Efficient Coordination Mechanisms for Unrelated Machine Scheduling

Ioannis Caragiannis*

Abstract

We present three new coordination mechanisms for scheduling n selfish jobs on m unrelated machines. A coordination mechanism aims to mitigate the impact of selfishness of jobs on the efficiency of schedules by defining a local scheduling policy on each machine. The scheduling policies induce a game among the jobs and each job prefers to be scheduled on a machine so that its completion time is minimum given the assignments of the other jobs. We consider the maximum completion time among all jobs as the measure of the efficiency of schedules. The approximation ratio of a coordination mechanism quantifies the efficiency of pure Nash equilibria (price of anarchy) of the induced game. Our mechanisms are deterministic, local, and preemptive in the sense that the scheduling policy does not necessarily process the jobs in an uninterrupted way and may introduce some idle time. Our first coordination mechanism has approximation ratio $O(\log m)$ and always guarantees that the induced game has pure Nash equilibria to which the system converges in at most n rounds. This result improves a recent bound of $O(\log^2 m)$ due to Azar, Jain, and Mirrokni and, similarly to their mechanism, our mechanism uses a global ordering of the jobs according to their distinct IDs. Next we study the intriguing scenario where jobs are anonymous, i.e., they have no IDs. In this case, coordination mechanisms can only distinguish between jobs that have different load characteristics. Our second mechanism handles anonymous jobs and has approximation ratio $O\left(\frac{\log m}{\log\log m}\right)$ although the game induced is not a potential game and, hence, the existence of pure Nash equilibria is not guaranteed by potential function arguments. However, it provides evidence that the known lower bounds for non-preemptive coordination mechanisms could be beaten using preemptive scheduling policies. Our third coordination mechanism also handles anonymous jobs and has a nice "cost-revealing" potential function. Besides in proving the existence of equilibria, we use this potential function in order to upper-bound the price of stability of the induced game by $O(\log m)$, the price of anarchy by $O(\log^2 m)$, and the convergence time to $O(\log^2 m)$-approximate assignments by a polynomial number of best-response moves. Our third coordination mechanism is the first that handles anonymous jobs and simultaneously guarantees that the induced game is a potential game and has bounded price of anarchy.

1 Introduction

We study the classical problem of *unrelated machine scheduling*. In this problem, we have m parallel machines and n independent jobs. Job i induces a (possibly infinite) positive processing time (or load) w_{ij} when processed by machine j. The load of a machine is the total load of the jobs assigned to it. The quality of an assignment of jobs to machines is measured by the makespan (i.e., the maximum) of the machine loads or, alternatively, the maximum completion time among all jobs. The optimization problem of computing an assignment of minimum makespan is a fundamental APX-hard problem, quite well-understood in terms of its offline [31] and online approximability [4, 9].

The approach we follow in this paper is both algorithmic and game-theoretic. We assume that each job is owned by a selfish agent. This gives rise to a *selfish scheduling* setting where each agent aims to minimize the completion time of her job with no regard to the globally optimal schedule. Such a selfish behaviour can lead to inefficient schedules from which no agent has an incentive to unilaterally deviate in order to improve the completion time of her job. From the algorithmic point of view, the designer of such a system can define a *coordination mechanism* [14], i.e., a *scheduling policy* within each machine in order to "coordinate" the selfish behaviour of the jobs. Our main objective is to design coordination mechanisms that guarantee that the assignments reached by the selfish agents are *efficient*.

The model. A scheduling policy simply defines the way jobs are scheduled within a machine and can be either *non-preemptive* or *preemptive*. Non-preemptive scheduling policies process jobs uninterruptedly according to some order. Preemptive scheduling policies do not necessarily have this feature and can also introduce some idle time (delay). Although this seems unneces-

*Research Academic Computer Technology Institute & Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Greece. Email: caragian@ceid.upatras.gr.
This work was partially supported by the European Union under IST FET Integrated Project AEOLUS "Algorithmic Principles for Building Efficient Overlay Computers" and COST Action IC0602 "Algorithmic Decision Theory", and by a "Caratheodory" basic research grant from the University of Patras.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

sary at first glance, as we show in this paper, it is a very useful tool in order to guarantee coordination. A coordination mechanism is a set of scheduling policies running on the machines. In the sequel, we use the terms coordination mechanisms and scheduling policies interchangeably.

A coordination mechanism defines (or induces) a game with the job owners as players. Each job has all machines as possible *strategies*. We call an *assignment* (of jobs to machines) or *state* any set of strategies selected by the players, with one strategy per player. Given an assignment of jobs to machines, the cost of a player is the completion time of her job on the machine it has been assigned; this completion time depends on the scheduling policy on that machine and the characteristics of all jobs assigned to that machine. Assignments in which no player has an incentive to change her strategy in order to decrease her cost given the assignments of the other players are called *pure Nash equilibria*. The global objective that is used in order to assess the efficiency of assignments is the *maximum completion time* over all jobs. A related quantity is the *makespan* (i.e., the maximum of the machine loads). Notice that when preemptive scheduling policies are used, these two quantities may not be the same. However, the optimal makespan is a lower bound on the optimal maximum completion time. The *price of anarchy* [36] is the maximum over all pure Nash equilibria of the ratio of the maximum completion time among all jobs over the optimal makespan. The *price of stability* [3] is the minimum over all pure Nash equilibria of the ratio of the maximum completion time among all jobs over the optimal makespan. The *approximation ratio* of a coordination mechanism is the maximum of the price of anarchy of the induced game over all input instances.

Four natural coordination mechanisms are the Makespan, Randomized, LongestFirst, and ShortestFirst. In the Makespan policy, each machine processes the jobs assigned to it "in parallel" so that the completion time of each job is the total load of the machine. Makespan is obviously a preemptive coordination mechanism. In the Randomized policy, the jobs are scheduled non-preemptively in random order. Here, the cost of each player is the expected completion time of her job. In the ShortestFirst and LongestFirst policies, the jobs assigned to a machine are scheduled in non-decreasing and non-increasing order of their processing times, respectively. In case of ties, a *global ordering* of the jobs according to their distinct IDs is used; this is necessary by any deterministic non-preemptive coordination mechanism in order to be well-defined. Note that no such information is required by the Makespan and Randomized policies; in this case, we say that they handle *anonymous jobs*. According to the terminology of [8], all these four coordination mechanisms are *strongly local* in the sense that the only information required by each machine in order to compute a schedule are the processing times of the jobs assigned to it. A *local* coordination mechanism may use all parameters (i.e., the load vector) of the jobs assigned to the same machine.

Designing coordination mechanisms with as small approximation ratio as possible is our main concern. But there are other issues related to efficiency. The price of anarchy is meaningful only in games where pure Nash equilibria exist. So, the primary goal of the designer of a coordination mechanism should be that the induced game *always has* pure Nash equilibria. Furthermore, these equilibria should be *easy to find*. A very interesting class of games in which the existence of pure Nash equilibria is guaranteed is that of *potential* games. These games have the property that a *potential function* can be defined on the states of the game so that in any two states differing in the strategy of a single player, the difference of the values of the potential function and the difference of the cost of the player have the same sign. This property guarantees that the state with minimum potential is a pure Nash equilibrium. Furthermore, it guarantees that, starting from any state, the system will reach (converge to) a pure Nash equilibrium after a finite number of *selfish moves*. Given a game, its *Nash dynamics* is a directed graph with the states of the game as nodes and edges connecting two states differing in the strategy of a single player if that player has an incentive to change her strategy according to the direction of the edge. The Nash dynamics of potential games do not contain any cycle. Another desirable property here is *fast* convergence, i.e., convergence to a pure Nash equilibrium in a polynomial number of selfish moves. A particular type of selfish moves that have been extensively considered in the literature [6, 15, 19, 33] is that of *best-response* moves. In a best-response move, a player having an incentive to change her strategy selects the strategy that yields the maximum decrease in her cost.

Potential games are strongly related to *congestion games* introduced by Rosenthal [37]. Rosenthal presented a potential function for these games with the following property: in any two states differing in the strategy of a single player, the difference of the values of the potential function *equals* the difference of the cost of the player. Monderer and Shapley [34] have proved that each potential game having this property is isomorphic to a congestion game. We point out that potential functions are not the only way to guarantee the existence of pure Nash equilibria. Several generalizations of

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

congestion games such as those with player-specific latency functions [32] are not potential games but several subclasses of them provably have pure Nash equilibria.

Related work. The study of the price of anarchy of games began with the seminal work of Koutsoupias and Papadimitriou [29] and has played a central role in the recently emerging field of Algorithmic Game Theory [35]. Several papers provide bounds on the price of anarchy of different games of interest. Our work follows a different direction where the impact of selfishness to efficiency is the *objective to be minimized* and, in this sense, it is similar in spirit to studies where the main question is how to change the rules of the game at hand in order to improve the quality of equilibria. Typical examples are the introduction of taxes or tolls in congestion games [11, 16, 20, 26, 44], protocol design in network and cost allocation games [12, 27], Stackelberg routing strategies [25, 28, 30, 38, 44], and network design [39].

Coordination mechanisms were introduced by Christodoulou, Koutsoupias, and Nanavati in [14]. They study the case where each player has the same load on each machine and, among other results, they consider the LongestFirst and ShortestFirst scheduling policies. We note that the Makespan and Randomized scheduling policies were used in [29] as models of selfish behaviour in scheduling, and since that paper, the Makespan policy has been considered as standard in the study of selfish scheduling games in models simpler than the one of unrelated machines and is strongly related to the study of congestion games (see [45, 40] and the references therein). Immorlica et al. [23] study these four scheduling policies under several scheduling settings including the most general case of unrelated machines. They prove that the Randomized and ShortestFirst policies have approximation ratio $O(m)$ while the LongestFirst and Makespan policies have unbounded approximation ratio. Some scheduling policies are also related to earlier studies of local-search scheduling heuristics. So, the fact that the price of anarchy of the induced game may be unbounded follows by the work of Schuurman and Vredeveld [41]. As observed in [23], the equilibria of the game induced by ShortestFirst correspond to the solutions of the ShortestFirst scheduling heuristic which is known to be m-approximate [22]. The Makespan policy is known to induce potential games [17]. The ShortestFirst policy also induces potential games as proved in [23]. We have examples (that will appear in the final version of the paper) showing that the scheduling policies LongestFirst and Randomized do not induce potential games.

Azar et al. [8] study non-preemptive coordination mechanisms for unrelated machine scheduling. They prove that any local non-preemptive coordination mechanism is at least $\Omega(\log m)$-approximate while any strongly local non-preemptive coordination mechanism is at least $\Omega(m)$-approximate; as a corollary, they solve an old open problem concerning the approximation ratio of the ShortestFirst heuristic. On the positive side, the authors of [8] present a non-preemptive local coordination mechanism (henceforth called AJM-1) that is $O(\log m)$-approximate although it may induce games without pure Nash equilibria. The extra information used by this scheduling policy is the *inefficiency* of jobs (defined in the next section). They also present a technique that transforms this coordination mechanism to a preemptive one that induces potential games with price of anarchy $O(\log^2 m)$. In their mechanism, the players converge to a pure Nash equilibrium in n rounds of best-response moves. We will refer to this coordination mechanism as AJM-2. Both AJM-1 and AJM-2 use the IDs of the jobs.

Our results. We present three new coordination mechanisms for unrelated machine scheduling. Our mechanisms are deterministic, preemptive, and local. The schedules in each machine are computed as functions of the characteristics of jobs assigned to the machine, namely the load of jobs on the machine and their inefficiency. In all cases, the functions use an integer parameter $p \geq 1$ and guarantee that the maximum completion time of equilibria is close to the ℓ_p norm of the machine loads which in turn is close to the ℓ_p norm of the machine loads of optimal assignments. The best choice of the parameter p for our coordination mechanisms is $p = O(\log m)$; in this case, our bounds follow since the ℓ_p norm of the machine loads approximates the makespan within a constant factor. Hence, our mechanisms use the number of machines m as input. This is also the case for the only known coordination mechanism AJM-2 that induces games with pure Nash equilibria and price of anarchy $o(m)$.

Motivated by previous work, we first consider the scenario where jobs have distinct IDs. Our first coordination mechanism ACOORD uses this information and is superior to the known coordination mechanisms that induce games with pure Nash equilibria. The game induced is a potential game, has price of anarchy $O(\log m)$, and the players converge to pure Nash equilibria in at most n rounds. Essentially, the equilibria of the game induced by ACOORD can be thought of as the solutions produced by the application of a particular online algorithm, similar to the greedy online algorithm for minimizing the ℓ_p norm of machine loads [4, 10]. Interestingly, the local objective of the greedy online algorithm for the ℓ_p norm may not translate to a completion time of jobs in feasible schedules; the online

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

algorithm implicit by ACOORD uses a slightly different local objective that meets this constraint. The related results are presented in Section 3.

Next we address the case where no ID information is associated to the jobs (anonymous jobs). This scenario is relevant when the job owners do not wish to reveal their identities or in large-scale settings where distributing IDs to jobs is infeasible. Definitely, an advantage that could be used for coordination is lost in this way but this makes the problem of designing coordination mechanisms more challenging. In Section 4, we present our second coordination mechanism BCOORD which induces a simple congestion game with player-specific polynomial latency functions of a particular form. The price of anarchy of this game is only $O\left(\frac{\log m}{\log \log m}\right)$. This result demonstrates that preemption may be useful in order to beat the $\Omega(\log m)$ lower bound of [8] for non-preemptive coordination mechanisms. On the negative side, we can show that the game induced may not be a potential game as the Nash dynamics may contain cycles.

Our third coordination mechanism CCOORD is presented in Section 5. The scheduling policy on each machine uses an interesting function on the loads of the jobs assigned to the machine and their inefficiency. The game induced by CCOORD is a potential game; the associated potential function is "cost-revealing" in the sense that it can be used to upper-bound the cost of approximate equilibria as well as the convergence time to efficient assignments. In particular, we show that the price of stability of the induced game is $O(\log m)$, the price of anarchy is $O(\log^2 m)$, while the Nash dynamics reach $O(\log^2 m)$-approximate assignments after at most $O\left(n \log^2 m\right)$ best-response moves. The coordination mechanism CCOORD is the first that handles anonymous jobs and simultaneously guarantees that the induced game is a potential game and has bounded price of anarchy.

We begin with technical definitions in Section 2 and conclude with interesting open questions in Section 6. Due to lack of space, many proofs have been omitted. They will appear in the final version of the paper.

2 Preliminaries

In this section, we present our notation and give some statements that will be useful later. We reserve n and m for the number of jobs and machines, respectively, and the indices i and j for jobs and machines, respectively. Unless specified otherwise, the sums $\sum_i$ and $\sum_j$ run over all jobs and over all machines, respectively. Assignments are denoted by N or O. With some abuse in notation, we use N_j to denote both the set of jobs assigned to machine j and the set of their loads on machine j. We use the notation $L(N_j)$ to denote the load of machine j under the assignment N. More generally, $L(A)$ denotes the sum of the elements for any set of non-negative reals A. For an assignment N which assigns job i to machine j, we denote the completion time of job i under a given scheduling policy by $\mathcal{P}(i, N_j)$. Note that, besides defining the completion times, we do not discuss the particular way the jobs are scheduled by the scheduling policies we present. However, we require that *feasible* schedules are computable. A natural sufficient and necessary condition is the following: for any job $i \in N_j$, the total load of jobs with completion time at most $\mathcal{P}(i, N_j)$ is at most $\mathcal{P}(i, N_j)$.

Our three coordination mechanisms use the inefficiency of jobs in order to compute schedules. We denote by $w_{i,\min}$ the minimum load of job i over all machines. Then, its inefficiency ρ_{ij} on machine j is defined as $\rho_{ij} = w_{ij}/w_{i,\min}$. If $\rho_{ij} > m$, our coordination mechanisms schedule job i on machine j so that it finishes at time ∞. So, the assignments that have to be considered are only those in which all jobs have inefficiency at most m; we call them *m-efficient assignments.* An *optimal* m-efficient assignment is the one with minimum makespan. The term *optimal makespan* refers to the minimum makespan over all assignments. The next lemma states that the restriction to m-efficient assignments does not harm the efficiency of schedules significantly.

LEMMA 2.1. *Given any assignment of jobs to machines of makespan T, there exists an m-efficient assignment of makespan $2T$.*

Proof. Consider an assignment of jobs to machines and let W be the total load of the jobs which have been assigned to machines where they have inefficiency more than m. By rescheduling them to the machine where they have inefficiency 1, we obtain a new assignment of makespan at most $T + \frac{W}{m}$. Since W/m is a lower bound for the makespan T of the original assignment, we have that the new assignment has makespan at most $2T$. ■

Our proofs are heavily based on the convexity of simple polynomials such as z^k for $k \geq 1$ and on the relation of Euclidean norms of machine loads and the makespan. Recall that the ℓ_k norm of the machine loads for an assignment N is $\left(\sum_j L(N_j)^k\right)^{1/k}$. The proof of the next lemma is trivial.

LEMMA 2.2. *For any assignment N,* $\max_j L(N_j) \leq \left(\sum_j L(N_j)^k\right)^{1/k} \leq m^{1/k} \max_j L(N_j)$.

In some of the proofs, we also use the Minkowski inequality (or triangle inequality for the ℓ_p norm).

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

LEMMA 2.3. (MINKOWSKI INEQUALITY) $\left(\sum_{t=1}^{s}(a_t+b_t)^k\right)^{1/k} \leq \left(\sum_{t=1}^{s} a_t^k\right)^{1/k} + \left(\sum_{t=1}^{s} b_t^k\right)^{1/k}$, *for any* $k \geq 1$ *and* $a_t, b_t \geq 0$.

3 The coordination mechanism ACOORD

The coordination mechanism ACOORD uses a global ordering of the jobs according to their distinct IDs. Without loss of generality, we may assume that the index of a job is its ID. Let N be an assignment and denote by N^i the restriction of N to the jobs with the i smallest IDs. ACOORD schedules job i on machine j so that it completes at time

$$\mathcal{P}(i, N_j) = \left(\frac{(L(N_j^i)+w_{ij})^{p+1} - L(N_j^i)^{p+1}}{(p+1)w_{i,\min}}\right)^{1/p}$$

if its inefficiency is $\rho_{ij} \leq m$, and at time ∞ otherwise.

Consider the sequence of jobs in increasing order of their IDs and assume that each job plays a best-response move. In this case, job i will select that machine j so that the quantity $(L(N_j^i)+w_{ij})^{p+1} - L(N_j^i)^{p+1}$ is minimized. Since the completion time of job i depends only on jobs with smaller IDs, no job will have an incentive to change its strategy and the resulting assignment is a pure Nash equilibrium. The following lemma extends this observation in a straightforward way.

THEOREM 3.1. *The game induced by the coordination mechanism* ACOORD *is a potential game. Furthermore, any sequence of* n *rounds of best-response moves converges to a pure Nash equilibrium.*

The sequence of best-response moves mentioned above can be thought of as an online algorithm that processes the jobs in increasing order of their IDs. The local objective is slightly different that the local objective of the greedy online algorithm for minimizing the ℓ_p norm of machine loads [7, 10]; in that algorithm, job i is assigned to a machine j so that the quantity $(L(N_j^{i-1})+w_{ij})^{p+1} - L(N_j^{i-1})^{p+1}$ is minimized. Observe that this local objective may not translate into a completion time for job i in a feasible schedule. This constraint is satisfied by the coordination mechanism ACOORD as stated by the following lemma. The lemma also states that the maximum completion time is not much higher than the makespan. Its proof uses the definition of ACOORD and the convexity of function z^k.

LEMMA 3.1. *For any assignment* N*, the coordination mechanism* ACOORD *computes a feasible schedule and, furthermore,*

$$\max_{j, i \in N_j} \mathcal{P}(i, N_j) \leq 2m^{1/p} \max_j L(N_j)$$

For $p = O(\log m)$, Lemmas 3.1 and 2.2 yield that the maximum completion time in a pure Nash equilibrium N is at most a constant times the ℓ_p norm of the machine loads in N. So, the main part of the proof of Theorem 3.2 establishes an upper bound of $O(p)$ on the ratio of the ℓ_p norms of the machine loads in N and in an optimal m-efficient assignment O. In doing so, we borrow techniques from the analysis of the greedy online algorithm for the ℓ_p norm in [10]. Then, the result follows since (by Lemmas 2.1 and 2.2) the ℓ_p norm of the machine loads of O is at least a constant times the optimal makespan.

THEOREM 3.2. *The price of anarchy of the game induced by the coordination mechanism* ACOORD *with* $p = O(\log m)$ *is* $O(\log m)$.

Our analysis is asymptotically tight; this follows by the connection to online algorithms mentioned above and the lower bound of [9].

4 The coordination mechanism BCOORD

We now turn our attention to coordination mechanisms that handle anonymous jobs. BCOORD schedules job i on machine j so that it finishes at time $(\rho_{ij})^{1/p} L(N_j)$ if $\rho_{ij} \leq m$, and at time ∞ otherwise. Since $\rho_{ij} \geq 1$, the schedules produced are always feasible.

We first present an upper bound on the price of anarchy of the induced game. Since we do not give guarantees about the existence of pure Nash equilibria (see the discussion at the end of this section), the next statement is conditional.

THEOREM 4.1. *If the game induced by the coordination mechanism* BCOORD *with* $p = O(\log m)$ *has pure Nash equilibria, then its price of anarchy is* $O\left(\frac{\log m}{\log\log m}\right)$.

Proof. Consider a pure Nash equilibrium N and an optimal m-efficient assignment O. Since no job has an incentive to change her strategy from N, for any job i that is assigned to machine j_1 in N and to machine j_2 in O, we have that $(\rho_{ij_1})^{1/p} L(N_{j_1}) \leq (\rho_{ij_2})^{1/p}(L(N_{j_2}) + w_{ij_2})$. Equivalently, by raising both sides to the power p and multiplying both sides with $w_{i,\min}$, we have that $w_{ij_1} L(N_{j_1})^p \leq w_{ij_2}(L(N_{j_2}) + w_{ij_2})^p$. Using the binary variables x_{ij} and y_{ij} to denote whether job i is assigned to machine j in the assignments N ($x_{ij} = 1$) and O ($y_{ij} = 1$), respectively, or not ($x_{ij} = 0$ and $y_{ij} = 0$, respectively), we can express this last inequality as follows:

$$\begin{aligned}\sum_j x_{ij} w_{ij} L(N_j)^p &\leq \sum_j y_{ij} w_{ij} (L(N_j) + w_{ij})^p \\ &\leq \sum_j y_{ij} w_{ij} (L(N_j) + L(O_j))^p\end{aligned}$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

where the second inequality follows since $w_{ij} \leq L(O_j)$ when $y_{ij} = 1$. By summing over all jobs, we have

$$\sum_i \sum_j x_{ij} w_{ij} L(N_j)^p \leq \sum_i \sum_j y_{ij} w_{ij} (L(N_j) + L(O_j))^p.$$

By exchanging the double sums and since $\sum_i x_{ij} w_{ij} = L(N_j)$ and $\sum_i y_{ij} w_{ij} = L(O_j)$, we have

$$\sum_j L(N_j)^{p+1} \leq \sum_j L(O_j)(L(N_j) + L(O_j))^p.$$

In order to complete the proof, we will use the following technical lemma which has been proved in [43]. Similar lemmas have been used in the analysis of the price of anarchy of congestion games with polynomial latency functions [2, 5, 13].

LEMMA 4.1. (SURI, TOTH, AND ZHOU [43]) *For any real numbers $\alpha, \beta \geq 0$ and integer $k > 0$, it holds that*

$$\beta(\alpha + \beta)^{k-1} \leq \frac{k-1}{k}\alpha^k + \frac{c_k^k}{k}\beta^k$$

for $c_k = O\left(\frac{k}{\ln k}\right)$.

By applying Lemma 4.1 with $\alpha = L(N_j)$, $\beta = L(O_j)$, and $k = p+1$, we obtain that

$$\sum_j L(N_j)^{p+1} \leq \frac{p}{p+1}\sum_j L(N_j)^{p+1} + \frac{c_{p+1}^{p+1}}{p+1}\sum_j L(O_j)^{p+1}$$

which yields that

$$\left(\sum_j L(N_j)^{p+1}\right)^{\frac{1}{p+1}} \leq c_{p+1}\left(\sum_j L(O_j)^{p+1}\right)^{\frac{1}{p+1}}$$

By this inequality, Lemma 2.2, and the definition of the coordination mechanism BCOORD, we obtain that the maximum completion time in N is at most $m^{2/p}c_{p+1}$ times the makespan of O. Since (by Lemma 2.1) the makespan of the optimal m-efficient assignment O is no more than twice the optimal makespan and since $c_{p+1} = O\left(\frac{p}{\ln p}\right)$, by setting $p = O(\log m)$ we obtain that the maximum completion time in N is at most $O\left(\frac{\log m}{\log \log m}\right)$ times the optimal makespan. ■

Note that the game induced by BCOORD with $p = 1$ is the same with the game induced by the coordination mechanism CCOORD (with $p = 1$) that we present in the next section. As such, it also has a potential function (also similar to the potential function of [21] for linear weighted congestion games) as we will see in Lemma 5.2. Unfortunately, the next theorem demonstrates that, for higher values of p, the Nash dynamics of the game induced by BCOORD may contain a cycle. Its proof will appear in the final version of the paper.

THEOREM 4.2. *The game induced by the coordination mechanism* BCOORD *with $p = 2$ is not a potential game.*

5 The coordination mechanism CCOORD

In this section we present and analyze the coordination mechanism CCOORD that handles anonymous jobs and guarantees that the induced game has pure Nash equilibria, price of anarchy at most $O(\log^2 m)$, and price of stability $O(\log m)$. In order to define the scheduling policy, we first define an interesting function.

DEFINITION 1. *For integer $k \geq 0$, the function Ψ_k mapping finite sets of reals to the reals is defined as follows: $\Psi_k(\emptyset) = 0$ for any integer $k \geq 1$, $\Psi_0(A) = 1$ for any (possibly empty) set A, and for any non-empty set $A = \{a_1, a_2, ..., a_n\}$ and integer $k \geq 1$,*

$$\Psi_k(A) = k! \sum_{1 \leq d_1 \leq \ldots \leq d_k \leq n} \prod_{t=1}^{k} a_{d_t}.$$

So, $\Psi_k(A)$ is essentially the sum of all possible monomials of total degree k on the elements of A. Each term in the sum has coefficient $k!$. Clearly, $\Psi_1(A) = L(A)$. For $k \geq 2$, compare $\Psi_k(A)$ with $L(A)^k$ which can also be expressed as the sum of the same terms, albeit with different coefficients in $\{1, ..., k!\}$, given by the multinomial theorem.

The coordination mechanism CCOORD schedules job i on machine j in an assignment N so that its completion time is $\mathcal{P}(i, N_j) = (\rho_{ij}\Psi_p(N_j))^{1/p}$ if $\rho_{ij} \leq m$, and ∞ otherwise.

Our proofs extensively use the following properties. Their proof is omitted. The first inequality implies that the schedule defined by CCOORD is always feasible.

LEMMA 5.1. *For any integer $k \geq 1$, any finite set of non-negative reals A, and any non-negative real b the following hold:*

a. $L(A)^k \leq \Psi_k(A) \leq k!L(A)^k$
b. $\Psi_{k-1}(A)^k \leq \Psi_k(A)^{k-1}$
c. $\Psi_k(A \cup \{b\}) = \sum_{t=0}^{k} \frac{k!}{(k-t)!} b^t \Psi_{k-t}(A)$
d. $\Psi_k(A \cup \{b\}) - \Psi_k(A) = kb\Psi_{k-1}(A \cup \{b\})$
e. $\Psi_k(A) \leq kL(A)\Psi_{k-1}(A)$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

The second property implies that $\Psi_k(A)^{1/k} \leq \Psi_{k'}(A)^{1/k'}$ for any integer $k' \geq k$. The third property suggests an algorithm for computing $\Psi_k(A)$ in time polynomial in k and $|A|$ using dynamic programming.

We first show that the game induced by the coordination mechanism CCOORD always has pure Nash equilibria. The proof defines a potential function on the states of the induced game that will be very useful later.

LEMMA 5.2. *The function $\Phi(N) = \sum_j \Psi_{p+1}(N_j)$ is a potential function for the game induced by the coordination mechanism* CCOORD. *Hence, this game always has a pure Nash equilibrium.*

Proof. Consider two assignments N and N' differing in the strategy of the player controlling job i. Assume that job i is assigned to machine j_1 in N and to machine $j_2 \neq j_1$ in N'. Observe that $N_{j_1} = N'_{j_1} \cup \{w_{ij_1}\}$ and $N'_{j_2} = N_{j_2} \cup \{w_{ij_2}\}$. By Lemma 5.1d, we have that $\Psi_{p+1}(N_{j_1}) - \Psi_{p+1}(N'_{j_1}) = (p+1)w_{ij_1}\Psi_p(N_{j_1})$ and $\Psi_{p+1}(N'_{j_2}) - \Psi_{p+1}(N_{j_2}) = (p+1)w_{ij_2}\Psi_p(N'_{j_2})$. Using these properties and the definitions of the coordination mechanism CCOORD and function Φ, we have

$$\begin{aligned} & \Phi(N) - \Phi(N') \\ = \ & \sum_j \Psi_{p+1}(N_j) - \sum_j \Psi_{p+1}(N'_j) \\ = \ & \Psi_{p+1}(N_{j_1}) + \Psi_{p+1}(N_{j_2}) - \Psi_{p+1}(N'_{j_1}) - \Psi_{p+1}(N'_{j_2}) \\ = \ & (p+1)w_{ij_1}\Psi_p(N_{j_1}) - (p+1)w_{ij_2}\Psi_p(N'_{j_2}) \\ = \ & (p+1)w_{i,\min}\left(\mathcal{P}(i,N_{j_1})^p - \mathcal{P}(i,N'_{j_2})^p\right) \end{aligned}$$

which means that the difference of the potentials of the two assignments and the difference of the completion time of player i have the same sign as desired. ∎

The next lemma relates the maximum completion time of an assignment to the makespan of another assignment provided that their potentials are close. Its proof is omitted. It extensively uses the properties of function Ψ_{p+1} (Lemma 5.1).

LEMMA 5.3. *Let N and O be m-efficient assignments such that $(\Phi(N))^{\frac{1}{p+1}} \leq \gamma\,(\Phi(O))^{\frac{1}{p+1}}$. Then,*

$$\max_{j,i\in N_j} \mathcal{P}(i,N_j) \leq \gamma(p+1)m^{2/p}\max_j L(O_j).$$

A first application of Lemma 5.3 is in bounding the price of stability of the induced game.

THEOREM 5.1. *The game induced by the coordination mechanism* CCOORD *with $p = O(\log m)$ has price of stability at most $O(\log m)$.*

Proof. Consider the optimal m-efficient assignment O and the pure Nash equilibrium N of minimum potential. We have $(\Phi(N))^{\frac{1}{p+1}} \leq (\Phi(O))^{\frac{1}{p+1}}$ and, by Lemma 5.3, we obtain that the maximum completion time in N is at most $(p+1)m^{2/p}$ times the makespan of O and, by Lemma 2.1, at most $2(p+1)m^{2/p}$ times the optimal makespan. Setting $p = O(\log m)$, the theorem follows. ∎

We now prove the upper bound on the price of anarchy of the induced game. We consider approximate equilibria, i.e., assignments from which deviations of players cannot improve their completion times significantly.

DEFINITION 2. *Consider an assignment N under which job i is assigned to machine j_1 and define $\Delta_i(N) = \max\{0, w_{ij_1}\Psi_p(N_{j_1}) - \min_{j\neq j_1} w_{ij}\Psi_p(N_j \cup \{w_{ij}\})\}$. Player i is called an α-equilibrium player if $\Delta_i(N) \leq \alpha w_{ij_1}\Psi_p(N_{j_1})$. Let $\Delta(N) = \sum_i \Delta_i(N)$. The assignment N is called an α-equilibrium if $\sum_{i\in C}\Delta_i(N) \leq \alpha\Phi(N)$ where C denotes the set of non-α-equilibrium players.*

Intuitively, $\Delta_i(N)$ gives a measure of the gain of player i (and a measure of the decrease of the potential) after a best-response move from her strategy in N. Clearly, an assignment is a pure Nash equilibrium of the induced game if and only if it is a 0-equilibrium. Note that our definition of approximate equilibria is inspired by [6] but is non-standard since it relates the gain of the non-α-equilibrium players directly with the potential.

LEMMA 5.4. *Let α be such that $0 \leq \alpha < \frac{1}{2(p+1)}$. At any α-equilibrium N, it holds that*

$$(\Phi(N))^{\frac{1}{p+1}} \leq \frac{p+1}{\ln(2-2\alpha(p+1))}(\Phi(O))^{\frac{1}{p+1}},$$

where O is the optimal m-efficient assignment.

Proof. Consider a player i that is assigned to machine j_1 in N and to machine j_2 in O. By the definition of $\Delta_i(N)$, we have that

$$w_{ij_1}\Psi_p(N_{j_1}) - \Delta_i(N) \quad \leq \quad w_{ij_2}\Psi(N_{j_2}\cup\{w_{ij_2}\}).$$

Using the binary variables x_{ij} and y_{ij} to denote whether job i is assigned to machine j in the assignment N ($x_{ij} = 1$) and O ($y_{ij} = 1$) or not ($x_{ij} = 0$ and $y_{ij} = 0$, respectively), we can express the last inequality as follows:

$$\sum_j x_{ij}w_{ij}\Psi_p(N_j) - \Delta_i(N) \quad \leq \quad \sum_j y_{ij}w_{ij}\Psi(N_j\cup\{w_{ij}\})$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

By summing over all jobs, we have

$$\begin{aligned}&\sum_i \left(\sum_j x_{ij} w_{ij} \Psi_p(N_j) - \Delta_i(N)\right)\\ \leq\ &\sum_i \sum_j y_{ij} w_{ij} \Psi(N_j \cup \{w_{ij}\})\end{aligned}$$

By exchanging the double sums and since $\sum_i x_{ij} w_{ij} = L(N_j)$ and $\sum_i \Delta_i(N) = \Delta(N)$, we obtain

$$\begin{aligned}&\sum_j L(N_j)\Psi_p(N_j) - \Delta(N)\\ \leq\ &\sum_j \sum_i y_{ij} w_{ij} \Psi_p(N_j \cup \{w_{ij}\})\end{aligned} \tag{5.1}$$

Now, denote by C the set of non-α-equilibrium players and by D the set of α-equilibrium players. By Definition 2, we have that

$$\begin{aligned}\Delta(N) &= \sum_{i\in C} \Delta_i(N) + \sum_{i \in D} \Delta_i(N)\\ &\leq \alpha\Phi(N) + \sum_{i\in D}\sum_j \alpha x_{ij} w_{ij} \Psi_p(N_j)\\ &\leq \alpha\Phi(N) + \alpha \sum_j \sum_i x_{ij} w_{ij} \Psi_p(N_j)\\ &= \alpha\Phi(N) + \alpha \sum_j L(N_j)\Psi_p(N_j)\\ &\leq \alpha\Phi(N) + \alpha \sum_j \Psi_{p+1}(N_j)\\ &= 2\alpha\Phi(N).\end{aligned}$$

The last inequality follows by Lemma 5.1b.

Using the above inequality and Lemma 5.1e in the first two lines of the following derivation, we have

$$\begin{aligned}&(2-2\alpha(p+1))\Phi(N)\\ =\ &\Phi(N) + \sum_j \Psi_{p+1}(N_j) - 2\alpha(p+1)\Phi(N)\\ \leq\ &\Phi(N) + (p+1)\sum_j L(N_j)\Psi_p(N_j) - (p+1)\Delta(N)\\ \leq\ &\Phi(N) + (p+1)\sum_j\sum_i y_{ij} w_{ij} \Psi_p(N_j \cup \{w_{ij}\})\\ =\ &\Phi(N)\\ &+(p+1)\sum_j\sum_i y_{ij} w_{ij} \sum_{t=0}^{p} \frac{p!}{(p-t)!}\Psi_{p-t}(N_j) w_{ij}^t\\ =\ &\Phi(N) + \sum_j \sum_{t=0}^{p} \frac{(p+1)!}{(p-t)!}\Psi_{p-t}(N_j)\sum_i y_{ij} w_{ij}^{t+1}\\ \leq\ &\Phi(N) + \sum_j \sum_{t=0}^{p} \frac{(p+1)!}{(p-t)!(t+1)!}\Psi_{p-t}(N_j)\Psi_{t+1}(O_j)\\ =\ &\Phi(N) + \sum_j \sum_{t=1}^{p+1} \binom{p+1}{t} \Psi_{p+1-t}(N_j)\Psi_t(O_j)\\ \leq\ &\Phi(N)\\ &+\sum_j \sum_{t=1}^{p+1} \binom{p+1}{t} \Psi_{p+1}(N_j)^{\frac{p+1-t}{p+1}} \Psi_{p+1}(O_j)^{\frac{t}{p+1}}\\ =\ &\Phi(N) + \sum_j \left(\left(\Psi_{p+1}(N_j)^{\frac{1}{p+1}} + \Psi_{p+1}(O_j)^{\frac{1}{p+1}}\right)^{p+1}\right.\\ &\left.-\Psi_{p+1}(N_j)\right)\\ =\ &\Phi(N) + \sum_j \left(\Psi_{p+1}(N_j)^{\frac{1}{p+1}} + \Psi_{p+1}(O_j)^{\frac{1}{p+1}}\right)^{p+1}\\ &-\sum_j \Psi_{p+1}(N_j)\\ \leq\ &\left(\left(\sum_j \Psi_{p+1}(N_j)\right)^{\frac{1}{p+1}} + \left(\sum_j \Psi_{p+1}(O_j)\right)^{\frac{1}{p+1}}\right)^{p+1}\\ =\ &\left((\Phi(N))^{\frac{1}{p+1}} + (\Phi(O))^{\frac{1}{p+1}}\right)^{p+1}\end{aligned}$$

The second inequality follows by inequality (5.1), the second equality follows by Lemma 5.1c, the third equality follows by exchanging the sums, the third inequality follows since the jobs i assigned to machine j are those for which $y_{ij} = 1$ and by the definition of function Ψ_{t+1} which yields that $\Psi_{t+1}(O_j) \geq (t+1)! \sum_i y_{ij} w_{ij}^{t+1}$, the fourth equality follows by updating the limits of the sum over t, the fourth inequality follows by Lemma 5.1b, the fifth equality follows by the binomial theorem, the sixth equality is obvious, the seventh equality follows by the definition of the potential $\Phi(N)$, and the fifth inequality follows by Minkowski inequality and by the definition of the potential $\Phi(N)$.

By the above inequality, we obtain that

$$\begin{aligned}(\Phi(N))^{\frac{1}{p+1}} &\leq \frac{1}{(2-2\alpha(p+1))^{\frac{1}{p+1}} - 1}(\Phi(O))^{\frac{1}{p+1}}\\ &\leq \frac{p+1}{\ln(2-2\alpha(p+1))}(\Phi(O))^{\frac{1}{p+1}}\end{aligned}$$

where the last inequality follows using the inequality $e^z \geq z+1$. ∎

We are now ready to bound the price of anarchy.

THEOREM 5.2. *The coordination mechanism* CCOORD *with* $p = O(\log m)$ *has approximation ratio at most* $O\left(\log^2 m\right)$.

Proof. Consider a pure Nash equilibrium N and let O be the optimal m-efficient assignment. Using Lemma 5.4 with $\alpha = 0$, we have that $(\Phi(N))^{\frac{1}{p+1}} \leq \frac{p+1}{\ln 2}(\Phi(O))^{\frac{1}{p+1}}$.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Hence, by Lemma 5.3, we obtain that the maximum completion time in N is at most $\frac{(p+1)^2}{\ln 2} m^{2/p}$ times the makespan of O, and, by Lemma 2.1, at most $2\frac{(p+1)^2}{\ln 2} m^{2/p}$ times the optimal makespan. Setting $p = O(\log m)$, the theorem follows. ■

In our last result, we consider a sequence of best-response moves of players so that each time the player that yields the maximum decrease in the potential plays. We show that, in this way, the players reach an efficient assignment after a polynomial number of moves.

THEOREM 5.3. *Starting from any m-efficient assignment, the game induced by the coordination mechanism* CCOORD *with* $p = O(\log m)$ *converges to an* $O\left(\log^2 m\right)$*-approximate assignment after at most* $O\left(n \log^2 m\right)$ *best-response moves.*

Proof. (*sketch*) Let O be the optimal m-efficient assignment. We can show that after $O(np \log m)$ best-response moves, an assignment N with potential such that $(\Phi(N))^{\frac{1}{p+1}} \leq 2(p+1)\,(\Phi(O))^{\frac{1}{p+1}}$ is reached. Then, by Lemma 5.3, we have that the maximum completion time in N is at most $2(p+1)^2 m^{2/p}$ times the makespan of O, and, by Lemma 2.1, at most $4(p+1)^2 m^{2/p}$ times the optimal makespan. The theorem will then follow by setting $p = O(\log m)$. ■

6 Open problems

Our work reveals several interesting questions. First of all, it leaves open the question of whether coordination mechanisms with constant approximation ratio exist. In particular, is there any coordination mechanism that handles anonymous jobs, guarantees that the induced game has pure Nash equilibria, and has constant price of anarchy? Based on the lower bound of [8], such a coordination mechanism (if it exists) must use preemption. Alternatively, is the case of anonymous jobs provably more difficult than the case where jobs have IDs? Investigating the limits of non-preemptive mechanisms is still interesting. Notice that AJM-1 is the only non-preemptive coordination mechanism that has approximation ratio $o(m)$ but it does not guarantee that the induced game has pure Nash equilibria; furthermore, the only known coordination mechanism that induces a potential game with bounded price of anarchy is ShortestFirst. So, is there any non-preemptive (deterministic or randomized) coordination mechanism that is simultaneously $o(m)$-approximate and induces a potential game? We also remark that Theorem 4.2 does not necessarily exclude a game induced by the coordination mechanism BCOORD from having pure Nash equilibria. Also, notice that both our coordination mechanisms and AJM-2 assume that the number of machines m is known. Is this information really necessary in order to obtain $o(m)$-approximate coordination mechanisms?

Finally, we believe that the games induced by the coordination mechanism CCOORD are of independent interest. We have proved that these games belong to the class PLS [24]. Furthermore, the result of Monderer and Shapley [34] and the proof of Lemma 5.2 essentially show that each of these games is isomorphic to a congestion game. However, they have a beautiful definition as games on parallel machines that gives them a particular structure. What is the complexity of computing pure Nash equilibria in such games? Even in case that these games are PLS-complete like several variations of congestion games that were considered recently [1, 18, 42], it is still interesting to study the convergence time to efficient assignments. Our convergence result (Theorem 5.3) is not absolutely satisfactory in this respect since the players act in a coordinated way. A series of recent papers [6, 15, 19, 33] consider adversarial rounds of best-response moves in potential games so that each player is given at least one chance to play in each round (this is essentially our assumption in Theorem 3.1 for the coordination mechanism ACOORD). Does the game induced by the coordination mechanism CCOORD converges to efficient assignments after a polynomial number of adversarial rounds of best-response moves? Although it is a potential game, it does not have the particular properties considered in [6] and, hence, proving such a statement probably requires different techniques.

References

[1] H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. *47th FOCS*, pp. 613-622, 2006.

[2] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. *23rd STACS*, LNCS 3884, pp. 218-229, 2006.

[3] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *45th FOCS*, pp. 295-304, 2004.

[4] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3): 486-504, 1997.

[5] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. *37th STOC*, pp. 57-66, 2005.

[6] B. Awerbuch, Y. Azar, A. Epstein, V. S. Mirrokni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. *9th EC*, pp. 264-273, 2008.

[7] B. Awerbuch, Y. Azar, E. F. Grove, M.-Y. Kao, P.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Krishnan, and J. S. Vitter. Load balancing in the L_p norm. *36th FOCS*, pp. 383-391, 1995.
[8] Y. Azar, K. Jain, and V. S. Mirrokni. (Almost) optimal coordination mechanisms for unrelated machine scheduling. *19th SODA*, pp. 323-332, 2008.
[9] Y. Azar, J. Naor, and R. Rom. The competitiveness of on-line assignments. *Journal of Algorithms*, 18(2): 221-237, 1995.
[10] I. Caragiannis. Better bounds for online load balancing on unrelated machines. *19th SODA*, pp. 972-981, 2008.
[11] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Taxes for linear atomic congestion games. *14th ESA*, LNCS 4168, pp. 184-195, 2006.
[12] H.-L. Chen, T. Roughgarden, and G. Valiant. Designing networks with good equilibria. *19th SODA*, pp. 854-863, 2008.
[13] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. *37th STOC*, pp. 67-73, 2005.
[14] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. *31st ICALP*, LNCS 3142, pp. 345-357, 2004.
[15] G. Christodoulou, V. S. Mirrokni, and A. Sidiropoulos. Convergence and approximation in potential games. *23rd STACS*, LNCS 3884, pp. 349-360, 2006.
[16] R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *Journal of Computer and System Sciences*, 72(3): 444-467, 2006.
[17] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibria. *30th ICALP*, LNCS 2719, pp. 502-513, 2003.
[18] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. *36th STOC*, pp. 604-612, 2004.
[19] A. Fanelli, M. Flammini, and L. Moscardelli. Speed of convergence in congestion games under best responce dynamics. *35th ICALP*, pp. 796-807, LNCS 5125, 2008.
[20] L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. *45th FOCS*, pp. 277-285, 2004.
[21] D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. *Theoretical Computer Science*, 340(3): 514-538, 2005.
[22] O. H. Ibarra and C. E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM*, 24(2): 280-289, 1977.
[23] N. Immorlica, L. Li, V. S. Mirrokni, and A. Schulz. Coordination mechanisms for selfish scheduling. *1st WINE*, LNCS 3828, pp. 55-69, 2005.
[24] D. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37: 79-100, 1988.
[25] A. C. Kaporis and P. G. Spirakis. The price of optimum in Stackelberg games on arbitrary single commodity networks and latency functions. *18th SPAA*, pp. 19-28, 2006.
[26] G. Karakostas, and S. Kolliopoulos. Edge pricing of multicommodity networks for heterogeneous selfish users. *45th FOCS*, pp. 268-276, 2004.
[27] Y. A. Korilis, A. A. Lazar, and A. Orda. Architecting noncooperative networks. *IEEE Journal on Selected Areas in Communications*, 13(7): 1241-1251, 1995.
[28] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5(1): 161-173, 1997.
[29] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *16th STACS*, LNCS 1563, pp. 404-413, 1999.
[30] V. S. Anil Kumar and M. V. Marathe. Improved results for Stackelberg scheduling strategies. *29th ICALP*, LNCS 2380, pp. 776-787, 2002.
[31] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46: 259-271, 1990.
[32] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13: 111-124, 1996.
[33] V. S. Mirrokni and A. Vetta. Convergence issues in competitive games. *APPROX-RANDOM*, LNCS 3122, pp. 183-194, 2004.
[34] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14: 124-143, 1996.
[35] N. Nissan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.
[36] C. H. Papadimitriou. Algorithms, games and the internet. *33rd STOC*, pp. 749-753, 2001.
[37] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2: 65-67, 1973.
[38] T. Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2): 332-350, 2004.
[39] T. Roughgarden. On the severity of Braess's Paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5): 922-953, 2006.
[40] T. Roughgarden. Routing games. In [35], Chapter 18, pp. 461-486, 2007.
[41] P. Schuurman and T. Vredeveld. Performance guarantees of local search for multiprocessor scheduling. *8th IPCO*, LNCS 2081, pp. 370-382, 2001
[42] A. Skopalik and B. Vöcking. Inapproximability of pure Nash equilibria. *40th STOC*, pp. 355-364, 2008.
[43] S. Suri, C. D. Toth, and Y. Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 47(1): 77-96, 2007.
[44] C. Swamy. The effectiveness of Stackelberg strategies and tolls for network congestion games. *18th SODA*, pp. 1133-1142, 2007.
[45] B. Vöcking. Selfish load balancing. In [35], Chapter 20, pp. 517-542, 2007.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.