

Equilibria for two parallel links: The strong price of anarchy versus the price of anarchy

Leah Epstein*

Abstract

Following recent interest in the *strong price of anarchy* (SPOA), we consider this measure, as well as the well known *price of anarchy* (POA) for the job scheduling problem on two uniformly related parallel machines (or links). The atomic players are the jobs, and the delay of a job is the completion time of the machine running it. The social goal is to minimize the maximum delay of any job. Thus the cost (or social cost) in this case is the makespan of the schedule. The selfish goal of each job is to minimize its delay, i.e., the delay of the machine that it chooses to run on.

A pure Nash equilibrium is a schedule where no job can obtain a smaller delay by selfishly moving to a different configuration (machine), while other jobs remain in their original positions. A strong equilibrium is a schedule where no (non-empty) subset of jobs exists, where all jobs in this subset can benefit from changing their configuration. We say that all jobs in a subset benefit from moving to a different machine if all of them have a strictly smaller delay as a result of moving (while the other jobs remain in their positions, and may possibly have a larger delay as a result).

The SPOA is the worst case ratio between the social cost of a (pure) *strong* equilibrium and the cost of an optimal assignment, that is, the minimum achievable social cost. The POA is a standard measure which takes into account not only strong equilibria but any (pure) equilibrium. These two measures consolidate and give the same results for some problems, whereas for other problems, the SPOA gives much more meaningful results than the POA.

We study the behavior of the SPOA versus the behavior of the POA for this scheduling problem and give tight results for both these measures. We find the exact SPOA for any possible speed ratio $s \geq 1$ of the machines, and compare it to the exact POA which we also find. We show that for a wide range of speeds ratios these two measures are very different ($1.618 < s < 2.247$), whereas for other values of s , these two measures give the exact same bound. We extend all our results for cases where a machine may have an initial load resulting from jobs that can only be assigned to this machine, and show tight bounds on the SPOA and the POA for three such variants as well.

1 Introduction

John F. Nash earned his Ph.D. degree in 1950 with a dissertation on non-cooperative games. The thesis contained the definition and properties of what would later be called the Nash equilibrium. In 1994, Nash has received the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel, which is unofficially and commonly known as the Nobel Prize in Economics. A book and a movie, both called *A beautiful mind*, were inspired by his life.

A Nash equilibrium (which was actually considered earlier by Cournot) is a kind of solution concept of a game, involving two or more players, where no player can gain anything by changing only his own strategy unilaterally. If each player has chosen a strategy and no player can benefit by changing his strategy

*Department of Mathematics, University of Haifa, 31905 Haifa, Israel. lea@math.haifa.ac.il.

while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium. If a player chooses to take one action with probability 1 then that player is playing a pure strategy, and otherwise a mixed strategy. If all players have pure strategies, the resulting equilibrium is called *pure* (see [18]).

Side by side with this late recognition by the spectators of the movie and readers of the book, computer scientists started to adopt some game theoretical concepts and terminology in their studies. A large number of studies of Nash Equilibria, for problems coming from the field of computer science, were carried out in the last few years. Koutsoupias and Papadimitriou [15] proposed to investigate the behavior of the worst case *coordination ratio*, which is the ratio between the cost of the worst Nash equilibrium and the social optimum.

Robert J. Aumann was awarded the 2005 Nobel memorial prize in economics. The prize was awarded to him for “having enhanced our understanding of conflict and cooperation through game-theory analysis”. Aumann was the first one to introduce a number of concepts in game theory. One of these concepts was a *strong equilibrium*, which is a pure Nash equilibrium, in which not only single players cannot benefit from changing their strategy (to a different pure strategy), but no non-empty subset of players can form a coalition, where a coalition means that all of them can change their strategies together, and all gain from the change (see [2, 1, 7]).

In this paper, we study pure Nash equilibria and strong equilibria for a scheduling problem on uniformly related machines. We next define the problem and the meaning of equilibria in this context.

Scheduling on uniformly related machines is a basic assignment problem. In such problems, a set of jobs $J = \{j_1, j_2, \dots, j_n\}$ is to be assigned to a set of m machines $M = \{M_1, \dots, M_m\}$, where machine M_i has a speed s_i . The size of job j_k is denoted by p_k and it is equal to its running time on a unit speed machine. Moreover, the running time of this job on a machine of speed s is $\frac{p_k}{s}$. An assignment or schedule is a function $S : J \rightarrow M$. The completion time of machine M_i , which is also called the delay of this machine, is $\sum_{k:S(j_k)=M_i} \frac{p_k}{s_k}$. The cost, or the *social cost* of a schedule is the maximum delay of any machine, i.e., the makespan.

In this paper we consider the case of two uniformly related machines. We assume (without loss of generality) that M_1 has unit speed and M_2 has speed $s \geq 1$. We consider pure Nash equilibria and strong equilibria. The delay of a job is defined to be the delay of the machine that runs it. Seeing this scheduling problem as a game, the players are the jobs who are selfishly interested in minimizing their own delays.

A schedule is a *Nash equilibrium* if there exists no job that can decrease its delay by migrating to a different machine. More precisely, consider an assignment $S : J \rightarrow \{M_1, M_2\}$. The class of schedules \mathcal{S} contains all schedules S_k that differ from S only in the assignment of j_k , that is $S_k(j_\ell) = S(j_\ell)$ for all $\ell \neq k$ and $S_k(j_k) \neq S(j_k)$, that is if $S(j_k) = M_1$ then $S_k(j_k) = M_2$ and otherwise $S_k(j_k) = M_1$. For cases where the number of machines is larger than 2, \mathcal{S} contains a wider class of schedules, allowing each job to move to any machine. We say that S is a (pure) Nash equilibrium if for any job j_k , the delay of j_k in S_k is no smaller than its delay in S . Pure Nash equilibria do not necessary exists for all games (as opposed to mixed Nash equilibria). It is known that for scheduling games of this type, a pure Nash equilibrium always exists [12, 8].

A schedule is a *strong equilibrium* if there exists no (non-empty) subset of jobs, such that if all jobs in this set migrate to a different machine simultaneously, this results in a smaller delay for each and every one of them. More precisely, given a schedule S , we can define a class of schedules \mathcal{S} which contains all schedules S_K , where $K \subseteq J$, $K \neq \emptyset$. For $\ell \notin K$, we have $S_K(j_\ell) = S(j_\ell)$ whereas for $\ell \in K$, we have $S_K(j_\ell) \neq S(j_\ell)$. S is a strong equilibrium if for any $K \neq \emptyset$, there exists at least of job $j_k \in K$ whose delay in S_K is no smaller than its delay in S . A strong equilibrium is always pure Nash equilibrium (by

definition). Strong equilibria do not necessarily exist. Andelman, Feldman and Mansour [1] were the first to study strong equilibria in the context of scheduling and proved that scheduling games (of a more general form) admit strong equilibria. More general studies of the classes of congestion games which admit strong equilibria were studied in [14, 22].

In general, there is a recent interest in studies that separate the effect of the lack of coordination between players from the effect of their selfishness (see e.g. [13]). A Nash equilibrium that is not a social optimum is a stable situation not only since users are selfish, but also since the type of moves they consider is unilateral moves. Strong equilibria are stable situations whose stability is only the result of selfishness, since coordination between players is possible.

We consider the following four variants of scheduling on two uniformly related machines. The first variant is the standard one where any job can run on any machine. Three other variants relate to the so called *restricted assignment* problem. In this problem, each job is associated not only with a size, but also with a list of machines it can be processed on. This means that each job can run on one of the three subsets $\{1, 2\}, \{1\}$ and $\{2\}$. Thus for the case of two machines, a job can either run on any machine, or is restricted to one of the machines. Therefore, this model is equivalent to the case where machines may have an initial load that cannot switch machines. This generalization was mentioned already in the seminal paper of Koutsoupias and Papadimitriou [15]. The two additional models are the hierarchical models (see [4]), in which every job is associated with a prefix (or suffix) of the machines. In the first hierarchical model, each job is associated with one of the sets $\{1, 2\}, \{1\}$, whereas in the second hierarchical model the sets are $\{1, 2\}, \{2\}$ (if $s = 1$, only one hierarchical model exists). We therefore consider four different variants. **1.** No machine may have an initial load. **2.** Any machine may have an initial load. **3.** Only M_2 may have initial load. **4.** Only M_1 may have initial load.

Let the initial load of machine i be e_i , and the total size of jobs assigned to machine i be h_i . The delay of a machine is defined to be the total size of jobs and initial loads on this machine, divided by its speed. Therefore, the delay of M_1 is $e_1 + h_1$ and the delay of M_2 is $\frac{e_2 + h_2}{s}$.

In our scheduling model, the *coordination ratio*, or *price of anarchy* (POA) (see [20]) is the worst case ratio between the cost of a pure Nash equilibrium and the cost (i.e., makespan) of an optimal schedule. The *strong price of anarchy* (SPOA) is defined similarly, but only strong equilibria are considered. Therefore we refer to the pure price of anarchy by POA and when we discuss the mixed price of anarchy we call it the mixed POA. Note that a pure equilibrium is a special case of mixed equilibria.

We study the *price of anarchy* (POA) and the *strong price of anarchy* (SPOA) for all these models as functions of s . We denote the POA and SPOA for the i -th variant by $\text{POA}_i(s)$ and $\text{SPOA}_i(s)$.

It is noted in a series of papers (e.g., [15, 17, 19, 6, 5]) the model we study is a simplification of problems arising in real networks, that seems appropriate for describing basic problems in networks.

Previous work. We mention several related results for similar models of scheduling. We survey the known results for the POA and SPOA and see that in some models these measures give the same results, whereas in other models the SPOA allows to obtain much more meaningful results.

The most general case is unrelated machines, where the time to run a job j_k on a machine M_i is a function of k and i . In this model the POA is unbounded [3], which holds already for a setting of two machines. Surprisingly the SPOA for this problem is bounded by the number of machines m , as shown by Fiat et al. [10], and this is tight [1]. The upper bound of 2 for two machines was already shown in [1] (an upper bound of $2m - 1$ for $m \geq 3$ was shown in that paper as well). It can be seen that in this case, separating the effect of lack of coordination from the effect of selfishness reveals a linear (in the number of machines) ratio between the cost of worst case equilibrium and the optimal cost.

Awerbuch et al. [3] focused on scheduling with restricted assignment and identical speed machines.

Each job can run on only a subset of the machines, and has a fixed running time on all machines that can run it. They show that the POA is $\Theta(\frac{\log m}{\log \log m})$ (and $\Theta(\frac{\log m}{\log \log \log m})$ for mixed strategies). Their result holds for the hierarchical machines model as well, which for m machines means that the subset of allowed machines is a prefix of the machines for every job. Levy [16] observed that the results on the (pure) POA in this case are valid for the SPOA as well.

For m identical machines, the POA is $\frac{2m}{m+1}$ which can be deduced from the results of [11] (the upper bound) and [21] (the lower bound). It was shown in [1] that the SPOA has the same value as the POA for every m . Note, however, that the mixed POA is non constant already in this case, and equals $\Theta(\frac{\log m}{\log \log m})$, where the lower bound was shown by Koutsoupias and Papadimitriou [15] and the upper bound by Chumaj and Vöcking [6]. Tight bounds of $\frac{3}{2}$ on the mixed POA for two identical machines were shown by [15].

We finish this section with the known results for scheduling on uniformly related machines, the model which we study in this paper. A number of papers studied this model [15, 17, 6, 9, 10]. It is typically assumed that there is no initial load on the machines. Chumaj and Vöcking [6] showed that the POA is $\Theta(\frac{\log m}{\log \log m})$ (and $\Theta(\frac{\log m}{\log \log \log m})$ for mixed strategies). Feldmann et al. [9] proved that the POA for $m = 2$ and $m = 3$ is $\frac{\sqrt{4m-3}+1}{2}$ which equals $\phi = \frac{\sqrt{5}+1}{2}$ for two machines and 2 for three machines. They did not investigate the POA as a function of the machine speeds. As for the mixed POA, it was shown in [15] that it is at least $1 + \frac{s}{s+1}$ for $s \leq \phi$. Recently, Fiat et al. showed that the SPOA for this model is $\Theta(\frac{\log m}{(\log \log m)^2})$.

2 Statement of Results

In this paper, we find tight bounds on all eight functions $\text{POA}_i(s)$ and $\text{SPOA}_i(s)$ (for $i = 1, 2, 3, 4$).

We define the following five functions.

$$F_A(s) = \begin{cases} 1 + \frac{s}{s+2}, & 1 \leq s \leq \sqrt{2} \approx 1.4142 \\ s, & \sqrt{2} \leq s \leq \phi = \frac{1+\sqrt{5}}{2} \approx 1.618 \\ 1 + \frac{1}{s}, & s \geq \phi, \end{cases}$$

$$F_B(s) = \begin{cases} 1 + \frac{1}{s+1}, & 1 \leq s \leq \sqrt{2} \\ s, & \sqrt{2} \leq s \leq \phi \\ 1 + \frac{1}{s}, & s \geq \phi, \end{cases}$$

$$F_C(s) = \begin{cases} 1 + \frac{s}{s+1}, & 1 \leq s \leq \phi \\ 1 + \frac{1}{s}, & s \geq \phi, \end{cases}$$

Let s_1 be the root of $s^3 - 2s^2 - s + 1 = 0$ in the interval $(2, 3)$, and let s_2 be the root of $3s^3 - 4s^2 - 3s + 2 = 0$ in the interval $(\frac{5}{3}, 2)$.

$$G_A(s) = \begin{cases} 1 + \frac{s}{s+2}, & 1 \leq s \leq \sqrt{2} \\ s, & \sqrt{2} \leq s \leq \phi \\ \frac{1}{s-1}, & \phi \leq s \leq \sqrt{3} \approx 1.732 \\ 1 + \frac{1}{s+1}, & \sqrt{3} \leq s \leq 2 \\ \frac{s^2}{2s-1}, & 2 \leq s \leq s_1 \approx 2.24698 \\ 1 + \frac{1}{s}, & s \geq s_1, \end{cases}$$

$$G_B(s) = \begin{cases} 1 + \frac{s}{s+1}, & 1 \leq s \leq \phi \\ \frac{1}{s-1}, & \phi \leq s \leq s_2 \approx 1.69152 \\ 1 + \frac{s^2}{2s^2+s-1}, & s_2 \leq s \leq s_1 \\ 1 + \frac{1}{s}, & s \geq s_1, \end{cases}$$

We prove the following theorems. A summary of the results can be found in Table 1. Graphs of the POA and SPOA functions can be found in Figures 1,2,3 and 4.

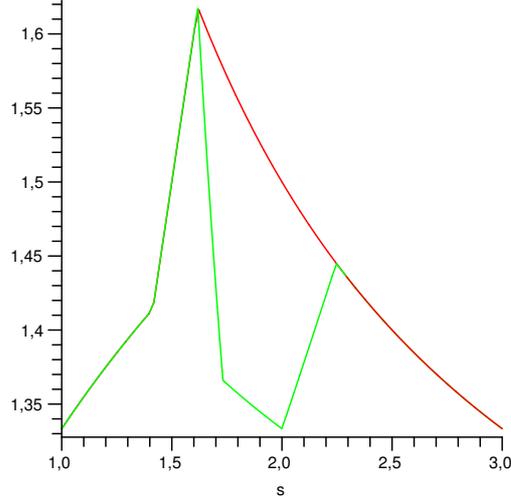


Figure 1: $POA_1(s)$ (top) and $SPOA_1(s)$ (bottom)

Theorem 1 *The price of anarchy as a function of $s \geq 1$ is exactly $F_A(s)$ if no initial load may exist on any of the machines, $F_B(s)$ if only M_2 may have an initial load, and $F_C(s)$ if M_1 may have an initial load.*

Theorem 2 *The Strong price of anarchy as a function of $s \geq 1$ is exactly $G_A(s)$ if no initial load may exist on any of the machines, $G_B(s)$ if M_1 may have an initial load but M_2 cannot have an initial load, $F_B(s)$ if M_2 may have an initial load but M_1 cannot have an initial load, and $F_C(s)$ if both M_2, M_1 may have an initial load.*

i	1	2	3	4
$POA(i)$	$F_A(s)$	$F_C(s)$	$F_B(s)$	$F_C(s)$
$SPOA(i)$	$G_A(s)$	$F_C(s)$	$F_B(s)$	$G_B(s)$

Table 1: Overview of Results

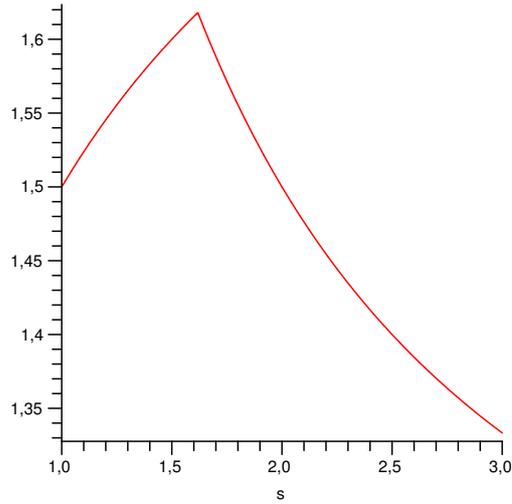


Figure 2: $POA_2(s) = SPOA_2(s)$

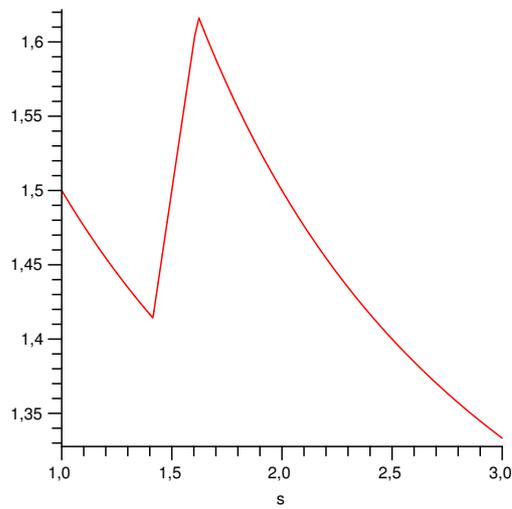


Figure 3: $POA_3(s) = SPOA_3(s)$

Note that for standard scheduling, we get that the SPOA and POA are equal for $s \leq \phi$ and for $s > s_1$. Thus for values of s that are close to 2, the two measures are different. On the other hand, for relatively small values of s and large values of s , the two measures give the same result. We encounter a similar situation for the case where M_1 (only) may have an initial load. The functions are different but the interval where $SPOA_4(s) \neq POA_4(s)$ is the same. In the other two cases we learn that the two measures give the same result. The overall bound (maximum value over all speeds) is ϕ in all the cases, which is achieved for $s = \phi$.

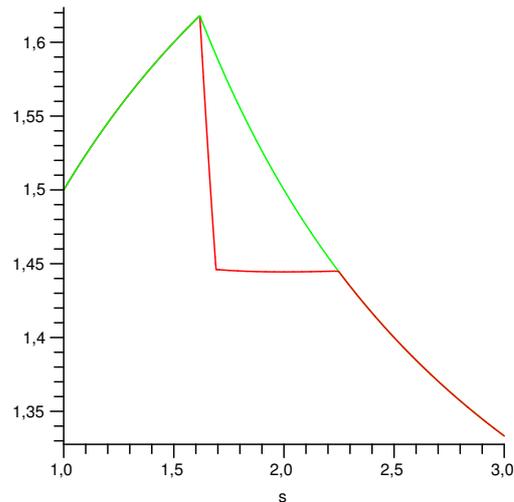


Figure 4: $POA_4(s)$ (top) and $SPOA_4(s)$ (bottom)

References

- [1] N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. In *Proc. of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, 2007. to appear.
- [2] Robert J. Aumann. Acceptable points in general cooperative n-person games. In A. W. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV, Annals of Mathematics Study 40*, pages 287–324. Princeton University Press, 1959.
- [3] B. Awerbuch, Y. Azar, Y. Richter, and D. Tsur. Tradeoffs in worst-case equilibria. *Theoretical Computer Science*, 361(2-3):200–209, 2006.
- [4] A. Bar-Noy, A. Freund, and J. Naor. On-line load balancing in a hierarchical server topology. *SIAM Journal on Computing*, 31(2):527–549, 2001.
- [5] A. Czumaj. Selfish routing on the internet. In J. Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 42. CRC Press, 2004.
- [6] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms*, 3(1), 2007.
- [7] A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost-sharing connection games. In *Proc. of the 8th ACM Conference on Electronic Commerce (EC'07)*, 2007. to appear.
- [8] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming (ICALP2003)*, pages 502–513, 2003.

- [9] R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Nashification and the coordination ratio for a selfish routing game. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming (ICALP2003)*, pages 514–526, 2003.
- [10] A. Fiat, H. Kaplan, M. Levy, and S. Olonetsky. Strong price of anarchy for machine load balancing. In *Proc. of the 34th International Colloquium on Automata, Languages and Programming (ICALP2007)*, 2007. to appear.
- [11] G. Finn and E. Horowitz. A linear time approximation algorithm for multiprocessor scheduling. *BIT Numerical Mathematics*, 19(3):312–320, 1979.
- [12] D. Fotakis, S. C. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *Proc. of the 29th International Colloquium on Automata, Languages and Programming (ICALP2002)*, pages 123–134, 2002.
- [13] D. Fotakis, S. C. Kontogiannis, and P. G. Spirakis. Atomic congestion games among coalitions. In *Proc. of the 33rd International Colloquium on Automata, Languages and Programming (ICALP2006) Part I*, pages 572–583, 2006.
- [14] R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21(1-2):85101, 1997.
- [15] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proc. of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, pages 404–413, 1999.
- [16] M. Levy. Private communication, 2007.
- [17] M. Mavronicolas and P. G. Spirakis. The price of selfish routing. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC2001)*, pages 510–519, 2001.
- [18] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- [19] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC2001)*, pages 749–753, 2001.
- [20] T. Roughgarden. *Selfish routing and the price of anarchy*. MIT Press, 2005.
- [21] P. Schuurman and T. Vredeveld. Performance guarantees of local search for multiprocessor scheduling. *Inform Journal on Computing*, 19(1):52–63, 2007.
- [22] M. Tennenholtz and O. Rozenfeld. Strong and correlated strong equilibria in monotone congestion games. In *Proc. of the 2nd International Workshop on Internet and Network Economics (WINE2006)*, page 7486, 2006.