The Price of Anarchy in Two-Stage Scheduling Games

Deshi Ye[1(✉)], Lin Chen[2], and Guochuan Zhang[1]

[1] College of Computer Science, Zhejiang University, Hangzhou 310027, China
{yedeshi,zgc}@zju.edu.cn
[2] Department of Computer Science, University of Houston, Houston, TX, USA
chenlin198662@gmail.com

Abstract. We consider a scheduling game, in which both the machines and the jobs are players. A job attempts to minimize its completion time by switching machines, while each machine would like to maximize its workload by choosing a scheduling policy from the given set of policies. We consider a two-stage game. In the first stage every machine simultaneously chooses a policy from some given set of policies, and in the second stage, every job simultaneously chooses a machine. In this work, we use the price of anarchy to measure the efficiency of such equilibria where each machine is allowed to use at most two policies. We provide nearly tight bounds for every combination of two deterministic scheduling policies with respect to two social objectives: minimizing the maximum job completion, and maximizing the minimum machine completion time.

Keywords: Price of anarchy · Scheduling · Coordination mechanisms

1 Introduction

Cloud provides an attractive platform for two entities: service providers (or machine owners) and users. Clearly it is reasonable to assume that the users are rational and are selfish. As mentioned in [1], it is now becoming common that the service providers are controlled by different agents too. A number of selfish users, each owning a job, aim to minimize the completion time of their own jobs by choosing a proper machine. On the other hand, in a market setting, the service providers get paid for running jobs, thus the service providers will attempt to attract more jobs by specifying a scheduling policy. A scheduling policy is an algorithm for the machine to schedule all the jobs that have been assigned to by the users.

Most previous games [21,22] on scheduling consider only one-side, either jobs are the players or machines are the players, but not both. To the best of our

Research was supported in part by NSFC (11671355, 11271325).

© Springer International Publishing AG 2017
X. Gao et al. (Eds.): COCOA 2017, Part II, LNCS 10628, pp. 214–225, 2017.
https://doi.org/10.1007/978-3-319-71147-8_15

knowledge, Ashlagi et al. [1] were the first ones to study the model that both the machines and the jobs are the selfish players. Actually, it is a two-stage game. In this game, a set of scheduling policies is given at the beginning. In the first stage, every machine picks a scheduling policy from the given set of policies with aims to maximize the total running time. In the second stage, every job simultaneously chooses a machine such that its own completion time is minimized. The game reaches a Nash equilibrium if no machine would like to change its policy, and no job has incentive to switch machines. Ashlagi et al. [1] proved that there always exists a pure strategy *Nash equilibrium* if the machines are restricted to use two deterministic strategies. Besides, they showed that there may not exist a pure strategy Nash equilibrium if the machines are allowed to use more than two deterministic policies.

It is worthy to note that Nash equilibrium does not always get the optimum of the social welfare function. Actually, selfish behavior might lead to highly inefficient outcome [10]. Moreover, there might exist many different equilibria. It is challenging to figure out the quality of such equilibria. The quality of an equilibrium is measured with respect to the social optimum. In this work, we consider two social objectives: minimizing the maximum completion time of the jobs (we call it the *Min-Max* model), and maximizing the minimum machine completion time (we call it the *Max-Min* model).

To measure the efficiency of such a game G with respect to a social objective, we adopt the *price of anarchy* (PoA) or the *coordination ratio* that was introduced by Koutsoupias and Papadimitriou [22]. The price of anarchy has been extensively studied in many game-theoretic models, such as job scheduling [21,25], selfish routing [27], network formation [13], facility location [29], congestion games [26], greedy combinatorial auctions [14,24].

Let $NE(G)$ be the set of Nash equilibria in the game G. The social cost of a game G is a function $C(v)$ for each $v \in NE(G)$ that numerically expresses the social objective of an outcome v of the game. The social optimum $OPT(G)$ is the optimal value in the corresponding optimization problem. The price of anarchy of a game G is the worst-case ratio over all equilibria to the social optimum. Formally, it is defined as

$$PoA(G) = \sup_{v \in NE(G)} \{\frac{C(v)}{OPT(G)}\},$$

if the social objective is a minimization function. Similarly, for a maximization objective function, we have

$$PoA(G) = \sup_{v \in NE(G)} \{\frac{OPT(G)}{C(v)}\}.$$

Related Work. To the best of our knowledge, there are not too many works on two-stage scheduling games. Besides Ashlagi et al. [1]'s work, recently Chen et al. [5] studied another two-stage scheduling game, in which each machine can reject some of their jobs (to accept more valuable jobs). Most scheduling games

focus on the scenario that the jobs are the players, where every job attempts to switch machines to minimize its own completion time, where the completion time of a job refers to the load of the machine it is assigned to. Immorlica et al. [21] proved that the price of anarchies are $2 - 2/(m+1)$, $\Theta(\log m/\log\log m)$, and unbounded for identical parallel machine scheduling, related machine scheduling, and unrelated machine scheduling, respectively, where m is the number of machines.

The price of anarchy is to measure the inefficiency of equilibrium points. In order to reduce the price of anarchy for selfish users, *coordination mechanism* was first proposed by Christodoulou et al. [6]. A coordination mechanism for a game is a set of local policies, one for each machine, such that the completion time of a job is determined by the policy of the machine that the job has been assigned to.

The scheduling policies can be Makespan (M), ShortestFirst (S), LongestFirst (L), Randomized (R) et al. (detailed definitions are given in Sect. 2). In contrast to our model, each machine in a coordination mechanism game does not change its policy during the whole game. Note that a coordination mechanism game with all machines use the policy Makespan (M) is exactly the classic scheduling game. The motivation of the makespan policy is that in some scenario all jobs in a machine will be completed at the time. Here the notation of Makespan is a bit overused. Sometimes it refers to the scheduling policy, sometimes it refers to the social objective. Anyway, we will point it out explicitly when we use it.

If the social objective is to minimize the makespan, i.e. minimizing the longest completion time of jobs, the price of anarchies in various coordination mechanism games are given as below. For identical parallel machines, the PoA of the game with the policies Makespan, ShortestFirst, LongestFirst, Randomized are $2 - 2/(m+1)$ [15,21], $2 - 1/m$ [17,21], $4/3 - 1/(3m)$ [6,18], $2 - 2/(m+1)$ [15], respectively, where m is the number of machines. For related machines, the PoA of the game with the ShortestFirst policy is $\Theta(\log m)$ [21], the PoA of the game with the LongestFirst policy is at least 1.52 and at most 1.59 [9,16], the PoAs of the game with Makespan policies and Randomized policies are $\Theta(\log m/\log\log m)$ [22]. For unrelated machines, there are a number of results, see e.g. [2,3,21]. We refer to surveys [19,25] for the study of selfish scheduling.

There were also many studies of scheduling games on the social objective of maximizing the minimum machine load. Deuermeyer et al. [8] investigated a coordination mechanism scheduling game with all machines using the policy LongestFirst. They showed the upper bound is of at most $4/3-1/3m$ for identical parallel machines. The scheduling game with policy Makespan were studied in [4, 11], it was shown that the PoA is bounded in 1.691 and 1.7. Furthermore, it was mentioned in [4,12] that the PoA was unbounded in the related machine model. Some restricted cases of the related machine scheduling game where the speed ratio is of at most two was studied in [12,23,28].

Finally, we mention that there are some works on the social objective of minimizing the (or weighted) sum of completion time. Cole et al. [7] showed that the PoA is 4 for unrelated machines. Hoeksma and Uetz [20] studied the

related machine scheduling under the SPT (shortest processing time first) rule, and presented an upper bound of 2 and a lower bound of $e/(e-1) \approx 1.58$.

Our Contribution. We concentrate on the pure strategies case where each user selects one machine to assign his job, and each machine selects one policy to schedule his jobs. In this work, the local policies for machines are limited to ShortestFirst, LongestFirst, and Makespan. As indicated in [1], there might not exist an equilibrium if machines are allowed to use more than two policies. They claimed that there exists Nash equilibrium when machines are limited to use two deterministic policies. However, this claim is inaccurate. Hence, we first show the existence of Nash equilibrium under a necessary assumption, even if the machines are restricted to use only two policies. Then, we give detailed analysis of the performance via price of anarchy. If the two policies are ShortestFirst and LongestFirst, we denote it as an (S, L)-game. Similarly, we can define the (S, M)-game and the (L, M)-game, respectively. Table 1 summarizes the results of the three games, where a single number presents a tight bound and an interval presents a lower bound and an upper bound.

Table 1. The PoAs of different games

Model	(S, L)	(S, M)	(L, M)
Min-Max $m = 2$	9/7	3/2	7/6
Min-Max $m \geq 3$	$2m/(m+1)$	$2 - 1/m$	$2m/(m+1)$
Max-Min $m \geq 2$	$[2 - 2/(m-1), 2 - 1/m]$	m	[1.691, 1.7]

In the remaining part of the paper, we first present the problem statement and settings in Sect. 2. Then we address the analysis of the price of anarchy on two social objectives in Sects. 3 and 4, respectively. We conclude the paper with open questions in Sect. 5.

2 The Game Settings

Let $\mathcal{M} = \{1, 2, \ldots, m\}$ be the set of identical machines, and $\mathcal{N} = \{1, 2, \ldots, n\}$ be the set of jobs. Both jobs and machines are selfish players. Each job j, associated with a processing time (or size) a_j, attempts to minimize his own completion time. Each machine can select a local scheduling policy to maximize the workload, which is the total processing time of all the jobs that have been assigned to that machine.

In this work, we consider three scheduling policies, namely ShortestFirst (S), LongestFirst (L), and Makespan (M). The *ShortestFirst* policy (the *LongestFirst* policy) executes jobs in the non-decreasing (non-increasing) order of processing times. The *Makespan* policy processes the jobs in a batch such that all the jobs complete at the same time, i.e. the processing time of every job is the workload

of the machine that has been assigned to. Actually, every policy determines a priority for each job that has been assigned to that machine. In ShortestFirst policy, a job with a shorter processing time has a higher priority. In LongestFirst policy, a job with a longer processing time has a higher priority. In Makespan policy, all jobs have the same priorities.

In policies mentioned above, ties are broken in favor of the job with the lowest index. This means that if two jobs have the same length then the one with the lower index has a higher priority.

The scheduling game can be described in *two stages*. At the first stage the machines publicize their own scheduling policies to all the jobs. Then in the second stage each job selects a machine such that its own completion time is minimized. A job may migrate to another machine if its completion time can be reduced. A job (or a machine) is called *satisfied* if a job (or a machine) does not have an incentive to move. Once all the jobs (or machines) get satisfied, we called it a *job equilibrium* (a machine equilibrium).

We say that the game reaches a pure Nash equilibrium if it is both in a job equilibrium and in a machine equilibrium. Ashlagi et al. [1] claimed that they have proved that there exists a pure Nash equilibrium if the machines are restricted to use any two deterministic policies. However, this claim is inaccurate. Figure 1 illustrates a scenario that the algorithm in [1] did not find an equilibrium. In details, there are two machines and four jobs with processing times 2, 3, 4, 5, respectively. Each sub-figure shows a job equilibrium. Initially, both the machines use the policy S, illustrated in sub-figure (a) of Fig. 1. There is a loop demonstrating the change of machine policies. Along the loop there is always a job equilibrium so that exactly one machine can get better by changing the current policy.

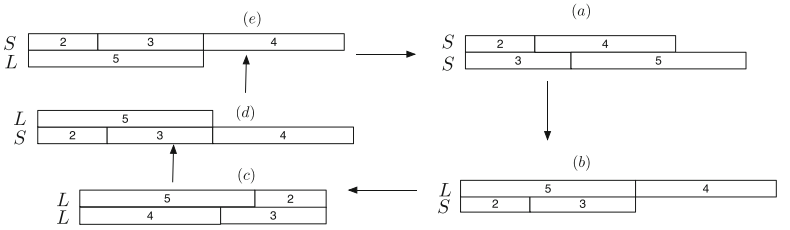

Fig. 1. An example to illustrate that there is no Nash equilibrium if each machine will change its policy once it benefits in one job equilibrium.

Note that the problem arises in above example is that given a profile of the machines, there might exist several job equilibria. To find an equlibrium for our problem, we need to make a assumption that a machine has an incentive to alter its policy if and only if its workload will strictly increase in all resulting job equilibria after the change. Violating this assumption, the existence of an equilibrium cannot be guaranteed. Under this assumption, this given example

reaches an equilibrium. As we know, sub-figures (b) and (e) of Fig. 1 are both job equilibria when one machine in sub-figure (a) changes its policy from S to L. In the sub-figure (e), the load of the machine with policy L is 5, which is less than the equilibrium in (a) with load of 6 or 8. Therefore, no machines in sub-figure (a) have incentive to change their policies under our assumption.

In the following, we will show the existence of a pure Nash equilibrium with an additional assumption, and the formal proof will be deferred to the full version of this paper.

Theorem 2.1. *In the two-stage game, there exists a pure Nash equilibrium, if machines are only allowed to use two deterministic policies, and a machine has an incentive to change its policy only if its workload will strictly increase in all resulting job equilibria after the change.*

3 The PoA in the Min-Max Model

In this section, we study the game with the social objective to minimize the makespan. The selfish action of machines may result in a high social cost, because the machines attempt to increase their workloads. The (S, M)-game, the (S, L)-game, and the (L, M)-game are considered respectively.

3.1 The (S, M)-game

In this game, the machines are only allowed to use either the ShortestFirst policy or the Makespan policy. The idea of analysis is similar as the list scheduling [17] by Graham. The detailed proof is deferred to the full version of this paper.

Theorem 3.1. *The price of anarchy of the* (S, M)*-game is* $2 - 1/m$.

3.2 The (S, L)-game

We show that the price of anarchy for the (S, L)-game is $2m/(m+1)$ for $m \geq 3$, while it becomes $9/7$ for $m = 2$.

Theorem 3.2. *The price of anarchy of the* (S, L)*-game is at most* $2m/(m+1)$, *for* $m \geq 3$.

Proof. Denote by Φ the set of machines that use the policy S, and Ψ the set of machines that use the policy L, respectively. Note that Φ can be empty (or Ψ is empty) which represents that all the machines use the same policy L (or S).

Now let us go back to the proof of this theorem. We consider the job k with processing time y, which is the last job on machine i that determines the makespan. Denote L_i to be the load of machine i. Let $x = L_i - y$ be the load of the machine i without job k. Again, denote C^{OPT} and C^{NE} to be the social cost of the optimal solution and the cost of the equilibrium, respectively.

One can easily check that the theorem follows immediately if $y \leq m/(m+1)C^{OPT}$. In the following we assume that $y > m/(m+1)C^{OPT}$ and the proof is done by contradiction. Actually, we can also assume that $y + x > 2m/(m+1)C^{OPT}$, since otherwise the theorem follows trivially. Now we have $\max\{y, x\} > m/(m+1)C^{OPT}$.

Case 1. Machine i uses the policy S, i.e., $i \in \Phi$. We know that the load of each machine in Φ other than i is not less than y. Otherwise this machine will change its policy to L and its load will be at least y, which contradicts with the fact that the current schedule is an equilibrium. We also note that $L_j \geq x$ from the point that job k does not move to machine j for any $j \neq i$. On the other hand, the load of each machine in Ψ is at least y, otherwise the job k will move that machine. Hence, we have $L_j \geq \max\{y, x\} \geq m/(m+1)C^{OPT}$ for any $j \neq i$. The total load of all machines is at least

$$\begin{aligned}\sum_{j \neq i} L_j + L_i &> ((m-1)m/(m+1) + 2m/(m+1))C^{OPT} \\ &= mC^{OPT},\end{aligned}$$

which is a not true since C^{OPT} is at least the average of total loads.

Case 2. Machine i uses the policy L, i.e., $i \in \Psi$. We get that $x \geq y \geq m/(m+1)C^{OPT}$. On the other hand, $L_j \geq x$ for any $j \neq i$ due to the equilibrium. Again, we have the total load is larger than mC^{OPT}, which is a contradiction. □

The preceding upper bound is valid for $m \geq 3$ machines, and we will provide the upper bound for two machines and lower bounds for general m machines in the full version of this paper.

Theorem 3.3. *The price of anarchy of the (S, L)-game for two machines is at most 9/7.*

Theorem 3.4. *The price of anarchy of the (S, L)-game is at least $2m/(m+1)$ for $m \geq 3$ and at least $9/7$ for $m = 2$.*

3.3 The (L, M)-game

We consider the (L, M)-game, where machines can use the policy LongestFirst or Makespan.

Theorem 3.5. *The price of anarchy of the (L, M)-game is $2m/(m+1)$ for $m \geq 3$.*

Proof. To prove the lower bound, we are given $(m-1)(m-2)$ jobs of processing times 1, $(m-1)$ jobs of processing time 2, one job of processing time $m-\epsilon$, and one job of processing time m, and the profile of machine policies is $(M, M, \ldots, M)$. All the machines use the policy M. The job configuration in Fig. 2(a) is an equilibrium. This is because all the jobs cannot reduce their completion times

if only one of them is moving. On the other hand, if one machine change to L, Fig. 2(b) is an equilibrium, in which the load of machine with policy L is m. Hence, Fig. 2(a) is an equilibrium for the (L, M)-game, which indicates that the PoA is at least $2m/(m+1)$ for $m \geq 3$.

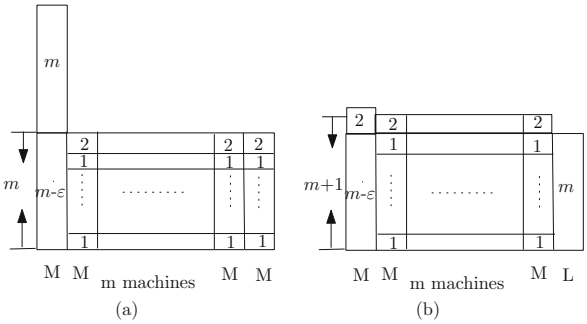

Fig. 2. job configuration in the profile (L, M)-game

The proof for the upper bound is shown as below. Again let us consider the job J' with processing time y that determines the makespan locates on the machine i. Let $x = L_i - y$ be the load on machine i without counting J'.

Each machine with policy M has load at least y, otherwise it will increase its load by changing to policy L. On the other hand, we know the load in machines with policy L is at least y. Similar as Theorem 3.2, if $y > m/(m+1)C^{OPT}$ and $x+y > 2m/(m+1)C^{OPT}$, where C^{OPT} is the optimal social solution, we will get a contradiction with total optimal load is within mC^{OPT}. Then we have either $y \leq m/(m+1)C^{OPT}$ or $x + y \leq 2m/(m+1)C^{OPT}$, which therefore the PoA is at most $2m/(m+1)$. □

From the construction of the lower bound for the (L, M)-game, we know that the result of Theorem 3.3 does not valid for $m = 2$. In the following, we obtain a new result when $m = 2$. The detailed proof will be given in the full version of this paper.

Theorem 3.6. *The price of anarchy of the (L, M)-game is 7/6 for $m = 2$.*

4 The PoA for Maximizing the Minimum Load

In this section, we study the game with the social objective of maximizing the minimum load among machines. The change of social objective does not affect the existence of equilibria. However, the price of anarchy might be very different.

4.1 The (S, L)-game

Theorem 4.1. *The lower bound of the price of anarchy of the (S, L)-game is at least $2 - 2/(m-1)$, and the upper bound is at most $2 - 1/m$, where m is the number of machines.*

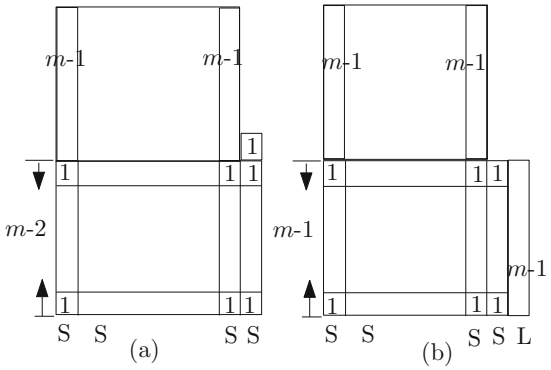

Fig. 3. Illustration of the lower bound for the (S, L)-game in the Max-Min model

Proof. The job instance is $m(m-2)+1$ jobs of processing time 1 and $m-1$ jobs of processing time $m-1$. Since all the machines use the policy S, we know that a job configuration in Fig. 3(a) is a job equilibrium. Figure 3(b) shows that no machine has incentive to change its policy to L. From this instance, $C^{NE} = m-1$ and $C^{OPT} = 2m-4$, therefore we obtain that $PoA \geq 2 - 2/(m-1)$.

The upper bound is shown as below. Denote L_j to be the load of machine j. Suppose that the machine i has the minimum load. Thus $C^{NE} = L_i$. Suppose that the machine k reaches the makespan and let $y = L_k - L_i$. Denote by J' the latest job on the machine k, whose processing time is z, then $z \geq y$, otherwise job J' would move to the machine i. Similar as Theorem 3.2, we know $L_i \geq \max\{z, L_k - z\}$. Thus, for any other machine $j \neq i$, we obtain that

$$L_j \geq L_i \geq \max\{z, L_k - z\} \geq y,$$

which implies that $y \leq L_i$. On the other hand,

$$\begin{aligned} mC^{OPT} &\leq mL_i + \sum_{j \neq i}(L_j - L_i) \\ &\leq mL_i + (m-1)y \\ &\leq (2m-1)L_i = (2m-1)C^{NE}, \end{aligned}$$

which gives $PoA = C^{OPT}/C^{NE} \leq 2 - 1/m$. □

4.2 The (S, M)-game

Lemma 4.2. *The price of anarchy of the (S, M)-game in the Max-Min model is at least m.*

Proof. There are total m jobs of processing time 1 and $m-1$ jobs of processing times m. All the machines use the policy S. From Fig. 4, no machine has incentive to change its policy to M. In this equilibrium, $C^{NE} = 1$, while $C^{OPT} = m$. □

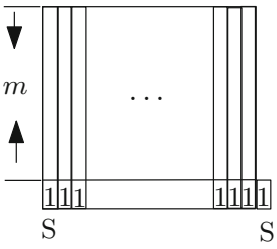

Fig. 4. Illustration of the lower bound for the (S, M)-game in the Max-min model

Theorem 4.3. *The price of anarchy of the (S, M)-game in the Max-Min model is at most m.*

Proof. Let j be the machine with the minimum load, and suppose its load is 1. Hence $C^{NE} = 1$. Denote y_k to be the size of the latest job in the machine k. Hence the load in machine $k \neq j$ is at most $1 + y_k$, otherwise that job with size y_k will switch to the machine j.

Suppose that we keep the last job of all that $m-1$ machines other than j, and move the other jobs to the machine j. In this case, machine j has load at most of m, and the other machines contain a single job. It is easy to check that the load of machine j is an upper bound of the optimal solution. As a consequence, $C^{OPT} \leq m$. □

4.3 The (L, M)-game

If all the machines are only allowed to use policy L, Deuermeyer et al. [8] showed the upper bound is at most of $4/3 - 1/3m$. Epstein et al. [11] proved that if all the machines use the policy M, the price of anarchy is bounded by 1.691 and 1.7. In this section, we prove that the price of anarchy of the (L, M)-game is also between 1.691 and 1.7. The technique for the upper bound proof is based on Epstein et al. [11], some analysis in that paper will be adopted as a black-box. The detailed proofs will be given the full version of this paper.

Theorem 4.4. *The lower bound of the price of anarchy for the (L, M)-game in the Max-Min model approaches to 1.691 when m is sufficiently large.*

Theorem 4.5. *The upper bound for the price of anarchy of the (L, M)-game in the Max-Min model is at most* 1.7.

5 Concluding Remarks

In this paper we have addressed the price of anarchy in two-stage identical parallel machine scheduling games. In these games, both the machines and the jobs are selfish players. The PoAs were explored with respect to two social objectives, the minimum of the makespan and the maximum of the minimum machine completion time. We provided nearly tight bounds of the (S, L)-game, the (S, M)-game, and the (L, M)-game, respectively, for these two social objectives.

Many directions for future work arise from these two-stage games. It is interesting to consider different individual value functions of the agents or different social objectives. In particular, one may consider the case that each machine attempts to minimize its load because each machine might get more profit in the long run since agents prefer a machine with the lightest load. One can extend our work to other measure of efficiencies, such as strong price of anarchy or price of stability. Another direction might be the extension work on uniformly related machine scheduling, or unrelated machine scheduling.

Acknowledgment. The authors thank anonymous referees for helpful comments and suggestions to improve the presentation of this paper.

References

1. Ashlagi, I., Tennenholtz, M., Zohar, A.: Competing schedulers. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI), pp. 691–696 (2010)
2. Azar, Y., Jain, K., Mirrokni, V.: (Almost) optimal coordination mechanisms for unrelated machine scheduling. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 323–332 (2008)
3. Caragiannis, I.: Efficient coordination mechanisms for unrelated machine scheduling. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 815–824 (2009)
4. Chen, X., Epstein, L., Kleiman, E., van Stee, R.: Maximizing the minimum load: the cost of selfishness. Theoret. Comput. Sci. **482**, 9–19 (2013)
5. Chen, X., Hu, X., Ma, W., Wang, C.: Efficiency of dual equilibria in selfish task allocation to selfish machines. In: Proceedings of the 6th International Conference on Combinatorial Optimization and Applications (COCOA), pp. 312–323 (2012)
6. Christodoulou, G., Koutsoupias, E., Nanavati, A.: Coordination mechanisms. In: Proceedings of the 31st International Colloquium on Automata, Languages, and Programming (ICALP), pp. 45–56 (2004)
7. Cole, R., Correa, J.R., Gkatzelis, V., Mirrokni, V., Olver, N.: Inner product spaces for minsum coordination mechanisms. In: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC), pp. 539–548 (2011)
8. Deuermeyer, B.L., Friesen, D.K., Langston, M.A.: Scheduling to maximize the minimum processor finish time in a multiprocessor system. SIAM J. Algebraic Discrete Meth. **3**(2), 190–196 (1982)
9. Dobson, G.: Scheduling independent tasks on uniform processors. SIAM J. Comput. **13**, 705–716 (1984)
10. Dubey, P.: Inefficiency of Nash equilibria. Math. Oper. Res. **11**(1), 1–8 (1986)

11. Epstein, L., Kleiman, E., Stee, R.: Maximizing the minimum load: the cost of selfishness. In: Proceedings of the 5th International Workshop on Internet and Network Economics (WINE), pp. 232–243 (2009)
12. Epstein, L., Kleiman, E., Stee, R.: The cost of selfishness for maximizing the minimum load on uniformly related machines. J. Comb. Optim. **27**(4), 767–777 (2014)
13. Fabrikant, A., Luthra, A., Maneva, E., Papadimitriou, C., Shenker, S.: On a network creation game. In: ACM Symposium on Principles of Distributed Computing (PODC), pp. 347–351 (2003)
14. Feldman, M., Immorlica, N., Lucier, B., Roughgarden, T., Syrgkanis, V.: The price of anarchy in large games. In: Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC), pp. 963–976 (2016)
15. Finn, G., Horowitz, E.: A linear time approximation algorithm for multiprocessor scheduling. BIT Numer. Math. **19**(3), 312–320 (1979)
16. Friesen, D.K.: Tighter bounds for LPT scheduling on uniform processors. SIAM J. Comput. **16**, 554–560 (1987)
17. Graham, R.L.: Bounds for certain multiprocessing anomalies. Bell Syst. Tech. J. **45**, 1563–1581 (1966)
18. Graham, R.L.: Bounds on multiprocessing timing anomalies. SIAM J. Appl. Math. **17**, 263–269 (1969)
19. Heydenreich, B., Müller, R., Uetz, M.: Games and mechanism design in machine scheduling-an introduction. Prod. Oper. Manag. **16**(4), 437–454 (2007)
20. Hoeksma, R., Uetz, M.: The price of anarchy for minsum related machine scheduling. In: Proceedings of the 9th International conference on Approximation and Online Algorithms (WAOA), pp. 261–273 (2011)
21. Immorlica, N., Li, L.E., Mirrokni, V.S., Schulz, A.S.: Coordination mechanisms for selfish scheduling. Theoret. Comput. Sci. **410**, 1589–1598 (2009)
22. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 404–413. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49116-3_38
23. Lin, L., Tan, Z.: Inefficiency of Nash equilibrium for scheduling games with constrained jobs: a parametric analysis. Theoret. Comput. Sci. **521**, 123–134 (2014)
24. Lucier, B., Borodin, A.: Price of anarchy for greedy auctions. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 537–553 (2010)
25. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
26. Roughgarden, T., Tardos, E.: Bounding the inefficiency of equilibria in nonatomic congestion games. Games Econ. Behav. **47**(2), 389–403 (2004)
27. Roughgarden, T., Tardos, E.: How bad is selfish routing? J. ACM **49**(2), 236–259 (2002)
28. Tan, Z., Wan, L., Zhang, Q., Ren, W.: Inefficiency of equilibria for the machine covering game on uniform machines. Acta Informatica **49**(6), 361–379 (2012)
29. Vetta, A.R.: Nash equilibria in competitive societies with applications to facility location, traffic routing and auctions. In: Symposium on the Foundations of Computer Science (FOCS), pp. 416–425 (2002)