Bounds on the Price of Anarchy in the KP Model

CS 364A Algorithmic Game Theory, Fall 2004
Paper Reading Project

Nikola Milosavljevic

1 Introduction

The topic of this report is the Koutsoupias-Papadimitriou (KP) game-theoretic model for a class of job-scheduling problems. It contains a presentation of the results from the original paper by Koutsoupias and Papadmitriou [1] and a follow-up paper by Czumaj and Vöcking [4]. The papers address the problem of bounding the ratio between the worst case Nash equilibrium and optimum allocation of a set of jobs to a set of machines.

First we introduce the details of the KP model, then give an overview of the results and present some of the proofs in more detail. The attempt is made to provide some intuition about the ideas and proof techniques used to obtain the sharp bounds. For a positive integer n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For any non-negative real x, we define $\log x := \max\{\log_2 x, 1\}$.

2 The KP Model and the Problem Statement

The KP model was first introduced in [4]. The original context was not job scheduling, but network routing. The authors want to study the behavior of a communication network with one source and one sink, whose users are allowed to behave selfishly in choosing source-sink paths along which to send their packets.

The attention is ultimately restricted to a very specific class of simple networks, in which the network links and network flows (from various users) can be identified with machines and jobs, respectively. Since both papers discuss only this subset of network topologies, throughout this report the context and the terminology of job-scheduling will be used.

Now we describe the details of the model. The basic setup of the game is the following

- There is a set of n jobs that act as selfish players. Let w_i ($i \in [n]$) denote the sizes (weights) of the jobs, which are fixed and known. Without loss of generality, we assume $w_1 \leq w_2 \leq \cdots \leq w_n$.
- There is a set of m machines on which the jobs are to be scheduled for execution. Let s_j ($j \in [m]$) denote the speeds of the machines (size of a job that can be executed in unit amount of time), also fixed and known parameters. Without loss of generality we assume $s_1 \geq s_2 \geq \cdots \geq s_m$.

- One job can be assigned to one machine only. Splitting jobs among several machines is not allowed. The outcome of the game is a particular assignment $(j_1, \dots, j_n) \in [m]^n$ of jobs to machines. As always, a deterministic outcome is equivalent to a *pure strategy* of the players.
- Given an outcome, the *load* of machine j is defined as the time needed to execute all jobs scheduled on it
$$C_j = \sum_{j_i = j} \frac{w_i}{s_j}.$$
- The *cost* that a given player i incurs in a given outcome is defined as the total load on its machine
- The *social cost* of an outcome is the maximum load over all machines, $\max\limits_{j \in [m]} C_j$
- As always, each player is interested in selecting a strategy which minimizes its own cost given the other players' strategies, and regardless of how that affects the social cost.

We also consider *mixed strategies*, i.e. probability distributions over $[m]$, one for each player. Each player chooses a machine, independently of the others, according to its mixed strategy. Let p_i^j be the probability that job i selects machine j. Clearly, any set of mixed strategies must satisfy $\sum_{j=1}^m p_i^j = 1$. Once we have mixed strategies, all the above defined quantities become random variables (not the social optimum, for obvious reasons), so we are interested in their expected values.

- The *expected load* of machine j
$$l_j = \mathrm{E}[C_j] = \sum_{i=1}^n \frac{w_i p_i^j}{s_j}.$$
- The *expected cost* for player i if it is assigned to machine j is actually the load of machine j conditioned on the event that job i is assigned to it
$$c_i^j = \mathrm{E}[C_j | j_i = j] = \frac{w_i + \sum_{k \neq i} w_k p_k^j}{s_j} = l_j + (1 - p_i^j)\frac{w_i}{s_j}.$$
- The *social cost* is the expectation of the maximum load over all machines, $C = \mathrm{E}\left[\max\limits_{j \in [m]} C_j\right]$

The rest of the report will deal only with the mixed strategy version of the game, noting as always that the pure strategy case is then recovered as a special case.

As noted above, each player ultimately wants to select a strategy that minimizes her own cost given the current strategies of other players. This goal is captured by standard notion of *Nash equilibrium*. In this case it is defined to be a set of probabilities p_i^j $(i \in [n], j \in [m])$ such that for job i, a non-zero probability p_i^j is assigned only to those machines j that minimize the cost c_i^j, i.e. that satisfy $c_i^j \leq c_i^{j'}$ for all $j' \in [m]$.

Optimal assignment (*social optimum*) is defined as a (deterministic) assignment that minimizes the maximum load.
$$\mathrm{OPT} = \min_{(j_1, j_2, \dots, j_n) \in [m]^n} \max_{j \in [m]} \frac{\sum\limits_{i : j_i = j} w_i}{s_j}$$

In this class of games, like in many others, a Nash equilibrium does not necessarily achieve the social optimum. Therefore the natural question to consider is the one of estimating *the price of anarchy*, defined in this case as $\mathrm{PoA} = \max(C/\mathrm{OPT})$, where the maximum is taken over all possible combinations of job sizes and machine speeds. The work presented in [1] and [4] successfully answers this question by showing tight upper and lower bounds on the price of anarchy in the job-scheduling model.

As already mentioned, the model is originally introduced in more general setting, namely that of network routing. In this framework, the players are the data packets that need to be transported from source to sink. A player's strategy is a routing path, and the incurred cost is the cost of the path. The cost is some function of the total traffic routed through individual links on the path. It is easy to see that for a general network topology and general cost functions the interactions among the players can be far more complicated that those in the job-scheduling model described above. However, when the network consists only of parallel links from source to sink, then selecting paths becomes equivalent to selecting individual edges. If we further assume that the cost functions of edges are linear in the amount of traffic, we recover the job-scheduling model.

Apparently, [4] was the paper to propose the price of anarchy in network routing as a useful measure of network performance. The justification is that in certain settings, such as the Internet, the network users indeed act without coordination. As a consequence, most of the time the network is actually in a Nash equilibrium. Then the proposed quantity is a measure of how much network performance is lost due to the lack of a central authority which would ensure that the social optimum is achieved by dictating the users' behavior. Although [1] and [4] solve the problem only for the case of several simple network topologies, the price of anarchy has been extensively studied for more general classes of networks [7]. The idea of investigating how bad a Nash equilibrium can be with respect to the socially optimal outcome turned out to be interesting in different settings [2] [3] [8], and lead to various nontrivial conclusions.

Besides introducing the general concept, [4] gives tight bounds for $m = 2$ and weak bounds for $m \geq 3$. The more recent paper of Czumaj and Vöcking [1] shows tight bounds on the price of anarchy for any m, thus completely answering the question for the case of scheduling. Next we list the more detailed descriptions of these contributions.

3 Overview of the Results

We first present the results of the Koutsoupias and Papadimitriou paper [4]. Most of them are special cases of the tight bound given in the second paper, so we won't bother to show the complete proofs, but only sketch and comment on some of them.

- If $m = 2$ and $s_1 = s_2$, then $\mathrm{PoA} = 3/2$, independent of n.
- If $m = 2$ and $s_2 \leq \phi s_1$, where $\phi = (1+\sqrt{5})/2$, then $\mathrm{PoA} \geq 1 + s_2/(s_1+s_2)$. In particular, the price of anarchy for two machines of different speeds can get as large as ϕ (when $s_2 = \phi s_1$). This result also does not depend on n. The paper does not give the corresponding upper bound for this case.
- For $m \geq 3$ and equal speeds, a lower bounds $\mathrm{PoA} = \Omega(\log m/\log\log m)$ is shown. The proof consists of simple observation that the mixed strategy given by $p_i^j = 1/m$ for all i, j is

equivalent to the occupancy problem of m balls being thrown at m bins uniformly at random. The claim then follows from the well known analysis of expected maximum occupancy.

- For $m \geq 3$, the authors also prove an upper bound PoA $\leq 3 + \sqrt{4m \ln m}$ if the speeds are equal and PoA $= O\left(\sqrt{\frac{s_m}{s_1}\sum_j \frac{s_j}{s_1}}\sqrt{\log m}\right)$ for the case of different speeds. The idea of this proof is essentially the same one that is used for its improved analog in [1]. The latter will be presented in full detail, and the differences between the two will be pointed out to show how the more careful analysis leads to a better bound.

Koutsoupias and Papadimitriou also consider a variant of the cost function with fixed *initial loads* on machines as additional parameters, as well as the *batch model* where each player's cost is halved, reflecting the fact that in this model tasks on one machine are executed in random batch order. We do not discuss these variations separately, because as pointed out in [1], they seem to have been abandoned in more recent literature, and they do not affect the sharp bounds.

The result of [1] are much more interesting, as they completely resolve the questions posed by [4], at least for the simple parallel networks. In particular, the paper gives an asymptotically tight upper and lower bounds on the price of anarchy for any m, both for the case of equal speeds and the case of different speeds. Specifically, it shows that the PoA is (asymptotically) upper bounded and lower bounded by the expression

$$\min\left(\frac{\log m}{\log\log\log m}, \frac{\log m}{\log\left(\frac{\log m}{\log(s_1/s_m)}\right)}\right)$$

In particular, this implies the asymptotic upper bound of $\log m/\log\log\log m$. Since this is the main contribution of the two papers, in the rest of the report we will go through (most of the) proofs in more detail. To that end, we first establish some infrastructure which appears to be the key ingredient in all the proofs.

Recall that the goal is to estimate the maximum ratio of C and OPT, taken over all possible input parameters (job and machine counts, job weights, machine speeds). This is achieved through an intermediate step, namely estimating *the maximum expected load* $c = \max_{j\in[m]} l_j$. Note that $c < C$ in general, that is maximum expected load is less than expected maximum load. However, c is also easier to compute and analyze, since the random variable under the expectation has a much simpler distribution. Fortunately, the authors are able to use the estimate of c to get both the upper and lower bound for C, with respect to OPT.

Another idea is to capture the structural properties of a Nash equilibrium by distinguishing certain machines at which important events happen. More precisely, imagine that the machines 1 to m are arranged in a sequence from left to right in the order of increasing indices (or nonincreasing speeds), then scan them from left to right and mark the positions at which the expected load drops below integer multiples of OPT for the first time. It turns out that most of the reasoning is based on looking at these transition points and their close vicinity. Tracking various quantities over this subset of machines gives us a handle on the equilibrium distribution of loads.

Let's also state this formally and introduce the notation, because it will make the main proofs much easier to present. For $k \geq 1$ define j_k to be the smallest index in $\{0\} \cup [m]$ such that $l_{j_k+1} < k\text{OPT}$, or if no such index exists $j_k = m$. We will often be referring to the obvious facts that $l_{j\leq j_k} \geq k\text{OPT}$ and $l_{j_k+1} < k\text{OPT}$, since they form the basis of many arguments.

4 The Proof of the Upper Bound

The goal of this section is to show the following asymptotic upper bound on the worst coordination ratio (price of anarchy)

$$\mathrm{PoA} = O\left(\min\left(\frac{\log m}{\log\log\log m}, \frac{\log m}{\log\left(\frac{\log m}{\log(s_1/s_m)}\right)}\right)\right) \tag{1}$$

The proof proceeds in two steps. The first step bounds the ratio c/OPT, which serves as an auxiliary quantity

$$c = \mathrm{OPT} \cdot O\left(\min\left\{\frac{\log m}{\log\log m}, \log\left(\frac{s_1}{s_m}\right)\right\}\right) \tag{2}$$

and the second step bounds C/OPT as a function of c/OPT.

$$C = \mathrm{OPT} \cdot O\left(\frac{\log m}{\log\left(\log m \cdot \frac{\mathrm{OPT}}{c}\right)}\right) \tag{3}$$

Combining (2) and (3) clearly yields the desired bound. We begin by separately proving the two estimates of c

Lemma 1 $c = OPT \cdot O\left(\frac{\log m}{\log\log m}\right)$

Lemma 2 $c = OPT \cdot O\left(\log\left(\frac{s_1}{s_m}\right)\right)$

which obviously imply (2) above.

Proof of Lemma 1 Before stating the formal proof we give a few intuitive arguments:

- Even though the expected load of the fastest machine is not necessarily highest in every Nash equilibrium, it must always be roughly within OPT from the maximum. If this was not the case, then the jobs on the most heavily loaded machine would certainly prefer the fastest machine. Therefore, to estimate c it suffices to estimate l_1.

- One expects that in a typical equilibrium the expected loads roughly decrease as the speeds decrease (from s_1 to s_m). Still, they cannot decrease too rapidly if we have a Nash equilibrium. Otherwise, jobs would want to migrate from faster to slower machines, if they are significantly less loaded.

These three facts together enable us to upper bound c. The speed in which the expected load decays in the machine sequence is captured by the relative distance between the "marked" machines $j_1, j_2, \ldots$.

Formally, one can show that $l_1 \geq c - \mathrm{OPT}$ (first intuitive point above). Any job is of size at most $s_1 OPT$, because otherwise no machine would be able to process it in OPT time. In particular, one such job has a positive probability for the most loaded (in expected sense) machine. Now if it were $l_1 < c - \mathrm{OPT}$, then it would be better for this job to switch to machine 1 (with probability 1), which is a contradiction.

Our notion of load that slowly decreases with speed is captured by the fact $j_k \geq (k+1)j_{k+1}$, which we show next. Consider the jobs with positive probability of one of the fist j_{k+1} machines. Their total weight is certainly at least as large as the expected total weight on this subset of machines, which is at least

$$(k+1)\mathrm{OPT}(s_1 + s_2 + \cdots + s_{j_{k+1}}) \geq \mathrm{OPT}(s_1 + s_2 + \cdots + s_{(k+1)j_{k+1}})$$

If it were $j_k < (k+1)j_{k+1}$, then it follows that at least one of these jobs is optimally allocated beyond j_k. This is either impossible (if $j_k = m$), or the size of this job is at most $s_{j_k+1}\mathrm{OPT}$. So its cost at $j_k + 1$ is at most

$$l_{j_k+1} + \frac{s_{j_k+1}\mathrm{OPT}}{s_{j_k+1}} < (k+1)\mathrm{OPT}$$

which violates the Nash condition.

So since $j_1 \leq m$ by definition, it follows $j_k = O(m/k!)$. Also, we have $j_{\lfloor c/\mathrm{OPT} \rfloor - 1} \geq 1$. This easily implies that c/OPT is roughly "inverse factorial" of m, which is $O(\log m / \log\log m)$, by Stirling formula. This completes the proof.

Proof of Lemma 2 The second estimate directly follows from the fact that the speeds decrease geometrically within the subsequence of marked machines, $s_{j_{k+2}+1}/s_{j_k+1} \geq 2$. This relation is established by considering a specific job i that has a non-zero probability in one of the heavily loaded machines ($j \leq j_{k+2}$), but its optimal placement is outside that set. Such job must exist, since for every $j \leq j_{k+2}$ the expected load is more than OPT. Once we this i, it gives us a way of comparing $s_{j_{k+2}+1}$ to s_{j_k+1}. We simply note that w_i is at most $s_{j_{k+2}+1}OPT$, to make the optimum achievable, and also at least $2s_{j_k+1}OPT$, to prevent the tendency of i to completely migrate from $j \leq j_{k+2}$ to some $j' \geq j_k$. This easily implies the claim about the geometric decrease in speed. Thus, there can be at most $O(\log(s_1/s_m))$ "markings" in the sequence. Every marking contributes an OPT toward l_1, which is very close to c, as shown before. This completes the proof of the lemma, at least at an intuitive level.

Next step is to upper bound C/OPT given c/OPT to obtain (3). This part of the proof has the familiar "maximum occupancy" flavor. Here is the outline of the steps, which should make the previous statement clear.

- Tail-bounding the probability that the actual loads on machines C_j exceed the maximum expected load c by a large amount. Hoeffding inequality is used in this case.
- Applying the union bound to get the same type of concentration result for the *maximum* load.
- Upper bounding $C = \mathrm{E}[\max_{j \in [m]} C_j]$ by plugging the previous result into the definition of expectation.

The most interesting part is bounding the deviation of C_j. This step deals with with the specifics of the model, while the other two are fairly straightforward calculations, the details of which we will skip occasionally. Even though both papers prove their upper bounds using the same high level plan described above, the proof in [1] does the tail-bounding part more carefully. That is where the sharpness of their bound comes from. Here are the details of the approach.

Note that each C_j is a sum of indicator variables

$$C_j = \sum_{i \in [n]} \frac{w_i}{s_j} J_i^j$$

where J_i^j is 1 with probability p_i^j and zero otherwise.

For the terms whose p_i^j is bounded away from zero, say $p_i^j > 1/4$, tail-bounding is fairly straightforward

$$C_j^{(1)} = \sum_{i:p_i^j>1/4} \frac{w_i}{s_j} J_i^j \leq \sum_{i:p_i^j>1/4} \frac{w_i}{s_j} \cdot 4\,\mathrm{E}[J_i^j] = 4\,\mathrm{E}\left[\sum_{i:p_i^j>1/4} \frac{w_i}{s_j} J_i^j\right] \leq 4c$$

This bound holds with probability 1. If p_i^j is arbitrarily close to zero, the previous approach does not work. However, each of the remaining terms with high probability does not contribute to the sum, in any case its contribution cannot exceed 12OPT, as Lemma 3 below shows. Then we can apply the Hoeffding inequality to bound the probability that the total contribution $C_j^{(2)}$ exceeds αc, where $\alpha > 1$ will be conveniently chosen later.

Lemma 3 $\quad 0 < p_i^j \leq \frac{1}{4} \quad \Rightarrow \quad \frac{w_i}{s_j} \leq 12\mathrm{OPT} \qquad \forall i \in [n], j \in [m]$

Proof The argument uses the result proved before, about the geometric decay of speeds. Intuitively, if j is between j_k+1 and j_{k-1} for some k, then w_i must be small enough to prevent migration to $j' = j_{k+2}+1$, since the speed of j' is at least twice that of j, and current expected load is greater by at most by 3OPT. The result follows just from comparing (the appropriate bounds on) the two expected loads

$$(k-1)\mathrm{OPT} + \frac{3}{4} \cdot \frac{w_i}{s_j} \leq (k+2)\mathrm{OPT} + \frac{w_i}{2s_j}$$

Note that what proof really requires for the probability is $p_i^j \leq 1/2 - \epsilon$, for $\epsilon > 0$. This is because the probabilities p_i^j and $p_i^{j'}$ also enter the equation when determining which machine is better for player i. In other words, from player i's perspective the speed ratio of 2 is modified by a multiplicative factor of $(1-p_i^j)/(1-p_i^{j'})$ which can be arbitrarily close to $1-p_i^j$. We do not want this factor to be too small and make the effective speed ratio less than or equal to 1, because that invalidates our argument (one of the necessary incentives for switching is gone).

Another technical point is dealing with the case when j is close enough to 1 so that the required j_{k+2} is not defined, i.e. there is no machine with expected load so high. This happens when $k = \lfloor c/\mathrm{OPT} \rfloor - 3$. In this case, taking machine 1 works just as well. Again, the result follows from the Nash condition

$$(c - 4\mathrm{OPT}) + \frac{3}{4} \cdot \frac{w_i}{s_j} \leq c + OPT$$

This concludes the proof of the lemma.

After applying the Hoeffding bound and doing some algebraic manipulation we get the result

$$\mathrm{P}[C_j \geq (4+\alpha)c] \leq (e/\alpha)^{\alpha c/(12\mathrm{OPT})} \qquad \text{for every } \alpha > 1$$

Then the union bound yields

$$\mathrm{P}\left[\max_{j \in [m]} C_j \geq (4+\alpha)c\right] \leq m(e/\alpha)^{\alpha c/(12\mathrm{OPT})} \qquad \text{for every } \alpha > 1 \tag{4}$$

By increasing α we can make the decay of (4) arbitrarily fast beyond $(4+\alpha)c$. Finally, it takes some more manipulation to deduce the value of α in which the expectation becomes bounded. It turns out that if we set

$$\alpha = \frac{\mathrm{OPT}}{c} \cdot \Theta\left(\frac{\log m}{\log\left(\frac{\mathrm{OPT}\cdot\log m}{c}\right)}\right)$$

the tail of the distribution adds at most a constant to the expectation. Thus we have shown the desired bound.

4.1 Identical Machines

For this case the authors show a simplified proof which also improves the bound for the general case (1). The improvement consists of showing that the multiplicative constant hidden in O-notation is actually at most 1, which implies that the worst price of anarchy is upper bounded by $\log m/\log\log m$ up to a constant *additive* factor.

Here is the basic idea of the simplified proof. Consider a machine j with really high expected load which is close to the cost of Nash equilibrium. When all speeds are equal, one can show that every job that has a positive probability on j, contributes to the expected load C_j roughly equally, in particular at least $C_j - \mathrm{OPT}$. This follows easily from the Nash condition. So when C_j is large fraction of OPT, then it is bounded just by bounding the number of participating jobs. After doing the calculations, this bound turns out to be what is needed, that is $\log m/\log\log m$ (notice no O-notation) with probability 1. For smaller C_j the Hoeffding bound can used to prove that the tail is small, and the authors claim that again a bound of $\log m/\log\log m + O(1)$ is obtained with high probability. Once the high probability bounds on each C_j are established, the proof can be easily completed like in the general case.

For the sake of comparison, it is worth mentioning that the approach of Koutsoupias and Papadimitriou yields a weaker corresponding bound of $O(\sqrt{m\log m})$ because the proof does not classify the jobs according to how much they contribute to the expected load on certain machine. Instead, each job's contribution is trivially bounded by its size, and the Azuma-Hoeffding bound is applied to the sum of the contributions. The authors themselves point out that the proof does not use most structural properties of Nash equilibria.

5 Proof of the Lower Bound

The goal of this section is to prove that the upper bound (1) from the previous section is asymptotically tight. This is done by first exhibiting a pure strategy Nash equilibrium that asymptotically matches the first component of the upper bound, i.e. achieves the worst case c/OPT ratio of equation (2). Of course, to match the other half of the bound we need a mixed strategy, since for pure strategies $c = C$ always. It turns out that (3) this can also be matched by introducing a small amount of randomization into the exiting pure strategy.

5.1 Lower Bound for Pure Strategies

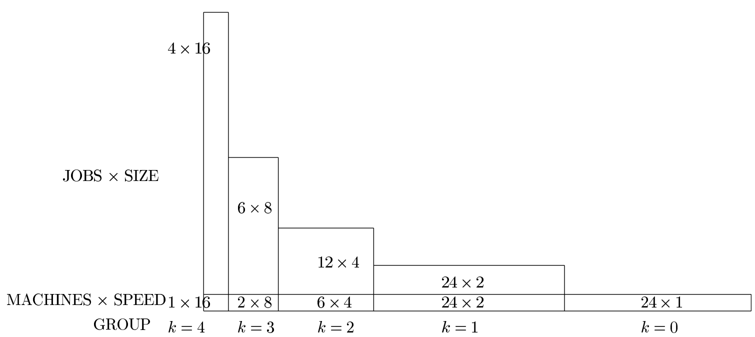
Fortunately, the key properties of the Nash equilibria that the previous proofs depend on (Lemmas 1 and 2) can be matched one by one, quite independently of each other, and in a very straightforward way. So the design is quite natural in the sense that each property can be easily mapped back to the lemmas that constitute the proofs of the upper bounds. Here is the description with the some intuitive justifications.

- There are $K+1$ groups of machines, numbered $0, 1, \ldots, K$. The value of K will be defined later. For each $0 \le k \le K$, the number of jobs on each machine in group k is equal to k. We are attempting to design the groups of machines according to the definition.

- The groups are of different sizes. For each $0 \le k \le K$, group $k+1$ has exactly $k+1$ times fewer machines than group k. Let the smallest group be of size 1, therefore groups 1 (and 0) have $K!$ machines. This construction aims to achieve the required distribution of the "markings" established by Lemma 1.

- Machines within a group have equal speeds. For $1 \le k \le K$, group $k+1$ is exactly 2 times slower than group k. Let the slowest group have unit speed, therefore group 0 has speed 2^K. This satisfies the requirement of geometric speed decay, Lemma 2.

Having defined the parameters and the strategies, we easily prove that the bound is actually achieved. Obviously, the maximum load is K, attained on any machine in group K. On the other hand, we get the social optimum of at most 2 (if we move each job from its current group prescribed by the Nash equilibrium, say k, to group $k-1$) and at least 1 (because of the large jobs of size 2^K cannot be processed faster than that by any machine). It is also easy to verify that the system is indeed in Nash equilibrium, because each group k is loaded just enough to prevent migration from group $k+1$. These are all very trivial calculations so we won't bother to show them formally. Perhaps it is enough to provide a pictorial explanation, see figure below



So the coordination ratio is $\Theta(K)$ in this case. We can choose any value of K such that the total number of machines does not exceed m which we are given beforehand. In particular, we can pick a K that satisfies $K! \le m/e$. Such K is $O(\log m / \log\log m)$, which is exactly what (2) predicts for this case.

5.2 Lower Bound for Mixed Strategies

Now allow players to randomize in such a way that the worst case gap between the maximum expectation c and the expected maximum C is achieved as a consequence. Once again, we can deduce a rather clean rationale underlying the construction. Intuitively the simplest approach would be to increase C without destroying the c/OPT ratio that we already have. Fortunately, this happens to be possible.

Since within each group things are uniform (equal job sizes, machines equally loaded), by allowing the jobs in single group to randomize we do not change the expectation for a single machine as long as the randomization is also uniform. This follows from the occupancy argument. The question is which group is the best choice. To answer this, we consider the situation in each group as an instance of the occupancy problem and invoke the results of the probabilistic analysis (for example, see [6]). We notice that the size of the possible gap between c and C that we can create increases with the group index, because so does the ratio of balls and bins. On the other hand, randomizing within group K of size 1 does not make sense. However, we can get around this by "scaling up" the group sizes in the pure strategy example by a suitable (integer) factor, so that the K-th group becomes large enough. It is straightforward to show that the first part of the proof remains valid if the machines are just replicated together with their loads.

Suppose the chosen factor is B (the number of "bins" in the K-th group). Then the expected maximum occupancy in group K is $\Theta\left(K + \frac{\log B}{\log(\log B/K)}\right)$ and we need to achieve $\Theta\left(\frac{\log m}{\log(\log m/K)}\right)$. Given the value of K computed above, the calculations show that $\log B = \Theta(\log m)$ is a good choice, so $B = m^{\alpha}$ where α is set to make B an integer. The only remaining constraint on α stems from the fact that now in order not to overflow the total number of machines it must be $BK! \leq m/e$. But setting $\alpha < 1$ allows K to remain the same asymptotically, so the above analysis (which assumes this order of K) still holds. The authors of [1] pick $\alpha = 1/2$ ($B = \sqrt{m}$), in the very beginning of their lower bound proof, anticipating the problems that arise in randomized version. The scaling is not strictly needed for the pure strategy example.

6 Remarks

Of course, the model that the papers mainly consider is far too simplified to be useful for the problems arising in real networks. The authors emphasize the importance of understanding the effects of the lack of coordination in the case of parallel networks, before more realistic settings can be studied.

Observe that none of the presented bounds depend on number of jobs or their sizes. By restricting the number of jobs or, say, the largest to smallest size ratio one might be able to get more optimistic worst case results. Note that the worst case example exhibited in the proof of the lower bound has a largest to smallest weight ratio of roughly 2^K, and given the choice of K, this grows almost as fast as the number of machines, in particular

$$2^K = 2^{\Theta(\log m/\log\log m)} = 2^{\omega(\log m/(\log m)^{\epsilon})} = \omega(m^{1-\epsilon})$$

for any small $\epsilon > 0$.

Another possible extension would be to keep the simple topology and scheduling interpretation, and consider more general class of cost functions. For example, one might allow speeds to vary depending on the load, or the size of the job being executed. As an extreme case, each job could have arbitrary different cost on each of the machines. For example, the case of unit weights and machine specific constant cost functions has been studied in [5]. In network setting, it has been proposed (also by [4]) to look at the "capacitated" cost functions of the form $1/(\text{cap} - \min(\text{cap}, \sum_i w_i))$, where cap is positive capacity of an edge, and $\sum_i w_i$ is the total traffic (load).

References

[1] Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibria. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 413–420. Society for Industrial and Applied Mathematics, 2002.

[2] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 347–351. ACM Press, 2003.

[3] R. Johari, S. Mannor, and J. Tsitsiklis. Efficiency loss in a network resource allocation game: the case of elastic supply, 2004.

[4] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Lecture Notes in Computer Science*, 1563:404–413, 1999.

[5] Igal Milchtaich. Congestion games with player-specific payoff functions.

[6] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms.* Cambridge University Press, 1995.

[7] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.

[8] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, page 416. IEEE Computer Society, 2002.