

# Nonpreemptive Coordination Mechanisms for Identical Machines

Konstantinos Kollias

Published online: 3 October 2012  
© Springer Science+Business Media New York 2012

**Abstract** We focus on the problem of scheduling  $n$  weighted selfish tasks on  $m$  identical parallel machines and we study the performance of nonpreemptive coordination mechanisms. A nonpreemptive coordination mechanism consists of  $m$  local scheduling policies that decide the processing order of the tasks on each machine without delays or interruptions. We discuss the existence of Nash equilibria for this setting and show that it is not a guaranteed property of all nonpreemptive coordination mechanisms. Next, we focus on the wider class of randomized Nash equilibria and prove lower bounds on the price of anarchy. Our lower bounds are presented in comparison to the currently best known coordination mechanism (which uses delays) and lead to the conclusion that preemption or delays are required in order to improve on the currently best known solution.

**Keywords** Coordination mechanisms · Selfish scheduling · Price of anarchy

## 1 Introduction

The advent of networking had a huge impact on the landscape of computer science and gave rise to environments where several self-interested entities share computational and communication resources. The rapid growth of the Internet has drawn attention to such settings and tools from game theory have been widely applied to model the interactions among selfish agents [26, 31].

---

A preliminary version of this work appeared in the *Proceedings of the 15th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2008)*, pages 197–208, Villars-sur-Ollon, June 2008.

---

K. Kollias (✉)  
Max-Planck-Institute for Informatics, Campus E1 4, Room 326, Saarbrücken 66125, Germany  
e-mail: [kkollias@mpi-inf.mpg.de](mailto:kkollias@mpi-inf.mpg.de)

In this framework, selfish entities are considered rational players who wish to minimize some defined personal cost (e.g. the time required for the processing of a task requested by the entity) and are expected to reach a solution state from which they are not willing to deviate. The most famous such concept is the *Nash equilibrium*, which is an outcome such that no player benefits by switching to a different choice unilaterally. The performance of the whole system is measured by a social cost (e.g. the maximum or the average of the players' costs) and, in general, Nash equilibria are not optimal from this perspective. This inefficiency is captured by the *price of anarchy* [18, 27, 30]. More specifically, the price of anarchy measures the worst case ratio of the social cost in a Nash equilibrium to the optimal social cost.

A natural question that arises is whether and how this inefficiency might be contained. In distributed settings, such as those encountered in networked environments, imposing a centralized control upon the users and forcing them to coordinate and produce a socially desirable outcome is not an available alternative. Instead, the designer is restricted to defining system-wide rules and protocols a priori. This idea is traditional in game theory and is known as *mechanism design* [24, 25]. In this framework various approaches have been proposed such as applying economic incentives [4, 7, 11] or the Stackelberg strategy [3, 16, 29, 33]. One fundamental characteristic of these approaches is that global knowledge is required and the communication costs involved are heavy. In such settings where communication is impossible or the communication costs involved are too high, *coordination mechanisms* [6, 17] remedy the situation. Coordination mechanisms define local rules that do not require any information other than the state on a specific shared resource of the system.

In the described framework consider a selfish task scheduling game (in literature one may also encounter the names selfish task allocation, selfish load balancing, and selfish routing on parallel links). The game consists of  $n$  selfish tasks expecting to be processed on a system of  $m$  machines. The objective of each task is to minimize its completion time by selecting, as a strategy, the machine that will process it. The coordination mechanism for this game defines the scheduling policies of the  $m$  machines. The scheduling policies are strictly local and have no knowledge about the state of other machines. We are interested in the performance of the system which is measured by the total makespan, i.e. the completion time of all tasks. Tasks may select a probability distribution among the available machines as a strategy and, in this case, the social cost will be the expected makespan. The price of anarchy of a coordination mechanism is, among all possible sets of selfish tasks, the worst case ratio of the expected makespan in a Nash equilibrium to the optimal makespan.

In this paper we will study selfish scheduling games where all tasks are allowed to select any machine, all tasks have complete information about the weights of the other tasks and about the scheduling policies, all machines have identical processing speed and all machines have nonpreemptive policies. A policy is nonpreemptive if no task is delayed or interrupted while it is being processed.

## 1.1 Related Work

The price of anarchy and the existence of Nash equilibria for selfish scheduling on identical machines have received significant attention. The study of selfish scheduling

was initiated in [18] by Koutsoupias and Papadimitriou. The model there is equivalent to a specific coordination mechanism, the one where every machine completes all tasks simultaneously at time equal to the sum of their processing times. This coordination mechanism, which we will call the KP mechanism has been studied extensively. For  $m = 2$ , the authors in [18] proved that the price of anarchy of the KP mechanism is  $\frac{3}{2}$  and for general  $m$ , they showed that it is  $\Omega(\frac{\log m}{\log \log m})$ . In [22], Mavronicolas and Spirakis managed to prove a tight upper bound for the KP mechanism regarding the fully mixed case, where all tasks play all machines with positive probability. The price of anarchy of the KP mechanism for general  $m$  was settled independently in [10] and [19] and was shown to be  $\Theta(\frac{\log m}{\log \log m})$ . In [12], Fotakis et al. show that the KP mechanism admits at least one pure strategy Nash equilibrium for any set of tasks. In [22], Mavronicolas and Spirakis show that the randomized Nash equilibrium, in which all tasks assign positive probability to all machines, is unique.

Subsequently attention was driven to coordination mechanisms other than the KP mechanism. In [6], Christodoulou et al. propose and study several coordination mechanisms for machines with identical processing speed. For 2 machines they propose the *Largest First–Shortest First* coordination mechanism. The policies of the mechanism are as follows. The first machine schedules the tasks it receives in decreasing weight order, while the second machine schedules the tasks in increasing weight order. The price of anarchy of this mechanism is  $\frac{4}{3}$ . For  $m \geq 2$ , Christodoulou et al. propose the *Largest First* coordination mechanism, where all machines schedule the tasks in decreasing weight order and each machine  $j$  imposes a delay  $j\epsilon$ , with  $\epsilon$  infinitesimal. The price of anarchy of *Largest First* is  $\frac{4}{3} - \frac{1}{3m}$  and is the best known among coordination mechanisms for selfish scheduling on machines with identical speed. Note that this mechanism does not fall in the nonpreemptive category since it uses delays. The mechanisms described above have a unique pure strategy Nash equilibrium for every set of tasks. The *Largest First–Shortest First* mechanism may also admit randomized Nash equilibria with social cost equal to that of the unique pure strategy Nash equilibrium.

We conclude this subsection by providing pointers to earlier work in the area of selfish scheduling, in settings other than that of machines with identical speed. Several such extensions of selfish scheduling have been studied, such as the case of machines with different processing speed [10, 18] and the restricted assignment model, where tasks are restricted to subsets of the machines [2]. In [15], Immorlica et al. study several coordination mechanisms, focusing on pure strategy Nash equilibria. Truthful coordination mechanisms, i.e., mechanisms such that no task benefits from declaring a false weight, are considered in [1, 6]. More general cases are studied in [5, 9, 20, 21] and different social costs, such as the average task cost, are studied in [5, 13, 14, 34]. Results in the area of selfish scheduling are surveyed in [8, 32].

## 1.2 Our Contributions

We focus on coordination mechanisms whose local scheduling policies simply decide the order in which the received tasks will be processed, i.e., nonpreemptive coordination mechanisms. We study the existence of Nash equilibria in games defined by a nonpreemptive coordination mechanism and a set of tasks. We prove that there exist

identical machine scheduling games with nonpreemptive coordination mechanisms that do not accept any pure strategy Nash equilibrium, so naturally we turn our discussion to the broader class of randomized Nash equilibria.

We compare the performance of nonpreemptive coordination mechanisms, under the price of anarchy metric, with the  $\frac{4}{3} - \frac{1}{3m}$  price of anarchy achieved by *Largest First*, which is the best known coordination mechanism. We prove constant lower bounds  $\frac{7}{6}$  for  $m = 2$  and  $\frac{3}{2}$  for  $m > 2$  and, this way, establish that we can't produce a nonpreemptive coordination mechanism that outperforms the *Largest First* mechanism. We also extend the lower bound on the price of anarchy of every nonpreemptive coordination mechanism to  $\Omega\left(\frac{\log \log m}{\log \log \log m}\right)$ .

Combining our lower bounds with upper bounds from earlier work, we derive that the price of anarchy of the best nonpreemptive coordination mechanism for  $m = 2$  is between  $\frac{4}{3}$  and  $\frac{7}{6}$ . For general  $m$  the price of anarchy of the best nonpreemptive coordination mechanism is  $O\left(\frac{\log m}{\log \log m}\right)$  and  $\Omega\left(\frac{\log \log m}{\log \log \log m}\right)$ .

## 2 The Model

A game of scheduling  $n$  weighted tasks on  $m$  parallel machines with identical speed is fully described by  $(c, w)$ , where  $c$  and  $w$  are as follows:

1.  $c$  is a coordination mechanism, i.e., a set of  $m$  local scheduling policies. The scheduling policy of each machine  $j$  observes the weights of the tasks allocated to  $j$  and decides the completion time of each task.
2.  $w = \{w_1, w_2, \dots, w_n\}$  is a vector of  $n$  task weights, with  $w_i$  being the weight of task  $i$ . We will assume that the tasks will be given in increasing weight order and that tasks with the same weight can be ordered from smallest to largest in a predefined manner.

The set of pure strategies of every task is the set of machines  $\{1, 2, \dots, m\}$ . A randomized strategy is naturally a probability distribution over the set  $\{1, 2, \dots, m\}$ . If  $p_{i,j}$  is the probability that task  $i$  assigns to machine  $j$ , then the  $n \times m$  array  $p = \{p_{i,j}\}$  describes an outcome of the game. For the game  $(c, w)$  and outcome  $p$ , we denote as  $t_{i,j}(c, w, p_{-i})$  the expected completion time of task  $i$  if all other tasks play their strategies as in  $p$  and  $i$  plays purely machine  $j$ . Outcome  $p$  is a Nash equilibrium in  $(c, w)$  if and only if for all  $i = 1, \dots, n$  and for all  $j = 1, \dots, m$ :

$$p_{i,j} > 0 \quad \Rightarrow \quad t_{i,j}(c, w, p_{-i}) \leq t_{i,k}(c, w, p_{-i}) \quad \text{for all } k = 1, \dots, m.$$

This means that every task assigns positive probability only to machines that minimize its expected completion time. Thus, no task can improve its completion time by changing its selected probability distribution in  $p$ .

The social cost of the system will be the expected maximum among the task completion times, i.e., the expected makespan. We will write  $s(c, w, p)$  for the expected makespan of outcome  $p$  in the game  $(c, w)$ . The price of anarchy of a coordination mechanism  $c$  is defined as the worst case ratio of the expected makespan in a Nash equilibrium to the optimal makespan, for all possible sets  $w$ . We will write  $\text{opt}(w, m)$

for the optimal makespan, which is given by the optimal allocation of the  $n$  tasks to the  $m$  machines, independent of the behavioral model. Summarizing we get that

$$\text{PoA}(c) = \max_w \max_{\text{Nash } p} \frac{s(c, w, p)}{\text{opt}(w, m)}$$

## 2.1 Nonpreemptive Coordination Mechanisms

In this subsection, we will define nonpreemptive coordination mechanisms and introduce some notation. A nonpreemptive coordination mechanism is a coordination mechanism that consists of  $m$  nonpreemptive policies. A nonpreemptive policy for some machine  $j$  is one that, given a set of task weights (the weights of the tasks allocated to machine  $j$ ), decides the processing order of the tasks. Such a policy is described by an infinite set  $c_j$  of task weight orderings, such that  $c_j$  includes exactly one ordering for every possible set of task weights received. The task whose weight has the  $k$ -th position in the ordering, is served  $k$ -th in order. Assume that  $c_j$  contains the ordering  $(w_{a_1}, w_{a_2}, \dots, w_{a_k})$ . The task with weight  $w_{a_i}$  has completion time  $w_{a_1} + w_{a_2} + \dots + w_{a_i}$ . Obviously,  $c_j$  can't contain any other ordering of the set  $\{w_{a_1}, w_{a_2}, \dots, w_{a_k}\}$ . Now consider the following example:

$$c_j = \{(1, 2), (5, 9, 3), (2, 4, 5), (4, 1, 2), \dots\}.$$

This policy means that if machine  $j$  receives tasks 1, 2, and 3 with weights  $w_1 = 1$ ,  $w_2 = 2$ , and  $w_3 = 4$ , then task 3 is processed first with completion time 4, task 1 is processed second with completion time 5, and task 2 is processed last with completion time 7.

If there are weights with equal value, the ordering makes them distinct by assigning subscripts from assumed smallest to assumed largest. This means that the ordering for weights 4, 4, and 3, could be  $(4_2, 3, 4_1)$ , where  $4_1$  is the weight of the task that is considered smaller among the two with weight 4. Note that the game might include more than two tasks with weight 4 but in case a machine receives any two of them it will name them  $4_1$  and  $4_2$  depending on which one is considered smaller ( $4_1$ ) and which one is considered larger ( $4_2$ ). Regarding the tie breaking rule, recall that the tasks are given in assumed increasing weight order. So if  $w_i = w_j$  and  $j > i$ , then  $w_j$  is the one considered larger by the scheduling policy. We will write, for example, that a policy includes  $(4, 3, 4)$  if it includes  $(4_1, 3, 4_2)$  or  $(4_2, 3, 4_1)$ .

We write  $c = \{c_1, c_2, \dots, c_m\}$ , with  $c_j$ ,  $j = 1, \dots, m$ , the policy of machine  $j$ , for an  $m$ -machine nonpreemptive coordination mechanism.

## 3 Equilibria and Nonpreemptive Coordination Mechanisms

Task scheduling games with coordination mechanisms can be thought of as variations of congestion games [17, 23, 28]. Congestion games consist of  $n$  players and  $m$  resources, with each player wishing to use a subset of the resources. The cost of each player on any resource is a function of the number of players using it. Thus, all players on a resource  $j$  suffer the same cost. The total cost of a player  $i$  is the

sum of the costs of the resources that  $i$  uses. In the task scheduling model, studied in this paper, we have  $n$  players,  $m$  resources and each player wishing to use exactly one resource. The difference with congestion games is that the cost of player  $i$  on resource  $j$  does not only depend on the number of players using  $j$ . It also depends on their weights. Moreover, the costs assigned to the players by resource  $j$  are not symmetrical and depend on the order in which the tasks will be processed. Thus, task scheduling games share many similarities with congestion games but they are not a subclass of them.

What is of interest to us in this section is that congestion games have been shown to possess at least one pure strategy Nash equilibrium [28]. Moreover, the coordination mechanisms for selfish scheduling that have been studied so far, also possess at least one pure strategy Nash equilibrium for every set of task weights  $w$  (see Sect. 1.1). Thus, it is natural to ask whether this is a rule for nonpreemptive coordination mechanisms. We get the following result.

**Theorem 1** *A game  $(c, w)$ , with  $c$  a nonpreemptive coordination mechanism and  $w$  a set of task weights, does not necessarily possess a pure strategy Nash equilibrium.*

*Proof* We prove this by presenting an example. Consider a game with  $n = 4$  tasks with weights  $w_1 = 1$ ,  $w_2 = 1$ ,  $w_3 = 2$ , and  $w_4 = 2$ . Note that the tasks are always given in (assumed) increasing weight order. Now suppose that the policies  $c_1, c_2$  of the coordination mechanism  $c$  are as follows.

$$c_1 = c_2 = \{(2, 1), (2_1, 2_2), (1_1, 2, 1_2), (2_1, 2_2, 1), \dots\}.$$

This game does not admit a pure strategy Nash equilibrium. Our reasoning is as follows:

1. For the case where all tasks select the same machine, it is trivial to see that there will be 3 tasks (all but the one served first) with incentive to switch to the other machine.
2. For the case where 3 tasks select one machine and 1 task the other one.
  - (a) In case the task that is alone has weight 1, the other task with weight 1 has an incentive to join it, since its completion time drops from 5 to at most 2.
  - (b) In case the task that is alone has weight 2, the task with weight 1<sub>2</sub> (task 2—remember that they are given in increasing weight order) has an incentive to join it, since its cost drops from 4 to 3.
3. For the case where each machine is assigned 2 tasks.
  - (a) In case the tasks with weight 2 go to the same machine, then the task with weight 2<sub>2</sub> (task 4) has an incentive to switch to the other machine, since its cost drops from 4 to 3.
  - (b) In case each machine receives tasks with weight 1 and 2, then the task with weight 1<sub>1</sub> (task 1) has an incentive to switch to the other machine, since its completion time drops from 3 to 1.

This exhausts all possibilities. □

The existence of at least one randomized Nash equilibrium follows directly from Nash's Theorem, since the number of players and the number of strategies are finite.

#### 4 Lower Bounds on the Price of Anarchy of Nonpreemptive Coordination Mechanisms

In this section we focus on the price of anarchy of nonpreemptive coordination mechanisms. First we examine nonpreemptive coordination mechanisms for  $m > 2$  machines. The following theorem gives an interesting preliminary result.

**Theorem 2** *The price of anarchy of every nonpreemptive coordination mechanism for  $m > 2$  machines is at least  $\frac{3}{2}$ .*

*Proof* The main idea behind the proof is that for every coordination mechanism  $c$  for  $m > 2$ , the game with  $n = m$  tasks with weight  $w_1 = w_2 = \dots = w_m = 1$  has a Nash equilibrium with expected makespan  $\frac{3}{2}$ . Here we describe how this equilibrium is reached. Players observe each policy  $c_j$  and are interested in whether it includes  $(1_1, 1_2)$  or  $(1_2, 1_1)$ . If  $(1_2, 1_1) \in c_j$  we will call  $j$  a *big-first* machine. Otherwise we will call  $j$  a *small-first* machine.

It is certain that we can find either two big-first machines, or two small-first machines, since the machines are at least 3 and the possible orderings for  $\{1_1, 1_2\}$  are 2. Let these two machines be  $j_1$  and  $j_2$ . Suppose that tasks start making their selections ignoring  $j_1$  and  $j_2$ . Assume that there are  $k_b$  big-first machines and  $k_s$  small-first machines, other than  $j_1$  and  $j_2$ . This means that  $k_b + k_s + 2 = m$ . The  $k_b$  tasks that are considered larger are allocated to the  $k_b$  big-first machines, one task per machine. The  $k_s$  tasks that are considered smaller are allocated to the  $k_s$  small-first machines, one task per machine. Since there are  $m$  tasks, there are 2 tasks  $i_1$  and  $i_2$  that have not selected their strategies. These two tasks play  $j_1$  and  $j_2$  with probability  $\frac{1}{2}$  each. Without loss of generality we may assume that  $i_1$  is the task that is processed first if it ends up in the same machine with  $i_2$ . This means that either it is considered larger and  $j_1, j_2$  are big-first or it is considered smaller and  $j_1, j_2$  are small-first.

This allocation has expected makespan  $\frac{3}{2}$  due to the possible event that  $i_1$  and  $i_2$  end up on the same machine. The optimal allocation is obviously one task per machine with makespan 1. We claim that the allocation of the previous paragraph is a Nash equilibrium, something that would prove the theorem. Our reasoning is as follows:

1. All tasks outside  $j_1$  and  $j_2$  are alone in a machine and have optimal cost.
2. Task  $i_1$  is processed first no matter what happens, so it also has optimal cost.
3. Task  $i_2$  has expected completion time  $\frac{3}{2}$  on both  $j_1$  and  $j_2$ . It will not benefit by switching to some other big-first machine, since all such machines have another task that is considered larger than  $i_2$  and it will have to wait for its processing for a total cost of 2. For a similar reason it will not benefit by switching to some other small-first machine.

So, it follows that no task can improve its cost by a unilateral move, thus, the outcome is a Nash equilibrium. This proves the theorem.  $\square$

This provides a constant lower bound that will prove useful in our discussion on the power of nonpreemptive coordination mechanisms. However this result is only preliminary and can be extended as the next theorem shows.

**Theorem 3** *The price of anarchy of every nonpreemptive coordination mechanism for  $m$  machines is  $\Omega\left(\frac{\log \log m}{\log \log \log m}\right)$ .*

*Proof* Consider a coordination mechanism  $c$  for  $m$  machines and a game with  $n = m$  tasks, such that all tasks have weight 1. The proof is similar to the one of Theorem 2. The idea is that we do not find 2 machines with identical policy for 2 tasks with weight 1, but  $k$  machines with identical policy for  $1, 2, \dots, k - 1$ , and  $k$  tasks with weight 1, for some  $k$  that depends on  $m$  as described in the next paragraph. The number of possible policies in Theorem 2 were just 2, namely  $\{(1), (1_1, 1_2), \dots\}$  and  $\{(1), (1_2, 1_1), \dots\}$ . The number of orderings for  $j$  tasks of length 1 is  $j!$ . Hence, the number of possible policies for  $1, 2, \dots, k - 1$ , and  $k$  tasks with weight is  $l_k = 1!2! \dots k!$ .

With  $m \geq l_k(k - 1) + 1$ , we are certain we can find  $k$  machines  $j_1, j_2, \dots, j_k$ , that have the same policy for up to  $k$  tasks with weight 1. Now, tasks start making their selections ignoring  $j_1, j_2, \dots, j_k$ . Again assume that there are  $k_b$  big-first machines (as defined in the proof of Theorem 2) and  $k_s$  small-first machines, other than  $j_1, j_2, \dots, j_k$ . This means that  $k_b + k_s + k = m$ . The  $k_b$  tasks that are considered larger are allocated to the  $k_b$  big-first machines, one task per machine. The  $k_s$  tasks that are considered smaller are allocated to the  $k_s$  small-first machines, one task per machine. Since there are  $m$  tasks in total, there are  $k$  tasks  $i_1, i_2, \dots, i_k$  that have not selected their strategies. These tasks play  $j_1, j_2, \dots, j_k$  with probability  $\frac{1}{k}$  each. We claim that this is a Nash equilibrium. This is verified as follows:

1. The tasks outside  $j_1, j_2, \dots, j_k$  are alone in a machine and have optimal cost.
2. Consider a task  $i$  among the ones that play  $j_1, j_2, \dots, j_k$ , with uniform probability. Assume that it is always processed last, in the purpose of obtaining an upper bound on the completion time of this task on any of the machines  $j_1, j_2, \dots, j_k$ . Then there is probability  $\frac{1}{k}$  that it will have to wait for each of the  $k - 1$  other tasks on the machine. Then its expected completion time would be  $1 + \frac{k-1}{k}$ , which is an upper bound on its actual completion time. Applying the same reasoning as the one used in the proof of Theorem 2, we get that there is no benefit for  $i$  in switching to any of the other machines, since its completion time will rise to 2.
3. Again consider a task  $i$  among the ones that play  $j_1, j_2, \dots, j_k$ , with uniform probability. All these machines have the exact same policy for up to  $k$  tasks with weight 1 and all  $k$  tasks play each of them with probability  $\frac{1}{k}$ . So everything is symmetric and task  $i$  has the same expected completion time on each of the  $k$  machines.

So this is a Nash equilibrium and we need to calculate its expected makespan. This setting, where  $k$  tasks play  $k$  machines with uniform probability, is identical to the setting where  $k$  balls are thrown to  $k$  bins at random. It is known that the expected maximum number of balls in a bin is  $\Theta\left(\frac{\log k}{\log \log k}\right)$ . This directly gives an  $\Omega\left(\frac{\log k}{\log \log k}\right)$  lower bound on the price of anarchy of  $c$ , since the optimal makespan of the game is 1. From  $m = l_k(k - 1) + 1 = 1!2! \dots k!(k - 1) + 1$ , we get

$$\log m = \Theta\left(\sum_{j=1}^k j \log j\right) = \Theta(k^2 \log k) \Rightarrow k^2 \log k = \Theta(\log m).$$



This gives  $\log k = \Theta(\log \log m)$ . Combining this with the  $\Omega\left(\frac{\log k}{\log \log k}\right)$  lower bound on the price of anarchy of  $c$ , we prove the theorem.  $\square$

Next, we focus on the case of  $m = 2$  machines. We will prove that the price of anarchy of every nonpreemptive coordination mechanism for  $m = 2$  is at least  $\frac{7}{6}$  by following an extensive case analysis. We begin with the following lemma.

**Lemma 1** *Suppose  $c = \{c_1, c_2\}$  is a nonpreemptive coordination mechanism such that both machines have the same policy for two tasks with weights  $w_1, w_2 \in \{2, 3\}$  and  $w_1 \leq w_2$ . Then  $\text{PoA}(c) \geq \frac{4}{3}$ .*

*Proof* For the described  $c$ , consider the game  $(c, \{w_1, w_2\})$ . There always exists the Nash equilibrium  $p$ , where both tasks play both machines with probability  $\frac{1}{2}$ . Noting that  $w_1 \leq w_2$ , we get that

$$s(c, w, p) = \frac{1}{2}(w_1 + w_2) + \frac{1}{2}w_2 \Rightarrow \text{PoA}(c) \geq 1 + \frac{w_1}{2w_2}.$$

For  $w_1, w_2 \in \{2, 3\}$ , this last inequality proves the lemma.  $\square$

Next, we need to examine mechanisms with the following property. One policy includes  $(2, 3)$  and the other includes  $(3, 2)$ , one policy includes  $(2_1, 2_2)$  and the other includes  $(2_2, 2_1)$ , and one policy includes  $(3_1, 3_2)$  and the other includes  $(3_2, 3_1)$ . We will refer to this as the *Mirror Property*. Without loss of generality we may assume that  $(2, 3) \in c_1$  which gives the following cases.

1.  $c_1 = \{(2, 3), (2_2, 2_1), (3_1, 3_2), \dots\}$  and  $c_2 = \{(3, 2), (2_1, 2_2), (3_2, 3_1), \dots\}$ .
2.  $c_1 = \{(2, 3), (2_1, 2_2), (3_1, 3_2), \dots\}$  and  $c_2 = \{(3, 2), (2_2, 2_1), (3_2, 3_1), \dots\}$ .
3.  $c_1 = \{(2, 3), (2_2, 2_1), (3_2, 3_1), \dots\}$  and  $c_2 = \{(3, 2), (2_1, 2_2), (3_1, 3_2), \dots\}$ .
4.  $c_1 = \{(2, 3), (2_1, 2_2), (3_2, 3_1), \dots\}$  and  $c_2 = \{(3, 2), (2_2, 2_1), (3_1, 3_2), \dots\}$ .

In the remainder of the section we will only examine mechanisms of the first form. We will prove that the price of anarchy of every one of them is at least  $\frac{7}{6}$  and we will show that the other 3 cases are symmetric. We will say that mechanisms that match the first case satisfy the *Specific Mirror Property*.

At this point we proceed as follows. We present and prove several lemmata, which lead to the conclusion that for any ordering of  $\{2_1, 2_2, 3\}$  in  $c_1$ , the price of anarchy of  $c$  is at least  $\frac{7}{6}$ . The next lemma examines the case where  $c_1$  includes  $(3, 2_2, 2_1)$  and the ones that follow examine the remaining 5 cases.

**Lemma 2** *If  $c$  satisfies the Specific Mirror Property and  $(3, 2_2, 2_1) \in c_1$  then  $\text{PoA}(c) \geq \frac{7}{6}$ .*

*Proof* We examine 3 cases for the described  $c$ .

1. Case where  $c_2$  includes (3, 2, 2). In this case the game  $(c, \{2, 2, 3, 3\})$  has the Nash equilibrium

$$p = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Confirmation that this is indeed a Nash equilibrium comes from computing all values  $t_{i,j}(c, w, p_{-i})$ . For this outcome we have

- (a)  $t_{1,1}(c, w, p_{-1}) = \frac{1}{2}7 + \frac{1}{2}2 = \frac{9}{2}$  and  $t_{1,2}(c, w, p_{-1}) \geq \frac{1}{2}5 + \frac{1}{2}5 = 5$ .
- (b)  $t_{2,1}(c, w, p_{-2}) = 5$  and  $t_{2,2}(c, w, p_{-2}) = 5$ .
- (c)  $t_{3,1}(c, w, p_{-3}) = \frac{1}{2}3 + \frac{1}{2}5 = 4$  and  $t_{3,2}(c, w, p_{-3}) \geq \frac{1}{2}6 + \frac{1}{2}3 = \frac{9}{2}$ .
- (d)  $t_{4,2}(c, w, p_{-4}) = 3$ , which is the optimal cost for task 4.

These values verify that every task assigns positive probability only to machines that minimize its expected completion time. So, this outcome is a Nash equilibrium with expected makespan 6, while the optimal makespan is 5. Thus, in this case the price of anarchy of  $c$  is at least  $\frac{6}{5}$ .

2. Case where  $c_2$  includes (2, 3, 2). In this case the game  $(c, \{2, 2, 2, 3\})$  has the Nash equilibrium

$$p = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

with expected makespan 6, while the optimal makespan is 5. So, also in this case  $\text{PoA}(c) \geq \frac{6}{5}$ .

3. Case where  $c_2$  includes (2, 2, 3). In this case the game  $(c, \{2, 2, 3\})$  has the Nash equilibrium

$$p = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

with expected makespan  $\frac{14}{3}$ , while the optimal makespan is 4. This proves that the price of anarchy of  $c$ , in this case, is at least  $\frac{7}{6}$ .

These 3 cases prove the lemma. □

**Lemma 3** *If  $c$  satisfies the Specific Mirror Property and  $(3, 2_1, 2_2) \in c_1$  then  $\text{PoA}(c) > \frac{7}{6}$ .*

*Proof* For the described  $c$ , the game  $(c, \{2, 2, 3\})$  has the Nash equilibrium

$$p = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix},$$

with expected social cost  $\frac{19}{4}$ , while the optimal is 4. Thus, we get that  $\text{PoA}(c) \geq \frac{19}{16} > \frac{7}{6}$ .  $\square$

**Lemma 4** *If  $c$  satisfies the Specific Mirror Property and  $(2, 2, 3) \in c_1$ , then  $\text{PoA}(c) > \frac{6}{5}$ .*

*Proof* Assume  $c$  is such that  $(3, 2, 3) \in c_2$  or  $(3, 3, 2) \in c_2$ . Then  $(c, \{2, 2, 3, 3\})$  has the equilibrium

$$p = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix},$$

with makespan 6, while the optimal makespan is 5. This proves that the price of anarchy of  $c$ , in this case, is at least  $\frac{6}{5}$ . So, we need to prove that the same holds when  $(2, 3, 3) \in c_2$ . Since  $c_1$  contains  $(2_1, 2_2, 3)$  or  $(2_1, 2_2, 3)$  and  $c_2$  contains  $(2, 3_1, 3_2)$  or  $(2, 3_2, 3_1)$ , there are 4 cases we need to examine. For each one of them, we will give an equilibrium of the game  $(c, \{2, 2, 3, 3\})$  with expected makespan greater than 6. This will complete the proof.

1. When  $(2_2, 2_1, 3) \in c_1$  and  $(2, 3_2, 3_1) \in c_2$  the equilibrium is

$$p = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \end{bmatrix}.$$

2. When  $(2_2, 2_1, 3) \in c_1$  and  $(2, 3_1, 3_2) \in c_2$  the equilibrium is

$$p = \begin{bmatrix} \frac{3}{7} & \frac{4}{7} \\ 1 & 0 \\ 0 & 1 \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}.$$

3. When  $(2_1, 2_2, 3) \in c_1$  and  $(2, 3_2, 3_1) \in c_2$  the equilibrium is

$$p = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ 1 & 0 \\ \frac{2}{5} & \frac{3}{5} \\ 0 & 1 \end{bmatrix}.$$

4. When  $(2_1, 2_2, 3) \in c_1$  and  $(2, 3_1, 3_2) \in c_2$  the equilibrium is

$$p = \begin{bmatrix} \frac{3}{7} & \frac{4}{7} \\ 1 & 0 \\ 0 & 1 \\ \frac{2}{5} & \frac{3}{5} \end{bmatrix}.$$

$\square$

**Lemma 5** *If  $c$  satisfies the Specific Mirror Property, with  $(2, 3, 2) \in c_1$  and  $(3, 2, 2) \in c_2$ , then  $\text{PoA}(c) \geq \frac{7}{6}$ .*

*Proof* For such  $c$ , the game  $(c, \{2, 2, 2, 3, 3\})$  always has a Nash equilibrium, in which machine 1 has load 5 (one task with weight 2 and one with weight 3) and machine 2 has load 7 (the remaining 3 tasks). Such an equilibrium has makespan 7, while the optimal is 6. We now present how to construct such an equilibrium.

1. We first decide which task with weight 3 will be allocated to machine 1 (with completion time 5). It must be selected such that it does not have incentive to switch to machine 2 in an allocation such as described above. So, we select the task that would have completion time at least 6 if it were on machine 2 together with 3 other tasks with weights 2, 2, and 3. So, this task is in a best response state. The other task with weight 3 is processed first in the allocation examined, so it is also in a best response.
2. Now we decide which task with weight 2 to allocate to machine 1. If  $(2_2, 3, 2_1) \in c_1$ , then we allocate task 3, while if  $(2_1, 3, 2_2) \in c_1$ , then we allocate task 1. We can easily confirm that they are in best response in both cases.

This completes the proof.  $\square$

**Lemma 6** *If  $c$  satisfies the Specific Mirror Property, with  $(2_1, 3, 2_2) \in c_1$  and  $(3, 2, 2) \notin c_2$ , then  $\text{PoA}(c) > \frac{7}{6}$ .*

*Proof* For the described  $c$ , the game  $(c, \{2, 2, 3\})$  has the equilibrium

$$p = \begin{bmatrix} \frac{2}{7} & \frac{5}{7} \\ \frac{3}{10} & \frac{7}{10} \\ 1 & 0 \end{bmatrix},$$

with social cost  $\frac{327}{70}$ , while the optimal is 4. This completes the proof.  $\square$

The final case that remains to be examined is when  $(2_2, 3, 2_1) \in c_1$ ,  $(3, 2_2, 2_1) \notin c_2$ , and  $(3, 2_1, 2_2) \notin c_2$ . This will be done in Lemma 9, which makes use of the results presented in Lemma 7 and Lemma 8 below.

**Lemma 7** *If  $c$  satisfies the Specific Mirror Property, with  $(2_1, 2_2, 2_3) \in c_1$ , then  $\text{PoA}(c) \geq \frac{5}{4}$ .*

*Proof* For the described  $c$ , the game  $(c, \{2, 2, 2, 2\})$  always has a Nash equilibrium with expected makespan 5, while the optimal is 4. We examine all possible cases and give a Nash equilibrium with expected makespan 5 for each one.

1. If  $(2_3, 2_1, 2_2) \in c_2$ , then the Nash equilibrium is the following.

$$p = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

2. If  $(2_3, 2_2, 2_1) \in c_2$ , then the Nash equilibrium is the following.

$$p = \begin{bmatrix} 1 & 0 \\ \frac{1}{3} & \frac{2}{3} \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}.$$

3. For any other  $c_2$ , the Nash equilibrium is the following.

$$p = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{bmatrix}.$$

This completes the proof.  $\square$

**Lemma 8** *If  $c$  satisfies the Specific Mirror Property, with  $(2_1, 2_3, 2_2) \in c_1$ , then  $\text{PoA}(c) \geq \frac{5}{4}$ .*

*Proof* For the described  $c$ , the game  $(c, \{2, 2, 2, 2\})$  always has a Nash equilibrium with expected makespan 5, while the optimal is 4. We examine all possible cases and give a Nash equilibrium with expected makespan 5 for each one.

1. If  $(2_2, 2_1, 2_3) \in c_2$ , then the Nash equilibrium is the following.

$$p = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

2. If  $(2_2, 2_3, 2_1) \in c_2$ , then the Nash equilibrium is the following.

$$p = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

3. For any other  $c_2$ , the Nash equilibrium is the following.

$$p = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

This completes the proof.  $\square$

**Lemma 9** *If  $c$  satisfies the Specific Mirror Property, with  $(2_2, 3, 2_1) \in c_1$ ,  $(3, 2_2, 2_1) \notin c_2$ , and  $(3, 2_1, 2_2) \notin c_2$ , then  $\text{PoA}(c) \geq \frac{6}{5}$ .*

*Proof* For the described  $c$ , we will examine all possible cases for the scheduling policy of  $c_2$ .

1. If  $(2_1, 3, 2_2) \in c_2$ , then the game  $(c, \{2, 2, 2, 3\})$  has the following Nash equilibrium.

$$p = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

The expected social cost in this case is 6, while the optimal would be 5.

2. If  $(2_2, 3, 2_1) \in c_2$ , then the game  $(c, \{2, 2, 2, 3\})$  has the following Nash equilibrium.

$$p = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ \frac{3}{8} & \frac{5}{8} \end{bmatrix}.$$

The expected social cost in this case is  $\frac{25}{4}$ , while the optimal would be 5.

3. If  $(2_2, 2_1, 3) \in c_2$ , then the game  $c, \{2, 2, 3, 3\}$  either

$$p = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{or} \quad p = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

is a Nash equilibrium, depending on the ordering of  $\{2, 3_1, 3_2\}$  that is included in  $c_2$ . In both cases, the expected social cost is 6, while the optimal would be 5.

4. Finally, if  $(2_1, 2_2, 3) \in c_2$ , and under the assumption that  $c_1$  does not include  $(2_1, 2_2, 2_3)$  or  $(2_1, 2_3, 2_2)$ , then the game  $(c, \{2, 2, 2, 3\})$  has the following Nash equilibrium.

$$p = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The expected social cost in this case is 6, while the optimal would be 5. Removing the assumption, we would get from Lemma 7 and Lemma 8 that the mechanism would have price of anarchy at least  $\frac{6}{5}$ .

This completes the proof. □

Now, we can see that Lemmas 2 through 9 cover all possible cases of mechanisms that satisfy the Specific Mirror Property, i.e.,  $c_1 = \{(2, 3), (2_2, 2_1), (3_1, 3_2), \dots\}$  and  $c_2 = \{(3, 2), (2_1, 2_2), (3_2, 3_1), \dots\}$ . We observe that the case analysis of Lemmas 2 through 9 can be modified to prove the remaining 3 cases of mechanisms that satisfy

the Mirror Property, mentioned earlier. If instead of  $(2_2, 2_1) \in c_1$  and  $(2_1, 2_2) \in c_2$ , we have  $(2_1, 2_2) \in c_1$  and  $(2_2, 2_1) \in c_2$ , we adjust the case analysis as follows. We reverse the “direction” of the tasks with weight 2 in all orderings of the mechanism. For example, if an ordering of a policy in our case analysis is  $(\dots, 2_3, \dots, 2_1, \dots, 2_2, \dots)$ , we turn it into  $(\dots, 2_1, \dots, 2_3, \dots, 2_2, \dots)$ , or if it is  $(\dots, 2_1, \dots, 2_2, \dots)$ , we turn it into  $(\dots, 2_2, \dots, 2_1, \dots)$ . We also change the strategies of the tasks with weight 2 as follows. The task which is placed last in the increasing order switches strategies with the task which is placed first, the task which is placed second to last switches with the task which is placed second and so on. This produces a case analysis that is symmetric to the one we followed previously and proves the same result. The same idea is applied if instead of  $(3_1, 3_2) \in c_1$  and  $(3_2, 3_1) \in c_2$ , we have  $(3_2, 3_1) \in c_1$  and  $(3_1, 3_2) \in c_2$ . This proves that all mechanisms which satisfy the Mirror Property have price of anarchy at least  $\frac{7}{6}$ . Combining this with Lemma 1, we get the following theorem.

**Theorem 4** *The price of anarchy of every nonpreemptive coordination mechanism for  $m = 2$  machines is at least  $\frac{7}{6}$ .*

## 5 Discussion

The results presented in this paper, derive two key properties. The first one is that games defined by nonpreemptive coordination mechanisms, do not necessarily possess pure Nash equilibria. The second property is tied to the performance of these mechanisms, with respect to randomized Nash equilibria, regarding the price of anarchy metric. We established constant lower bounds  $\frac{7}{6}$  for 2 machines and  $\frac{3}{2}$  for 3 or more machines. These lower bounds lead to the following corollary.

**Corollary 1** *The price of anarchy of every nonpreemptive coordination mechanism is at least  $\frac{4}{3} - \frac{1}{3m}$ . As a consequence, we are not able to design a nonpreemptive coordination mechanism that performs better than the preemptive Largest First coordination mechanism of [6].*

This suggests that if there exists a coordination mechanism that has price of anarchy better than  $\frac{4}{3} - \frac{1}{3m}$ , which is the best known, it must use preemption or task delays.

Regarding the case of general  $m$ , we provide an  $\Omega\left(\frac{\log \log m}{\log \log \log m}\right)$  lower bound, on the price of anarchy of nonpreemptive coordination mechanisms, in this paper. It is trivial to design a nonpreemptive coordination mechanism that has price of anarchy  $O\left(\frac{\log m}{\log \log m}\right)$  by just setting the same scheduling policy to all machines. Then, the bound follows directly as discussed by Christodoulou et al. in [6]. Combining this with the  $\frac{4}{3}$  price of anarchy of the *Largest First–Shortest First* nonpreemptive coordination mechanism of [6], we conclude our discussion with the following corollary.

**Corollary 2** *The price of anarchy of the best nonpreemptive coordination mechanism for 2 machines is at least  $\frac{7}{6}$  and at most  $\frac{4}{3}$ . For general  $m$  it is  $\Omega\left(\frac{\log \log m}{\log \log \log m}\right)$  and  $O\left(\frac{\log m}{\log \log m}\right)$ .*

## References

1. Angel, E., Bampis, E., Pascual, F.: Truthful algorithms for scheduling selfish tasks on parallel machines. *Theor. Comput. Sci.* **369**(1–3), 157–168 (2006)
2. Awerbuch, B., Azar, Y., Richter, Y., Tsur, D.: Trade-offs in worst-case equilibria. In: *Proc. WAOA*, pp. 41–52 (2003)
3. Bagchi, A.: *Stackelberg Differential Games in Economic Models*. Lecture Notes in Control and Information Sciences, vol. 64. Springer, Berlin (1984)
4. Beckmann, M., McGuire, C.B., Winstein, C.B.: *Studies in the Economics of Transportation*. Yale University Press, New Haven (1956)
5. Caragiannis, I., Flammini, M., Kaklamanis, C., Kanellopoulos, P., Moscardelli, L.: Tight bounds for selfish and greedy load balancing. In: *Proc. ICALP*, pp. 311–322 (2006)
6. Christodoulou, G., Koutsoupias, E., Nanavati, A.: Coordination mechanisms. In: *Proc. ICALP*, pp. 345–357 (2004)
7. Cole, R., Dodis, Y., Roughgarden, T.: How much can taxes help selfish routing? In: *Proc. ACM EC*, pp. 98–107 (2003)
8. Czumaj, A.: Selfish routing on the Internet. In: *Handbook of Scheduling: Algorithms, Models, and Performance Analysis* (2004). Chapter 42
9. Czumaj, A., Krysta, P., Vocking, B.: Selfish traffic allocation for server farms. In: *Proc. STOC*, pp. 287–296 (2002)
10. Czumaj, A., Vocking, B.: Tight bounds for worst-case equilibria. In: *Proc. SODA*, pp. 413–420 (2002)
11. Fleischer, L., Jain, K., Mahdian, M.: Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In: *Proc. FOCS*, pp. 277–285 (2004)
12. Fotakis, D., Kontogiannis, S., Koutsoupias, E., Mavronicolas, M., Spirakis, P.: The structure and complexity of Nash equilibria for a selfish routing game. In: *Proc. ICALP*, pp. 123–134 (2002)
13. Gairing, M., Lucking, T., Mavronicolas, M., Monien, B.: Computing Nash equilibria for scheduling on restricted parallel links. In: *Proc. STOC*, pp. 613–622 (2004)
14. Gairing, M., Lucking, T., Mavronicolas, M., Monien, B.: The price of anarchy for polynomial social cost. In: *Proc. MFCS*, pp. 574–585 (2004)
15. Immorlica, N., Li, L., Mirokni, V.S., Schulz, A.: Coordination mechanisms for selfish scheduling. In: *Proc. WINE*, pp. 55–69 (2005)
16. Korilis, Y.A., Lazar, A.A., Orda, A.: Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Trans. Netw.* **5**(1), 161–173 (1997)
17. Koutsoupias, E.: Coordination mechanisms for congestion games. In: *Sigact News*, vol. 35, pp. 58–71 (2004)
18. Koutsoupias, E., Papadimitriou, C.H.: Worst-case equilibria. In: *Proc. STACS*, pp. 404–413 (1999)
19. Koutsoupias, E., Mavronicolas, M., Spirakis, P.: Approximate equilibria and ball fusion. *Theory Comput. Syst.* **36**(6), 683–693 (2003)
20. Libman, L., Orda, A.: The designer’s perspective to atomic noncooperative networks. *IEEE/ACM Trans. Netw.* **7**(6), 875–884 (1999)
21. Libman, L., Orda, A.: Atomic resource sharing in noncooperative networks. *Telecommun. Syst.* **17**(4), 385–409 (2001)
22. Mavronicolas, M., Spirakis, P.: The price of selfish routing. In: *Proc. STOC*, pp. 510–519 (2001)
23. Monderer, D., Shapley, L.S.: Potential games. *Games Econ. Behav.* **14**, 124–143 (1996)
24. Nisan, N.: Introduction to mechanism design (for computer scientists). In: *Algorithmic Game Theory*, pp. 209–242. Cambridge University Press, Cambridge (2007)
25. Nisan, N., Ronen, A.: Algorithmic mechanism design. *Games Econ. Behav.* **35**, 166–196 (2001)
26. Osborne, M.J., Rubinstein, A.: *A Course in Game Theory*. MIT Press, Cambridge (1994)
27. Papadimitriou, C.H.: Algorithms, games and the Internet. In: *Proc. STOC*, pp. 749–753 (1999)
28. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. *Int. J. Game Theory* **2**, 65–67 (1973)



29. Roughgarden, T.: Stackelberg scheduling strategies. In: Proc. STOC, pp. 104–113 (2001)
30. Roughgarden, T., Tardos, E.: How bad is selfish routing? *J. ACM* **49**(2), 236–259 (2002)
31. Tardos, E., Vazirani, V.: Basic solution concepts and computational issues. In: *Algorithmic Game Theory*, pp. 3–28. Cambridge University Press, Cambridge (2007)
32. Vocking, B.: Selfish load balancing. In: *Algorithmic Game Theory*, pp. 517–542. Cambridge University Press, Cambridge (2007)
33. von Stackelberg, H.: *Marktform und Gleichgewicht*. Springer, Berlin (1934). English translation entitled *The Theory of the Market Economy*
34. Suri, S., Toth, C., Zhou, Y.: Selfish load balancing and atomic congestion games. In: Proc. SPAA, pp. 188–195 (2005)