CrossMark

An Almost Ideal Coordination Mechanism for Unrelated Machine Scheduling

Ioannis Caragiannis[1] ⓘ **· Angelo Fanelli**[2]

Abstract Coordination mechanisms aim to mitigate the impact of selfishness when scheduling jobs to different machines. Such a mechanism defines a scheduling policy within each machine and naturally induces a game among the selfish job owners. The desirable properties of a coordination mechanism includes simplicity in its definition and efficiency of the outcomes of the induced game. We present a broad class of coordination mechanisms for unrelated machine scheduling that are simple to define and we identify one of its members (mechanism DCOORD) that is superior to all known mechanisms. In particular, DCOORD induces potential games with logarithmic price of anarchy and only constant price of stability. Both bounds are almost optimal.

✉ Ioannis Caragiannis
caragian@ceid.upatras.gr

Angelo Fanelli
angelo.fanelli@unicaen.fr

1   Department of Computer Engineering and Informatics, University of Patras, 26504 Rion-Patras, Greece

2   CNRS (UMR-6211), Université de Caen Basse-Normandie, Caen France

1 Introduction

We consider a selfish scheduling setting where each job owner acts as a non-cooperative player and aims to assign her job to one of the available machines so that the completion time of the job is as low as possible. An algorithmic tool that can be utilized by the designer of such a system is a *coordination mechanism* [9]. The coordination mechanism uses a *scheduling policy* within each machine that aims to mitigate the impact of selfishness to performance.

We focus on *unrelated machine scheduling*. There are m available machines and n players, each controlling a distinct job. The job (owned by player) u has a (possibly infinite) positive processing time (or load) $w_{u,j}$ when processed by machine j. A scheduling policy defines the way jobs are scheduled within a machine. In its simplest form, such a policy is *non-preemptive* and processes jobs uninterruptedly according to some order. *Preemptive* scheduling policies (which is our focus here) do not necessarily have this feature (e.g., they may process jobs in parallel) and may even introduce some idle time.

Naturally, a coordination mechanism induces a game with the job owners as players. Each player has all machines as possible *strategies*. The term *assignment* is used for a snapshot of the game, where each player has selected a strategy, i.e., she has selected a particular machine to process her job. Given an assignment, the cost a player experiences is the completion time of her job on the machine she has selected. This is well-defined by the scheduling policy of the machine and typically depends on the characteristics of all jobs assigned to the machine.

Assignments in which no player has any incentive to change her strategy are called *pure Nash equilibria* (or, simply, *equilibria*). When studying a coordination mechanism, we are interested in bounding the inefficiency of equilibria of the game induced by the mechanism. We use the *maximum completion time* among all jobs to measure the social cost. A related quantity is the load of a machine which is defined as the total processing time of the jobs assigned to the machine. The *makespan* of an assignment is the maximum load over all machines. Clearly, the makespan of an assignment is a lower bound on the maximum completion time. The *price of anarchy* (respectively, *price of stability*) of the game induced by a coordination mechanism is defined as the worst (respectively, best) ratio of the maximum completion time over all equilibria over the optimal makespan.

We prefer mechanisms that induce games that always have equilibria. Furthermore, we would like these equilibria to be easy to find. A highly desirable property that ensures that equilibria can be reached by the players themselves (for example, with best-response play) is the existence of a *potential function*. A potential function is defined over all possible assignments and has the property that, in any two assignments that differ in the strategy of a single player, the difference of the two values of the potential and the difference of the completion time of the deviating player have the same sign.

Coordination mechanisms for scheduling were introduced by Christodoulou et al. [9]. Immorlica et al. [12] were the first to consider coordination mechanisms in the unrelated machine setting and studied several intuitive mechanisms, including `ShortestFirst` and `Makespan`. In `ShortestFirst`, the jobs in each

machine are scheduled non-preemptively, in monotone non-decreasing order of their processing time. Since ties are possible, the mechanism has to distinguish between jobs with identical processing times, e.g., using distinct IDs for the jobs. This is necessary for every deterministic non-preemptive coordination mechanism in order to be well-defined. In contrast, in Makespan, each machine processes the jobs assigned to it "in parallel" so that they all have the same completion time. So, no ID information is required by Makespan. We use the term *anonymous* to refer to coordination mechanisms having this property. These two coordination mechanisms are *strongly local* in the sense that the only information that is required to compute the schedule of jobs within a machine is the processing time of the jobs on that machine only. A *local* coordination mechanism may use all parameters of the jobs that are assigned to a machine (e.g., the whole load vector of each job).

Azar et al. [4] prove lower bounds of $\Omega(m)$ and $\Omega(\log m)$ on the price of anarchy for any strongly local and local non-preemptive coordination mechanism, respectively. On the positive side, they presented two local coordination mechanisms with price of anarchy $o(m)$. Their first coordination mechanism (henceforth called AFJMS-1) is non-preemptive and may induce games without equilibria. When the induced game has equilibria, the price of anarchy is at most $\mathcal{O}(\log m)$. Their second coordination mechanism (henceforth called AFJMS-2) is preemptive, induces potential games (and, hence, the existence of equilibria is guaranteed), and has price of anarchy $\mathcal{O}(\log^2 m)$. Both mechanisms are not anonymous.

Caragiannis [7] presents three more coordination mechanisms. The mechanism ACOORD, induces potential games with price of anarchy $\mathcal{O}(\log m)$. The mechanism uses the distinct IDs of the jobs to ensure that the equilibria of the game are essentially assignments that are reached by a greedy-like online algorithm for minimizing the p-norm of machine loads. [3] and [6] study this online scheduling problem; the results therein imply that the price of stability of mechanism ACOORD is $\Omega(\log m)$ as well. A different coordination mechanism with similar characteristics (called Balance) is presented in [10]. The coordination mechanism BCOORD (defined also in [7]) has even better price of anarchy $O\left(\frac{\log m}{\log \log m}\right)$ (matching a lower bound due to Abed and Huang [2] for all deterministic coordination mechanisms) but the induced games are not potential ones and may not even have equilibria. However, the price of anarchy bound for BCOORD indicates that preemption may be useful in order to beat the $\Omega(\log m)$ lower bound for non-preemptive mechanisms from [4]. Interestingly, this mechanism is anonymous. The third mechanism CCOORD is anonymous as well, induces potential games, and has price of anarchy and price of stability $\mathcal{O}(\log^2 m)$ and $\mathcal{O}(\log m)$, respectively. To the best of our knowledge, this is the only anonymous mechanism that induces potential games and has polylogarithmic price of anarchy.[1] Table 1 summarizes the known local coordination mechanisms.

In the discussion above, we have focused on papers that define the social cost as the maximum completion time (among all players). An alternative social cost that has

---

[1]Even though their mechanism Balance heavily uses job IDs, Cohen et al. [10] claim that it is anonymous. This is certainly false according to our terminology since anonymity imposes that two jobs with identical load vectors should be indistinguishable.

Table 1 A comparison between DCOORD and other local coordination mechanisms from the literature. A dash (-) indicates that no bound is known (or is implied by related results in the literature). The question mark (?) indicates that it is not known whether all games induced by mechanism BCOORD have equilibria

Coordination mechanism	PoA	PoS	PNE	Pot.	IDs	Preempt.	Reference
AFJMS-1	$\Theta(\log m)$	$-$	No	No	Yes	No	[4]
AFJMS-2	$\mathcal{O}(\log^2 m)$	$-$	Yes	Yes	Yes	Yes	[4]
ACOORD	$\mathcal{O}(\log m)$	$\Theta(\log m)$	Yes	Yes	Yes	Yes	[7]
Balance	$\mathcal{O}(\log m)$	$\Theta(\log m)$	Yes	Yes	Yes	Yes	[10]
BCOORD	$\Theta(\frac{\log m}{\log \log m})$	$-$	?	No	No	Yes	[7]
CCOORD	$\mathcal{O}(\log^2 m)$	$\mathcal{O}(\log m)$	Yes	Yes	No	Yes	[7]
DCOORD	$\mathcal{O}(\log m)$	$\mathcal{O}(1)$	Yes	Yes	No	Yes	this paper

received much attention is the *weighted average completion time*; see [1, 5, 8, 11] for some recent related results. Interestingly, the design principles that lead to efficient mechanisms in their case are considerably different.

Our contribution is as follows. We introduce a quite broad class (called $\mathcal{M}(d)$) of local anonymous coordination mechanisms that induce potential games. The class contains the coordination mechanism CCOORD as well as the novel coordination mechanism DCOORD, which has additional *almost ideal* properties. In particular, we prove that it has logarithmic price of anarchy (i.e., only an $\mathcal{O}(\log \log m)$ factor away from the lower bound of Abed and Huang [2]) and only constant price of stability. A (qualitative and quantitative) comparison of DCOORD to other known local coordination mechanisms is depicted in Table 1.

The rest of the paper is structured as follows. We begin with preliminary definitions in Section 2. Section 3 is devoted to the definition of the class of mechanisms $\mathcal{M}(d)$ and to the proof that all mechanisms in this class induce potential games. Then, the novel mechanism DCOORD from this class is defined in Section 4; its feasibility as well as preliminary statements that are useful for the analysis are also presented there. Finally, in Section 5, we prove the bounds on the price of anarchy and stability.

2 Definitions and Preliminaries

Throughout the paper, we denote the number of machines by m. The index j always refers to a machine; the sum $\sum_j$ runs over all available machines. An assignment is a partition $N = (N_1, ..., N_m)$ of the players to the m machines. So, N_j is the set of players assigned to machine j under N. We use the notation $L_j(N_j)$ to refer to the load of machine j, i.e., $L_j(N_j) = \sum_{u \in N_j} w_{u,j}$. Recall that $w_{u,j}$ is the (possibly infinite) positive processing time (or load) that the job (owned by player) u has when processed by machine j.

A coordination mechanism uses a scheduling policy per machine. For every set of jobs assigned to machine j, the scheduling policy of the machine defines a detailed

schedule of the jobs in the machine, i.e., it defines which job is executed in each point in time, whether more than one jobs are executed in parallel, or whether a machine stays idle for particular time intervals. Instead of defining coordination mechanisms at this level of detail, it suffices to focus on the definition of the completion time $\mathcal{P}(u, N_j)$ for the job of each player $u \in N_j$. This definition should correspond to some feasible detailed scheduling of jobs in the machine. A sufficient condition that guarantees *feasibility* is to define completion times that are never smaller than the machine load.

Like the coordination mechanisms in [4, 7, 10], our coordination mechanisms are local. The completion time $\mathcal{P}(u, N_j)$ of the job belonging to player u in machine j depends on the processing times the jobs in N_j have on machine j, as well as on the minimum processing time $w_u = \min_j w_{u,j}$ of job u over all machines.

Our proofs exploit simple facts about Euclidean norms of machine loads. Recall that, for $p \geq 1$, the p-norm of the vector of machine loads $L(N) = (L_1(N_1), L_2(N_2), ..., L_m(N_m))$ under an assignment N is $\|L(N)\|_p = \left(\sum_j L_j(N_j)^p\right)^{1/p}$. By convention, we denote the makespan $\max_j L_j(N_j)$ as $\|L(N)\|_\infty$. The following property follows easily by the definition of norms; we use it extensively in the following.

Lemma 1 *For any $p \geq 1$ and any assignment N, $\|L(N)\|_\infty \leq \|L(N)\|_p \leq m^{1/p}\|L(N)\|_\infty$.*

We also use the well-known Minkowski inequality (or triangle inequality for the p-norm). For machine loads, it reads as follows:

Lemma 2 (Minkowski inequality) *For every $p \geq 1$ and two assignments N and N', $\|L(N) + L(N')\|_p \leq \|L(N)\|_p + \|L(N')\|_p$.*

The notation $L(N) + L(N')$ denotes the m-entry vector with $L_j(N_j) + L_j(N'_j)$ at the j-th entry. Another necessary technical lemma follows by the convexity properties of polynomials; see [7] for a proof.

Lemma 3 *For $r \geq 1$, $t \geq 0$, positive integer p, and $a_i \geq 0$ for $i = 1, ..., p$, it holds*

$$\sum_{i=1}^{p} \left((t + a_i)^r - t^r\right) \leq \left(t + \sum_{i=1}^{p} a_i\right)^r - t^r.$$

3 A Broad Class of Coordination Mechanisms

In this section, we show that the coordination mechanism CCOORD from [7] can be thought of as belonging to a broad class of coordination mechanisms, which we call $\mathcal{M}(d)$. This class contains also our novel coordination mechanism DCOORD, which will be presented in Section 4.

The definition of CCOORD uses a positive integer $d \geq 2$ and the functions Ψ_j that map sets of players to the non-negative reals as follows. For any machine j, $\Psi_j(\emptyset) = 0$ and for any non-empty set of players $U = \{u_1, u_2, ..., u_\ell\}$,

$$\Psi_j(U) = d! \sum_{t_1+t_2+...+t_\ell=d} \prod_{k=1}^{\ell} w_{u_k,j}^{t_k}.$$

The sum runs over all multi-sets of non-negative integers $\{t_1, t_2, ..., t_\ell\}$ that satisfy $t_1 + t_2 + ... + t_\ell = d$. So, $\Psi_j(U)$ is the sum of all possible degree-d monomials of the processing times of the jobs belonging to players from U on machine j, with each term in the sum having a coefficient of $d!$. CCOORD schedules the job of player u_i on machine j in an assignment N so that its completion time is

$$\mathcal{P}(u_i, N_j) = \left(\frac{w_{u_i,j} \Psi_j(N_j)}{w_{u_i}} \right)^{1/d}.$$

We will extend CCOORD to define a broad class of coordination mechanisms; we use $\mathcal{M}(d)$ to refer to this class, where $d \geq 2$ is a positive integer. Each member of $\mathcal{M}(d)$ is identified by a *coefficient function* γ. The coefficient functions are defined over multi-sets of non-negative integers that have sum equal to $d + 1$ and take non-negative values. An important property of the coefficient functions is that they are *invariant to zeros* that requires that for a multi-set A of integers that sum up to $d + 1$, $\gamma(A) = \gamma(A \cup \{0\})$. Hence, the value returned by γ depends only on the non-zero elements in the multiset it takes as argument.

The definition of a coordination mechanism in $\mathcal{M}(d)$ uses the quantity $\Lambda_{u_i,j}(U)$, which is defined as follows for a machine j and a job u_i from a set of jobs $U = \{u_1, u_2, ..., u_\ell\}$:

$$\Lambda_{u_i,j}(U) = \sum_{\substack{t_1+t_2+...+t_\ell=d+1 \\ t_i \geq 1}} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell} w_{u_k,j}^{t_k}. \tag{1}$$

The sum runs over all multi-sets of non-negative integers, with each integer corresponding to a distinct player of U, so that the integer t_i corresponding to player u_i is strictly positive. Notice that γ is defined over (unordered) multi-sets; this implies that symmetric monomials have the same coefficient. For example, for $d = 2$, the set of players $U = \{u_1, u_2\}$, and a machine j,

$$\Lambda_{u_1,j}(U) = \gamma(\{3, 0\}) w_{u_1,j}^3 + \gamma(\{2, 1\}) w_{u_1,j}^2 w_{u_2,j} + \gamma(\{1, 2\}) w_{u_1,j} w_{u_2,j}^2.$$

Clearly, $\{2, 1\}$ and $\{1, 2\}$ denote the same multi-set and, hence, the coefficients of the (symmetric) second and third monomial are identical.

A coordination mechanism of $\mathcal{M}(d)$ sets the completion time of player u_i to

$$\mathcal{P}(u_i, N_j) = \left(\frac{\Lambda_{u_i,j}(N_j)}{w_{u_i}} \right)^{1/d}. \tag{2}$$

when her job is scheduled on machine j under assignment N.

By simply setting $\gamma(A) = d!$ for every multi-set A of non-negative integers summing up to $d + 1$, we obtain CCOORD. Indeed, it is easy to see that $\Lambda_{u_i,j}(U) = w_{u_i,j}\Psi_j(U)$ in this case.

The definition of $\mathcal{M}(d)$ guarantees that all its members satisfy two important properties. First, every coordination mechanism in $\mathcal{M}(d)$ is anonymous. This is due to the fact that the definition of the completion time in (2) does not depend on the identity of a player and the jobs of two different players u and u' that have equal processing times $w_{u,j} = w_{u',j}$ at machine j and the same minimum processing time (over all machines) will enjoy identical completion times therein, when each is scheduled together with a set U of other players (i.e., $\mathcal{P}(u, U \cup \{u\}) = \mathcal{P}(u', U \cup \{u'\})$) or when the set of players N_j assigned to machine j contains both u and u' ($\mathcal{P}(u, N_j) = \mathcal{P}(u', N_j)$ in this case).

Another important property of the coordination mechanisms in $\mathcal{M}(d)$ is that they always induce potential games. We will prove this in a while, after defining the function $\Lambda_j(U)$, again for a machine j and a set of players $U = \{u_1, u_2, ..., u_\ell\}$, as follows:

$$\Lambda_j(U) = \sum_{t_1+t_2+...+t_\ell=d+1} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell} w_{u_k,j}^{t_k}. \tag{3}$$

Compared to the definition of $\Lambda_{u_i,j}(U)$ in (1), the sum in (3) runs just over all multi-sets of non-negative integers (corresponding to players in U) that sum up to $d + 1$, without any additional constraint.

We will sometimes use the informal term Λ-*functions* to refer to the functions defined in both (1) and (3). We can now state and prove the following property of Λ-functions that we will use several times in our analysis below.

Lemma 4 *Consider a machine j and a set of players $U = \{u_1, u_2, ..., u_\ell\}$. Then, for every player $u_i \in U$,*
$$\Lambda_j(U) = \Lambda_{u_i,j}(U) + \Lambda_j(U \setminus \{u_i\}).$$

Proof Without loss of generality, let us assume that $i = 1$. Using the definition of Λ-functions in (1) and (3), we obtain

$$\begin{aligned}
\Lambda_j(U) &= \sum_{t_1+t_2+...+t_\ell=d+1} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell} w_{u_k,j}^{t_k} \\
&= \sum_{t_1+t_2+...+t_\ell=d+1 t_1\geq 1} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell} w_{u_k,j}^{t_k} \\
&\quad + \sum_{t_1+t_2+...+t_\ell=d+1 t_1=0} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell} w_{u_k,j}^{t_k} \\
&= \Lambda_{u_1,j}(U) + \sum_{t_2+...+t_\ell=d+1} \gamma(\{t_2, ..., t_\ell\}) \prod_{k=2}^{\ell} w_{u_k,j}^{t_k} \\
&= \Lambda_{u_1,j}(U) + \Lambda_j(U \setminus \{u_1\}).
\end{aligned}$$

In the third equality, we have used the fact that the coefficient function is invariant to zeros.   □

Lemma 4 is particularly useful in order to prove that mechanisms of $\mathcal{M}(d)$ induce potential games as our next Theorem 5 states.

Theorem 5 *The non-negative function* Φ*, which is defined over assignments of players to machines as* $\Phi(N) = \sum_j \Lambda_j(N_j)$*, is a potential function for the game induced by any coordination mechanism in* $\mathcal{M}(d)$*.*

Proof Consider two assignments N and N' that differ in the assignment of a single player u. Assume that player u is assigned to machine j_1 and j_2 in the assignments N and N', respectively. Using the definition of function Φ and Lemma 4, we have

$$\Phi(N) - \Phi(N') = \sum_j \Lambda_j(N_j) - \sum_j \Lambda_j(N'_j)$$
$$= \Lambda_{j_1}(N_{j_1}) + \Lambda_{j_2}(N_{j_2}) - \Lambda_{j_1}(N'_{j_1}) - \Lambda_{j_2}(N'_{j_2})$$
$$= \Lambda_{u,j_1}(N_{j_1}) + \Lambda_{j_1}(N_{j_1} \setminus \{u\}) + \Lambda_{j_2}(N_{j_2})$$
$$\quad - \Lambda_{j_1}(N'_{j_1}) - \Lambda_{u,j_2}(N'_{j_2}) - \Lambda_{j_2}(N'_{j_2} \setminus \{u\}).$$

Now observe that $N_{j_1} \setminus \{u\} = N'_{j_1}$ and $N'_{j_2} \setminus \{u\} = N_{j_2}$. Hence, using this observation and the definition of the completion time for u in assignments N and N', the above derivation becomes

$$\Phi(N) - \Phi(N') = \Lambda_{u,j_1}(N_{j_1}) - \Lambda_{u,j_2}(N'_{j_2})$$
$$= w_u \left(\mathcal{P}(u, N_{j_1})^d - \mathcal{P}(u, N'_{j_2})^d \right),$$

which implies that the difference in the potentials and the difference $\mathcal{P}(u, N_{j_1}) - \mathcal{P}(u, N'_{j_2})$ in the completion time of the deviating player u in the two assignments have the same sign as desired.   □

Additional desirable properties for coordination mechanisms in class $\mathcal{M}(d)$, such as feasibility and low price of anarchy/stability, strongly depend on the coefficient function γ. The next two sections are devoted to proving such properties specifically for mechanism DCOORD.

4 The Coordination Mechanism DCOORD

Like CCOORD, our new coordination mechanism DCOORD belongs to class $\mathcal{M}(d)$. It uses the coefficient function defined as

$$\gamma(\{t_1, t_2, ..., t_\ell\}) = \begin{cases} 1 & \text{if } \exists i \text{ such that } t_i = d + 1 \\ \frac{d! d}{t_1! t_2! ... t_\ell!} & \text{otherwise} \end{cases}$$

for every multi-set of integers $\{t_1, t_2, ..., t_\ell\}$ such that $t_1 + t_2 + ... + t_\ell = d + 1$.

Observe that $\gamma(\{t_1, t_2, ..., t_\ell\})$ is very similar (but not identical) to the multinomial coefficient defined as $\binom{d+1}{t_1, t_2, ..., t_\ell} = \frac{(d+1)!}{t_1!...t_\ell!}$. This is exploited in the proof of the next statement.

Lemma 6 *Consider a machine j and a set of players $U = \{u_1, u_2, ..., u_\ell\}$. Then,*

$$\Lambda_j(U) = \frac{d}{d+1} L_j(U)^{d+1} + \frac{1}{d+1} \sum_{u \in U} w_{u,j}^{d+1}.$$

Proof By the definition of $\Lambda_j(U)$, $L_j(U)$, and the coefficient function γ, we have

$$\Lambda_j(U) = \sum_{t_1+t_2+...+t_\ell=d+1} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell w_{u_k,j}^{t_k}}$$

$$= \frac{d}{d+1} \sum_{t_1+t_2+...+t_\ell=d+1} \binom{d+1}{t_1, t_2, ..., t_\ell} \prod_{k=1}^{\ell w_{u_k,j}^{t_k}} + \frac{1}{d+1} \sum_{u \in U} w_{u,j}^{d+1}$$

$$= \frac{d}{d+1} L_j(U)^{d+1} + \frac{1}{d+1} \sum_{u \in U} w_{u,j}^{d+1}$$

as desired.   □

We proceed with two properties which relate Λ-functions to machine loads. The first one follows as a trivial corollary of Lemma 6 after observing that $\sum_{u \in U} w_{u,j}^{d+1} \leq L_j(U)^{d+1}$.

Corollary 7 *Consider a machine j and a set of players U. Then,*

$$\frac{d}{d+1} L_j(U)^{d+1} \leq \Lambda_j(U) \leq L_j(U)^{d+1}.$$

The second one will be very useful in proving that DCOORD is feasible and in bounding its price of anarchy.

Lemma 8 *Let $U = \{u_1, ..., u_\ell\}$ be a set of players. For every player $u_i \in U$ and every machine j, it holds that*

$$w_{u_i,j} L_j(U)^d \leq \Lambda_{u_i,j}(U) \leq d \cdot w_{u_i,j} L_j(U)^d.$$

Proof Without loss of generality, let us assume that $i = 1$. We will first expand the quantities $\Lambda_{u_1,j}(U)$ and $w_{u_1,j} L_j(U)^d$. We have

$$\Lambda_{u_1,j}(U) = \sum_{t_1+t_2+...+t_\ell=d+1|t_1\geq 1} \gamma(\{t_1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell w_{u_k,j}^{t_k}}$$

$$= w_{u_1,j} \cdot \sum_{t_1+t_2+...+t_\ell=d} \gamma(\{t_1 + 1, t_2, ..., t_\ell\}) \prod_{k=1}^{\ell w_{u_k,j}^{t_k}} \qquad (4)$$

and

$$w_{u_1,j} \cdot L_j(U)^d = w_{u_1,j} \cdot \left(\sum_{k=1}^{\ell w_{u_k,j}} \right)^d$$

$$= w_{u_1,j} \cdot \sum_{t_1+t_2+...+t_\ell=d} \binom{d}{t_1, t_2, ..., t_\ell} \prod_{k=1}^{\ell w_{u_k,j}^{t_k}}. \qquad (5)$$

We can prove the two desired inequalities by comparing the corresponding coefficients of each monomial in (4) and (5). Recall that, when $t_1 + t_2 + ... + t_\ell = d$, the coefficient $\gamma(\{t_1 + 1, t_2, ..., t_\ell\})$ from (4) is equal to 1 when $t_1 = d$. In this case, the corresponding coefficient in (5) is $\binom{d}{d,0,...,0} = 1$ as well. Otherwise,

$$\gamma(\{t_1 + 1, t_2, ..., t_\ell\}) = \frac{d}{t_1 + 1} \binom{d}{t_1, t_2, ..., t_\ell}.$$

Since t_1 is non-negative and at most $d - 1$, we have that

$$\binom{d}{t_1, t_2, ..., t_\ell} \leq \gamma(\{t_1 + 1, t_2, ..., t_\ell\}) \leq d \cdot \binom{d}{t_1, t_2, ..., t_\ell},$$

which, together with (4) and (5), concludes the proof. □

Feasibility of DCOORD follows easily now.

Theorem 9 *DCOORD produces feasible schedules.*

Proof Consider player u_1 and any assignment N which assigns it to machine j together with $\ell - 1$ other players $u_2, u_3, ..., u_\ell$. By the leftmost inequality of Lemma 8, we have that

$$\mathcal{P}(u_1, N_j) = \left(\frac{\Lambda_{u_1,j}(N_j)}{w_{u_1}} \right)^{1/d} \geq \left(\frac{w_{u_1,j}}{w_{u_1}} \right)^{1/d} L_j(N_j) \geq L_j(N_j),$$

as desired. The inequality holds since, by definition, $w_{u_1, j} \geq w_{u_1}$. $\qquad\square$

5 Bounding the Price of Anarchy and Stability

For proving the price of anarchy bound, we will need the following lemma which relates the load of any machine at an equilibrium with the optimal makespan.

Lemma 10 *Let N be an equilibrium and N^* an assignment of optimal makespan. Then, for every machine j, it holds that*

$$L_j(N_j) \leq m^{\frac{1}{d+1}} \frac{d+1}{\ln 2} \|L(N^*)\|_\infty.$$

Proof Consider a player u that is assigned to machine j in the equilibrium assignment N and to machine j' in the assignment $\widetilde{N}$ that minimizes the $(d+1)$-norm of the machine loads. First, consider the case where $j \neq j'$. In the equilibrium assignment N, player u has no incentive to deviate from machine j to machine j' and, hence, $\mathcal{P}(u, N_j) \leq \mathcal{P}(u, N_{j'} \cup \{u\})$. By the definition of DCOORD, we obtain that $\Lambda_{u,j}(N_j) \leq \Lambda_{u,j'}(N_{j'} \cup \{u\})$. Using this observation, Lemma 4, and Lemma 6, we get

$$
\begin{aligned}
\Lambda_{u,j}(N_j) &\leq \Lambda_{u,j'}(N_{j'} \cup \{u\}) = \Lambda_{j'}(N_{j'} \cup \{u\}) - \Lambda_{j'}(N_{j'}) \\
&= \frac{d}{d+1} L_{j'}(N_{j'} \cup \{u\})^{d+1} - \frac{d}{d+1} L_{j'}(N_{j'})^{d+1} + \frac{1}{d+1} w_{u,j'}^{d+1} \\
&= \frac{d}{d+1}(L_{j'}(N_{j'}) + w_{u,j'})^{d+1} - \frac{d}{d+1} L_{j'}(N_{j'})^{d+1} + \frac{1}{d+1} w_{u,j'}^{d+1}
\end{aligned}
$$

We will now prove that the same inequality holds when $j = j'$. In this case, together with Lemmas 4 and 6, we need to use a different argument that exploits a convexity property of polynomials. We have

$$
\begin{aligned}
\Lambda_{u,j}(N_j) &= \Lambda_{u,j'}(N_{j'}) = \Lambda_{j'}(N_{j'}) - \Lambda_{j'}(N_{j'} \setminus \{u\}) \\
&= \frac{d}{d+1} L_{j'}(N_{j'})^{d+1} - \frac{d}{d+1} L_{j'}(N_{j'} \setminus \{u\})^{d+1} + \frac{1}{d+1} w_{u,j'}^{d+1} \\
&= \frac{d}{d+1} L_{j'}(N_{j'})^{d+1} - \frac{d}{d+1}(L_{j'}(N_{j'}) - w_{u,j'})^{d+1} + \frac{1}{d+1} w_{u,j'}^{d+1} \\
&\leq \frac{d}{d+1}(L_{j'}(N_{j'}) + w_{u,j'})^{d+1} - \frac{d}{d+1} L_{j'}(N_{j'})^{d+1} + \frac{1}{d+1} w_{u,j'}^{d+1}.
\end{aligned}
$$

The last inequality follows since $z^{d+1} - (z-\alpha)^{d+1} \leq (z+\alpha)^{d+1} - z^{d+1}$ for every $z \geq \alpha \geq 0$, due to the convexity of the polynomial function z^{d+1}. Notice that we have used $z = L_{j'}(N_{j'})$ and $\alpha = w_{u,j'}$.

Let us sum the above inequality over all players. We obtain

$$\sum_j \sum_{u \in N_j} \Lambda_{u,j}(N_j)$$

$$\leq \frac{d}{d+1} \sum_j \sum_{u \in \widetilde{N}_j} \left(\left(L_j(N_j) + w_{u,j} \right)^{d+1} - L_j(N_j)^{d+1} \right) + \frac{1}{d+1} \sum_j \sum_{u \in \widetilde{N}_j} w_{u,j}^{d+1}$$

$$\leq \frac{d}{d+1} \sum_j \left(\left(L_j(N_j) + \sum_{u \in \widetilde{N}_j} w_{u,j} \right)^{d+1} - L_j(N_j)^{d+1} \right) + \frac{1}{d+1} \sum_j L_j(\widetilde{N}_j)^{d+1}$$

$$= \frac{d}{d+1} \|L(N) + L(\widetilde{N})\|_{d+1}^{d+1} - \frac{d}{d+1} \|L(N)\|_{d+1}^{d+1} + \frac{1}{d+1} \|L(\widetilde{N})\|_{d+1}^{d+1}$$

$$\leq \frac{d}{d+1} \left(\|L(N)\|_{d+1} + \|L(\widetilde{N})\|_{d+1} \right)^{d+1} - \frac{d-1}{d+1} \|L(N)\|_{d+1}^{d+1}. \qquad (6)$$

The second inequality follows by Lemma 3 and since $\sum_{u \in \widetilde{N}_j} w_{u,j}^{d+1} \leq L_j(\widetilde{N})^{d+1}$. The equality follows by the definition of $(d+1)$-norms and the last inequality follows by Minkowski inequality (Lemma 2) and by the fact that $\|L(\widetilde{N})\|_{d+1} \leq \|L(N)\|_{d+1}$.

Using the definition of norms and Lemma 8, we also have

$$\|L(N)\|_{d+1}^{d+1} = \sum_j L_j(N_j)^{d+1} = \sum_j \sum_{u \in N_j} w_{u,j} L_j(N_j)^d \leq \sum_j \sum_{u \in N_j} \Lambda_{u,j}. \qquad (7)$$

By combining (6) and (7), we have

$$2\|L(N)\|_{d+1}^{d+1} \leq \left(\|L(N)\|_{d+1} + \|L(\widetilde{N})\|_{d+1} \right)^{d+1}$$

and, equivalently,

$$\|L(N)\|_{d+1} \leq \frac{1}{2^{\frac{1}{d+1}} - 1} \|L(\widetilde{N})\|_{d+1} \leq \frac{d+1}{\ln 2} \|L(N^*)\|_{d+1}$$

$$\leq m^{\frac{1}{d+1}} \frac{d+1}{\ln 2} \|L(N^*)\|_{\infty}.$$

The second inequality follows since, by definition, $\|L(\widetilde{N})\|_{d+1} \leq \|L(N^*)\|_{d+1}$ and by the inequality $e^z \geq z + 1$. The third inequality follows by Lemma 1. Since $\|L(N)\|_{d+1} \geq L_j(N_j)$ for every machine j, the lemma follows. □

For the price of stability bound, we will use a qualitatively similar (to Lemma 10) relation between machine loads at a particular equilibrium and the optimal makespan.

Lemma 11 *Let N be the equilibrium that minimizes the potential function and N^* an assignment of optimal makespan. Then, for every machine j, it holds that*

$$L_j(N_j) \leq \left(\frac{d+1}{d} m \right)^{\frac{1}{d+1}} \|L(N^*)\|_{\infty}.$$

Proof Observe that $\Phi(N) \leq \Phi(N^*)$ since every equilibrium that is reached when players repeatedly best-respond starting from assignment N^* has potential at most

$\Phi(N^*)$. Using this observation, the definition of norms, Corollary 7, and the definition of the potential function (see the statement of Theorem 5), we have

$$
\begin{aligned}
\|L(N)\|_{d+1}^{d+1} = \sum_j L_j(N_j)^{d+1} &\leq \frac{d+1}{d} \sum_j \Lambda_j(N_j) = \frac{d+1}{d} \Phi(N) \\
&\leq \frac{d+1}{d} \Phi(N^*) = \frac{d+1}{d} \sum_j \Lambda_j(N_j^*) \leq \frac{d+1}{d} \sum_j L_j(N_j^*)^{d+1} \\
&= \frac{d+1}{d} \|L(N^*)\|_{d+1}^{d+1}.
\end{aligned}
$$

Hence, for every machine j, by exploiting Lemma 1, we have

$$
L_j(N_j) \leq \|L(N)\|_{d+1} \leq \left(\frac{d+1}{d}\right)^{\frac{1}{d+1}} \|L(N^*)\|_{d+1} \leq \left(\frac{d+1}{d} m\right)^{\frac{1}{d+1}} \|L(N^*)\|_\infty
$$

as desired. $\square$

We are now ready to complete the proofs of the price of anarchy and stability bounds for DCOORD. We will do so by comparing the completion time of any player to the optimal makespan $\|L(N^*)\|_\infty$.

Theorem 12 *By setting $d = \mathcal{O}(\log m)$, DCOORD has price of anarchy $\mathcal{O}(\log m)$ and price of stability $\mathcal{O}(1)$.*

Proof Consider a player u that is assigned to machine j at some equilibrium N and satisfies $w_u = w_{u,j^*}$ for some machine j^*. We use the fact that player u (is either already at or) has no incentive to deviate to machine j^* at equilibrium. In this way, we can bound its completion time as follows:

$$
\begin{aligned}
\mathcal{P}(u, N_j) \leq \mathcal{P}(u, N_{j^*} \cup \{u\}) &= \left(\frac{\Lambda_{u,j^*}(N_{j^*} \cup \{u\})}{w_u}\right)^{1/d} \\
&\leq \left(\frac{d w_{u,j^*} L_{j^*}(N_{j^*} \cup \{u\})^d}{w_u}\right)^{1/d} \leq d^{1/d}(L_{j^*}(N_{j^*}) + w_u).
\end{aligned}
$$

The equality follows by the definition of DCOORD, and the second inequality follows by Lemma 8. The third inequality follows since $w_u = w_{u,j^*}$ and by observing that $L_{j^*}(N_{j^*} \cup \{u\}) = L(N_{j^*}) + w_u$ if $u \notin N_{j^*}$ (i.e., $j \neq j^*$) and $L_{j^*}(N_{j^*} \cup \{u\}) = L(N_{j^*})$ otherwise.

Now, using Lemma 10 to bound $L_{j^*}(N_{j^*})$, we obtain that

$$
\mathcal{P}(u, N_j) \leq d^{1/d} \left(m^{\frac{1}{d+1}} \frac{d+1}{\ln 2} + 1\right) \|L(N^*)\|_\infty.
$$

If the equilibrium N is a potential-minimizing assignment, Lemma 11 can be further used to obtain the better guarantee

$$
\mathcal{P}(u, N_j) \leq d^{1/d} \left(\left(\frac{d+1}{d} m\right)^{\frac{1}{d+1}} + 1\right) \|L(N^*)\|_\infty.
$$

The theorem follows since, by setting $d = \Theta(\log m)$, the factors (ignoring $\|L(N^*)\|_\infty$ in the rightmost expressions become $\mathcal{O}(\log m)$ and $\mathcal{O}(1)$, respectively. So, in general, we have that the completion time of any player at equilibrium is at most $\mathcal{O}(\log m)$ times the optimal makespan (hence, the price of anarchy bound) while there exists a particular equilibrium where the completion time of any player is at most $\mathcal{O}(1)$ times the optimal makespan (hence, the price of stability bound). □

References

1. Abed, F., Correa, J.R., Huang, C.-C.: Optimal coordination mechanisms for multi-job scheduling games. In: Proceedings of the 22nd European Symposium on Algorithms (ESA), LNCS 8737, Springer, pages 13–24 (2014)
2. Abed, F., Huang, C.: Preemptive coordination mechanisms for unrelated machines. In: Proceedings of the 20th Annual European Symposium on Algorithms (ESA), LNCS 7501, Springer, pages 12–23 (2012)
3. Awerbuch, B., Azar, Y., Grove, E.F., Kao, M.-Y., Krishnan, P., Vitter, J.S.: Load balancing in the L_p norm. In: Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 383-391 (1995)
4. Azar, Y., Fleischer, L., Jain, K., Mirrokni, V.S., Svitkina, Z.: Optimal coordination mechanisms for unrelated machine scheduling. Oper. Res. **63**(3), 489–500 (2015)
5. Bhattacharya, S., Im, S., Kulkarni, J., Munagala, K.: Coordination mechanisms from (almost) all scheduling policies. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS), pages 121–134 (2014)
6. Caragiannis, I.: Better bounds for online load balancing on unrelated machines. In: Proceedings of the 19th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA), pages 972-981 (2008)
7. Caragiannis, I.: Efficient coordination mechanisms for unrelated machine scheduling. Algorithmica **66**, 512–540 (2013)
8. Caragiannis, I., Gkatzelis, V., Vinci, C.: Coordination mechanisms, cost-sharing, and approximation algorithms for scheduling. In: Proceedings of the 13th International Conference on Web and Internet Economics (WINE), LNCS 10660, Springer, pages 74–87 (2017)
9. Christodoulou, G., Koutsoupias, E., Nanavati, A.: Coordination mechanisms. Theor. Comput. Sci. **410**(36), 3327–3336 (2009)
10. Cohen, J., Dürr, C., Thang, N.K.: Smooth inequalities and equilibrium inefficiency in scheduling games. In: Proceedings of the 8th International Workshop on Internet and Network Economics (WINE), LNCS 7695, Springer, pages 350–363 (2012)
11. Cole, R., Correa, J.R., Gkatzelis, V., Mirrokni, V.S., Olver, N.: Decentralized utilitarian mechanisms for scheduling games. Games and Econ. Behav. **92**, 306–326 (2015)
12. Immorlica, N., Li, L., Mirrokni, V.S., Schulz, A.S.: Coordination mechanisms for selfish scheduling. Theor. Comput. Sci. **410**(17), 1589–1598 (2009)