Operations Research Letters 47 (2019) 7–11

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Selfish load balancing for jobs with favorite machines

Cong Chen [a,*], Yinfeng Xu [a,b]

[a] School of Management, Xi'an Jiaotong University, Xi'an, China
[b] The State Key Lab for Manufacturing Systems Engineering, Xi'an, China

ARTICLE INFO

Article history:
Received 10 July 2018
Received in revised form 7 November 2018
Accepted 8 November 2018
Available online 15 November 2018

Keywords:
Load balancing game
Favorite machine
(Strong) Price of Anarchy

ABSTRACT

This paper studies the load balancing game for the *favorite machine* model, where each job has a certain set of favorite machines with the shortest processing time for the job. We obtain tight bounds on the *Strong Price of Anarchy* (strong PoA) for the general favorite machine model and a special case of the model. Our results generalize the well-known bounds on the strong PoA for the unrelated machine and identical machine models.

1. Introduction

The *unrelated* machine model is one of the most fundamental models of load balancing games (scheduling games), in which there are n jobs and m machines corresponding to *self-interested* users and resources, respectively. The processing time of a job changes from machine to machine. Each job will choose one machine to minimize its own cost (the load of the chosen machine), which will result in some *equilibrium*. Usually, the unrelated machine model will end up with some bad equilibrium, i.e., high values of *Price of Anarchy* and *Strong Price of Anarchy* (defined below), while the *identical* and the *related* machine models have much better performance in terms of the PoA and the strong PoA.

However, many practical problems are neither as simple as the identical or related machine cases nor as complicated as the general unrelated machine. In a sense, many practical problems are "intermediate" cases, such as the two products model [20] and the CPU–GPU cluster model [15,5]. In the above examples, each job has one processing time for some of the machines and a different processing time for the other machines.

Inspired by these examples, we propose the *favorite machine* model, where each job has a minimum processing time on a certain set of machines, called *favorite machines*, and longer processing times on other machines, called *non-favorite machines*. Denote the processing time of job j on machine i by p_{ji} and the minimum processing time of job j by $p_j = \min_i p_{ji}$. Thus, the set of favorite machines of job j is defined as $K_j = \left\{i |\, p_{ji} = p_j\right\}$, and the favorite machine model is as follows:

(*Favorite machine model*) This model is simply the unrelated machine model when every job has at least k favorite machines ($|K_j| \geq k$). The processing time of job j on a favorite machine is p_j and on any non-favorite machine i is an arbitrary value $p_{ji} \geq p_j$.

Note that this model interpolates between the *unrelated* machine model, where possibly only one machine has the minimal processing time for the job ($k = 1$) and the *identical* machine model ($k = m$). It can also be regarded as a "relaxed" version of *restricted assignment* problem, where each job j can only be allocated to a subset K_j of machines: the restricted assignment is essentially the case where the processing time of a job on a non-favorite machine is always ∞.

1.1. Contributions

As shown in Section 2 (Theorem 1), the *Price of Anarchy* (PoA) for the favorite machine model is unbounded (except for the identical machine case). Thus, it motivates the study of the *Strong Price of Anarchy* (strong PoA), which is the ratio between the cost of the worst *strong equilibrium* and the optimum. Tight bounds on the strong PoA are obtained for both a special case and the general favorite machine model (as shown in Sections 3 and 4, respectively). The results generalize the well-known bounds on the strong PoA for the unrelated and the identical machine models and illustrate the impact of the number of favorite machines on the performance of the model in terms of the strong PoA.

In Section 3, we first consider a special case of the favorite machine model as a warm-up. This case is an extension of prior work [4] moving from a two-machine case to a m-machine case. In this model, each job has exactly one favorite machine and the processing time on a non-favorite machine is s (> 1) times the processing time on the favorite machine. This type of setting can

* Corresponding author.
E-mail address: chencong2779@stu.xjtu.edu.cn (C. Chen).

https://doi.org/10.1016/j.orl.2018.11.004
0167-6377/

be observed in several studies, such as the related machine model where only one machine has a different speed [18,14,6] and the one-sided case of the CPU–GPU cluster model [5]. Since the PoA is already unbounded for the two-machine case [4], we focus on the strong PoA and obtain a tight bound of $\Theta\left(\frac{m}{\log m}\right)$.

In Section 4, we consider the general favorite machine model, where each job has at least k favorite machines and the processing time on a non-favorite machine is arbitrarily larger than on a favorite machine. The model is a natural restriction of the unrelated machine model and is a generalization of several classical models, such as the identical machine model, the two products model [20], and the CPU–GPU cluster model [15,5]. For the general favorite machine model, we show that the strong PoA is $\Theta(m/k)$, which characterizes the impact of k on the performance of the model in terms of the strong PoA. The result generalizes the well-known bounds on the strong PoA for the unrelated machine and identical machine models. Specifically, when $k = 1$, the model is the unrelated machine model and strong PoA $= \Theta(m)$; when $k = \Theta(m)$, the model has a constant strong PoA, like the identical machine model.

1.2. Related work

We provide below related work about the PoA and the strong PoA for the *unrelated*, *related*, and *identical* machine models and the *restricted assignment*. Note that several results regarding the *mixed* (strong) PoA that denote the (strong) PoA for *mixed* equilibria also appear in this section. A mixed equilibrium results when players choose a probability distribution over strategy. However, we focus in this paper only on the pure equilibria of the favorite machine model, where all players play pure strategies.

For the most general model, the *unrelated* machine, the PoA is unbounded [3] and the strong PoA is exactly m (i.e., the number of machines) [1,10]. Hence, several restrictions of the unrelated machine model drew more attention from researchers. One of the most studied restrictions is the *related* machine model, in which the processing time of a job on a machine is the size of the job divided by the speed of the machine. For the *related* machine model, Czumaj and Vöcking [8] showed that PoA $= \Theta\left(\frac{\log m}{\log\log m}\right)$ and *mixed* PoA $= \Theta\left(\frac{\log m}{\log\log\log m}\right)$, while strong PoA $= \Theta\left(\frac{\log m}{(\log\log m)^2}\right)$ was given by Fiat et al. [10]. The above bounds for the related machine model are still non constant, whereas the simple *identical* machine model can achieve constant bounds for pure equilibria. The PoA for the identical machine model is $2-\frac{2}{m+1}$, where the upper and lower bounds can be deduced from Finn and Horowitz [11] and Schuurman and Vredeveld [19], respectively. Andelman et al. [1] showed that the strong PoA has the same ratio as PoA. However, the mixed PoA is not constant which is $\Theta\left(\frac{\log m}{\log\log m}\right)$ [17,16,8].

There is another interesting restriction called the *restricted assignment*, where each job can only be processed with a fixed processing time by a subset of the machines. This model is also a special case of the favorite machine model, i.e., $k = 1$ and $p_{ji} = \infty$ if $i \notin K_j$ for the favorite machine model. Awerbuch et al. [3] showed that PoA $= \Theta\left(\frac{\log m}{\log\log m}\right)$ and *mixed* PoA $= \Theta\left(\frac{\log m}{\log\log\log m}\right)$, where the bounds are similar to the related machines. The result of (mixed) PoA also appeared in Gairing et al. [13].

2. Preliminaries

2.1. Model and definitions

The *favorite* machine model is defined as follows: There are m machines to process n jobs, denoted by $M := \{1, 2, \ldots, m\}$ and $J := \{1, 2, \ldots, n\}$, respectively. Let p_{ji} be the processing time of job j on machine i, and $p_j = \min_{i\in M} p_{ji}$ be the minimum processing time of job j. For each job j, we call the machines with minimum processing time p_j the set $K_j \subseteq M$ of *favorite machines*. We assume that each job has at least $k\,(\geq 1)$ favorite machines, i.e., $|K_j| \geq k$ for $j \in J$. Note that the processing time of job j on its favorite machines equals the minimum processing time p_j, while the processing time on its non-favorite machines can be any value that greater than p_j, i.e.:

$$p_{ji} = \begin{cases} p_j, & \text{if } i \in K_j \\ p_{ji} > p_j, & \text{if } i \notin K_j\,. \end{cases}$$

Each job can only be processed by one machine, and each machine can process only one job at a time.

Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ denote a schedule of the jobs in J, where σ_j is the machine to which job j is allocated. For a schedule σ, we denote by σ_{-j} the schedule for all jobs except job j, i.e., $\sigma_{-j} = (\sigma_1, \ldots, \sigma_{j-1}, \sigma_{j+1}, \ldots, \sigma_n)$. Similarly, we denote by $\sigma_{-\Gamma}$ the schedule for all jobs except jobs Γ, where $\Gamma \subset J$ is a subset of jobs J.

The *load* of machine i in schedule σ, denoted by $\ell_i(\sigma)$, is the total processing time of the jobs allocated to machine i, i.e., $\ell_i(\sigma) := \sum_{j\in J:\sigma_j=i} p_{ji}$. The *makespan* is the maximum load over all machines, denoted by $\ell_{\max}(\sigma) := \max_{i\in M} \ell_i(\sigma)$. For simplicity, we drop the parameters and simply use ℓ_i and $\ell_{\max}$ when there is no ambiguity.

In load balancing games, each job is a player and chooses a machine aiming at minimizing its *own cost*, the load of the machine it chooses. This will result in some *equilibrium*, such as *Nash equilibrium* and *strong equilibrium*:

Nash equilibrium (NE). A schedule σ is an NE if there is no *single* job $j \in J$ can benefit from changing its choice, i.e., $\forall \sigma'_j$: $\ell_j(\sigma_{-j}, \sigma'_j) \geq \ell_j(\sigma)$ where σ'_j could be any machine.

Strong equilibrium (SE) [2]. A schedule σ is an SE if there is no *coalition* Γ of jobs that can all simultaneously benefit from deviating from their choices, i.e., $\forall \sigma'_\Gamma: \exists j \in \Gamma, \ell_j(\sigma_{-\Gamma}, \sigma'_\Gamma) \geq \ell_j(\sigma)$ where σ'_Γ could be any schedule of the jobs Γ.

Note that NE and SE always exist for load balancing games [9,12,1].

An equilibrium is not necessarily optimal in terms of the *makespan*. In fact, the inefficiency of equilibrium is a central concept in algorithmic game theory, as it quantifies the efficiency loss resulting from selfish behaviors of the players. In particular, the following two notions have received significant attention:

Price of Anarchy (PoA) [17]. PoA is the ratio between the cost of the *worst* NE and the *optimum* NE.

Strong Price of Anarchy (strong PoA) [1]. Strong PoA is the ratio between the cost of the *worst* SE and the *optimum* SE.

Specifically, PoA $= \max_{\sigma\in \mathsf{NE}} \frac{\ell_{\max}(\sigma)}{\mathrm{OPT}}$ and strong PoA $= \max_{\sigma\in \mathsf{SE}} \frac{\ell_{\max}(\sigma)}{\mathrm{OPT}}$, where NE is the set of all pure NEs, SE is the set of all SEs and OPT is the optimal cost.

2.2. A lower bound of PoA

For the case $k = m$, the favorite machine model is the model of identical machine and *PoA* $= 2 - \frac{2}{m+1} < 2$ [11,19]. For $1 \leq k \leq m-1$, the PoA is *unbounded*, as shown by the following theorem.

Theorem 1. *For $1 \leq k \leq m-1$, the PoA of the favorite machine model is unbounded.*

Proof. We give an instance in which there are exactly m jobs $J = \{1, 2, \ldots, m\}$ (m is the number of machines). Each job j has only one non-favorite machine, which is machine j. The processing time of each job is 1 on favorite machines and $s\,(> 1)$ on the non-favorite machine. We give a schedule where each job j is on its non-favorite machine j. Observe that the cost of each job is s and it cannot be improved by moving the job to any other machine, i.e., the schedule is an NE. The optimal cost is 1, in which each job $j \in [1, m-1]$ chooses machine $j+1$ and job m chooses machine 1. Therefore, we have PoA $\geq s$, and PoA is unbounded if s is unbounded. This instance holds for any $1 \leq k \leq m-1$, because the number of favorite machines of each job is exactly $m-1 \geq k$. □

3. A special case

As a warmup, we consider a special case in this section, which is a direct extension of the model in Chen et al. [4] (from two-machine to m-machine). In this case, each job has exactly one favorite machine (i.e. $|K_j| = 1$), and the processing time on non-favorite machines is $s\,(> 1)$ times that on favorite machines, i.e.,

$$p_{ji} = \begin{cases} p_j\,, & i \in K_j\,;\\ s \cdot p_j\,, & \text{otherwise.}\end{cases}$$

We will provide almost tight bounds on the strong PoA for this case, i.e., $\frac{m-2}{W(m-2)} <$ strong PoA $\leq \frac{m-2}{W(m-2)} + 2$, where $W(m)$ is the Lambert-W function, which yields strong PoA $= \Theta\left(\frac{m}{\log m}\right)$.

The following lemma shows how to bound the root of function $\left(\frac{x}{x-1}\right)^y = x$ in terms of x, which is useful in generating the upper and lower bounds.

Lemma 1. *Denote by $x^\star$ the root of equation $\left(\frac{x}{x-1}\right)^y = x$ in terms of x, where $x \geq 1$ and $y \geq 0$. It holds that $\frac{y}{W(y)} < x^\star < \frac{y}{W(y)} + 1$, where $W(y)$ is the Lambert-W function which is defined to be the function satisfying $W(y)e^{W(y)} = y$.*

Proof. To solve $\left(\frac{x}{x-1}\right)^y = x$, we take ln of both sides of the equation so that $y \ln \frac{x}{x-1} = \ln x$. It can be rewritten as

$$\frac{\ln x - \ln(x-1)}{x-(x-1)} = \frac{\ln x}{y}\,. \tag{1}$$

To handle this equation, we define a function $f(z) = \ln z$, whose derivative is $f'(z) = \frac{\mathrm{d}\ln z}{\mathrm{d}z} = \frac{1}{z}$. For the function $f(z)$, the following property holds: for arbitrary $0 < a < b$, there is some $c\ (a < c < b)$ that satisfies $f'(c) = \frac{f(b)-f(a)}{b-a}$. From this property, the left hand side of (1) implies that there is an x' satisfying $x-1 < x' < x$ and $f'(x') = \frac{\ln x - \ln(x-1)}{x-(x-1)}$. Noting that $f'(x') = \frac{1}{x'}$ and (1), we have that $\frac{1}{x'} = \frac{\ln x}{y}$, i.e., $y = x' \ln x$. Since $x-1 < x' < x$, we have $(x-1)\ln x < y < x \ln x$, which implies

$$(x-1)\ln(x-1) < y < x \ln x\,. \tag{2}$$

According to the second inequality of (2), $y < x \ln x$, which is $\frac{y}{x} < \ln x$, it holds that $e^{\frac{y}{x}} < e^{\ln x} = x$, i.e., $\frac{y}{x}e^{\frac{y}{x}} < y$. Noting that $W(y)e^{W(y)} = y$ (the definition of the Lambert-W function), we have $\frac{y}{x}e^{\frac{y}{x}} < W(y)e^{W(y)}$. As ze^z increases in x, it follows that $\frac{y}{x} < W(y)$, i.e., $x > \frac{y}{W(y)}$. Similarly, we obtain that $x - 1 < \frac{y}{W(y)}$ by the first inequality of (2). Therefore, it holds that $\frac{y}{W(y)} < x < \frac{y}{W(y)} + 1$. □

3.1. Upper bound

Denote the schedule of the worst *strong equilibrium* by σ and the optimal schedule σ^*. Without loss of generality, we assume that the load ℓ_i of machine i in schedule σ satisfies

$$\ell_1 \leq \ell_2 \leq \ldots \leq \ell_m,$$

and the makespan of the optimal schedule σ^* is 1, i.e., $\ell_i^* \leq 1$ for all $i \in M$, where ℓ_i^* is the load of machine i in schedule σ^*.

Lemma 2. *$\ell_i \leq \left(\frac{s}{s-1}\right)^{i-1}$, where $i \in M\backslash\{m\}$.*

Proof (*By induction*)**.** For $i = 1$, $\ell_1 \leq 1$ holds; otherwise, all jobs will benefit from moving to the optimal schedule σ^*. We then show that if $\ell_{i-1} \leq \left(\frac{s}{s-1}\right)^{i-2}$ holds, $\ell_i \leq \left(\frac{s}{s-1}\right)^{i-1}$ is also true.

Let J_{im} be the jobs that are on machines i to m in schedule σ. We separate the proof into two cases.

Case 1. There exists a job in J_{im} whose favorite machine is among machines 1 to $i-1$.

Suppose $j \in J_{im}$ is the job whose favorite machine i'' satisfies $1 \leq i'' \leq i-1$ and job j itself is on machine i' ($i \leq i' \leq m$). We know that machine i' must be a non-favorite machine of job j (because each job has only one favorite machine), which yields $p_{ji'} = s \cdot p_j \leq \ell_{i'}$ so that $p_j \leq \frac{\ell_{i'}}{s}$. To guarantee that job j does not have the incentive to move to its favorite machine i'', it holds that $\ell_{i'} \leq \ell_{i''} + p_{ji''} = \ell_{i''} + p_j \leq \ell_{i''} + \frac{\ell_{i'}}{s}$, implying $\ell_{i'} \leq \frac{s}{s-1}\ell_{i''}$. Therefore,

$$\ell_i \leq \frac{s}{s-1}\ell_{i-1} \leq \left(\frac{s}{s-1}\right)^{i-1},$$

since $\ell_i \leq \ell_{i'}$, $\ell_{i''} \leq \ell_{i-1}$ and $\ell_{i-1} \leq \left(\frac{s}{s-1}\right)^{i-2}$.

Case 2. The favorite machine of each job in J_{im} is among machines i to m.

We construct a *new schedule* of the jobs in J_{im}. Let $J'_{im} \subset J_{im}$ be the jobs that are on machines 1 to $i-1$ in the optimal schedule σ^*.

New schedule First, arrange the jobs of J'_{im} to their favorite machines. Then, arrange the rest of jobs $J_{im}\backslash J'_{im}$ following the optimal schedule σ^*. Note that the rearrangement is carried out only among machines i to m and the schedule of jobs on machines 1 to $i-1$ remains the same.

Denote by R the maximum load over machines i to m under the *new schedule*. We have $\ell_i \leq R$; otherwise, jobs J_{im} will follow the *new schedule* and all of them will improve.

Let $S = \sum_{j \in J'_{im}} p_j$ denote the total minimum processing time of jobs J'_{im}. It holds that $R \leq S + 1$, since jobs $J_{im}\backslash J'_{im}$ follow the optimal schedule and create a load no greater than 1. Note that machines 1 to $i-1$ are non-favorite machines of any job in J'_{im}. As the jobs J'_{im} choose their non-favorite machines in schedule σ^* and the makespan of schedule σ^* is 1, we obtain that $S \leq \frac{i-1}{s}$. Since $\ell_i \leq R \leq S + 1$, we have

$$\ell_i \leq R \leq S + 1 \leq \frac{i-1}{s} + 1 \leq \left(\frac{s}{s-1}\right)^{i-1},$$

where the last inequality can easily be proven by induction, as follows.

For $i = 1$, the base case is trivial. Suppose $\frac{i-1}{s} + 1 \leq \left(\frac{s}{s-1}\right)^{i-1}$ holds for $i = u$, i.e., $\frac{u-1}{s} + 1 \leq \left(\frac{s}{s-1}\right)^{u-1}$. For $i = u+1$, it follows that $\frac{u}{s} + 1 = \frac{u-1}{s} + 1 + \frac{1}{s} \leq \left(\frac{s}{s-1}\right)^{u-1} + \frac{1}{s} = \left(\frac{s}{s-1}\right)^u - \frac{1}{s}\left(\frac{s}{s-1}\right)^u + \frac{1}{s} \leq \left(\frac{s}{s-1}\right)^u$. Thus, $\frac{i-1}{s} + 1 \leq \left(\frac{s}{s-1}\right)^{i-1}$ holds for $i \in \mathbb{N}^+$. □

Lemma 3. *$\ell_m \leq \min\left\{\left(\frac{s}{s-1}\right)^{m-2} + 1,\ s + 1\right\}$.*

Proof. We separate the proof into two cases:

Case 1. There exists a job j on machine m whose favorite machine is $i' \neq m$.

Observe that $\ell_m \leq \ell_{i'} + p_j$; otherwise, job j can benefit from moving to machine i'. Since $\ell_{i'} \leq \ell_{m-1} \leq \left(\frac{s}{s-1}\right)^{m-2}$ (by Lemma 2) and $p_j \le 1$, we have $\ell_m \le \left(\frac{s}{s-1}\right)^{m-2} + 1$. In addition, it holds that $\ell_m \le \ell_1 + s \cdot p_j \le s+1$, as job j will not move to machine 1. To sum up, we have $\ell_m \le \min\left\{\left(\frac{s}{s-1}\right)^{m-2} + 1,\ s+1\right\}$.

Case 2. Machine m is the favorite machine of all the jobs on that machine.

If all the jobs on machine m are on the same machine in the optimal schedule σ^*, it holds that $\ell_m \le \ell_m^* \le 1$. Hence, we consider the case that there is at least one job j on machine m in schedule σ that is on a different machine $i' \ne m$ in the optimal schedule σ^*. Note that machine i' is a non-favorite machine of job j, so $s \cdot p_j \le \ell_{i'}^* \le 1$. Since job j would not move to machine 1, it holds that $\ell_m \le \ell_1 + s \cdot p_j \le 2 \le \min\left\{\left(\frac{s}{s-1}\right)^{m-2} + 1,\ s+1\right\}$. $\square$

Lemma 4. $\ell_m \le \frac{m-2}{W(m-2)} + 2$.

Proof. As the first term of $\min\left\{\left(\frac{s}{s-1}\right)^{m-2} + 1,\ s+1\right\}$ (Lemma 3) decreases in s and the second increases in s for $s \ge 1$, we know that $\ell_m \le s^\star + 1$, where $s^\star$ satisfies $\left(\frac{s^\star}{s^\star-1}\right)^{m-2} = s^\star$. By Lemma 1, we obtain $s^\star < \frac{m-2}{W(m-2)} + 1$; hence, $\ell_m \le \frac{m-2}{W(m-2)} + 2$. $\square$

Note that asymptotically $W(z) \sim \log z$. Thus, Lemma 4 yields the following theorem:

Theorem 2. *For the case $|K_j| = 1$, $p_{ji} = p_j$ for $i \in K_j$ and $p_{ji} = s \cdot p_j$ for $i \notin K_j$, it holds that* strong PoA $= \mathcal{O}\left(\frac{m}{\log m}\right)$.

3.2. Lower bound

Lemma 5. Strong PoA $\ge \frac{m-2}{W(m-2)}$.

Proof. We give a schedule that the load of each machine $i \in M$ is $\ell_i = \left(\frac{s}{s-1}\right)^{i-2}$, where s is set to be the root of equation $\left(\frac{s}{s-1}\right)^{m-2} = s$. By Lemma 1, we have $s > \frac{m-2}{W(m-2)}$; hence, the load of machine m is $\ell_m = \left(\frac{s}{s-1}\right)^{m-2} = s > \frac{m-2}{W(m-2)}$.

In the schedule, there are m jobs $J = \{1, 2, \ldots, m\}$ (m is the number of machines), and each job $j \in J$ is allocated to machine j. Job 1 (on machine 1) has the minimum processing time $p_1 = \frac{s-1}{s}$, and its favorite machine is machine 1; that is, $\ell_1 = \frac{s-1}{s}$. For job j ($2 \le j \le m$), it satisfies $p_j = \frac{1}{s}\left(\frac{s}{s-1}\right)^{j-2}$ and $K_j = \{j-1\}$. Thus, job $j \ge 2$ is allocated to non-favorite machine j with the processing time $p_{jj} = \left(\frac{s}{s-1}\right)^{j-2}$; that is, $\ell_i = \left(\frac{s}{s-1}\right)^{i-2}$ for $2 \le i \le m$. For any $j \ge 2$, if job j moves to its favorite machine $j-1$, it will have cost $\left(\frac{s}{s-1}\right)^{j-3} + \frac{1}{s}\left(\frac{s}{s-1}\right)^{j-2} = \left(\frac{s}{s-1}\right)^{j-2}$. Therefore, no job has the incentive to move, and it is easy to obtain that this schedule is an SE. The optimal schedule simply allocates all jobs to their favorite machines with a makespan no greater than 1. $\square$

Lemma 5 yields the following theorem:

Theorem 3. *For the case $|K_j| = 1$, $p_{ji} = p_j$ for $i \in K_j$, and $p_{ji} = s \cdot p_j$ for $i \notin K_j$, it holds that* strong PoA $= \Omega\left(\frac{m}{\log m}\right)$.

4. The general case

In this section, we consider the general model where each job has at least k favorite machine(s), i.e., $|K_j| \ge k$. The processing time on job j on machine i is $p_{ji} = p_j$ for $i \in K_j$, and $p_{ji} > p_j$ for $i \notin K_j$. We present the tight bound $\Theta(m/k)$ on the strong PoA, which characterizes the impact of the factor k.

Similarly to the previous section, let σ be the schedule of the worst SE, and σ^* be the optimal schedule. We assume that $\ell_1 \le \ell_2 \le \ldots \le \ell_m$ and $\ell_i^* \le 1$ for all $i \in M$.

Lemma 6. $\ell_i \le \max\left\{\ell_{i-\lceil k/\tau\rceil} + 1,\ \frac{2\tau-1}{\tau-1}\right\}$ *for any $\tau \in (1, k]$.*

Proof. Denote by J_{im} the jobs that on machines i to m in schedule σ. Let $J'_{im} \subset J_{im}$ be the jobs that are on machines 1 to $i-1$ in the optimal schedule σ^*. Define a set of machines $U := \{u \in M \mid \exists j \in J'_{im},\ 1 \le u \le i-1,\ p_{ju} \le 1\}$. The smallest element in U is denoted by $\bar{u}$. Let $j \in J'_{im}$ be the corresponding job whose processing time on machine $\bar{u}$ is no greater than 1, i.e., $p_{j\bar{u}} \le 1$. As job j will not move to machine $\bar{u}$, it holds that $\ell_{i'} \le \ell_{\bar{u}} + p_{j\bar{u}}$, where i' is the machine chosen by job j in schedule σ. Since $p_{j\bar{u}} \le 1$ and $\ell_i \le \ell_{i'}$, it follows that $\ell_i \le \ell_{\bar{u}} + 1$. Therefore, if $|U| \ge \lceil k/\tau\rceil$ (where τ is a constant and $1 < \tau \le k$), we have $\ell_i \le \ell_{i-\lceil k/\tau\rceil} + 1$.

For the case when $|U| \le \lfloor k/\tau\rfloor$, we construct a *new schedule* of jobs J_{im}.

New schedule For the jobs in $J_{im} \backslash J'_{im}$, arrange them following the optimal schedule σ^*, after which the load of each machine i' ($i \le i' \le m$) is no greater than 1. For the jobs in J'_{im}, pair them with their favorite machines among machines i to m while *minimizing* the makespan of machines i to m. (It is shown in the following that any job in J'_{im} has at least one of machines i to m as its favorite machine, so that the new schedule is feasible.)

Denote by R the maximum load over machines i to m under the *new schedule*. We obtain that $\ell_i \le R$; otherwise, jobs J_{im} will follow the *new schedule*, and all will improve.

Let $S = \sum_{j \in J'_{im}} p_j$ be the total minimum processing time of jobs J'_{im}. By the definition of U and J'_{im}, we know that all the jobs in J'_{im} are on machines in U in schedule σ^*. Otherwise, the processing time of jobs J'_{im} will be greater than 1 and the optimal makespan will not be 1. Therefore, it holds that $S \le |U|$. Furthermore, since $|U| \le \lfloor k/\tau\rfloor < k$ and each job has at least k favorite machines, we know that each job in J'_{im} must have at least $k - |U|$ favorite machines among machines i to m. Recall that in the *new schedule*, the jobs in J'_{im} are assigned to their favorite machines and the makespan is minimized. Thus, a job j on the machine with maximum load R will have $k - |U| - 1$ other favorite machines. Since the makespan is minimized, the load of each of the $k - |U| - 1$ machines is at least $R - 1$; otherwise, the makespan can be further reduced by moving job j to another favorite machine with a load less than $R-1$. Therefore, the above $k - |U|$ machines have a total load of at least $(k-|U|-1)(R-1)+R$. Since the jobs in $J_{im} \backslash J'_{im}$ can create a load of at most 1 on each machine, the rest of the loads are all created by jobs J'_{im}. Therefore, $S \ge (k-|U|-1)(R-1)+R-(k-|U|)$. According to $S \le |U|$, we have $|U| \ge (k-|U|-1)(R-1)+R-(k-|U|)$ implying $R \le \frac{2k-|U|-1}{k-|U|}$, that is, $\ell_i \le \frac{2k-|U|-1}{k-|U|} \le \frac{2k-\lfloor k/\tau\rfloor-1}{k-\lfloor k/\tau\rfloor} < \frac{2\tau-1}{\tau-1}$. $\square$

Theorem 4. *For the favorite machine scheduling game,* strong PoA $\le \lceil \tau \cdot m/k\rceil + \frac{\tau}{\tau-1}$ *for any $\tau \in (1, k]$. Specifically,* strong PoA $\le \lceil 2m/k\rceil + 2$ *when $\tau = 2$.*

Proof. According to Lemma 6 and $\ell_1 \le 1$, it holds that

$$\ell_{1+\lceil k/\tau\rceil} \le \max\left\{2,\ \frac{2\tau-1}{\tau-1}\right\} = \frac{2\tau-1}{\tau-1}.$$

We have $\ell_m \le \frac{2\tau-1}{\tau-1} + \left\lceil \frac{m-(1+\lceil k/\tau\rceil)}{\lceil k/\tau\rceil} \right\rceil \le \lceil \tau \cdot m/k\rceil + \frac{\tau}{\tau-1}$ by applying Lemma 6 $\left\lceil \frac{m-(1+\lceil k/\tau\rceil)}{\lceil k/\tau\rceil} \right\rceil$ times. $\square$

Theorem 5. *For the favorite machine scheduling game,* strong PoA $\ge \lceil m/k\rceil$.

Proof. We provide an instance containing m jobs and show that there is an SE schedule of the m jobs with a makespan of $\lceil m/k\rceil$, while the optimal schedule has a makespan of 1.

For the first k jobs, each has $k+1$ favorite machines. Job j $(1 \le j \le k)$ has favorite machines $K_j = \{1, 2, \ldots, k, m-k+j\}$. The processing time of job j on a favorite machine is 1, while on a non-favorite machine it is ∞, i.e., $p_{ji} = 1$ for $i \in K_j$ and $p_{ji} = \infty$ for $i \notin K_j$, where $1 \le j \le k$.

For the remaining $m-k$ jobs, each has k favorite machines. Job j $(k+1 \le j \le m)$ has favorite machines $K_j = \{j-k, j-k+1, \ldots, j-1\}$. The processing time is defined as

$$p_{ji} = \begin{cases} 1, & \text{for } i \in K_j\,; \\ \left\lceil \frac{j}{k} \right\rceil, & \text{for } i = j\,; \\ \infty, & \text{otherwise.} \end{cases}$$

The SE schedule is that each job j chooses machine j so that the load of machine m is $\lceil m/k \rceil$. It is not difficult to check that the schedule is an SE, since the first k jobs already have their best possible cost and no coalition of jobs can all benefit without the first k jobs. However, the optimal schedule is the one where job $j \in [1, k]$ chooses machine $m-k+j$, and job $j \in [k+1, m]$ chooses machine $j-k$. $\square$

5. Conclusion

We obtain tight bounds on the strong PoA for the *favorite* machine model. Specifically, the bound $\Theta\left(\frac{m}{k}\right)$ generalizes the results for the unrelated and the identical machine models, and captures the impact of k, the minimum number of favorite machines, on the performance of the model. This result is a supplement to some classical problems and reveals the relations between the two classical models: the unrelated and identical machine models.

To further the research in this paper, one could design some *coordination mechanisms* [7] for the problem. As the best-known coordination mechanism for the unrelated machine model has PoA $= \Theta(\log m)$, we conjecture that it would be PoA $= \Theta\left(\log \frac{m}{k}\right)$ for the favorite machine model.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [71601152, 71301168]; and the China Postdoctoral Science Foundation [2016M592811].

References

[1] N. Andelman, M. Feldman, Y. Mansour, Strong price of anarchy, Games Econom. Behav. 2 (65) (2009) 289–317.
[2] R.J. Aumann, Acceptable points in general cooperative n-person games, in: Contributions to the Theory of Games (AM-40), volume IV, Princeton University Press, Princeton, NJ, 1959, pp. 287–324.
[3] B. Awerbuch, Y. Azar, Y. Richter, D. Tsur, Tradeoffs in worst-case equilibria, Theoret. Comput. Sci. 361 (2) (2006) 200–209.
[4] C. Chen, P. Penna, Y. Xu, Selfish jobs with favorite machines: price of anarchy vs. strong price of anarchy, in: Proc. of the 11th Annual International Conference on Combinatorial Optimization and Applications (COCOA), 2017, pp. 226–240.
[5] L. Chen, D. Ye, G. Zhang, Online scheduling of mixed cpu-gpu jobs, Internat. J. Found Comput. Sci. 25 (06) (2014) 745–761.
[6] Y. Cho, S. Sahni, Bounds for list schedules on uniform processors, SIAM J. Comput. 9 (1) (1980) 91–103.
[7] G. Christodoulou, E. Koutsoupias, A. Nanavati, Coordination mechanisms, in: International Colloquium on Automata, Languages, and Programming (ICALP), Springer, Berlin, Heidelberg, 2004, pp. 345–357.
[8] A. Czumaj, B. Vöcking, Tight bounds for worst-case equilibria, ACM Trans. Algorithms 3 (1) (2007) 4.
[9] E. Even-Dar, A. Kesselman, Y. Mansour, Convergence time to nash equilibrium in load balancing, ACM Trans. Algorithms 3 (3) (2007) 32.
[10] A. Fiat, H. Kaplan, M. Levy, S. Olonetsky, Strong price of anarchy for machine load balancing, in: International Colloquium on Automata, Languages, and Programming (ICALP), vol. 4596, Springer, Berlin, Heidelberg, 2007, pp. 583–594.
[11] G. Finn, E. Horowitz, A linear time approximation algorithm for multiprocessor scheduling, BIT Numer. Math. 19 (3) (1979) 312–320.
[12] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, P. Spirakis, The structure and complexity of nash equilibria for a selfish routing game, Theoret. Comput. Sci. 410 (36) (2009) 3305–3326.
[13] M. Gairing, T. Lücking, M. Mavronicolas, B. Monien, The price of anarchy for restricted parallel links, Parallel Process. Lett. 16 (01) (2006) 117–131.
[14] T. Gonzalez, O. Ibarra, S. Sahni, Bounds for lpt schedules on uniform processors, SIAM J. Comput. 6 (1) (1977) 155–166.
[15] C. Imreh, Scheduling problems on two sets of identical machines, Computing 70 (4) (2003) 277–294.
[16] E. Koutsoupias, M. Mavronicolas, P. Spirakis, Approximate equilibria and ball fusion, Theory Comput. Syst. 36 (6) (2003) 683–693.
[17] E. Koutsoupias, C. Papadimitriou, Worst-case equilibria, in: Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS), Springer-Verlag, Bern, 1999, pp. 404–413.
[18] J.W.S. Liu, C.L. Liu, Bounds on scheduling algorithms for heterogeneous computing systems, in: Proceedings of IFIP Congress, 1974, pp. 349–353.
[19] P. Schuurman, T. Vredeveld, Performance guarantees of local search for multiprocessor scheduling, INFORMS J. Comput. 19 (1) (2007) 52–63.
[20] N. Vakhania, J.A. Hernandez, F. Werner, Scheduling unrelated machines with two types of jobs, Int. J. Prod. Res. 52 (13) (2014) 3793–3801.