Strong Price of Anarchy for Machine Load Balancing

Amos Fiat*, Haim Kaplan*, Meital Levy**, and Svetlana Olonetsky*

Tel Aviv University

Abstract. As defined by Aumann in 1959, a strong equilibrium is a Nash equilibrium that is resilient to deviations by coalitions. We give tight bounds on the strong price of anarchy for load balancing on related machines. We also give tight bounds for k-strong equilibria, where the size of a deviating coalition is at most k.

Key words: Game theory, Strong Nash equilibria, Load balancing, Price of Anarchy

1 Introduction

Many concepts of game theory are now being studied in the context of computer science. This convergence of different disciplines raises new and interesting questions not previously studied in either of the original areas of study. Much of this interest in game theory within computer science is due to the seminal papers of Nisan and Ronen [20] and Koutsoupias and Papadimitriou [17].

A *Nash equilibrium* ([19]) is a state in a noncooperative game that is stable in the sense that no agent can gain from unilaterally switching strategies. There are many "solution concepts" used to study the behavior of selfish agents in a non-cooperative game. Many of these are variants and extensions of the original ideas of John Nash from 1951.

One immediate objection to Nash equilibria as a solution concept is that agents may in fact collude and jointly choose strategies so as to "profit". There are many possible interpretations of the statement that a set of agents "profit" from collusion. One natural interpretation of this statement is the notion of a *strong equilibrium* due to Aumann [5], where no coalition of players have any joint deviation such that every member strictly benefits. Whereas mixed strategy Nash equilibria always exist for finite games [19], this is not in general true for strong equilibria.

* School of computer science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: {fiat,haimk,levymeit,olonetsk}@post.tau.ac.il. Research partially supported by the Israel Science Foundation and the German Israel Foundation.

** The research has been supported by the Eshkol Fellowship funded by the Israeli Ministry of Science.

Dagstuhl Seminar Proceedings 07261
Fair Division
http://drops.dagstuhl.de/opus/volltexte/2007/1225

Holzman and Law-Yone [16] characterized the set of congestion games that admit strong equilibria. The class of congestion games studied was extended by Rozenfeld and Tennenholtz in [21]. [21] also considered mixed strong equilibria and correlated mixed strong equilibria under various deviation assumptions, pure, mixed and correlated. Variants of strong equilibria include limiting the set of possible deviations (coalition-proof equililbria [8]) and assuming static predefined coalitions ([15, 14]).

The term *price of anarchy* was coined by Koutsoupias and Papadimitriou [17]. This is the ratio between the cost of the worst-case Nash equilibria and the cost of the social optimum. A related notion is the price of stability defined in [3], the ratio between the cost of the best Nash equilibria and the cost of the social optimum. These concepts have been extensively studied in numerous settings, machine load balancing [17, 18, 11, 7, 10], network routing [22, 6, 9], network design [4, 12, 1, 3, 13], etc.

Andelman *et al.* [2] initiated the study of the *strong price of anarchy* (SPoA), the ratio of the worst case strong equilibria to the social optimum. The authors also define the notion of a k-strong equilibrium, where no coalition of size up to k has any joint deviation where all strictly benefit. Analogous definitions can be made for the k-strong price of anarchy.

One may argue that the strong price of anarchy (which is never worse than the price of anarchy) removes the element of poor coordination and is entirely due to selfishness. Likewise, the k-strong price of anarchy measures the cost of selfishness and restricted coordination (up to k agents at once).

Our work here is a direct continuation of the work of Andelman *et al.* [2], and addresses many of the open problems cited there, in particular in the context of a load balancing game. In this setting agents (jobs) choose a machine, and job j placed on machine i contributes $w_j(i)$ to the load on machine i. Agents seek machines with small load, and the social cost usually considered is the makespan, *i.e.*, the maximal load on any machine. Whereas [2] considered strong price of anarchy and k-strong price of anarchy for unrelated machines, herein we primarily consider the strong price of anarchy for related machines (machines having an associated speed).

Our results.

1. Czumaj and Vocking [11] showed that the price of anarchy for load balancing on related machine is $\Theta(\log m/\log\log m)$, we show that the strong price of anarchy for load balancing on related machine is $\Theta\left(\log m/(\log\log m)^2\right)$. This is our most technically challenging result.
2. We also give tight results for the problems considered by [2]:
   (a) In [2] the strong price of anarchy for load balancing on m unrelated machines was shown to lie between m and $2m-1$. We prove that the true value is always m.
   (b) In [2], the k-strong price of anarchy for load balancing of n jobs on m unrelated machines is between $O(nm^2/k)$ and $\Omega(n/k)$. We prove that the k-strong price of anarchy falls in between and is $\Theta(m(n-m+1)/(k-m+1))$.

2 Preliminaries

A load balancing game consists of a set $M = \{M_1, \ldots, M_m\}$ of *machines*, a set $N = \{1, \ldots, n\}$ of *jobs* (agents). We use the terms machine i or machine M_i interchangeably. Each job j has a weight function $w_j()$ such that $w_j(i)$ is the running time of job j on machine M_i. When the machines are *unrelated* then $w_j()$ is an arbitrary positive real function. For *related* machines, each job j has weight, denoted by w_j, and each machine M_i has a speed, denoted by $v(i)$. The running time of job j on machine i is $w_j(i) = w_j/v(i)$ where w_j is the weight of job j. In game theoretic terms, the set of strategies for job j is the set of machines M. A state S is an assignment of jobs to machines. Let $m_S(j)$ be the machine chosen by job j in state S. The *load* on machine M_i in state S is $\sum_{j|M_i=m_S(j)} w_j(i)$.

Given a state S in which job j is assigned to machine M_i, we say that the *load observed by job* j is the load on machine M_i in state S. The *makespan* of a state S is the maximum load of a machine in S. Jobs seek to minimize their observed load. The state OPT (the social optimum) is a state with minimal makespan. We also denote the makespan in state OPT by OPT, and the usage would be clear from the context.

A *strong equilibrium* is a state where no group of jobs can jointly choose an alternative set of strategies so that every job in the group has a reduced observed load in the new state. In a *k-strong equilibrium* we restrict such groups to include no more than k agents. The *strong price of anarchy* is the ratio between the makespan of the worst strong equilibrium and OPT. The *k-strong price of anarchy* is the ratio between the makespan of the worst k-strong equilibrium and OPT.

3 Related Machines

Czumaj and Vocking [11] show that the price of anarchy of load balancing on related machines is $\Theta(\log m/ \log\log m)$. We show that the lower bound construction of [11] is not in strong equilibrium. We also give a somewhat weaker (and tight) lower bound of $\Omega(\log m/(\log\log m)^2)$. We first present the lower bound of [11] and claim that it is not resilient to deviation by coalitions.

The lower bound of [11]:

Consider the following instance in which machines are partitioned into $\ell + 1$ groups. Let these groups be G_0, G_1, …, G_ℓ with m_j machines in group G_j. We define $m_0 = 1$, and $m_{j+1} = (\ell - j) \cdot m_j$ for $j = 1, \ldots, \ell$. Since the total number of machines $m = \sum_{j=0}^{\ell} m_j$ and $m_\ell = \ell!$, it follows that $\ell \sim \log m/ \log\log m$. Suppose that all machines in G_j have speed $2^{(\ell-j)}$.

Consider the following Nash equilibrium, every machine in G_j receives $\ell - j$ jobs, each of weight $2^{(\ell-j)}$. Each such job contributes 1 to the load of its machine. The total load on every machine in G_j is therefore $\ell - j$. Machines in group G_ℓ have no jobs assigned to them.

The makespan is ℓ, obtained on machines of group G_0. Consider some job assigned to a machine from group G_j, this job has no incentive to migrate to a machine in a group of lower index since the load there is already higher than the load it currently observes. If a job on a machine in G_j migrate to a machine in G_{j+1} then it would observe a load of $\ell - (j+1) + 2^{(\ell-j)}/2^{(\ell-(j+1))} = \ell - j + 1 > \ell - j$ and even higher loads on machines of groups $G_{j+2}, \ldots, G_\ell$.

For the minimal makespan, move all jobs on machines in G_j to machines in G_{j+1} (for $j = 0, \ldots, \ell - 1$). There are sufficiently many machines so that no machine gets more than one job, and the load on all machines is $2^{\ell-j}/2^{\ell-(j+1)} = 2$. The price of anarchy is therefore $\Omega(\log m / \log\log m)$. (This is also shown to be an upper bound).

However, this is not a strong Nash equilibrium.

Lemma 1. *The above Nash equilibrium is not in strong equilibrium for $\ell > 8$.*

Proof. Consider the following scenario. We consider a deviation by a coalition consisting of all ℓ jobs on some machine of group G_0 (machine M) along with 3 jobs from each of ℓ different machines from group G_2 (machines $N_1, N_2, \ldots, N_\ell$). We describe the actions of the coalition in two stages, and argue that all members of the coalition benefit from this deviation.

All ℓ jobs located on machine $M \in G_0$ migrate to separate machines $N_1, N_2, \ldots, N_\ell$ in group G_2.

Following this migration, the load on machines N_i is $\ell + 2$ (it was $\ell - 2$, we added a job from machine M that contributed an extra 4 to the load). The load on machine M has dropped to zero (all jobs were removed).

Now, remove 3 original jobs (with weight $2^{\ell-2}$) from each of the N_i machines and place them on machine M. The load on machine N_i has dropped to $\ell - 1$, so the job that migrated from machine M to machine N_i is now experience lower load than before. The load on machine M is now $3\ell \cdot 2^{\ell-2}/2^\ell = 3\ell/4 < \ell - 2$, for $\ell > 8$. Thus, the jobs that migrated from machines in G_2 to machine M also benefit from this coalition.

3.1 Lower bound on strong price of anarchy for related machines.

Theorem 1. *The strong price of anarchy for m related machines and n jobs is $\Omega\left(\log m / (\log\log m)^2\right)$.*

Proof. Consider the following instance in which machines are partitioned into $\ell + 1$ groups. Let these groups be G_0, G_1, …, G_ℓ. We further subdivide each group G_i, $0 \le i < \ell$, into $\log \ell$ subgroups, where all machines within the same subgroup have the same speed, but machines from different subgroups of a group differ in speed. The group G_ℓ consists of a single subgroup $F_{\ell \log \ell}$. In total, we have $\ell \log \ell + 1$ subgroups $F_0, F_1, \ldots, F_{\ell \log \ell}$, where subgroups $F_0, \ldots, F_{\log \ell - 1}$ are a partition of G_0, $F_{\log \ell}, \ldots, F_{2 \log \ell - 1}$ are the subgroups of G_1, etc. The speed of each machine in subgroup F_j is $2^{(\ell \log \ell - j)}$.

Let m_j denote the number of machines in subgroup F_j, $0 \le j \le \ell \log \ell$. Then $m_0 = 1$, and for subgroup F_{j+1} such that $F_j \subset G_i$ we define $m_{j+1} = (\ell - i) \times m_j$. It follows that the number of machines in subgroup $F_{\ell \log \ell}$ is at least $(\ell!)^{\log \ell}$ and therefore $m \ge (\ell!)^{\log \ell}$ and $\ell \sim \log m / (\log\log m)^2$.

Consider the following state, S. Each machine of group G_i is assigned $\ell - i$ jobs. Jobs that are assigned to machines in subgroup F_j have weight $2^{(\ell \log \ell - j)}$. As the speed of such machines is $2^{(\ell \log \ell - j)}$, it follows that each such job contributes one to the load of the machine it is assigned to. *I.e.*, the load on all machines in G_i is $\ell - i$. Machines of $F_{\ell \log \ell}$ have no jobs assigned to them.

The load on the machines in group G_0 is ℓ which is also the makespan in S. The minimal makespan (OPT) is attained by moving the jobs assigned to machines from F_j each to a separate machine of subgroup F_{j+1}, for $0 \leq j < \ell \log \ell$. The load on all machines is now $2^{\ell \log \ell - j}/2^{\ell \log \ell - (j+1)} = 2$.

State S is a Nash equlibrium. A job assigned to a machine of subgroup F_j has no incentive to migrate to a machine with a lower indexed subgroup since the current load there is equal or higher to the current load it observes. There is no incentive to migrate to a higher indexed subgroup as it observes a load of at least $\ell - j + 1 > \ell - j$. We now argue that state S is not only a Nash Equilibrium but also a strong Nash equilibrium.

First, note that jobs residing on machines of group G_i, $0 \leq i \leq \ell - 2$, have no incentive to migrate to machines of group G_j, for $j \geq i + 2$. This follows since the speed of each machine in group G_j is smaller by a factor of more than $2^{\log \ell} = \ell$ from the speed of any machine in group G_i. Thus, even if the job is alone on such a machine, the resulting load is higher than the load currently observed by the job (current load is $\leq \ell$). Thus, any deviating coalition has the property that jobs assigned to machines from group G_i may only migrate to machines from groups G_j, for $j \leq i + 1$.

Suppose that jobs that participate in a deviating coalition are from machines in groups G_i, G_{i+1}, …, G_j, $1 \leq i \leq j \leq \ell$. The load on machines from group G_i holding participating jobs must strictly decrease since either jobs leave (and the load goes down) or jobs from higher or equal indexed groups join (and then the load must strictly go down too). If machines from group G_i have their load decrease, and all deviating jobs belong to groups i through j, $i < j$, then there must be some machine $M \in G_p$, $i < p \leq j$, with an increase in load. Jobs can migrate to machine M either from a machine in group G_{p-1}, or from a machine in group G_j for some $j \geq p$.

If a deviating job migrates from a machine in G_j for some $j \geq p$ then this contradicts the increase in the load on M. The contradiction arises as such jobs will only join the coalition if they become strictly better off, and for this to happen the load on M should decrease.

However, this holds even if the deviating job migrates to M from a machine in G_{p-1}. The observed load for this job prior to deviating was $\ell - (p - 1)$ and it must strictly decrease. A job that migrates to machine M from G_{p-1} increases the load by an integral value. A job that migrates away from machine M decreases the load by an integral value too. This implies that the new load on M must be an integer smaller than $\ell - (p - 1)$, which contradicts the increase in load on M. □

4 Upper bound on strong price of anarchy for related machines.

We assume that machines are indexed such that $v(i) \geq v(j)$ for $i < j$. We also assume that the speeds of the machines are scaled so that OPT is 1. Let S be an arbitrary strong Nash equilibrium, and let $\ell_{\max}$ be the maximum load of a machine in S. Our goal is to give an upper bound on $\ell_{\max}$. When required, we may assume that $\ell_{\max}$ is a sufficiently large constant, since otherwise an upper bound follows trivially. Recall that machines are ordered such that $v(1) \geq v(2) \geq \cdots \geq v(m) > 0$. Let $\ell(i)$ be the load on machine M_i, i.e., the total weight of jobs assigned to machine M_i is $\ell(i)v(i)$.

4.1 Sketch of the proof

We prove that $m = \Omega({\ell_{max}}^{\ell_{\max} \log \ell_{\max}})$, which implies $\ell_{\max} = O(\log m/(\log\log m)^2)$. To show that $m = \Omega({\ell_{\max}}^{\ell_{\max} \log \ell_{\max}})$ we partition the machines into consecutive disjoint "phases" (Definition 1), with the property that the number of machines in phase i is $\Omega(\ell)$ times the number of machines in phase $i-1$ (Lemma 4.3), where ℓ is the minimal load in phases 1 through i.

For technical reasons we introduce shifted phases (s-phases, Definition 2) which are in one-to-one correspondence to the phases. We focus on the s-phases of faster machines, so that the total drop in load amongst the machines of these s-phases is about $\ell_{\max}/2$. We next partition the s-phases into consecutive blocks. Let δ_i be the load difference between slowest machine in block $i-1$ and the slowest machine in block i. By construction we get that $\sum \delta_i = \Theta(\ell_{\max})$.

We map s-phases to blocks such that each s-phase is mapped to at most one block as follows (Lemmas 7 and 8), see Fig. 1.

- If $\delta_i < 1/\log \ell_{\max} \Rightarrow$ we map a single $(1 = \lceil \delta_i \log \ell_{\max} \rceil)$ s-phase to block i
- If $\delta_i \geq 1/\log \ell_{\max} \Rightarrow$ we map $\Omega(\delta_i \log \ell_{\max})$ s-phases to block i

Therefore the total number of s-phases is at least $\sum \delta_i \log \ell_{\max} = \Omega(\ell_{\max} \log \ell_{\max})$. Given the one-to-one mapping from s-phases to phases, this also gives us a lower bound of $\Omega(\ell_{\max} \log \ell_{\max})$ on the number of phases.

In Lemma 4.3 we prove that the number of machines in phase i is $\Omega(\ell_{\max})$ times the number of machines in phase $i-1$. This allows us to conclude that the total number of machines $m = \Omega({\ell_{max}}^{\ell_{\max} \log \ell_{\max}})$, or that $\ell_{\max} = O(\log m/(\log\log m)^2)$.

4.2 Excess Weight and Excess Jobs

Given that the makespan of OPT is 1, the total weight of all jobs assigned to machine σ in OPT cannot exceed $v(\sigma)$, the speed of machine σ. We define the excess weight on machine $1 \leq \sigma \leq m$ to be $X(\sigma) = (\ell(\sigma) - 1)v(\sigma)$. (Note that excess weight can be positive or negative).

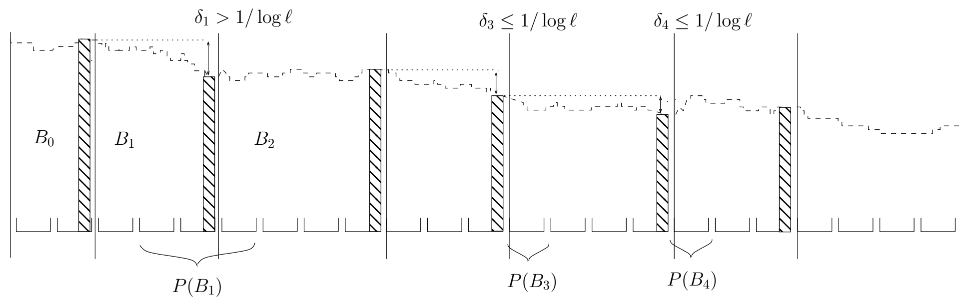

Fig. 1. The machines are sorted in order of decreasing speed (and increasing index), and partitioned into s-phases. The s-phases are further partitioned into blocks B_i. The s-phases that are mapped to block i are marked $P(B_i)$.

Given a set $R \subset \{1, \dots, m\}$, we define the excess weight on R to be

$$X(R) = \sum_{\sigma \in R} X(\sigma). \tag{1}$$

For clarity of exposition, we use intuitive shorthand notation for sets R forming consecutive subsequences of $1, \dots, m$. In particular, we use the notation $X(\sigma)$ for $X(\{\sigma\})$, $X(\le w)$ for $X(\{\sigma | 1 \le \sigma \le w\})$, $X(w \dots y)$ for $X(\{w \le \sigma \le y\})$, etc.

Given that some set of machines R has excess weight $X(R) > 0$, it follows that there must be some set of jobs $J(R)$, of total weight at least $X(R)$, that are assigned to machines in R by S, but are assigned to machines in $\{1, \dots, m\} - R$ by OPT. Given sets of machines R and Q, let $J(R \mapsto Q)$ be the set of jobs that are assigned by S to machines in R but assigned by OPT to machines in Q, and let $X(R \mapsto Q)$ be the weight of the jobs in $J(R \mapsto Q)$. Let R_1, and R_2 be a partition of the set of machines R. Then we have

$$X(R) \le X(R \mapsto \{1, \dots, m\} \setminus R) = X(R \mapsto R_1) + X(R \mapsto R_2)\ . \tag{2}$$

In particular, using the shorthand notation above, we have that for $1 \le y < \sigma \le m$,

$$X(\le y) \le X(\le y \mapsto > y) = X(\le y \mapsto y+1 \dots \sigma) + X(\le y \mapsto \sigma+1 \dots m)\ . \tag{3}$$

Similarly, for $1 \le \sigma < y \le m$ we have

$$X(\le y) \le X(\le \sigma) + X(\sigma + 1 \dots y)\ . \tag{4}$$

4.3 Partition into phases

Definition 1. *We partition the machines $1, \dots, m$ into disjoint sets of consecutive machines called phases, Φ_1, Φ_2, ..., where machines of Φ_i precede those of Φ_{i+1}. We define $\rho_0 = 0$ and $\rho_i = \max\{j \mid j \in \Phi_i\}$ for $i \ge 1$. Thus, it follows that $\Phi_i = \{\rho_{i-1} + 1, \dots, \rho_i\}$. It also follows that machines in Φ_i are no slower than those of Φ_{i+1}. Let n_i be number of machines in the ith phase,* i.e.*, $n_i = \rho_i - \rho_{i-1}$, for $i \ge 1$.*

To determine Φ_i it suffices to know ρ_{i-1} and ρ_i. For $i = 1$ we define $\rho_1 = 1$, as $\rho_0 = 0$ it follows that $\Phi_1 = \{1\}$. We define ρ_{i+1} inductively using both ρ_i and ρ_{i-1} as follows.

$$\rho_{i+1} = \operatorname{argmin}_\sigma \left\{ X(\leq \rho_i \mapsto > \sigma) < X(\leq \rho_{i-1}) + \frac{X(\Phi_i)}{2} \right\}. \tag{5}$$

The phases have the following properties.

Lemma 2. *Let ℓ be the minimal load of a machine in phases $1, \ldots, i$, ($\ell = \min\{\ell(\sigma) | 1 \leq \sigma \leq \rho_i\}$), then $n_{i+1} \geq n_i(\ell - 1)/2$.*

Proof. By the inductive definition of ρ_{i+1} above (Equation 5), we have that

$$X(\leq \rho_i \mapsto > \rho_{i+1}) < X(\leq \rho_{i-1}) + \frac{X(\Phi_i)}{2}.$$

Now, since $X(\leq \rho_i) \leq X(\leq \rho_i \mapsto \Phi_{i+1}) + X(\leq \rho_i \mapsto > \rho_{i+1})$, we have

$$\begin{aligned} X(\leq \rho_i \mapsto \Phi_{i+1}) &\geq X(\leq \rho_i) - X(\leq \rho_i \mapsto > \rho_{i+1}) \quad (6) \\ &> X(\leq \rho_{i-1}) + X(\Phi_i) - \left(X(\leq \rho_{i-1}) + \frac{X(\Phi_i)}{2} \right) \quad (7) \\ &= \frac{X(\Phi_i)}{2}; \quad (8) \end{aligned}$$

Equation (6) follows by rewriting Equation (2). Equation (7) follows from the definition of ρ_{i+1} (Equation (5)), and the rest is trivial manipulation.

Since the speed of any machine in Φ_i is no smaller than $v(\rho_i)$ (the lowest speed of any machine in phases $1, \ldots, i$), and we have chosen ℓ to be the minimum load of any machine in the set $\leq \rho_i$, for every machine $\sigma \in \Phi_i$ the excess weight $X(\sigma) = (\ell(\sigma) - 1)v(\sigma) \geq (\ell - 1)v(\rho_i)$. Therefore by substituting this into (8) we get

$$X(\leq \rho_i \mapsto \Phi_{i+1}) > \frac{n_i(\ell - 1)v(\rho_i)}{2}.$$

In OPT, no machine can have a load greater than one. Therefore since the speed of any machine in Φ_{i+1} is no larger than $v(\rho_i)$ at most $v(\rho_i)$ of the weight is on one machine in Φ_{i+1}, so there are at least $(\ell - 1)n_i/2$ machines in Φ_{i+1}. □

Lemma 3. *Let $j < i$ be two phases. If the minimal load of a machine $\rho_{j-1} \leq k \leq \rho_i$ is at least 3 then $X(\Phi_i) > X(\Phi_j)$.*

Proof. Clearly it suffices to prove that $X(\Phi_{i+1}) > X(\Phi_i)$ for every $i > 0$.

Since in OPT the load of every machine is at most one we have that

$$X(\leq \rho_i \mapsto \Phi_{i+1}) \leq \sum_{\sigma \in \Phi_{i+1}} v(\sigma). \tag{9}$$

This together with (8) gives that

$$\sum_{\sigma \in \Phi_{i+1}} v(\sigma) > \frac{X(\Phi_i)}{2} \ . \tag{10}$$

Let ℓ be the minimal load of a machine in Φ_{i+1}. From our definition follows that

$$X(\Phi_{i+1}) \geq (\ell - 1) \sum_{\sigma \in \Phi_{i+1}} v(\sigma) \ . \tag{11}$$

The lemma now follows by combining (10) and (11) together with the assumption that $\ell > 3$. □

Let ℓ be the minimal load among machines $1, \ldots, \rho_i$. Let Γ_i be the subset of Φ_i that have at least $(\ell-1)/2$ of their load contributed by jobs of weight $w \leq v(\rho_{i+1})$.

Lemma 4. *For $i > j$, $\sum_{\sigma \in \Gamma_i} v(\sigma) \geq v(\rho_j) n_j (\ell - 1)/(\ell + 3)$.*

Proof. First we want to estimate $X(\Phi_i \mapsto \geq \rho_{i+1})$. By rewriting Equation (2) we get that

$$X(\Phi_i \mapsto \geq \rho_{i+1}) = X(\leq \rho_i \mapsto \geq \rho_{i+1}) - X(\leq \rho_{i-1} \mapsto \geq \rho_{i+1}) \ .$$

Since $X(\leq \rho_{i-1} \mapsto \geq \rho_{i+1}) \leq X(\leq \rho_{i-1} \mapsto > \rho_i)$, we also have that

$$X(\Phi_i \mapsto \geq \rho_{i+1}) \geq X(\leq \rho_i \mapsto \geq \rho_{i+1}) - X(\leq \rho_{i-1} \mapsto > \rho_i) \ . \tag{12}$$

From the definition of ρ_{i+1} follows that

$$X(\leq \rho_i \mapsto \geq \rho_{i+1}) \geq X(\leq \rho_{i-1}) + \frac{X(\Phi_i)}{2} \ , \tag{13}$$

Similarly, from the definition of ρ_i follows that

$$X(\leq \rho_{i-1} \mapsto > \rho_i) < X(\leq \rho_{i-2}) + \frac{X(\Phi_{i-1})}{2} \ . \tag{14}$$

Substituting Equations (13) and (14) into Equation (12) we get that

$$\begin{aligned} X(\Phi_i \mapsto \geq \rho_{i+1}) &\geq X(\leq \rho_{i-1}) + \frac{X(\Phi_i)}{2} - \left(X(\leq \rho_{i-2}) + \frac{X(\Phi_{i-1})}{2} \right) \\ &\geq X(\leq \rho_{i-2}) + X(\Phi_{i-1}) + \frac{X(\Phi_i)}{2} - \left(X(\leq \rho_{i-2}) + \frac{X(\Phi_{i-1})}{2} \right) \\ &\geq \frac{X(\Phi_{i-1})}{2} + \frac{X(\Phi_i)}{2} \ . \end{aligned} \tag{15}$$

Let $A(\sigma)$ be the total weight of jobs j on machine σ with $w_j \leq v(\rho_{i+1})$ and let $A(\Phi_i) = \sum_{\sigma \in \Phi_i} A(\sigma)$. Since every job in $J(\Phi_i \mapsto \geq \rho_{i+1})$ has weight of at most $v(\rho_{i+1})$, it follows that $X(\Phi_i \mapsto \geq \rho_{i+1}) \leq A(\Phi_i)$, and by Equation (15)

$$A(\Phi_i) \geq \frac{X(\Phi_{i-1})}{2} + \frac{X(\Phi_i)}{2} \ . \tag{16}$$

We claim that every machine σ with $A(\sigma) > 0$ (*i.e.* the machine has at least one job j with $w_j \leq v(\rho_{i+1})$) has load of at most $\ell + 1$. To prove the claim, let $q \leq \rho_{i+1}$ be a machine that has load greater than $\ell + 1$ and a job j with $w_j \le v(\rho_{i+1})$, and let q' be the machine among $1, \ldots, \rho_i$ with load ℓ. This state is not a Nash equilibrium since if job j switches to machine q' it would have a smaller cost. We get that

$$\begin{aligned} A(\Phi_i) &\le \sum_{\sigma \in \Gamma_i} v(\sigma)(\ell+1) + \sum_{\sigma \in \Phi_i - \Gamma_i} v(\sigma)\frac{\ell-1}{2} \\ &= \sum_{\sigma \in \Gamma_i} v(\sigma)\frac{\ell+3}{2} + \sum_{\sigma \in \Phi_i} v(\sigma)\frac{\ell-1}{2} \\ &\le \sum_{\sigma \in \Gamma_i} v(\sigma)\frac{\ell+3}{2} + \frac{X(\Phi_i)}{2} \ . \end{aligned} \tag{17}$$

Inequality (17) holds ℓ is the smallest load of a machine in $1, \ldots, \rho_i$. Combining (17) and (16) we get that

$$\sum_{\sigma \in \Gamma_i} v(\sigma)\frac{\ell+3}{2} + \frac{X(\Phi_i)}{2} \ge \frac{X(\Phi_{i-1})}{2} + \frac{X(\Phi_i)}{2} \ , \tag{18}$$

and therefore

$$\sum_{\sigma \in \Gamma_i} v(\sigma)\frac{\ell+3}{2} \ge \frac{X(\Phi_{i-1})}{2} \ge \frac{X(\Phi_j)}{2} \ge v(\rho_j) n_j \frac{\ell-1}{2} \ . \tag{19}$$

The first inequality in (19) follows from (18), the second follows from Lemma 3, and the third inequality follows since ℓ is the smallest load of a machine in $1, \ldots, \rho_i$ and since $v(\rho_j)$ is the smallest speed in Φ_j. From (19) the lemma clearly follows. □

Recall that $\ell_{\max}$ is the maximum load in S. Define k to be $\min\{i \mid \ell(i) < \ell_{\max}/2\}$. Let t be the phase such that $\rho_t < k$ and $\rho_{t+1} \ge k$. Consider machines $1, \ldots, \rho_t$. From now on ℓ would be the minimal load of a machine in this set of machines. Then, $\ell = \Theta(\ell_{\max})$, and we may assume that ℓ is large enough.

Definition 2. *We define another partition of the machines into shifted phases (s-phases) Ψ_1, Ψ_2, … based on the partition to phases Φ_1, Φ_2, … as follows. We define $\varphi_0 = 0$. Let φ_i be the slowest machine in Φ_i such that at least $(\ell-1)/2$ of its load is contributed by jobs with weight $w \le v(\rho_{i+1})$ (there exists such a machine by Lemma 4). We define $\Psi_i = \{\varphi_{i-1}+1, \ldots, \varphi_i\}$.*

Note that there is a bijection between the s-phases $\Psi_1, \Psi_2, \ldots$ and the phases $\Phi_1, \Phi_2, \ldots$. Furthermore, all machines in Φ_i such that at least $(\ell-1)/2$ of their load is contributed by jobs of weight $\le v(\rho_{i+1})$ are in Ψ_i.

Lemma 5. *The load difference between machines φ_2 and φ_t, $\ell(\varphi_2) - \ell(\varphi_t) > \ell_{\max}/4 + 4$.*

Proof. According to the definition of Ψ_i, there is a job on machine φ_i with weight $w \le v(\rho_{i+1}) \le v(\varphi_{i+1})$ and therefore it contributes load of at most 1 to machine φ_{i+1}. As S is a Nash equilibrium, the load difference between machines φ_i and φ_{i+1} is at most 1. The load on the fastest machine φ_1 is $\ell(\varphi_1) > \ell_{\max} - 1$ since every job contributes a load of at most 1 to it. Thus, the load on machine $\varphi_2 \in \Phi_2$ is at least $\ell(\varphi_2) \ge \ell(\varphi_1) - 1 \ge \ell_{\max} - 2$.

The load on machine φ_t is $\ell(\varphi_t) \le \ell_{\max}/2 + 1$, since there is a job with weight of at most $v(\varphi_{t+1})$ on φ_s and by the definition of φ_i there is a machine $k \le \varphi_{t+1}$ with load less than $\ell_{\max}/2$.

Therefore, the load difference between machines φ_2 and φ_t is at least $(\ell_{\max}-2)-(\ell_{\max}/2+1) > \ell_{\max}/4+4$, for $\ell_{\max}$ sufficiently large (> 28). □

We define z_{i+1} to be $v(\varphi_i)/v(\varphi_{i+1})$. Notice that $z_{i+1} \geq 1$. We redefine Γ_b to be the subset of machines of Ψ_b such that for every such machine, at least $(\ell-1)/2$ of the load is contributed by jobs with weight $w \leq v(\varphi_{b+1})$. For two s-phases Ψ_a, and Ψ_b the lemma below relates the difference in load of φ_a and φ_b, to the ratio of speeds $v(\varphi_a)$ and $v(\varphi_b)$.

Lemma 6. *Consider s-phases Ψ_a and Ψ_b such that $a < b$. Let ℓ be the minimal load in Ψ_a and Ψ_b. If $v(\varphi_a)/v(\varphi_b) \leq z_{a+1}(\ell-1)/5$ then $\ell(\varphi_a) \leq \ell(\varphi_b) + 4/z_{b+1}$.*

Proof. Proof by contradiction, assume that $\ell(\varphi_a) = \ell(\varphi_b) + \alpha/z_{b+1}$, for some $\alpha > 4$, and that $v(\varphi_a)/v(\varphi_b) \leq z_{a+1}(\ell-1)/5$. We exhibit a deviating coalition all of whose members reduce their observed loads, contradicting the assumption that the current state is a strong equilibrium.

We observe that for every machine $\sigma \in \Gamma_b$ we have $\ell(\sigma) \leq \ell(\varphi_b) + 1/z_{b+1}$.(From this also follows that $\ell(\sigma) < \ell(\varphi_a)$.) If not, take any job j located on σ, such that $w_\sigma \leq v(\varphi_{b+1})$ and send it to machine φ_b, the contribution of job j to the load of φ_b is at most $v(\varphi_{b+1})/v(\varphi_b) = 1/z_{b+1}$, *i.e.*, the current state is not even a Nash equilibrium. Similarly, we have $\ell(\varphi_b) \leq \ell(\sigma) + 1/z_{b+1}$.

We group jobs on φ_a in a way such that the current load contribution of each group is greater than $1/(2z_{a+1})$ and no more than $1/z_{a+1}$. *I.e.*, for one such group of jobs G, $1/(2z_{a+1}) < \sum_{j\in G} w_j/v(\varphi_a) \leq 1/z_{a+1}$. At least $z_{a+1}(\ell-1)/2$ such groups are formed. Every such group is assigned a unique machine in Γ_b and all jobs comprising the group migrate to this machine. Let $\Gamma \subseteq \Gamma_b$ be a subset of machines that got an assignment, $|\Gamma| = \min\{z_{a+1}(\ell-1)/2, |\Gamma_b|\}$. The load contributed by migrating jobs to the target machine, $\sigma \in \Gamma_b$, is therefore

$$\sum_{j\in G} \frac{w_j}{v(\sigma)} \leq \sum_{j\in G} \frac{w_j}{v(\varphi_b)},$$

we also know that $v(\varphi_a)/v(\varphi_b) \leq z_{a+1}(\ell-1)/5$ and $\sum_{j\in G} w_j/v(\varphi_a) \leq 1/z_{a+1}$, this gives us that

$$\sum_{j\in G} \frac{w_j}{v(\varphi_b)} \leq \sum_{j\in G} \frac{w_j}{v(\varphi_a)} \cdot \frac{v(\varphi_a)}{v(\varphi_b)} \leq (\ell-1)/5.$$

Therefore, after migration, the load on $\sigma \in \Gamma_b$ is $\leq \ell(\sigma) + (\ell-1)/5 \leq \ell(\varphi_a) + (\ell-1)/5$. It is also at least $\ell(\varphi_a)$ (otherwise S is not a Nash equilibrium).

Additionally, jobs will also migrate from machines $\sigma \in \Gamma$ to machine φ_a (not the same jobs previously sent the other way). We choose jobs to migrate from $\sigma \in \Gamma$ to φ_a, so that the final load on σ is strictly smaller than $\ell(\varphi_a)$ and at least $\ell(\varphi_a) - 1/z_{b+1} = \ell(\varphi_b) + (\alpha-1)/z_{b+1}$. It has to be smaller than $\ell(\varphi_a)$ to guarantee that every job migrating from φ_a to σ observes a load strictly smaller than the load it observed before the deviation. We want it to be at least $\ell(\varphi_b) + (\alpha-1)/z_{b+1}$, so that a job migrating to φ_a from σ would observe a smaller load as we will show below. To achieve this, slightly more than $(\ell-1)/5$ of the load of $\sigma \in \Gamma$ has to migrate back to φ_a.

The jobs that migrate from $\sigma \in \Gamma$ to φ_a are those jobs with load $\leq 1/z_{b+1}$ on σ. Therefore, each such job which leaves σ reduces the load of σ by at most $1/z_{b+1}$. Since the total load of these jobs on σ is $(\ell-1)/2 > (\ell-1)/5$, we can successively send jobs from σ to φ_a until the load drops below to some value y such that $\ell(\varphi_b) + (\alpha-1)/z_{b+1} \leq y < \ell(\varphi_b) + \alpha/z_{b+1}$.

We argued that prior to any migration, the load $\ell(\sigma) \leq \ell(\varphi_b) + 1/z_{b+1}$ for $\sigma \in \Gamma_b$. Following the migrations above, the new load $\bar{\ell}(\sigma)$ on machine σ is $\bar{\ell}(\sigma) \geq \ell(\varphi_b) + (\alpha-1)/z_{b+1}$. Thus, the load on every such machine has gone up by at least $(\alpha-2)/z_{b+1}$.

If $|\Gamma| = z_{a+1}(\ell-1)/2$ the net decrease in load on machine φ_a is at least

$$\begin{aligned}\sum_{\sigma\in\Gamma} \frac{\alpha-2}{z_{b+1}} \cdot \frac{v(\sigma)}{v(\varphi_a)} &\geq \sum_{\sigma\in\Gamma} \frac{\alpha-2}{z_{b+1}} \cdot \frac{v(\varphi_b)}{v(\varphi_a)} \\ &\geq \frac{z_{a+1}(\ell-1)}{2} \cdot \frac{\alpha-2}{z_{b+1}} \cdot \frac{5}{z_{a+1}(\ell-1)} \\ &\geq \frac{2.5(\alpha-2)}{z_{b+1}} > \frac{\alpha+1}{z_{b+1}}.\end{aligned}$$

If $|\Gamma| < z_{a+1}(\ell-1)/2$, then $\Gamma = \Gamma_b$ and the net decrease in load on machine φ_a is at least

$$\begin{aligned}\sum_{\sigma\in\Gamma_b} \frac{\alpha-2}{z_{b+1}} \cdot \frac{v(\sigma)}{v(\varphi_a)} &\geq \frac{\alpha-2}{z_{b+1}v(\varphi_a)} \sum_{\sigma\in\Gamma_b} v(\sigma) \\ &\geq \frac{\alpha-2}{z_{b+1}} \cdot \frac{n_a(\ell-1)}{\ell+3} && (20) \\ &\geq \frac{2.5(\alpha-2)}{z_{b+1}} > \frac{\alpha+1}{z_{b+1}}. && (21)\end{aligned}$$

Inequality (20) follows from Lemma 4. Inequality (21) holds for $a > 1$ (according to Lemma for $a > 1$ we get that $n_a > (\ell-1)/2$) and for $\ell \geq 10$.

Thus, the new load $\bar{\ell}(\varphi_a)$ on machine φ_a is at most

$$\bar{\ell}(\varphi_a) < \ell(\varphi_b) + \alpha/z_{b+1} - (\alpha+1)/z_{b+1} = \ell(\varphi_b) - 1/z_{b+1},$$

which ensures that the jobs that migrate to machine φ_a could form a coalition, benefiting all members, in contradiction to the strong equilibrium assumption. □

We define a partition of the s-phases into blocks $B_0, B_1, \ldots$. The first block B_0 consists of the first two s-phases. Given blocks $B_0, \ldots, B_{j-1}$, define B_j as follows: For all i, let a_i be the first s-phase of block B_i and let b_i be be the last s-phase of block B_i. The first s-phase of B_j is s-phase $b_{j-1}+1$, *i.e.*, $a_j = b_{j-1}+1$.

To specify the last phase of B_j we define a consecutive set of s-phases denoted by P_1, where $b_j \in P_1$. The first s-phase in P_1 is a_j. The last s-phase of P_1 is the first phase, indexed p, following a_j, such that $v(\varphi_{b_{j-1}})/v(\varphi_p) > z_{a_j}(\ell-1)/5$. Note that P_1 always contains at least two s-phases. Let m_1 be an s-phase in $P_1 \setminus \{a_j\}$ such that $z_{m_1} \geq z_i$ for every i in $P_1 \setminus \{a_j\}$. We consider two cases:

– Case 1: $z_{m_1} \geq \log \ell$. We define $b_j = m_1 - 1$. In this case we refer to B_j as a block of a type I.
– Case 2: $z_{m_1} < \log \ell$. We define P_2 to be the suffix of P_1 containing all s-phases i for which $v(\varphi_{b_{j-1}})/v(\varphi_i) \geq z_{a_j}((\ell-1)/5)^{2/3}$. Note that s-phase p is in P_2 and s-phase a_j is not in P_2. Let m_2 be an s-phase in P_2 such that $z_{m_2} \geq z_i$ for every i in P_2. We define b_j to be $m_2 - 1$. In this case we refer to B_j as a block of type II.

If $v(\varphi_{b_{j-1}})/v(\varphi_t) \leq z_{a_j}(\ell-1)/5$ we do not define B_j and B_{j-1} is the last block.

For each block B_j let $P(B_j)$ be the s-phases which we map to B_j. In Case 1 we define $P(B_j) = m_1 = a_{j+1}$. In Case 2 we define $P(B_j) = P_2$.

Lemma 7. *The number of s-phases associated with block B_j, $|P(B_j)|$, is $\Omega(\log \ell / z_{a_{j+1}})$.*

Proof. If $z_{m_1} \geq \log \ell$ then $P(B_j)$ consists of a single phase. As $\log \ell / z_{m_1} < 1$, the claim trivially follows. Assume that $z_{m_1} < \log \ell$. Let s be the first s-phase in P_2, then

$$v(\varphi_{b_{j-1}})/v(\varphi_{s-1}) \leq z_{a_j} \left(\frac{\ell-1}{5}\right)^{2/3} . \tag{22}$$

Let k be the last s-phase of P_2 (which is also the last s-phase of P_1), we have that

$$v(\varphi_{b_{j-1}})/v(\varphi_k) \geq z_{a_j} \frac{\ell-1}{5} . \tag{23}$$

If we divide (22) by (23) we obtain that $v(\varphi_k)/v(\varphi_{s-1}) \geq ((\ell-1)/5)^{1/3}$. Let q be the number of s-phases in P_2. Since $z_{m_2} \geq z_i$ for all $i \in P_2$ it follows that $(z_{m_2})^q \geq ((\ell-1)/5)^{1/3}$. We conclude that $q = \Omega(\log \ell / z_{m_2}) = \Omega(\log \ell / z_{a_{j+1}})$, as $\log x \leq x$ for all x. □

The following lemma shows each s-phase is mapped into at most one block.

Lemma 8. *For every pair of blocks B, and B' we have $P(B) \bigcap P(B') = \emptyset$.*

Proof. This is clear if B is of type I and B' is of type II since we map to blocks of type I s-phase i for which $z_i < \log \ell$ and we map to blocks of type II s-phases i for which $z_i < \log \ell$.

The statement is also holds if both B and B' are of type I since each block is of size at least two. So we are left with case where both B and B' are of type II.

It is enough to prove it for two consecutive blocks B_j and B_{j+1}.

Let x_1,y_1 be the first and the last phase in $P(B_j)$, and let x_2 be the first phase in $P(B_{j+1})$.

From the definition of $P(B_j)$ follows:

$$v(\varphi_{b_{j-1}}) \ / \ v(\varphi_{x_1}) \geq z_{a_j}((\ell-1)/5)^{2/3} , \tag{24}$$

$$v(\varphi_{b_{j-1}}) \ / \ v(\varphi_{y_1}) > z_{a_j}(\ell-1)/5 , \tag{25}$$

$$v(\varphi_{b_{j-1}}) \ / \ v(\varphi_{y_1-1}) \leq z_{a_j}(\ell-1)/5 . \tag{26}$$

Since both blocks B_j and B_{j+1} are of type II, we have $v(\varphi_{y_1-1})/v(\varphi_{y_1}) < \log \ell$ and therefore,

$$v(\varphi_{b_{j-1}})/v(\varphi_{y_1}) < z_{a_j} \log \ell(\ell-1)/5 \ . \tag{27}$$

Using Inequality (27) and Inequality (24), we get

$$v(\varphi_{y_1}) > \frac{v(\varphi_{x_1})}{\log \ell((\ell-1)/5)^{1/3}} \ . \tag{28}$$

According to the definition, $b_j \geq x_1 - 1$ and therefore $v(\varphi_{b_j}) = z_{a_{j+1}} v(\varphi_{x_1})$. By substituting it in (28) we get,

$$v(\varphi_{y_1}) > \frac{v(\varphi_{b_j})}{z_{a_{j+1}} \log \ell((\ell-1)/5)^{1/3}} \ . \tag{29}$$

From the definition of $P(b_{j+1})$ follows:

$$v(\varphi_{b_j})/v(\varphi_{x_2}) \geq z_{a_{j+1}}((\ell-1)/5)^{2/3} \ . \tag{30}$$

Therefore, we get that

$$v(\varphi_{x_2}) \leq \frac{v(\varphi_{b_j})}{z_{a_{j+1}}((\ell-1)/5)^{2/3}} \ . \tag{31}$$

In order to avoid collision it have to be $v(\varphi_{x_2}) < v(\varphi_{y_1})$, so it is enough to show that

$$\frac{v(\varphi_{b_j})}{z_{a_{j+1}} \log \ell((\ell-1)/5)^{1/3}} > \frac{v(\varphi_{b_j})}{z_{a_{j+1}}((\ell-1)/5)^{2/3}} \tag{32}$$

The inequality holds for $\log \ell < ((\ell-1)/5)^{1/3}$. □

We now conclude the proof of the upper bound of the strong price of anarchy. By definition, we have that $v(\varphi_{b_{j-1}})/v(\varphi_{b_j}) \leq z_{a_j}(\ell-1)/5$, so using Lemma 6 we get that

$$\ell(\varphi_{b_{j-1}}) - \ell(\varphi_{b_j}) \leq 4/z_{a_j} \ . \tag{33}$$

Let f be the index of the last block. Then, b_f is the last phase of this block. We have that $v(\varphi_{b_f})/v(\varphi_t) \leq z_{b_f+1}(\ell-1)/5$, (where t is the last phase with minimal load $> \ell_{\max}/2$) so by Lemma 6, $\ell(\varphi_{b_f}) \leq \ell(\varphi_t) + 4$. By Lemma 5, $\ell(\varphi_2) - \ell(\varphi_t) \geq \ell_{\max}/4 + 4$. Therefore, $\ell(\varphi_2) - \ell(\varphi_{b_f}) \geq \ell_{\max}/4$. This together with Equation (33) gives that

$$\Theta(\ell_{\max}) = \ell(\varphi_2) - \ell(\varphi_{b_f}) = \sum_{j=1,\ldots,f} \big(\ell(\varphi_{b_{j-1}}) - \ell(\varphi_{b_j})\big) \leq \sum_{j=1,\ldots,f} 4/z_{b_j+1} \ . \tag{34}$$

Using Lemma 7 and Inequality (34) the total number of s-phases is

$$\sum_{i=1,\ldots,f} \Omega(\log \ell)/z_{b_i+1} = \log \ell \sum_{i=1,\ldots,f} 1/z_{b_i+1} = \Omega(\ell_{\max} \log \ell_{\max}) \ .$$

As described in the proof sketch this gives $\ell_{\max} = O(\log m/(\log\log m)^2)$ as required. We conclude:

Theorem 2. *The strong price of anarchy for m related machines is $\Theta\left(\log m/(\log\log m)^2\right)$.*

5 Unrelated Machines

5.1 Strong Price of Anarchy

We can show that the strong price of anarchy for m unrelated machine load balancing is at most m, improving the $2m-1$ upper bound given by Andelman *et al.* [2]. Our new upper bound is tight since it matches the lower bound shown in [2].

Theorem 3. *The strong price of anarchy for m unrelated machine load balancing is at most m.*

Proof. Omitted. Let s be a strong equilibrium. Let $M_1, \ldots, M_m$ be the machines ordered by decreasing loads in s, and let $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_m$ be their loads in s.

Note that $\ell_m \leq$ OPT. If $\ell_m >$ OPT then all jobs benefit from cooperating and moving to the optimal state OPT.

Next, we argue that $\ell_i \leq \ell_{i+1} +$ OPT for all $1 \leq i \leq m-1$. Assume that for some i, $\ell_{i+1} = x$ and $\ell_i > x + OPT$. Consider a coalition of all jobs running on machines M_j, $1 \leq j \leq i$, where each such job migrates to run on the machine in which it runs in state OPT. A job which migrates to a machine M_k for $k \geq i+1$ observes a load of at most $x +$ OPT which is strictly smaller than what it had previously observed. A job that migrates to machine M_k for $1 \leq k \leq i$ observes a load of $\leq$ OPT which is also strictly smaller than the previously observed load. This contradicts the assumption that s was a strong equilibrium.

Applying the argument above repeatedly, we conclude that $\ell_1 \leq \ell_m + (m-1)OPT$. Combining this with the fact that $\ell_m \leq$ OPT concludes the proof. □

5.2 *k*-Strong Price of Anarchy

In this section we consider coalitions of size at most k, where $k \geq m$ (for $k < m$ the upper bound is unbounded). Andelman et al. [2] show that for m machines and $n \geq m$ players the k-strong price of anarchy is $O(nm^2/k)$ and $\Omega(n/k)$. We give a refined analysis:

Theorem 4. *The k-strong price of anarchy for m unrelated machine load balancing, $k \geq m$, and given n jobs, $c = \Theta(m(n-m)/(k-m))$, more precisely, $(m-1)(n-m+1)/(k-m+1) \leq c \leq 2m(n-m+1)/(k-m+2)$.*

Proof. If $k < m$ then the k-strong price of anarchy is unbounded as shown in [2]. Thus, $k \geq m$. Consider the following scenario with m unrelated machines and n jobs. Each of the jobs has a finite weight only on two machines. Let $x = (m-1)(n-m+1)/(k-m+1)$. For $i \in \{1, \ldots, m-1\}$, the weight of job i on machine i is 1 and its weight on machine $i+1$ is $x - m + i + 1$. For $i \in \{m, \ldots, n\}$ the weight of job i on machine m is $1/(n-m+1)$ and its weight on machine 1 is $x/(n-m+1)$. See Figure 2.

The optimal solution assigns job i to machine i for $i \in \{1, \ldots, m-1\}$, and all other jobs are assigned to machine m. So, the load on all machines is 1 and the makespan is 1.

	1	2	..	$m-2$	$m-1$	m, ... ,n
M_1	1	∞	∞	∞	∞	$\frac{x}{n-m+1}$
M_2	$x-m+2$	1	∞	∞	∞	∞
M_3	∞	$x-m+3$	..	∞	∞	∞
..	∞	∞	..	∞	∞	∞
M_{m-2}	∞	∞	..	1	∞	∞
M_{m-1}	∞	∞	∞	$x-1$	1	∞
M_m	∞	∞	∞	∞	x	$\frac{1}{n-m+1}$

Fig. 2. This example shows that the k-strong price of anarchy $\geq (m-1)(n-m+1)/(k-m+1)$.

Consider the following state which we claim is a k-strong equilibrium. Assign job i, $i \in \{1, \ldots, m-1\}$, to machine $i+1$. Job i is the only job that runs on machine $i+1$ which therefore has load of $x-m+i+1$, all other jobs run on machine 1 with a total load of x, see Figure 3.

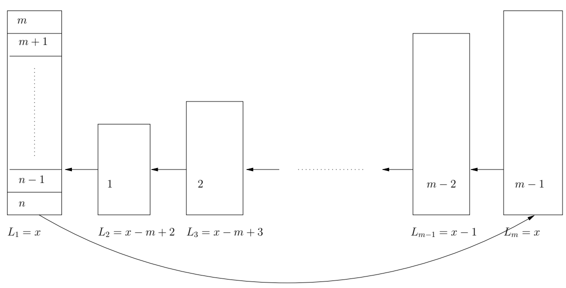

Fig. 3. Example of a Strong Equilibrium.

Since machine m has load of x which is the same as the load on machine 1 no job $i \in \{m, \ldots, n\}$ has incentive to migrate to machine m unless job $m-1$ leaves machine m. Furthermore, in order for job i, $i \in \{2, \ldots, m-1\}$, to join a coalition and move from machine $i+1$ to machine i, job $i-1$ has to leave machine i. So any coalition must include at least one job from each machine.

The load on machine 2 is $x-m+2$ and the load on machine 1 is x. Hence the load of machine 1 must decrease by more than $m-1$ so it would be beneficial for job 1 to migrate from machine 2 to machine 1. In order to reduce the load of machine 1 by more than $m-1$ units of weight, more than $(m-1)(n-m+1)/x = k-m+1$ jobs have to migrate from machine 1. Thus, such a coalition must include $> k-m+1$ jobs from machine 1 and all jobs $1 \leq i \leq m-1$, jointly these are more than k jobs. Since the largest allowable coalition is of size k, this deviation is illegal and, therefore, this state is a k-strong equilibrium. $\square$

Remark: We can improve the lower bound of Theorem 4 to $((m-1)(n-m+1)+1)/(k-m+1)$ by a slightly more careful choice of parameters.

Definition 3. *Let $M_1, \ldots, M_m$ be the machines sorted in decreasing order of load in state s. We say that M_i and M_j, are* directly connected, *and denote this by $M_i \rightarrowtail_s M_j$, if $i < j$ and there is a job that runs on M_j in* OPT *and runs on M_i in s.*

We say that machines M_i and M_j, $i < j$, are connected *in state s if there exist machines $M_{i'}$ and $M_{j'}$ such that $i' \leq i, j \leq j'$, and $M_{i'}$ and $M_{j'}$ are directly connected.*

Let $C(s) = M_1, \ldots, M_\ell$ denote the maximal prefix of machines (when ordered by decreasing loads), such that M_{i+1} is connected to M_i in state s.

A variation of this definition was also used by [2]. We use the following lemma which is proved in [2].

Lemma 9. *[2] Let s be a Nash equilibrium. Let $M_1, \ldots, M_m$ be the machines sorted by decreasing load in s, and let ℓ_i be the load on machine M_i. If $M_i \rightarrowtail_s M_j$ then $\ell_i \leq \ell_j + OPT$. In addition, for any $i, j \in C(s)$ we have $\ell_i \leq \ell_j + (m-1)OPT$.*

Proof. The proof can be found in [2].

Theorem 5. *For any job scheduling game with m unrelated machines, n jobs, $k \geq m$, and $n \geq m$, the k-strong price of anarchy is at most $2m(n-m+1)/(k-m+2)$.*

Proof. Let s be a strong Nash equilibrium with the largest makespan amongst all Nash equilibria and let $C(s) = M_1, \ldots, M_\ell$. Also let $\ell_{\max}$ be the load on M_1, the machine with the largest load, and let ℓ_{min} be the load on M_ℓ, the machine with the smallest load in $C(s)$. Note that if $\ell_{min} \leq$ OPT, then, by Lemma 9, the k-strong price of anarchy $\leq m \leq 2m(n-m+1)/(k-m+2)$. So we may assume for the rest of the proof that $\ell_{min} >$ OPT.

For every $i \leq \ell$ let S_i be a subset of jobs that run on machine $M_i \in C(s)$ of minimal cardinality, such that $\sum_{j \in S_i} w_i(j) > \ell_i - \ell_{min} +$ OPT. Let $s_i = |S_i|$.

We claim that $\sum_{i:M_i \in C(s)} s_i > k$. To establish this claim we show that if $\sum_{i:M_i \in C(s)} s_i \leq k$ then the jobs in $\cup_{i=1}^{\ell} S_i$ can jointly migrate so that they all benefit. This contradicts the assumption that s is a k-strong equilibrium. If $k = n$ then we cannot have $\sum_{i:M_i \in C(s)} s_i > k$ which means that ℓ_{min} could not have been larger than OPT and therefore the k-strong price of anarchy is $\leq m$.

To prove the claim let each job $j \in \cup_{i=1}^{\ell} S_i$ migrate to the machine on which it runs in state OPT. Consider a machine $M_i \in C(s)$. Since $\sum_{j \in S_i} w_i(j) > \ell_i - \ell_{min} +$ OPT the sum of the loads of the jobs leaving machine M_i is at least $\ell_i - \ell_{min} +$ OPT. But the sum of the loads of the jobs migrating to machine M_i is at most OPT. So the new load on machine M_i is less than $\ell_i - (\ell_i - \ell_{min} + \text{OPT}) + \text{OPT} = \ell_{min}$. This means that every job migrating to a machine in $C(s)$ sees an improvement. By definition of $C(s)$, no job migrates to a machine M_i, $i > \ell$, the claim follows.

Let $k' = \sum_{i:M_i \in C(s)} s_i$. We now know that $k' > k$. Note that the number of machines in $C(s)$ is ℓ and let n_i be the number of jobs running on machine M_i in s for every $1 \leq i \leq \ell$.

For every i such that $s_i > 1$ the weight of any $s_i - 1$ jobs is at most $\ell_i - \ell_{min} + \text{OPT}$ otherwise S_i is not of minimal cardinality. Therefore the total load on M_i is

$$\ell_i \leq \frac{n_i(\ell_i - \ell_{min} + \text{OPT})}{s_i - 1} . \tag{35}$$

Let x be the number of machines in $C(s)$ for which $s_i > 1$. Since $k' > m$ there exists some i such that $s_i > 1$ and hence $x \geq 1$. The total number of jobs on machines with $s_i > 1$ is $\sum_{i:s_i>1} n_i \leq n - (\ell - x)$. Also $\sum_{i:s_i>1} s_i = k' - (\ell - x)$.

We argue that there exists a machine M_i with $s_i > 1$ such that

$$n_i \leq \frac{s_i(n - \ell + x)}{k' - \ell + x}. \tag{36}$$

Indeed, if this is not true, for every such i we have $n_i > s_i(n - \ell + x)/(k' - \ell + x)$. Summing over all machines we obtain that

$$\sum_{i:s_i>1} n_i > \sum \frac{s_i(n - \ell + x)}{k' - \ell + x} = \frac{(n - \ell + x)\sum s_i}{k' - \ell + x} = n - \ell + x .$$

which is a contradiction.

Let M_p be a machine for which Equation (36) holds. Then using Equations (35) and (36) we obtain that

$$\ell_p \leq \frac{n_p(\ell_p - \ell_{min} + \text{OPT})}{s_p - 1} \leq \frac{s_p(n - \ell + x)(\ell_p - \ell_{min} + \text{OPT})}{(s_p - 1)(k' - \ell + x)} .$$

Since $\ell_{\max} = \ell_p + (\ell_{\max} - \ell_p)$ we have that

$$\ell_{\max} \leq \frac{s_p(n - \ell + x)(\ell_p - \ell_{min} + \text{OPT})}{(s_p - 1)(k' - \ell + x)} + (\ell_{\max} - \ell_p) ,$$

and since $\frac{s_p(n-\ell+x)}{(s_p-1)(k'-\ell+x)} > 1$ we obtain that

$$\ell_{\max} \leq \frac{s_p(n - \ell + x)(\ell_{\max} - \ell_{min} + \text{OPT})}{(s_p - 1)(k' - \ell + x)} .$$

Recall that by Lemma 9, $\ell_{\max} - \ell_{min} \leq (\ell - 1)\text{OPT}$, so we obtain that

$$\ell_{\max} \leq \frac{s_p(n - \ell + x)\ell}{(s_p - 1)(k' - \ell + x)}\text{OPT} .$$

Since $s_p/(s_p - 1) \leq 2$

$$\ell_{\max} \leq \frac{2(n - \ell + x)\ell}{k' - \ell + x}\text{OPT} ,$$

and since $\frac{(n-\ell+x)\ell}{k'-\ell+x}$ is maximized for $x = 1$, $\ell = m$, and $k' = k + 1$, the lemma follows. □

References

1. S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On nash equilibria for a network creation game. In *SODA*, pages 89–98, 2006.
2. N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. In *SODA*, 2007.
3. Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS*, pages 295–304, 2004.
4. Elliot Anshelevich, Anirban Dasgupta, Eva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *STOC*, pages 511–520, 2003.
5. R.J. Aumann. Acceptable points in general cooperative n-person games. In A.W. Tucker and R.D. Luce, editors, *Contribution to the Thoery of Games, Vol. IV (Annals of Mathematics Studies, 40)*, pages 287–324. 1959.
6. Baruch Awerbuch, Yossi Azar, and Amir Epstein. Large the price of routing unsplittable flow. In *STOC*, pages 57–66, 2005.
7. Baruch Awerbuch, Yossi Azar, Yossi Richter, and Dekel Tsur. Tradeoffs in worst-case equilibria. In *WAOA*, volume 2909, pages 41–52, 2003.
8. M Whinston BD Bernheim, B Peleg. Coalition-proof nash equilibria i: Concepts. *Journal of Economic Theory*, 1987.
9. George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *STOC*, pages 67–73, 2005.
10. George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. In *ICALP*, volume 3142, pages 345–357, 2004.
11. Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibria. In *SODA*, pages 413–420, 2002.
12. Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *PODC*, pages 347–351, 2003.
13. Amos Fiat, Haim Kaplan, Meital Levy, Svetlana Olonetsky, and Ronen Shabo. On the price of stability for designing undirected networks with fair cost allocations. In *ICALP*, pages 608–618, 2006.
14. Dimitris Fotakis, Spyros C. Kontogiannis, and Paul G. Spirakis. Atomic congestion games among coalitions. In *ICALP*, pages 572–583, 2006.
15. Ara Hayrapetyan, Éva Tardos, and Tom Wexler. The effect of collusion in congestion games. In *STOC*, pages 89–98. ACM Press, 2006.
16. Ron Holzman and Nissan Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21(1-2):85–101, 1997.
17. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.
18. M. Mavronicolas and P. Spirakis. The price of selfish routing. In *STOC*, pages 510–519, 2001.
19. John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
20. N. Nisan and A. Ronen. Algorithmic mechanism design (extended abstract). In *STOC*, 1999.
21. Moshe Tennenholtz Ola Rozenfeld. Strong and correlated strong equilibria in monotone congestion games. In *WINE*, pages 74–86, 2006.
22. Tim Roughgarden and Eva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.