: The curse of simultaneity. In: Proceedings of Innovations in Theoretical Computer Science (ITCS), pp. 60–67 (2012)
26. Roughgarden, T.: Intrinsic robustness of the price of anarchy. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), pp. 513–522 (2009)
27. Schuurman, P., Vredeveld, T.: Performance guarantees of local search for multiprocessor scheduling. INFORMS J. Comput. **19**(1), 52–63 (2007)