Resource Allocation Games of Various Social Objectives

Bo Chen[*]   Sinan Gürel
ORMS/DIMAP, University of Warwick

July 10, 2009

Abstract

In this paper, we study resource allocation games of two different cost components for individual game players and various social costs. The total cost of each individual player consists of the congestion cost, which is the same for all players sharing the same resource, and resource activation cost, which is proportional to the individual usage of the resource. The social costs we consider are, respectively, the total of costs of all players and the maximum congestion cost plus total resource activation cost.

Using the social costs we assess the quality of Nash equilibria in terms of the price of anarchy (PoA) and the price of stability (PoS). For each problem, we identify one or two problem parameters and provide parametric bounds on the PoA and PoS. We show that they are unbounded in general if the parameters involved are not restricted.

Keywords: resource allocation game, congestion cost, cost-sharing, price of anarchy, price of stability

1 Introduction

Problems of resource allocation often involve decentralized decision making. A typical example is, in terms of machine scheduling, allocation of machines to jobs (or assignment of jobs to machines) in which selfish agents, representing individual jobs, select machines for processing their own jobs. In the long run, decisions of the agents, motivated by individual interests, usually result in a *Nash equilibrium* (NE) at which no individual agent will benefit from any unilateral deviation for the current resource allocation. In terms of a given social objective, such an equilibrium is not necessarily, indeed, can often be far from optimal. It is important, therefore, to analyze the quality of NE solutions in terms of social optimality.

The resource allocation games we consider in this paper are as follows. Given a set of jobs, each of which has a positive weight and is controlled by a selfish agent. Each agent decides on which of the identical machines available to assign his job to. We consider two game models: the *load balancing* model and the *proportional cost sharing* model.

[*]Corresponding author: Warwick Business School, University of Warwick, Coventry, CV4 7AL, United Kingdom. `b.chen@warwick.ac.uk`

In the load balancing game, the cost of each agent is caused by a congestion, which is defined as the load of the machine, i.e., the sum of weights of the jobs assigned to it. This is the classical load balancing game as described in [15] and has been studied extensively as can be seen below. Our second game model is a generalization of the first, in which there are unlimited number of identical machines available, but usage of each machine comes with an additional set-up or *activation* cost, which is proportionally shared by all agents who assign their jobs to the machine according to the job weights. This model is recently introduced and studied in [8]. Therefore, in this model, each agent pays both the common congestion cost and the shared activation cost.

While cost functions of individual game players are the same as in those studies already existed in the literature, in our study of the aforementioned two models, we consider different social objectives in our assessment of the quality of NE solutions. For the load balancing model, our social objective is to minimize the total of individual players' congestion costs, which is also the total of machine loads. The rational of this social objective is that all jobs on a machine are released in a batch at the same time after the last job is finished. Hence, all agents who have assigned their jobs to the same machine will get their jobs done at the same time. Since this social cost gives the total time spent in the system by all jobs, it is closely related to the work-in-process inventory for the system. On the other hand, for the proportional cost-sharing model, we consider two different social costs: the total of congestion costs and activation costs; and the maximum congestion cost plus total activation costs.

As in any other games, the social cost of an NE solution in the resource allocation games is often not minimum, whose corresponding solutions are called *optimal*. In this paper, we use the commonly accepted notions of the *price of anarchy* (PoA) and the *price of stability* (PoS) to analyze the quality of NE solutions. As introduced by Papadimitriou in [13], the PoA (respectively, PoS) is defined as the ratio of the social cost of the worst (respectively, best) NE solution and the corresponding optimal social cost. There have been many studies such as [11, 15] and those we are to mention below that use the PoA and the PoS to study NE solutions in scheduling games. Heydenreich et al. [10] give a survey on the studies and Czumaj [5] gives a survey on studies of different congestion models, to which scheduling games also apply.

In the load balancing model, the utilitarian social cost of the total congestion caused by all job agents can be interpreted as the total time spent by all jobs. To the best of our knowledge, minimization of this social cost has not been considered in the literature. However, similar social objectives have been considered recently. Suri et al. [14] and Christodoulou and Koutsoupias [4] study the model where all jobs are identical. The social cost they consider is the total *latency* in the system, where the latency of each individual job is defined as a function of the number of jobs assigned to a machine (or a route on a network) to which the job belong. Awerbuch et al. [3] study the general case of the same model, i.e., jobs have different weights. They consider linear and polynomial latency functions and define the social cost as the sum of machine loads weighted by the latencies of the machines. Aland et al. [1] provide exact bounds on the PoAs for the social cost of the total latency with polynomial latency functions. Gairing et al. [9] consider mixed strategies for the same model and provide a bound on the PoA. Lücking et al. [12] use as social cost the sum of squared flows divided by capacities on the edges for routing games.

Fair cost sharing in network routing and design has been considered in [7, 2, 6]. For resource allocation games, Feldman and Tamir [8] recently introduce a proportional cost-sharing scheme when machine activation cost is considered. The social cost they consider is the maximum of individual costs. In this paper, we extend this study to two new social objectives as mentioned above.

Model descriptions and main results

A set of jobs $J = \{1, 2, \ldots, n\}$ is to be assigned to a number of identical parallel machines. Each job $j \in J$ has a processing time (or weight) of $p_j > 0$. For a given job assignment $\mathcal{A}$, we denote the set (respectively, number) of jobs assigned to machine i by $J^{\mathcal{A}}[i]$ (respectively, $n_i^{\mathcal{A}}$). The load of machine i under assignment $\mathcal{A}$ is then $L_i^{\mathcal{A}} = \sum\{p_j : j \in J^{\mathcal{A}}[i]\}$. We omit the indication of $\mathcal{A}$ from the notation unless there is a confusion. If the assignment $\mathcal{A}$ is optimal, we use $J^*[i]$, n_i^* and L_i^* to denote the above quantities, respectively.

In the load balancing model, we assume that there are fixed number m of machines. In this model, the cost to a job is the load of the machine the job is assigned to. The social cost of a given overall job assignment is:

$$C_1 = \sum_{i=1}^{m} n_i L_i.$$

In the proportional cost-sharing model, let B denote the cost of activating each machine. Given an overall job assignment, if job j with weight p_j is assigned to machine i of load L_i, then its cost is

$$L_i + \frac{p_j}{L_i} B,$$

where the first term is its congestion cost and the second term represents its share of the cost of activating machine i, which is in proportion to its weight with respect to the total weight of its machine. In this model, we consider two social costs:

$$\begin{aligned} C_2 &= \sum_{i=1}^{m} n_i L_i + mB, \\ C_3 &= L_{\max} + mB, \end{aligned}$$

where m is the number of activated machines in the given job assignment and $L_{\max} = \max_{1 \le i \le m} L_i$. As can be seen easily, the cost of C_2 is the total of individual costs of all jobs and C_3 is the maximum congestion cost plus the sum of total activation cost incurred in the system.

In each π of the three problems defined above ($\pi = 1, 2, 3$), we will use C_π^e and C_π^* to denote the social cost of an NE and an optimal solution, respectively. It is convenient to partition all jobs into two categories, large and small: $J_l = \{j \in J : p_j > B\}$ and $J_s = \{j \in J : p_j \le B\}$. Let $P = \sum_{j \in J} p_j$ and denote the minimum, average and maximum job weights respectively by $p_{\min} = \min_{j\in J} p_j$, $p_{\text{avg}} = P/n$ and $p_{\max} = \max_{j\in J} p_j$. Let

$$\begin{cases} \rho_1 &= p_{\text{avg}}/p_{\min}, \\ \rho_2 &= B/\max\{p_j : j \in J_s\}, \\ \rho_3 &= B/p_{\min}. \end{cases} \tag{1}$$

Clearly, $\rho_1, \rho_2, \rho_3 \geq 1$ and $\rho_2 \leq \rho_3$ whenever $J_s \neq \emptyset$. In the next three sections, we will establish the following three sets of main results.

In the load balancing model of social cost of C_1 and the proportional cost-sharing model with social costs of respective C_2 and C_3, we have the following respective sets of bounds with all the upper bounds (asymptotically) tight:

$$\begin{array}{ll} \rho_1 \leq \text{PoS} \leq \text{PoA} \leq 1 + \rho_1, & \text{(Theorem 2.1)} \\ \Omega(\sqrt[3]{\rho_3}) \leq \text{PoS} \leq \text{PoA} \leq (\rho_3 + 1)/(2\sqrt{\rho_2}), & \text{(Theorem 3.1)} \\ \begin{cases} \text{PoA} \leq \frac{1}{2}\sqrt{\frac{P}{p_{\min}}} + O(1), \\ \text{PoS} \leq \sqrt{\frac{P}{B}} + O(1), \end{cases} & \text{(Theorems 4.1 and 4.2)} \end{array}$$

where in the big O notation above, we assume B is fixed and $p_{\max} = O(\sqrt{P})$ if $P \to +\infty$.

Notice that, for each of the three problems, the parametric bounds above show that the PoS and hence the PoA can be very large if the parameters are not restricted.

2 Load balancing

We start with a direct observation of a simple property of NE assignments.

Lemma 2.1. *Given any NE assignment, machine loads satisfy the following inequalities:*

$$L_i \leq L_k + p_j, \ \forall j \in J[i], 1 \leq i, k \leq m.$$

Lemma 2.1 simply means that, in any NE assignment, no job can reduce its cost by unilaterally changing its machine. Using Lemma 2.1 we next prove an upper bound on C_1^e.

Lemma 2.2. *Given any NE assignment, its total cost C_1^e satisfies the following:*

$$C_1^e = \sum_{i=1}^{m} n_i L_i \leq \left(\frac{n}{m} + 1\right) P.$$

Proof. For any fixed i ($1 \leq i \leq m$), we choose k and j in Lemma 2.1 such that

$$L_k = \min_{1\leq k'\leq m} L_{k'} \leq \frac{P}{m}; \text{ and } p_j = \min_{j'\in J[i]} p_{j'} \leq \frac{L_i}{n_i}.$$

The two inequalities above are due to the facts that

$$\sum_{i'=1}^{m} L_{i'} = P \text{ and } \sum_{j'\in J[i]} p_{j'} = L_i; |J[i]| = n_i.$$

Therefore, we obtain

$$L_i \leq \frac{P}{m} + \frac{L_i}{n_i},\ i = 1, \ldots, m,$$

which leads directly to our conclusion. $\square$

The upper bound on C_1^e in Lemma 2.2 depends on the total processing time of the jobs, the number of jobs and the number of machines. The following lemma is a direct conclusion from the convexity of function $f(x) = x^2$.

Lemma 2.3. *For any real values $x_1, \ldots, x_m$, we have*

$$\sum_{i=1}^{m} x_i^2 \geq \frac{1}{m} \left(\sum_{i=1}^{m} x_i \right)^2.$$

With Lemmas 2.2 and 2.3, we can establish our first upper bound on PoA.

Theorem 2.1. *Let ρ_1 be defined as in (1). Then*

$$\rho_1 \leq PoS \leq PoA \leq 1 + \rho_1.$$

Proof. Fix any NE and optimal assignments. From Lemma 2.2 and the fact that $P = \sum_i L_i^* \leq C_1^*$, we have

$$\frac{C_1^e}{C_1^*} \leq 1 + \frac{n}{m} \frac{P}{\sum_{i=1}^{m} n_i^* L_i^*}.$$

With the following inequality

$$L_i^* \geq n_i^*\, p_{\min},$$

and Lemma 2.3 and noticing that $\sum_{i=1}^{m} n_i^* = n$, we get

$$\frac{C_1^e}{C_1^*} \leq 1 + \frac{n}{m} \frac{P}{p_{\min} \sum_{i=1}^{m} (n_i^*)^2} \leq 1 + \rho_1.$$

The following example provides the lower bound for the PoS. $\square$

Example 2.1. Consider an instance of m machines, m large jobs of unit processing time and n small jobs of processing time $1/n$. Assume that $n > m$ and let n be a multiple of $m(m-1)$.

Consider the assignment in which all large jobs are assigned to a single machine and all small jobs are evenly assigned to the remaining machines. The social cost of this assignment is an upper bound on the optimal social cost:

$$C_1^* \leq m^2 + \frac{n}{m-1}.$$

Now consider the NE assignment $\mathcal{A}_1$ in which one large job and n/m small jobs are assigned to each machine. The social cost of the assignment is

$$C_1^e(\mathcal{A}_1) = (n+m) \frac{m+1}{m}.$$

Then we get

$$\frac{C_1^e(\mathcal{A}_1)}{C_1^*} \geq \frac{(n+m)(m+1)(m-1)}{m(m^2(m-1)+n)},$$

which approaches $m - 1/m$ as $n \to +\infty$.

Now consider another NE assignment $\mathcal{A}_2$ in which two large jobs are assigned to one of the machines, all small jobs are assigned to another machine, and one large job is assigned to each of the remaining machines. The social cost of assignment $\mathcal{A}_2$ is

$$C_1^e(\mathcal{A}_2) = (m+n+2).$$

Therefore,

$$\frac{C_1^e(\mathcal{A}_2)}{C_1^*} \geq \frac{(m+n+2)(m-1)}{(m^2(m-1)+n)}$$

which approaches $m-1$ as $n \to +\infty$.

There are no NE assignments other than the two given above, because in any NE assignment, no more than two large jobs can be assigned to a single machine, and no two machines can each receive two large jobs at the same time. It is easy to see that $C_1^e(\mathcal{A}_1) > C_1^e(\mathcal{A}_2)$ since $n > m$. In this example, we have

$$\rho_1 = \frac{(m+1)n}{m+n}$$

which approaches $(m+1)$ as $n \to \infty$. We can observe that with increased m both $C_1^e(\mathcal{A}_1)/C_1^*$ and $C_1^e(\mathcal{A}_2)/C_1^*$ are bounded below by ρ_1.

3 Cost sharing: Sum of loads

In the next two sections, we study proportional cost sharing model, in which each machine used incurs an additional activation cost of B. Recall that $J_l = \{j \in J : p_j > B\}$ and $J_s = \{j \in J : p_j \leq B\}$. Soon we shall see that the problem becomes trivial if all jobs are large: $J_s = \emptyset$. For notational convenience and without loss of generality, we will assume $B = 1$ in the remainder of the section, as we can achieve this by dividing all processing times with the activation cost B.

It is easy to observe the following property of NE assignments for large jobs.

Lemma 3.1. *Any large job will be assigned to a dedicated machine in any NE assignment.*

Similarly, the following lemma characterizes optimal assignments.

Lemma 3.2. *In any optimal assignment, if $L_i^* > 1$, then $n_i^* = 1$.*

Proof. Suppose that $L_i^* > 1$ and $n_i^* \geq 2$. Let $j \in J^*[i]$. Then moving job j from machine i to a dedicated machine will result in a new assignment with a reduced objective value C_2':

$$\begin{aligned} C_2' - C_2^* &= (1 + (n_i - 1)(L_i - p_j) + 1 + p_j) - (1 + n_i L_i) \\ &= 1 - L_i - (n_i - 2)p_j < 0, \end{aligned}$$

which contradicts the optimality of the original schedule. □

Lemma 3.2 implies that an optimal assignment assigns all large jobs to dedicated machines.

Lemmas 3.1 and 3.2 together imply that, if $J_s = \emptyset$, then any NE assignment is optimal and vice versa. Therefore, without loss of generality, we assume $J_s \neq \emptyset$ and hence our definition of ρ_2 in (1) is valid. On NE assignments of small jobs only, we have

Lemma 3.3. *For machines of small jobs, $L_i \leq 1$ holds in any NE assignment.*

Proof. Suppose that $L_i > 1$ holds for machine i. Consider job j on machine i. The cost of job j is

$$L_i + \frac{p_j}{L_i}.$$

If job j activates a new machine, its cost will be $1 + p_j$. Then the cost change is

$$\begin{aligned} \Delta &= 1 + p_j - L_i - \frac{p_j}{L_i} = (1 - L_i) + p_j \frac{L_i - 1}{L_i} \\ &= \frac{1 - L_i}{L_i}(L_i - p_j) < 0. \end{aligned}$$

□

Lemma 3.3 implies that in an NE assignment, no machines other than dedicated ones can have a load greater than 1. Given Lemmas 3.1 and 3.3, let $A = \sum_{j \in J_l} p_j + |J_l|$ and denote $n_s = |J_s|$, we have the following result:

Lemma 3.4. *Any NE assignment has a social cost C_2^e that is bounded above as follows:*

$$C_2^e \leq n_s + P_s + A.$$

Proof. Suppose that there are m activated machines in the NE assignment. According to Lemma 3.1, we assume that all small jobs are assigned to the first m_s machines. Then $L_i = 1 - \theta_i \leq 1$ for some $\theta_i \geq 0$ for $i = 1, \ldots, m_s$. We have

$$\begin{aligned} C_2^e &= \sum_{i=1}^{m_s}(n_i L_i + 1) + A = \sum_{i=1}^{m_s}((n_i - 1)L_i + 1) + P_s + A \\ &= \sum_{i=1}^{m_s}((n_i - 1)(1 - \theta_i) + 1) + P_s + A \\ &= n_s - \sum_{i=1}^{m_s}(n_i - 1)\theta_i + P_s + A \leq n_s + P_s + A. \end{aligned}$$

□

It is easy to see that the above bound is tight. On the other hand, we can provide a lower bound on C_2^*.

Lemma 3.5. *Any optimal assignment has a social cost C_2^* such that*

$$C_2^* \geq 2P_s\sqrt{\rho_2} + A.$$

Proof. Since $n_i^* \geq \rho_2 L_i^*$, we have

$$C_2^* = \sum_{i=1}^{m_s^*} n_i^* L_i^* + m_s^* + A \geq \sum_{i=1}^{m_s^*} \rho_2 L_i^{*2} + m_s^* + A.$$

Since $\sum_{i=1}^{m_s^*} L_i^* = P_s$, from Lemma 2.3 we have $\sum_{i=1}^{m_s^*} L_i^{*2} \geq P_s^2/m_s^*$. Then,

$$C_2^* \geq \rho_2 \frac{P_s^2}{m_s^*} + m_s^* + A.$$

The fact that the right-hand side of the above inequality is at its minimum when $m_s^* = P_s\sqrt{\rho_2}$ implies

$$C_2^* \geq 2P_s\sqrt{\rho_2} + A.$$

□

With the above lemma, we are now able to establish our second main result.

Theorem 3.1. *The PoS and PoA are bounded as follows:*

$$\Omega(\sqrt[3]{\rho_3}) \leq PoS \leq PoA \leq \frac{\rho_3 + 1}{2\sqrt{\rho_2}} \left(\leq \frac{\rho_3 + 1}{2} \right).$$

Furthermore, the upper bound is tight.

Proof. Given the upper bound on C_2^e and the lower bound on C_2^* in Lemmas 3.4 and 3.5, respectively, noticing that $n_s \leq \rho_3 P_s$, we obtain

$$\frac{C_2^e}{C_2^*} \leq \frac{n_s + P_s + A}{2P_s\sqrt{\rho_2} + A} \leq \frac{\rho_3 + 1}{2\sqrt{\rho_2}}.$$

Since the above inequality holds for any instance, the upper bound in the theorem is established. The tightness of the upper bound and the lower bound are shown in the following two examples, respectively. □

Example 3.1. Consider an instance of a^{2k} jobs, each having a processing time $\frac{1}{a^{2k}}$, where $a, k > 1$ are fixed integers. An NE assignment is that all jobs are on a single machine, which has the following social cost:

$$C_2^e = a^{2k} + 1.$$

It is easy to see that an optimal solution distributes all these identical jobs evenly on all activated machines, i.e., $|n_u^* - n_v^*| \leq 1$ for any two activated machines u, v. Suppose that m machines are activated in a solution and there are equal numbers of jobs on all these machines. Then the social cost of such a solution is

$$C_2 = m + \frac{a^{2k}}{m} \frac{1}{a^{2k}} a^{2k} = m + \frac{a^{2k}}{m},$$

which is minimized if $m^* = a^k$. Hence, there exists an optimal solution with a^k machines activated of the following social cost:

$$C_2^* = 2a^k.$$

Therefore, we conclude that

$$\frac{C_2^e}{C_2^*} = \frac{a^{2k}+1}{2a^k} = \frac{\rho_3+1}{2\sqrt{\rho_2}},$$

which equals the upper bound in Theorem 3.1.

Example 3.2. Consider an instance of $n+1$ jobs ($n \geq 2$) with machine activation cost $B = 1$. The job processing times are $p_j = 1/n^3$ for $j = 1, \ldots, n$ and $p_{n+1} = 1 - 1/n^2 - \alpha$ with $0 < \alpha \leq 1/n^2$.

Let $\mathcal{S}$ be the assignment in which all jobs are assigned to the same machine, say machine 1. Then $L_1 = 1 - \alpha$, giving a social cost

$$C_2(\mathcal{S}) = n_1 L_1 + 1 = (n+1)(1-\alpha) + 1.$$

Proposition 1. $\mathcal{S}$ is the unique NE assignment.

Proof. $\mathcal{S}$ is an NE assignment since $L_1 < 1$ so that no job can reduce its cost by activating a new machine. We show that $\mathcal{S}$ is the unique NE assignment.

Claim 1. In any NE assignment with at least two activated machines, job $n+1$ cannot have a dedicated machine.

Suppose that job $n+1$ is on a dedicated machine, say, machine 1. Then the cost of job $n+1$ is

$$1 - \frac{1}{n^2} - \alpha + 1,$$

and $L_1 = p_{n+1}$. On the other hand, there is another machine i of load L_i such that

$$\frac{1}{n^3} \leq L_i \leq \frac{1}{n^2}. \tag{2}$$

Then job $n+1$ would benefit by deviating from machine 1 to machine i since

$$L_i + \frac{p_{n+1}}{L_i + p_{n+1}} < 1, \tag{3}$$

which can be easily verified by noticing that the LHS of inequality (3) is (a) convex in L_i and hence maximized at the end points of interval (2), and (b) strictly increasing in p_{n+1}, which is smaller than $1 - 1/n^2 < 1 - 1/n^3$.

Claim 2. In any NE assignment with at least two activated machines, no small job (of $\{1, \ldots, n\}$) is assigned to the same machine with job $n+1$.

Suppose to the contrary that such NE assignment exists and a subset of small jobs $1, \ldots, n$ are on machine 1, which also contains job $n+1$. Consider moving a small job on machine 1 to another activated machine i. With the same argument as in Claim 1 above we can see that the moving small job will benefit from such a deviation, contradicting that the original assignment is an NE.

Claims 1 and 2 together imply that no NE assignment will activate more than one machine, which proves our proposition. □

Let $\mathcal{S}^*$ be the assignment in which machine 1 is dedicated to job $n+1$ and machine 2 accommodates all other jobs $1, ..., n$. The social cost of $\mathcal{S}^*$ is

$$C_2(\mathcal{S}^*) = \sum_{i=1,2} n_i L_i + 2 = 3 + \frac{1}{n} - \frac{1}{n^2} - \alpha.$$

Proposition 2. $\mathcal{S}^*$ is an optimal assignment.

Proof. An assignment with at least three activated machines will have a social cost more than $4 - 1/n^2 - \alpha$, which is greater than $C_2(\mathcal{S})$. On the other hand, $C_2(\mathcal{S}) > C_2(\mathcal{S}^*)$ since $n \geq 2$ and $\alpha \leq 1/n^2$. □

Consequently, the lower bound on the PoS in Theorem 3.1 is implied by Propositions 1 and 2 and the fact that

$$C_2(\mathcal{S})/C_2(\mathcal{S}^*) = \Omega(n) = \Omega(\sqrt[3]{\rho_3}), \text{ as } n \to \infty.$$

4 Cost sharing: Maximum load

In this section, we consider minimizing social cost $C_3 = L_{\max} + mB$, the sum of maximum load in the system and the total machine activation cost. We first observe a lower bound on the optimal social cost.

Lemma 4.1. *Given any instance, the optimal social cost $C_3^* \geq 2\sqrt{PB}$.*

Proof. The social cost C_3 of any assignment with m machines activated is

$$C_3 \geq \frac{P}{m} + mB.$$

The RHS is minimized when the number of machines is $m^* = \sqrt{P/B}$. Therefore, the social cost of any optimal assignment is

$$C_3^* \geq \frac{P}{\sqrt{\frac{P}{B}}} + B\sqrt{\frac{P}{B}} = 2\sqrt{PB}.$$

□

Now let us move on to considering NE assignments and establishing a relationship between the machine loads and the number of jobs in any such assignment. The following lemma says that, in an NE assignment, jobs of very small weights tend to concentrate on machines of very small loads.

Lemma 4.2. *In any NE assignment, if there are two non-empty machine loads L_{i_1} and L_{i_2} such that $L_{i_1} \leq L_{i_2} < \frac{1}{k}B$ for some integer $k \geq 1$, then there are at least k jobs on machine i_1.*

Proof. It is trivial if $k = 1$. Consider first that $k = 2$. Suppose to the contrary that $L_{i_1} = p_j$ for some job j. Then $0 < p_j \le L_{i_2} < B/2$ implies that the cost for job j will decrease if it migrates to machine i_2:

$$\frac{p_j}{L_{i_2} + p_j} B + L_{i_2} + p_j < B + p_j,$$

a contradiction. Now assume that $k \ge 3$. Let job j be the longest among all jobs on machine i_1. If there are at most $k-1$ jobs on machine i_1, then $p_j \ge \frac{1}{k-1} L_{i_1}$, which together with $L_{i_1} \le L_{i_2} < \frac{1}{k} B$ implies that

$$\frac{p_j}{L_{i_2} + p_j} B + L_{i_2} + p_j < \frac{p_j}{L_{i_1}} B + L_{i_1}.$$

The above inequality again contradicts to the fact that the current assignment is an NE. □

Corollary 4.1. All but possibly one activated machine of small jobs will have a load of at least $\sqrt{p_{\min} B}$.

Proof. The claim is trivial if $p_{\min} > B$. Assume $k = \sqrt{B/p_{\min}} \ge 1$ (for simplicity we assume k is an integer). Suppose to the contrary that we have two machines, each of a load less than $\frac{1}{k} B$. Let L be the smaller load, which contains at least k jobs according to Lemma 4.2. Hence,

$$\sqrt{p_{\min} B} > L \ge k p_{\min},$$

which is a contradiction. □

Theorem 4.1. *Let $p_{\max} = O(\sqrt{P})$ and B be a fixed parameter. Then*

$$PoA \le \frac{1}{2}\sqrt{\frac{P}{p_{\min}}} + O(1).$$

The bound is tight.

Proof. If $p_{\min} > B$, then all jobs are large and the above bound is easily derived from Lemma 4.1 and the fact that each job occupies a dedicated machine in any NE assignment according to Lemma 3.1, which is also applicable here as the objective of individual jobs is the same. Therefore, we assume in the remaining proof that $p_{\min} \le B$.

From Corollary 4.1 we conclude that the number of machines activated in the NE assignment is at most

$$\frac{P_s}{\sqrt{p_{\min} B}} + 1 + n_l \le \frac{P_s}{\sqrt{p_{\min} B}} + 1 + \frac{P_l}{B} \le \frac{P}{\sqrt{p_{\min} B}} + 1.$$

Therefore, with Lemma 4.1, we have

$$\text{PoA} \le \frac{\left(\frac{P}{\sqrt{p_{\min} B}} + 1\right) B + \max\{B, p_{\max}\}}{2\sqrt{PB}} = \frac{1}{2}\sqrt{\frac{P}{p_{\min}}} + O(1).$$

The tightness of the bound is demonstrated by the following example of $n = k^2$ jobs of length $1/k^2$ each for some large integer $k \geq 2$ and $B = 1$. It is easy to check that the assignment of k jobs to each of k machines is an NE and the optimal assignment is to activate only one machine. Therefore, the NE cost and optimal cost are $k + 1/k$ and 2, respectively, giving a ratio of $k/2 + 1/(2k) = \frac{1}{2}\sqrt{\frac{P}{p_{\min}}} + O(1)$. $\square$

The tight upper bound on the PoA in Theorem 4.1 is proportional to $\sqrt{P}$. We will establish a similar bound for the PoS. Let us start with a special NE assignment.

Lemma 4.3 ([8]). *If $P > B$ then there always exists an NE assignment in which $L_{i_1} + L_{i_2} > B$ for any pair $\{i_1, i_2\}$ of activated machines.*

Such an NE assignment as stated in the above lemma can be constructed by an algorithm based on LPT [8]. Lemma 4.3 immediately leads to the following observation.

Lemma 4.4. *There exists an NE for which $m < 2P/B$.*

Similar to Theorem 4.1, we have the following theorem for the PoS.

Theorem 4.2. *As in Theorem 4.1, let $p_{\max} = O(\sqrt{B})$ and B be a fixed parameter. Then*

$$PoS \leq \sqrt{\frac{P}{B}} + O(1).$$

The bound is tight.

Proof. Denote by C_3^e the social cost of an NE assignment such that it satisfies the condition in Lemma 4.4. Lemmas 3.1 and 3.3 together with Lemma 4.4 imply

$$C_3^e \leq \max\{p_{\max}, B\} + 2P,$$

which in turn along with Lemma 4.1 implies that

$$\frac{C_3^e}{C_3^*} \leq \frac{\max\{p_{\max}, B\} + 2P}{2\sqrt{PB}} = \sqrt{\frac{P}{B}} + O(1),$$

from which the upper bound in the theorem follows with the definition of the PoS. The following example shows the tightness of the bound. $\square$

Example 4.1. Consider an example with n jobs, each having a length of $1/2 + 1/n$. Suppose that $B = 1$. Then, the only possible NE solution is to assign each job to a different machine, which gives $C_3^e = n + 1/2 + 1/n$. As n increases, C_3^e and C_3^* approach $2P$ and $2\sqrt{P}$, respectively, and hence the PoS approaches $\sqrt{P}$.

5 Concluding remarks

A natural measure for the efficiency of NE solutions is the utilitarian social cost, i.e., the sum of individual cost. In this paper, we have considered such social cost for two models, one of which is the classical machine load balancing and the other is the proportional cost sharing. For the latter model, we have also considered an egalitarian cost $L_{\max} + mB$. We have shown that, for each of the three problems, NE solutions can be arbitrarily worse than the optimal social costs. On the other hand, we have also provided a tight parametric upper bound on the PoA in each case. Our work fills a gap in the literature in the well studied load balancing game and extends the results to the proportional cost-sharing games recently introduced.

References

[1] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. In *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science, LNCS 3884*, pages 218–229. Springer-Verlag, February 2006.

[2] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[3] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 57–66. ACM, May 2005.

[4] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 67–73. ACM, May 2005.

[5] A. Czumaj. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter Selfish Routing on the Internet, pages 517–542. CRC Press, Boca Raton, FL, USA, 2004.

[6] A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost sharing connection games. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 84–92. ACM, 2007.

[7] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21 – 41, 2001.

[8] M. Feldman and T. Tamir. Conflicting congestion effects in resource allocation games. In *Proceedings of the 4th International Workshop on Internet and Network Economics, LNCS 5385*, pages 109–117. Springer-Verlag, December 2008.

[9] M. Gairing, T. Lücking, M. Mavronicolas, B. Monien, and M. Rodea. Nash equilibria in discrete routing games with convex latency functions. *Journal of Computer and System Sciences*, 74:1199–1225, 2008.

[10] B. Heydenreich, R. Müller, and M. Uetz. Games and mechanism design in machine scheduling–an introduction. *Production and Operations Management*, 16(4):437–454, 2007.

[11] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science, LNCS 1563*, pages 404–413. Springer-Verlag, May 1999.

[12] T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. A new model for selfish routing. *Theoretical Computer Science*, 406(3):187–206, 2008.

[13] C. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 749–753. ACM Press, 2001.

[14] S. Suri, C.D. Tóth, and Y. Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 47:79–96, 2007.

[15] B. Vöcking. *Algorithmic Game Theory*, chapter Selfish Load Balancing, pages 517–542. Cambridge University Press, 2007.