# Coordinating Oligopolistic Players in Unrelated Machine Scheduling

Fidaa Abed and Chien-Chung Huang

Max-Planck-Institut für Informatik, Saarbrücken, Germany.
{fabed,villars}@mpi-inf.mpg.de

**Abstract.** We consider the following machine scheduling game. Jobs, controlled by selfish players, are to be assigned to unrelated machines. A player cares only about the finishing time of his job(s), while disregarding the welfare of other players. The outcome of such games is measured by the makespan. Our goal is to design coordination mechanisms to schedule the jobs so as to minimize the price of anarchy.
We introduce *oligopolistic players*. Each such player controls a set of jobs, with the aim of minimizing the sum of the completion times of his jobs. Our model of oligopolistic players is a natural generalization of the conventional model, where each player controls only a single job.
In our setting, previous mechanisms designed for players with single jobs are inadequate, e.g., having large price of anarchy, or not guaranteeing pure Nash equilibria. To meet this challenge, we design three mechanisms that are adapted/generalized from Caragiannis' ACOORD. All our mechanisms induce pure Nash equilibria while guaranteeing relatively small price of anarchy.

**Keywords:** Unrelated Machine Scheduling, Coordination Mechanisms, Price of Anarchy.

## 1 Introduction

We consider the game-theoretic version of the following machine scheduling problem. A set $\mathcal{I}$ of jobs and a set $\mathcal{M}$ of machines are given. A job $i \in \mathcal{I}$ has size $p_{ij}$ on machine $j \in \mathcal{M}$. Jobs are to be assigned to machines and the goal is to minimize the makespan. This is the classical *unrelated machine scheduling* problem with the makespan objective [21].

A natural question from the angle of game theory is what happens if the jobs are controlled by selfish players. The strategy space of a player is the set of machines. A player cares only about the finishing time of his job while disregarding the welfare of other players. In such games, researchers especially focus on the *Nash equilibrium* [22], a stable situation where no player can unilaterally change his strategy to *strictly* improve the finishing time of his job. In case that all players use pure strategies only, the Nash equilibrium resulted is called the pure Nash equilibrium (PNE). In this paper, we consider only pure strategies and PNEs.

A central question in algorithm game theory is the quality of equilibria. In particular, people analyze the *price of anarchy* (PoA) [20], which in our context is defined as the worst ratio between the makespan of a PNE against that of the optimal solution.

The PoA is obviously determined by the rules of the game. Here the "rules" mean the scheduling policies of the machines. In the literature, the rules are formally called the *coordination mechanisms* [8]. Ideally, we would like to have good coordination mechanisms so as to minimize the PoA.

The design space of coordination mechanism depends on a number of parameters, e.g., whether preemption is allowed, whether jobs have unique IDs and so on. However, the following two conditions on coordination mechanisms are (implicitly) assumed by all previous works (and the current one).

1. **Physical Feasibility**. Suppose that a set of jobs $\mathcal{I}' \subseteq \mathcal{I}$ are assigned to machine $j$. At any point of time $t$, if a subset of jobs $\mathcal{I}'' \subseteq \mathcal{I}'$ are finished by machine $j$, then $\sum_{i \in \mathcal{I}''} p_{ij} \leq t$.
2. **Locality of Scheduling Decision**. A machine decides its schedule based only on the information of the incoming jobs, while, where the other jobs go to, and how the other machines schedule them is irrelevant.

The first condition is self-evident; the second condition is motivated by the fact that in a fluid environment, such as the Internet, a machine may not be able to coordinate with other machines in a timely manner. Azar, Jain, and Mirrokni [5] differentiate two classes of mechanisms: a mechanism is *local* if a machine schedules its job based only on the information of the incoming jobs (but notice that a machine is allowed to look at the sizes of its jobs on other machines); a mechanism is *strongly local* if a machine schedules its jobs only based on the sizes of the incoming job on it. It is known that the PoAs of strongly local mechanisms and of local mechanisms can be significantly different when preemption is disallowed [5, 15].

**Oligopolistic Players**. All previous works assume that a player controls a single job. A natural and more realistic extension is to assume a player can control multiple jobs and we refer to such players as *oligopolistic* players. A question that arises in our model is: what would be the local objective of an oligopolistic player? This is a non-issue when a player controls a single job. However, when he has multiple jobs, several objectives are possible. For instance, it could be his makespan (the latest finishing time of his jobs), or it could be the sum of completion times of his jobs.

In this work, we assume that each player aims to minimize the sum of the completion times of his jobs. This assumption is motivated by the observation that a player would care about the *collective welfare* of his jobs. If moving a job from one machine to another machine decreases the finishing time of that job, the controlling player would have incentive to do so—even if the latest finishing time of his jobs is not really decreased.

To evaluate the overall system performance, there can be two natural candidates: makespan (the latest finishing time of a job), or the weighted completion times of the jobs (jobs are given weights and the cost is computed as the weighted sum of their completion times.) In a companion paper of this work [1], we use the weighted completion times of the jobs to measure the system performance. In this work, we instead consider the makespan.

In general, in terms of PoA, it is harder to design mechanisms when the global objective is the makespan than when it is the weighted sum of completion times of all jobs. In the original model where each player controls a single job, with weighted sum global objective, Cole et al. [11] show several mechanisms achieving constant PoA; on the other hand, when the global objective is the makespan, it is known [2, 5, 15] that constant PoA is impossible. As our model is a generalization of the single-job player model, we also cannot hope to achieve constant PoA.

We observe that previous mechanisms designed for players with single jobs are inadequate in our model of oligopolistic players. In some cases (ACOORD/BCOORD/CCOORD), the PoA becomes significantly worse; in some cases, they no longer guarantee PNEs (ShortestFirst/AJM-2/CCOORD/BALANCE). See Table 1 for a summary of the properties of the known mechanisms in our new model. Our challenge here is to design coordination mechanisms that simultaneously guarantee the existence of PNEs *and* still maintain small PoA.

### 1.1 Our Contribution

We propose three mechanisms, $A_1$-COORD, $A_2$-COORD, and $A_3$-COORD, which are presented in Sections 4-6. These mechanisms make use of preemption and assume that players and jobs are not anonymous, namely, each player and each job has a unique ID. When a job is assigned to a machine, the machine can make schedule decisions based on this job's ID and the ID of its owner.

Our mechanisms are adapted/generalized from Caragiannis' ACOORD (hence the naming). See Table 2 for a summary of their properties. All of them induce PNEs. Under $A_1$-COORD and $A_2$-COORD, such PNEs can be computed in

| Mechanisms | PoA | | PNE | |
|---|---|---|---|---|
| | $C = 1$ | $C > 1$ | $C = 1$ | $C > 1$ |
| ShortestFirst [19] | $\Theta(m)$ | $\Omega(m)$ | Yes | No$\star$ |
| LongestFirst [19] | Unbounded | Unbounded | No | No |
| Makespan [19] | Unbounded | Unbounded | Yes | No$\star$ |
| RANDOM [19] | $\Theta(m)$ | $\Omega(m)$ | No | No |
| EQUI [9] | $\Theta(m)$ | $\Omega(m)$ | Yes | Yes |
| AJM-1 [5] | $\Theta(\log m)$ | $\Omega(\log m)$ | No | No |
| AJM-2 [5] | $\Theta(\log^2 m)$ | $\Omega(\log^2 m)$ | Yes | No$\star$ |
| BALANCE [10] | $\Theta(\log m)$ | $\Omega(\log m)$ | Yes | No$\star$ |
| ACOORD [7] | $O(p \cdot m^{1/p})$ | $\Omega(C^{(1-\epsilon)(p+1)} m/p^2)\star$ | Yes | Yes |
| BCOORD [7] | $O(p \cdot m^{1/p}/\log p)$ | $\Omega(C^{(1-\epsilon)(p+1)} m/p^2)\star$ | No | No$\star$ |
| CCOORD [7] | $O(p^2 \cdot m^{1/p})$ | $\Omega(C^{(1-\epsilon)(p+1)} m/p^2)$ when $p = 1\star$ | Yes | No$\star$ |

**Table 1.** Summary of the properties of known mechanisms in our model. $m = |\mathcal{M}|$ is the number of machines, and $C$ is the largest number of jobs controlled by a player. The results marked by $\star$ are proved in the appendix. For the last three mechanisms, $p \geq 1$, and $\epsilon$ is some small constant where $\epsilon > 0$. If $p = \Theta(\log m)$ and $C = 1$, then the PoAs for ACOORD, BCOORD, and CCOORD are $\Theta(\log m)$, $\Theta(\frac{\log m}{\log \log m})$, and $O(\log^2 m)$ respectively.

polynomial time; furthermore, each player can compute his own optimal strategy in polynomial time.

| Mechanisms | PoA | PNE | Note |
|---|---|---|---|
| $A_1$-COORD | $O(C^{q+1} m^{\frac{1}{q+1}})$ for any $q > 0$ | Yes | Local |
| $A_2$-COORD | $O(C^{\frac{2q}{q+1}} m^{\frac{1}{q+1}})$ for any $0 < q \leq 1$, $1/q \in \mathbb{Z}$ | Yes | Local |
| $A_3$-COORD | $O(\min\{W\sqrt{m}, \min_{\gamma \in \mathbb{Z}_{\geq 1}}\{m^{\frac{\gamma+1}{2\gamma}} + W^\gamma\}\})$, $O(\log m + \log W)$ when $C = 1$ | Yes | Strongly local |

**Table 2.** Summary of our mechanisms. $m = |\mathcal{M}|$ is the number of machines, and $C$ is the largest number of jobs controlled by a player. In $A_3$-COORD, $W = \frac{\max_{i \in \mathcal{I}} \min_{j \in \mathcal{M}} p_{ij}}{\min_{i \in \mathcal{I}} \min_{j \in \mathcal{M}} p_{ij}}$. In $A_1$-COORD and $A_2$-COORD, PNEs can be computed in polynomial time.

In terms of PoA, our three mechanisms perform differently depending on the situation. Let $m = |\mathcal{M}|$ be the number of machines and $C$ be the largest number of jobs controlled by a player. $A_1$-COORD achieves the PoA of $O(C^{q+1} m^{\frac{1}{q+1}})$, for any chosen $q > 0$. $A_1$-COORD is better suited for the situation when $C$ is some bounded constant (in this case we can get a PoA of $O(m^\epsilon)$). When $C$ is relatively large, $A_2$-COORD is a better mechanism, with the PoA of $O(C^{\frac{2q}{q+1}} m^{\frac{1}{q+1}})$, for any chosen $q$, $0 < q \leq 1$ so that $1/q$ is an integer. For example, if $m$ is bounded by a constant and $C$ is very large, then we can get a PoA of $O(C^{\epsilon})$.

Our third mechanism, $A_3$-COORD, has the PoA independent of $C$ and, in some cases, is superior to the previous two. Let $W = \frac{\max_{i \in \mathcal{I}} \min_{j \in \mathcal{M}} p_{ij}}{\min_{i \in \mathcal{I}} \min_{j \in \mathcal{M}} p_{ij}}$, i.e., the largest ratio of sizes of two jobs when they are both assigned to the most efficient machines. Then $A_3$-COORD guarantees the PoA of $O(\min\{W\sqrt{m}, \min_{\gamma \in \mathbb{Z}_{\geq 1}}\{m^{\frac{\gamma+1}{2\gamma}} + W^{\gamma}\}\})$. Unlike the previous two mechanisms that are local, $A_3$-COORD is strongly local, thus more "frugal" in terms of the information it needs. Additionally, when $C = 1$, (i.e., the original model), $A_3$-COORD achieves the PoA of $O(\log m + \log W)$. Previously, Cohen, Dürr, and Thang [9] raised the question whether it is possible to design a strongly local mechanism that achieves the PoA of $O(polylog(m))$. Here we give a partial positive answer—as long as $W = O(m^{polylog(m)})$.

How our mechanisms guarantee PNEs is similar to the original ACOORD[1], using a simple idea: the finishing times of the jobs of the $k$-th player is dependent only on the strategies of the first $k-1$ players and the $k$-th player himself. This idea also ensures the game converges to PNEs in polynomial steps. The main technical challenge of this work is in the analysis of PoA. To prove that our mechanisms have the claimed PoAs, we introduce several non-trivial extensions of Caragiannis' ideas in the analysis of his ACOORD.

## 2 Related Work

The design of coordination mechanisms for machine scheduling has been intensively studied in recent years [2, 4, 5, 7, 9, 11, 13, 19]. All these works focus on the setting where a player controls a single job.

Our three mechanisms are adapted from ACOORD mechanism [7]. This mechanism uses a global ordering of the jobs according to their distinct IDs. The finishing time of a job is the total load of the jobs preceding it and itself, modified by a certain *inefficiency* parameter. The game induced by this mechanism is a potential game which guarantees the existence of a PNE. Moreover, the convergence to a PNE is fast. Our three mechanisms are generalized from ACOORD by fine tuning the inefficiency parameter.

Though so far not directly considered in the machine scheduling context, the notion of "oligopolistic players" has in fact been studied in different settings. For instance, in a version of selfish routing [23], an atomic player controls a splittable flow. Such a player can be regarded as an oligopolistic player. See [6, 12] and the references therein for an overview of such games. Another example of an oligopolistic player is a coalition of players. In [3, 14], a partition equilibrium, where the agents are partitioned into coalitions, and only deviations by the prescribed coalitions are considered, is proved to exist in resource selection games. In [16, 18], the authors assume that once a set of players form a coalition, they care only about their collective welfare while disregarding their own

[1] We note that when $c = 1$, $A_1$-COORD and $A_2$-COORD reduce to ACOORD.

outcomes (thus there is no backstabbing or double-crossing). A coalition, under such assumptions, is equivalent to an oligopolistic player.

Table 1 summarizes the performance of various known mechanisms in our setting. As mentioned before, the difficulty is to guarantee both the existence of PNEs and a small PoA. Only ACOORD and EQUI guarantee PNEs in our model. To ensure that ACOORD has a PNE, we just need to index jobs in such a way that all jobs belonging to the same player have consecutive indices. EQUI was originally designed to guarantee a *strong* Nash equilibrium, when players control single jobs. Interestingly, in our model, it can be shown that it still induces a potential game (thus guaranteeing PNEs). We leave it as an open question regarding its real PoA when $C > 1$.

## 3 Preliminary

We first introduce some necessary notations to facilitate our discussion. Throughout the paper, we use $N$ to denote an assignment and $O$ the optimal assignment. $N_j$(resp. $O_j$) is the set of jobs assigned to machine $j$ in $N$(resp. $O$). $L(N_j) = \sum_{i \in N_j} p_{ij}$ is the total load of jobs assigned to $j$ in assignment $N$. For each job $i$, let $p_{i,\min}$ be its smallest size, $p_{i,\min} = \min_{j \in \mathcal{M}} p_{ij}$, and $\phi_{ij}$ its *inefficiency* on machine $j$, defined as $\frac{p_{ij}}{p_{i,\min}}$. Note that only local mechanisms can make use of the inefficiencies of the jobs in scheduling, while strongly local mechanisms cannot. We assume that the set $\mathcal{P}$ of players are indexed consecutively, from 1, 2, $\cdots$, up to $|\mathcal{P}|$. Given job $i \in \mathcal{I}$, $\pi(i)$ denotes the player controlling it.

**Proposition 1.** [7] *For an assignment $N$ and any $p \geq 1$,*
$\max_{j \in \mathcal{M}} L(N_j) \leq (\sum_{j \in \mathcal{M}} L(N_j)^p)^{\frac{1}{p}} \leq m^{\frac{1}{p}} \max_{j \in \mathcal{M}} L(N_j)$.

**Proposition 2.** *Let $a_i, b_i \geq 0$, $p \geq 1$, and let $f(x)$ be a convex function.*

- *Minkowski's inequality:* $(\sum_{i=1}^{s} (a_i + b_i)^p)^{1/p} \leq (\sum_{i=1}^{s} a_i^p)^{1/p} + (\sum_{i=1}^{s} b_i^p)^{1/p}$.
- *Jensen's inequality:* $\sum_{i=1}^{s} f(a_i) \geq s f(\frac{\sum_{i=1}^{s} a_i}{s})$.

The next proposition is an easy consequence of Minkowski's inequality.

**Proposition 3.** *Let $p \geq 1$ be some integer. Then $a^{1/p} + b^{1/p} \geq (a+b)^{1/p}$.*

*Proof.* By re-writing $a^{1/p}$ as $((a^{1/p})^p)^{1/p}$ and $b^{1/p}$ as $((b^{1/p})^p)^{1/p}$, we can apply Minkowski's inequality

$$((a^{1/p})^p)^{1/p} + ((b^{1/p})^p)^{1/p} \geq ((a^{1/p} + b^{1/p})^p)^{1/p}$$
$$= [a + b + \sum_{t=1}^{p-1} \binom{p}{t} (a^{1/p})^t (b^{1/p})^{p-t}]^{1/p} \geq (a+b)^{1/p}.$$

□

**Proposition 4.** [7] *Suppose that* $p \geq 1$, $A \geq 0$, *and* $B_i \geq 0$ *for* $i = 1, \cdots, s$. *Then* $\sum_{i=1}^{s}((A + B_i)^p - A^p) \leq (A + \sum_{i=1}^{s} B_i)^p - A^p$.

The following proposition is slightly modified from Caragiannis [7]. Specifically, we replace the constrain on the exponent $p \geq 1$ with $p \geq 0$ so as to design a larger set of mechanisms. The proof is entirely the same as in [7].

**Proposition 5.** [7] *For any* $z_0 \geq 0$, $\alpha \geq 0$, *and* $p \geq 0$, *the following holds.*

$$(p+1)\alpha z_0^p \leq (z_0 + \alpha)^{p+1} - z_0^{p+1} \leq (p+1)\alpha(z_0 + \alpha)^p.$$

*Proof.* If $\alpha = 0$, the lemma holds trivially. If $\alpha > 0$, the lemma holds by observing that the function $z^{p+1}$ is convex for any $p \geq 0$. Therefore, the slope of the line crossing $(z_0, z_0^{p+1})$ and $(z_0 + \alpha, (z_0 + \alpha)^{p+1})$ is between its derivatives at points $z_0$ and $z_0 + \alpha$. □

**Proposition 6.** *Let* $A \geq 0$, $B_i \geq 0$ *for* $1 \leq i \leq s$, *and* $p \geq 0$. *Then*

$$(A + \sum_{i=1}^{s} B_i)^{1+p} - A^{1+p} \leq (1+p)s^p \sum_{i=1}^{s} B_i(A + B_i)^p.$$

*Proof.* Observe that the function $g(x) = x(A + x)^p$ is convex when $x \geq 0$, since $g''(x) = p(A + x)^{p-2}(2A + xp + x) \geq 0$. Therefore,

$$\sum_{i=1}^{s} B_i(A + B_i)^p = \sum_{i=1}^{s} g(B_i) \geq sg(\frac{\sum_{i=1}^{s} B_i}{s}) = \sum_{i=1}^{s} B_i(A + \frac{\sum_{t=1}^{s} B_t}{s})^p,$$

where the inequality follows from the convexity of $g(x)$ and Jensen's inequality.

Using the above inequality, we have

$$\begin{aligned}
&(1+p)s^p \sum_{i=1}^{s} B_i(A + B_i)^p \\
\geq\ &(1+p)s^p \sum_{i=1}^{s} B_i(A + \frac{\sum_{t=1}^{s} B_t}{s})^p \\
=\ &(1+p) \sum_{i=1}^{s} B_i(As + \sum_{t=1}^{s} B_t)^p \\
\geq\ &(1+p) \sum_{i=1}^{s} B_i(A + \sum_{t=1}^{s} B_t)^p \\
=\ &(1+p)(\sum_{i=1}^{s} B_i)(A + \sum_{i=1}^{s} B_i)^p \\
\geq\ &(A + \sum_{i=1}^{s} B_i)^{1+p} - A^{1+p},
\end{aligned}$$

where the last inequality follows from Proposition 5 by setting $\sum_{i=1}^{s} B_i = \alpha$, and $z_0 = A$. The proof follows. $\square$

## 4 $A_1$-COORD

In this and the next section, let $N_j^k$ denote the set of jobs assigned to $j$ belonging to the first $k$ players, for any $k \in \mathcal{P}$. Observe that $N_j^{|\mathcal{P}|} = N_j$ and $N_j^0 = \emptyset$. Finally, let $L(N_j^k) = \sum_{i:\pi(i) \leq k, i \in N_j} p_{ij}$, the total load of jobs belonging to the first $k$ players on machine $j$ in $N$.

> $A_1$-COORD: Let $N$ be the assignment. Suppose that job $i \in N_j$. Then the completion of job $i$ is set as $P(i, N_j) = C(\phi_{ij})^{\frac{1}{q}}(L(N_j^{\pi(i)-1}) + p_{ij})$, for some $q > 0$.

As in the original ACOORD, the term $(\phi_{ij})^{\frac{1}{q}}$ is used to encourage a player to assign his job to a more efficient machine. The term $L(N_j^{\pi(i)-1})$ is the total load of jobs belonging to the first $\pi(i)-1$ players on machine $j$. So a job's completion time is unaffected by jobs belonging to players with indices larger than $\pi(i)$. This property will be used when we argue that $A_1$-COORD has a PNE.

The important idea behind our mechanism is that the jobs belonging to the same player, even if they are assigned to the same machine $j$, *would have their completion times independent of each other*. This follows from the simple observation that the sum $L(N_j^{\pi(i)-1}) + p_{ij}$ does not include other jobs belonging to the player $\pi(i)$. This property is a key part in our analysis of PoA; also it allows each player to compute his own optimal strategy in polynomial time (see Theorem 1). Finally, the multiplicative factor $C$ is introduced to make sure that $A_1$-COORD produces feasible schedules.

**Lemma 1.** *The schedule decided by $A_1$-COORD is feasible.*

*Proof.* Recall that to prove a schedule is feasible, we need to show that at time $f$, if a set $\mathcal{I}' \subseteq N_j$ of jobs are finished, then $\sum_{i \in \mathcal{I}'} p_{ij} \leq f$. It is easy to see that we only need to consider those times $f$ where some job $i \in N_j$ are finished.

Now suppose that in assignment $N$ player $k$ puts jobs $i_1$, $i_2$, $\cdots$, $i_{x \leq C}$ on machine $j$, where the jobs are indexed by their non-decreasing completion times. We argue that the completion time $P(i_y, N_j)$ of job $i_y$, $y \leq x$, is at least as large as the total load of jobs finishing no later than $i_y$. In the case that multiple jobs among $i_1$, $i_2$, $\cdots$, $i_{x \leq C}$ finish at the same time as $i_y$, w.l.o.g., we can assume that $i_y$ has the largest index. Consider two possibilities.

1. Suppose that none of the jobs belonging to players in $\mathcal{P} \backslash \{1, 2, \cdots, k\}$ finishes earlier than $i_y$. Since jobs $i_1, \cdots, i_{y-1}$ finish no later than $i_y$, $(\phi_{i_t j})^{1/q}(L(N_j^{k-1}) + p_{i_t j}) \leq (\phi_{i_y j})^{1/q}(L(N_j^{k-1}) + p_{i_y j})$, for $1 \leq t \leq y-1$. As a result,

$$P(i_y, N_j) = C(\phi_{i_y j})^{1/q}(L(N_j^{k-1}) + p_{i_y j}) \geq$$
$$\sum_{t=1}^{y}(\phi_{i_t j})^{1/q}(L(N_j^{k-1}) + p_{i_t j}) \geq L(N_j^{k-1}) + \sum_{t=1}^{y} p_{i_t j},$$

which is no less than the total load of jobs finishing as early as $i_y$.

2. Suppose that some players in $\mathcal{P}\backslash\{1, 2, \cdots, k\}$ have jobs on machine $j$ that finish earlier than $i_y$. Let player $k+s$ be such player with the largest index. Suppose further that player $k+s$ has jobs $i_1^*$, $i_2^*$, $\cdots$, $i_z^*$ finish as early as $i_y$ on machine $j$. W.L.O.G., assume that $i_1^*$ is the job with the largest size among the jobs $i_1^*$, $i_2^*$, $\cdots$, $i_z^*$. Then

$$P(i_y, N_j) \geq P(i_1^*, N_j) = C(\phi_{i_1^* j})^{1/q}(L(N_j^{k+s-1}) + p_{i_1^* j}) \geq$$
$$(\phi_{i_1^* j})^{1/q}(L(N_j^{k+s-1}) + \sum_{t=1}^{z} p_{i_t^* j}) \geq L(N_j^{k+s-1}) + \sum_{t=1}^{z} p_{i_t^* j},$$

which is no less than the total load of jobs finishing as early as $i_y$. The proof follows. □

**Lemma 2.** *Suppose that $N$ is a PNE under $A_1$-COORD. Then* $\max_{j\in\mathcal{M}, i\in N_j} P(i, N_j) \leq C[(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + \max_{j\in\mathcal{M}} L(O_j)]$.

*Proof.* Let $i^*$ be the job with the largest completion time in assignment $N$. Suppose that it is assigned to $j_1$ in assignment $N$ and has an inefficiency 1 on machine $j_2$ ($j_2$ could be the same as $j_1$). Either $j_2 \neq j_1$, then the player $\pi(i^*)$ controlling $i^*$ has no incentive to move $i^*$ to $j_2$; or $j_2 = j_1$. In both cases, we have

$$P(i^*, N_{j_1}) \leq C(L(N_{j_2}^{\pi(i^*)-1}) + p_{i^* j_2}) \leq$$
$$C(L(N_{j_2}) + p_{i^* j_2}) \leq C(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + C \max_{j\in\mathcal{M}} L(O_j),$$

where the last inequality follows from Proposition 1 and the fact that in the optimal, one machine would have load at least $p_{i^* j_2}$. The proof follows. □

**Lemma 3.** *Suppose that $N$ is a PNE under $A_1$-COORD. Then*

$$(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} \leq 4(q+1)C^q m^{\frac{1}{q+1}} \max_{j\in\mathcal{M}} L(O_j).$$

*Proof.* Suppose that job $i$ is assigned to machine $j_1$ in assignment $N$ and machine $j_2$ in the optimal assignment $O$. As $N$ is a PNE, $C(\phi_{ij_1})^{1/q}(L(N_{j1}^{\pi(i)-1}) + p_{ij_1}) \leq$

$C(\phi_{ij_2})^{1/q}(L(N_{j2}^{\pi(i)-1}) + p_{ij_2})$. Canceling $C$, raising both sides to the power of $q$, and multiplying them by $p_{i,\min}$, we have

$$p_{ij_1}(L(N_{j_1}^{\pi(i)-1}) + p_{ij_1})^q \leq p_{ij_2}(L(N_{j_2}^{\pi(i)-1}) + p_{ij_2})^q \tag{1}$$

Define $x_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in assignment $N$; similarly define $y_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in the optimal $O$. Then the above inequality can be re-written as

$$\begin{aligned}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{\pi(i)-1}) + x_{ij}p_{ij})^q &\leq \sum_{j\in\mathcal{M}} y_{ij}p_{ij}(L(N_j^{\pi(i)-1}) + y_{ij}p_{ij})^q \\ &\leq \sum_{j\in\mathcal{M}} y_{ij}p_{ij}(L(N_j) + y_{ij}p_{ij})^q.\end{aligned}$$

Summing the above inequality over all jobs $i \in \mathcal{I}$, we have

$$\begin{aligned}&(q+1)\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{\pi(i)-1}) + x_{ij}p_{ij})^q \\ \leq &\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} (q+1)y_{ij}p_{ij}(L(N_j) + y_{ij}p_{ij})^q \\ \leq &\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} [L(N_j) + 2y_{ij}p_{ij})^{q+1} - (L(N_j) + y_{ij}p_{ij}]^{q+1} \\ \leq &\sum_{j\in\mathcal{M}}\sum_{i\in\mathcal{I}}[L(N_j) + 2y_{ij}p_{ij})^{q+1} - (L(N_j)]^{q+1} \\ \leq &\sum_{j\in\mathcal{M}} [L(N_j) + 2\sum_{i\in\mathcal{I}} y_{ij}p_{ij})^{q+1} - (L(N_j)]^{q+1} \\ \leq &[(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + 2(\sum_{j\in\mathcal{M}} L(O_j)^{q+1})^{\frac{1}{q+1}}]^{q+1} - \sum_{j\in\mathcal{M}} L(N_j)^{q+1}, \end{aligned} \tag{2}$$

where the second inequality follows from Proposition 5 by setting $\alpha = y_{ij}p_{ij}$ and $z_0 = L(N_j) + y_{ij}p_{ij}$, and $p = q$; the fourth inequality from Proposition 4 by setting $A = L(N_j)$, $B_i = 2y_{ij}p_{ij}$ and $p = 1 + q$; and the fifth inequality from Minkowski's inequality.

We next bound $\sum_{j\in\mathcal{M}} L(N_j)^{q+1}$ by writing it as a telescopic sum:

$$
\begin{aligned}
\sum_{j\in\mathcal{M}} L(N_j)^{q+1} &= \sum_{j\in\mathcal{M}}\sum_{k=1}^{|\mathcal{P}|} L(N_j^k)^{q+1} - L(N_j^{k-1})^{q+1} \\
&= \sum_{j\in\mathcal{M}}\sum_{k=1}^{|\mathcal{P}|} [L(N_j^{k-1}) + \sum_{i:\pi(i)=k, i\in N_j} p_{ij}]^{q+1} - L(N_j^{k-1})^{q+1} \\
&\le \sum_{j\in\mathcal{M}}\sum_{k=1}^{|\mathcal{P}|} \sum_{i:\pi(i)=k, i\in N_j} (1+q)|\{i|\pi(i)=k, i\in N_j\}|^q p_{ij}(L(N_j^{k-1}) + p_{ij})^q \\
&\le \sum_{j\in\mathcal{M}}\sum_{k=1}^{|\mathcal{P}|} \sum_{i:\pi(i)=k, i\in N_j} (1+q)C^q p_{ij}(L(N_j^{k-1}) + p_{ij})^q \\
&= (1+q)C^q \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{\pi(i)-1}) + x_{ij}p_{ij})^q \qquad (3) \\
&\le C^q\{[(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + 2(\sum_{j\in\mathcal{M}} L(O_j)^{q+1})^{\frac{1}{q+1}}]^{q+1} - \sum_{j\in\mathcal{M}} L(N_j)^{q+1}\} \qquad (4)
\end{aligned}
$$

where the first inequality follows from Proposition 6 by setting $A = L(N_j^{k-1})$, $B_i = p_{ij}$ and $p = q$; the second inequality from the fact that a player has at most $C$ jobs on machine $j$ in assignment $N$; the third equality from a double counting argument; and the last inequality from (2).

Rearranging terms in Inequality (4), we have

$$
\begin{aligned}
(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} \le \frac{2(\sum_{j\in\mathcal{M}} L(O_j)^{q+1})^{\frac{1}{q+1}}}{(\frac{1}{C^q}+1)^{\frac{1}{q+1}} - 1} &= \frac{2(\sum_{j\in\mathcal{M}} L(O_j)^{q+1})^{\frac{1}{q+1}}}{((\frac{1}{C^q}+1)^{C^q})^{\frac{1}{C^q(q+1)}} - 1} \\
&\le \frac{2m^{\frac{1}{q+1}} \max_{j\in\mathcal{M}} L(O_j)}{((\frac{1}{C^q}+1)^{C^q})^{\frac{1}{C^q(q+1)}} - 1},
\end{aligned}
$$

where the second inequality follows from Proposition 1.

Observe that as $C \ge 1$, $C^q \ge 1$. Then by calculus, $\sqrt{e} < 2 \le (\frac{1}{C^q}+1)^{C^q} \le e$. Thus, the term $(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}}$ can be further upper-bounded as follows,

$$
\begin{aligned}
(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} \le \frac{2m^{\frac{1}{q+1}} \max_{j\in\mathcal{M}} L(O_j)}{((\frac{1}{C^q}+1)^{C^q})^{\frac{1}{C^q(q+1)}} - 1} &< \frac{2m^{\frac{1}{q+1}} \max_{j\in\mathcal{M}} L(O_j)}{e^{\frac{1}{2C^q(q+1)}} - 1} \\
&\le 4(q+1)C^q m^{\frac{1}{q+1}} \max_{j\in\mathcal{M}} L(O_j),
\end{aligned}
$$

where the last inequality holds because of the well-known inequality that $e^z - 1 \ge z$. Hence the proof. □

**Theorem 1.** *$A_1$-COORD guarantees a PNE. Such a PNE can be computed in polynomial time and each player can compute his optimal strategy in polynomial time. Moreover, for any fixed $q > 0$, it guarantees that PoA of $O(C^{q+1}m^{\frac{1}{q+1}})$.*

*Proof.* For the first part, we can construct a PNE as follows. Let all players 1, 2, $\cdots$, $|\mathcal{P}|$, in this order, choose their optimal strategies one at a time. For any player $k$, his strategy under $A_1$-COORD is only dependent on the strategies of previous players. No matter how the later players choose their strategies, player $k$ has no reason to deviate. Therefore, the outcome is a PNE.

To see that each player can compute his optimal strategy in polynomial time and the aforementioned PNE can be constructed in polynomial time, observe that under $A_1$-COORD, each of his jobs has completion time independent of his other jobs. Therefore, he can simply assign each of his jobs to the machine that causes the least finishing time of that job. The outcome of such assignment is clearly his optimal strategy and can be computed in polynomial time.

The last part of the theorem follows from Lemmas 2 and 3. □

## 5 $A_2$-COORD

In this section we modify $A_1$-COORD so as to achieve better PoA when $C$ is relatively large compared to $m$.

> $A_2$-COORD: Let $N$ be the assignment. Suppose that job $i \in N_j$. Then the completion of job $i$ is set as $P(i, N_j) = (\phi_{ij})^{\frac{1}{q}}(L(N_j^{\pi(i)-1}) + Cp_{ij})$, for some $0 < q \leq 1$ and $1/q$ is an integer.

**Lemma 4.** *The schedule decided by $A_2$-COORD is feasible.*

*Proof.* Suppose that in assignment $N$ player $k$ puts jobs $i_1$, $i_2$, $\cdots$, $i_{x \leq C}$ on machine $j$, where the jobs are indexed by their non-decreasing completion times. We need to argue that the completion time $P(i_y, N_j)$ of job $i_y$, $y \leq x$, is at least as large as the total load of jobs finishing no later than $i_y$. In the case that multiple jobs among $i_1$, $i_2$, $\cdots$, $i_{x \leq C}$ finish at the same time as $i_y$, W.L.O.G., we can assume that $i_y$ has the largest index. Consider two possibilities.

1. Suppose that none of the jobs belonging to player $\mathcal{P} \backslash \{1, 2, \cdots, k\}$ finishes as early as $i_y$. Let $i_z$ be the heaviest job among $i_1$, $i_2$, $\cdots$, $i_y$. Then as $P(i_y, N_j) \geq P(i_z, N_j)$,

$$P(i_y, N_j) = (\phi_{i_y j})^{1/q}(L(N_j^{k-1}) + Cp_{i_y j}) \geq$$
$$(\phi_{i_z j})^{1/q}(L(N_j^{k-1}) + Cp_{i_z j}) \geq L(N_j^{k-1}) + \sum_{t=1}^{y} p_{i_t j}.$$

   Observe that the last term in the inequality is at least as large as the total load of the jobs finishing no later than $i_y$.

2. Suppose that some players in $\mathcal{P}\backslash\{1,2,\cdots,k\}$ have jobs on machine $j$ that are finished as early as $i_y$. Let player $k+s$ be such player with the largest index. Suppose further that player $k+s$ has jobs $i_1^*$, $i_2^*$, $\cdots$, $i_z^*$ finish as early as $i_y$ on machine $j$. W.L.O.G., assume that $i_1^*$ is the job with the largest size among the jobs $i_1^*$, $i_2^*$, $\cdots$, $i_z^*$. Then

$$P(i_y, N_j) \geq P(i_1^*, N_j) = (\phi_{i_1^* j})^{1/q}(L(N_j^{k+s-1}) + Cp_{i_1^* j}) \geq L(N_j^{k+s-1}) + \sum_{t=1}^{z} p_{i_t^* j},$$

which is no less than the total load of jobs finishing as early as $i_y$. The proof follows. □

**Lemma 5.** *Suppose that $N$ is a PNE under $A_2$-COORD. Then* $\max_{j\in\mathcal{M}, i\in N_j} P(i,N_j) \leq (\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + C\max_{j\in\mathcal{M}} L(O_j)$

*Proof.* Let $i^*$ be the job with the largest completion time. Suppose that it is assigned to $j_1$ in assignment $N$ and has inefficiency 1 on machine $j_2$ ($j_2$ could be the same as $j_1$). Either $j_2 \neq j_1$, then the player $\pi(i^*)$ controlling $i^*$ has no incentive to move $i^*$ to $j_2$; or $j_2 = j_1$. In both cases, we have

$$P(i^*, N_{j_1}) \leq L(N_{j_2}^{\pi(i^*)-1}) + Cp_{i^* j_2} \leq L(N_{j_2}) + Cp_{i^* j_2} \leq (\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + C\max_{j\in\mathcal{M}} L(O_j),$$

where the last inequality follows from Proposition 1 and the fact that in the optimal, one machine would have load at least $p_{i^* j_2}$. The proof follows. □

**Lemma 6.** *Suppose that $N$ is a PNE under $A_2$-COORD. Then*

$$(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} \leq \frac{[(3q+2)C^{2q}]^{\frac{1}{q+1}} m^{\frac{1}{q+1}}}{(1-q/2)^{\frac{1}{q+1}}} \max_{j\in\mathcal{M}} L(O_j).$$

*Proof.* Suppose that job $i$ is assigned to machine $j_1$ in assignment $N$ and machine $j_2$ in the optimal assignment $O$. As $N$ is a PNE, $(\phi_{ij_1})^{1/q}(L(N_{j_1}^{\pi(i)-1}) + Cp_{ij_1}) \leq (\phi_{ij_2})^{1/q}(L(N_{j_2}^{\pi(i)-1}) + Cp_{ij_2})$. Raising both sides by the power of $q$, and multiplying them by $p_{i,\min}$, we have

$$p_{ij_1}(L(N_{j_1}^{\pi(i)-1}) + Cp_{ij_1})^q \leq p_{ij_2}(L(N_{j_2}^{\pi(i)-1}) + Cp_{ij_2})^q$$

Using the above inequality, we derive

$$p_{ij_1}(L(N_{j_1}^{\pi(i)-1}) + p_{ij_1})^q \leq p_{ij_1}(L(N_{j_1}^{\pi(i)-1}) + Cp_{ij_1})^q \leq p_{ij_2}(L(N_{j_2}^{\pi(i)-1}) + Cp_{ij_2})^q \leq p_{ij_2}(L(N_{j_2}) + Cp_{ij_2})^q.$$

Define $x_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in assignment $N$; similarly define $y_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in the optimal $O$. Then the above inequality, if summed over all jobs, can be expressed as

$$\begin{aligned}
&\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{\pi(i)-1}) + x_{ij}p_{ij})^q \\
&\leq \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} y_{ij}p_{ij}(L(N_j) + Cy_{ij}p_{ij})^q \\
&\leq \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} y_{ij}p_{ij}L(N_j)^q + y_{ij}p_{ij}(y_{ij}p_{ij}C)^q \\
&= \sum_{j\in\mathcal{M}} L(N_j)^q \sum_{i:i\in O_j} p_{ij} + \sum_{j\in\mathcal{M}} C^q \sum_{i:i\in O_j} p_{ij}^{1+q} \\
&\leq \sum_{j\in\mathcal{M}} L(N_j)^q L(O_j) + \sum_{j\in\mathcal{M}} C^q L(O_j)^{1+q} \\
&\leq \sum_{j\in\mathcal{M}} \frac{(1/q-1)L(O_j)^{1+q} + 2C^q L(O_j)^{1+q} + \frac{1}{2C^q}L(N_j)^{1+q}}{1/q+1} + \sum_{j\in\mathcal{M}} C^q L(O_j)^{1+q} \\
&= \sum_{j\in\mathcal{M}} \frac{L(N_j)^{1+q}}{2C^q(1+1/q)} + L(O_j)^{1+q}[C^q(1+\frac{2}{1+1/q}) + \frac{1-q}{1+q}], \qquad (5)
\end{aligned}$$

where the second inequality follows from Proposition 3 by setting $p = 1/q$, $a = L(N_j^{\pi(i)-1})$ and $b = Cy_{ij}p_{ij}$; the third inequality from Proposition 4 by setting $A = 0$, $B_i = p_{ij}$ and $p = 1+q$ (so that $\sum_{i:i\in O_j} p_{ij}^{1+q} \leq (\sum_{i:i\in O_j} p_{ij})^{1+q} = L(O_j)^{1+q}$); and the fourth one from the arithmetic-geometric mean inequality (of $1/q+1$ terms).

We now bound the term $\sum_{j\in\mathcal{M}} L(N_j)^{q+1}$.

$$\begin{aligned}
\sum_{j\in\mathcal{M}} L(N_j)^{q+1} &\leq (1+q)C^q \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{\pi(i)-1}) + x_{ij}p_{ij})^q \\
&\leq (1+q)C^q\{\frac{\sum_{j\in\mathcal{M}} L(N_j)^{1+q}}{2C^q(1/q+1)} + L(O_j)^{1+q}[C^q(1+\frac{2}{1+1/q}) + \frac{1-q}{1+q}]\} \\
&= q/2\sum_{j\in\mathcal{M}} L(N_j)^{1+q} + (C^{2q}((1+q)+2q) + C^q(1-q))\sum_{j\in\mathcal{M}} L(O_j)^{1+q} \\
&\leq q/2\sum_{j\in\mathcal{M}} L(N_j)^{1+q} + C^{2q}(3q+2)\sum_{j\in\mathcal{M}} L(O_j)^{1+q},
\end{aligned}$$

where the first inequality follows from the same derivation as in (3); the second from (5); the third from the fact that $C^q(1-q) \leq C^{2q}$.

Rearranging terms in the above inequality, and raising both sides to the power of $1/(1+q)$,

$$(1-q/2)^{\frac{1}{1+q}}(\sum_{j\in\mathcal{M}} L(N_j)^{1+q})^{\frac{1}{q+1}} \leq (C^{2q}(3q+2))^{\frac{1}{1+q}}(\sum_{j\in\mathcal{M}} L(O_j)^{1+q})^{\frac{1}{1+q}} \leq$$
$$(C^{2q}(3q+2))^{\frac{1}{1+q}} m^{\frac{1}{1+q}} \max_{j\in\mathcal{M}} L(O_j),$$

where the second inequality follows from Proposition 1. The proof follows. □

**Theorem 2.** *$A_2$-COORD guarantees a PNE. Such a PNE can be computed in polynomial time and each player can compute his optimal strategy in polynomial time. Moreover, for any fixed q, $0 < q \leq 1$ so that $1/q$ is an integer, it guarantees the PoA of $O(C^{\frac{2q}{q+1}} m^{\frac{1}{q+1}})$.*

*Proof.* The existence of the PNE, and the poly-time computability of such PNEs and of a player's optimal strategy follow the same arguments as in the proof of Theorem 1. The last part of the theorem is due to Lemmas 5 and 6. □

## 6 $A_3$-COORD

In this section, we assume that jobs are indexed in such a way that each player controls jobs with consecutive indices. Precisely, let $c_k$ be the number of jobs controlled by player $k$. Then his jobs are indexed as $1+\sum_{t=1}^{k-1} c_t$, $2+\sum_{t=1}^{k-1} c_t$, $\cdots$, $c_k+\sum_{t=1}^{k-1} c_t$. Unlike the previous two sections, here $N_j^i$ denotes the set of jobs with index at most $i$ that are assigned to machine $j$ in $N$. Accordingly, $L(N_j^i)$ is their total load: $L(N_j^i) = \sum_{i': i' \leq i, i' \in N_j} p_{i'j}$. Let $W = \frac{\max_{i\in\mathcal{I}} \min_{j\in\mathcal{M}} p_{ij}}{\min_{i\in\mathcal{I}} \min_{j\in\mathcal{M}} p_{ij}}$. We assume that all job sizes are rescaled so that $\min_{i\in\mathcal{I}} \min_{j\in\mathcal{M}} p_{ij} = 1$. Then $p_{ij} \geq 1$ for all $i$, $j$.

We now introduce $A_3$-COORD.

> $A_3$-COORD: Let $N$ be the assignment. Suppose that job $i \in N_j$. Then the completion of job $i$ is set as $P(i, N_j) = (p_{ij})^{\frac{1}{q}}(L(N_j^{i-1}) + p_{ij})$. When $C > 1$, we set $q = 1$. When $C = 1$, we set $q = \theta(\log mW)$.

Unlike $A_1$-COORD and $A_2$-COORD, here $C$ is absent in the completion time $P(i, N_j)$. Also notice that we replace the inefficiency $\phi_{ij}$ with the size $p_{ij}$, hence $A_3$-COORD is strongly local.

**Lemma 7.** *The schedule decided by $A_3$-COORD is feasible.*

*Proof.* Let $i_1$, $i_2$, $\cdots$, $i_x$ be the set of jobs assigned to $j$ in $N$, assuming that their finishing times are non-decreasing. We argue that when job $i_{y\leq x}$ finishes, the total load of jobs $i_{y'}$, $y' \leq y$, is no larger than $P(i_y, N_j)$. W.L.O.G., if multiple jobs finish at the same time with $i_y$, then $i_y$ is the one with the largest index.

Suppose that all jobs $i_{y'}$, $y' < y$, have indices smaller than $i_y$, then $P(i_y, N_j) = (p_{i_y j})^{1/q}(L(N_j^{i_y-1}) + p_{i_y j}) \geq L(N_j^{i_y})$, which is at least as large as the total load of jobs $i_{y'}$, for all $y' \leq y$. On the other hand, suppose that a job $i_{y'}$, $y' < y$, has index larger than $i_y$. W.L.O.G., let $i_{y'}$ be such job with the largest index. Then $P(i_y, N_j) \geq P(i_{y'}, N_j) \geq L(N_j^{i_{y'}}) > \sum_{t=1}^{y} p_{i_t j}$ and the proof follows. □

**Lemma 8.** *Suppose that $N$ is a PNE under $A_3$-COORD. Suppose that $q = 1$ (thus $C > 1$). Then given any $\gamma \in \mathbb{Z}^+$,*

$$\max_{j \in \mathcal{M}, i \in N_j} P(i, N_j) \leq \min \left\{ \begin{array}{c} W(\sqrt{\sum_{j \in \mathcal{M}} L(N_j)^2} + \max_{j \in \mathcal{M}} L(O_j)), \\ \frac{\gamma}{\gamma+1}(\sqrt{\sum_{j \in \mathcal{M}} L(N_j)^2})^{\frac{\gamma+1}{\gamma}} + (W + \frac{W^\gamma}{\gamma+1}) \max_{j \in \mathcal{M}} L(O_j). \end{array} \right\}.$$

*Suppose that $q = \Theta(\log mW)$ (thus $C = 1$). Then*

$$\max_{j \in \mathcal{M}, i \in N_j} P(i, N_j) \leq W^{1/q}[(\sum_{j \in \mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + \max_{j \in \mathcal{M}} L(O_j)].$$

*Proof.* Let $i^*$ be the job with the largest completion time in assignment $N$. Suppose that it is controlled by player $\pi(i^*)$, is assigned to $j_1$ in assignment $N$, and has the least size on machine $j_2$ ($j_2$ could be the same as $j_1$). Below we only prove the case of $q = 1$. See the appendix for the case of $q = \Theta(\log mW)$.

Let $I_x$ denote the union of job $i^*$ and the set of jobs controlled by player $\pi(i^*)$ that are assigned to $j_2$ in $N$. First assume that $j_2 \neq j_1$. As player $\pi(i^*)$ has no incentive to move $i^*$ to $j_2$,

$$\begin{aligned} P(i^*, N_{j_1}) + \sum_{i' \in I_x \setminus \{i^*\}} P(i', N_{j_2}) &\leq p_{i^* j_2}(L(N_{j_2}^{i^*-1}) + p_{i^* j_2}) \\ + \sum_{i' \in I_x, i' < i^*} P(i', N_{j_2}) &+ \sum_{i' \in I_x, i' > i^*} [P(i', N_{j_2}) + p_{i' j_2} p_{i^* j_2}]. \end{aligned}$$

Notice that the RHS of the inequality is the sum of the costs of the jobs in set $I_x$ if $i^*$ moved to $j_2$. Observe that the above inequality holds as well when $j_2 = j_1$. Canceling the term $\sum_{i' \in I_x \setminus \{i^*\}} P(i', N_{j_2})$ from both sides of the inequality, we obtain

$$\begin{aligned} P(i^*, N_{j_1}) &\leq p_{i^* j_2}(L(N_{j_2}^{i^*-1}) + \sum_{i' \in I_x, i' \geq i^*} p_{i' j_2}) \\ &\leq p_{i^* j_2}(L(N_{j_2}) + p_{i^* j_2}) \leq W L(N_{j_2}) + W^2. \end{aligned} \tag{6}$$

We can further bound the expression $WL(N_{j_2}) + W^2$ in two different ways. First, note that

$$WL(N_{j_2}) + W^2 \leq W(\sqrt{\sum_{j \in \mathcal{M}} L(N_j)^2} + W) \leq W(\sqrt{\sum_{j \in \mathcal{M}} L(N_j)^2} + \max_{j \in \mathcal{M}} L(O_j)), \tag{7}$$

where the first inequality follows from Proposition 1 and the second by the fact that one machine would have load at least $W$ in the optimal. A second way to bound $WL(N_{j_2}) + W^2$ is as follows.

$$
\begin{aligned}
WL(N_{j_2}) + W^2 &\leq \frac{W^{\gamma+1} + \sum_{t=1}^{\gamma} (L(N_{j_2})^{\frac{1}{\gamma}})^{\gamma+1}}{\gamma+1} + W^2 \\
&= \frac{\gamma}{\gamma+1} L(N_{j_2})^{\frac{\gamma+1}{\gamma}} + (W + \frac{W^\gamma}{\gamma+1})W \\
&\leq \frac{\gamma}{\gamma+1} (\sqrt{\sum_{j\in\mathcal{M}} L(N_j)^2})^{\frac{\gamma+1}{\gamma}} + (W + \frac{W^\gamma}{\gamma+1}) \max_{j\in\mathcal{M}} L(O_j), \quad (8)
\end{aligned}
$$

where the first inequality follows from the arithmetic-geometric mean inequality (of $\gamma + 1$ terms), and the second inequality from the same reason as in (7). The inequalities (6), (7), and (8) together give the first part of lemma.

When $q = \Theta(\log mW)$, either $j_2 \neq j_1$, then the player $\pi(i^*)$ controlling $i^*$ has no incentive to move $i^*$ to $j_2$; or $j_2 = j_1$. In both cases,

$$
\begin{aligned}
P(i^*, N_{j_1}) \leq (p_{i^*j_2})^{1/q} (L(N_{j_2}^{i^*-1}) + p_{i^*j_2}) \leq W^{1/q}(L(N_{j_2}) + W) \leq \\
W^{1/q}[(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + \max_{j\in\mathcal{M}} L(O_j)],
\end{aligned}
$$

where the last inequality follows from the same reason as in (7). The last part of the lemma follows. □

In the next two lemmas, we show how to bound the term $(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}}$, when $q = 1$ and when $q = \Theta(\log mW)$, separately.

**Lemma 9.** *Suppose that $N$ is a PNE under $A_3$-COORD and $q = 1$ (thus $C > 1$). Then*

$$
\sqrt{\sum_{j\in\mathcal{M}} L(N_j)^2} \leq \frac{\sqrt{m}}{\sqrt{3/2} - 1} \max_{j\in\mathcal{M}} L(O_j).
$$

*Proof.* Let $x_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in assignment $N$. Caragiannis [7, Theorem 7] showed the following inequality.

$$
\sum_j \frac{L(N_j)^2}{2} \leq \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{M}} x_{ij} p_{ij} (L(N_j^{i-1}) + x_{ij} p_{ij}). \quad (9)
$$

The RHS of the inequality is exactly the sum of costs of all players in $\mathcal{P}$ in assignment $N$. Therefore,

$$
\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{M}} x_{ij} p_{ij} (L(N_j^{i-1}) + x_{ij} p_{ij}) = \sum_{k\in\mathcal{P}} \sum_{i:\pi(i)=k, j:i\in N_j} P(i, N_j). \quad (10)
$$

Consider player $k \in \mathcal{P}$. He has no incentive to re-assign his jobs to the machines where they belong to in the optimal. Therefore,

$$\sum_{i:\pi(i)=k,j:i\in N_j} P(i,N_j) \leq$$
$$\sum_{j\in\mathcal{M}}[\sum_{i:\pi(i)=k,i\in O_j} p_{ij}L(N_j) + \frac{(\sum_{i:\pi(i)=k,i\in O_j} p_{ij})^2 + \sum_{i:\pi(i)=k,i\in O_j} w_{ij}^2}{2}] \leq$$
$$\sum_{j\in\mathcal{M}}[\sum_{i:\pi(i)=k,i\in O_j} p_{ij}L(N_j) + (\sum_{i:\pi(i)=k,i\in O_j} p_{ij})^2],$$

where the first inequality is derived by assuming (pessimistically) that all jobs of player $k$ in $O_j$ have indices larger than the jobs of all other players in $N_j$. Summing the above inequality over all players,

$$\begin{aligned}
\sum_{k\in\mathcal{P}}\sum_{i:\pi(i)=k,j:i\in N_j} P(i,N_j) &\leq \sum_{j\in\mathcal{M}}[L(O_j)L(N_j) + \sum_{k\in\mathcal{P}}(\sum_{i:\pi(i)=k,i\in O_j} p_{ij})^2] \\
&\leq \sum_{j\in\mathcal{M}} L(O_j)L(N_j) + L(O_j)^2 \\
&\leq \sum_{j\in\mathcal{M}} (L(N_j)+L(O_j))^2 - \sum_{j\in\mathcal{M}} L(N_j)^2 \\
&\leq (\sqrt{\sum_{j\in\mathcal{M}} L(N_j)^2} + \sqrt{\sum_{j\in\mathcal{M}} L(O_j)^2})^2 - \sum_{j\in\mathcal{M}} L(N_j)^2 \quad (11)
\end{aligned}$$

where the second inequality follows from Proposition 4 (by setting $A = 0$ and $p = 1$), and the last inequality from Minkowski inequality. By (9), (10), (11), and re-arranging terms, we have

$$(\sqrt{3/2}-1)\sqrt{\sum_{j\in\mathcal{M}} L(N_j)^2} \leq \sqrt{\sum_{j\in\mathcal{M}} L(O_j)^2} \leq \sqrt{m}\max_{j\in\mathcal{M}} L(O_j),$$

where the last inequality follows from Proposition 1. The proof follows. □

**Lemma 10.** *Suppose that $N$ is a PNE under $A_3$-COORD and $q = \Theta(\log mW)$ (thus $C = 1$). Then*

$$(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} \leq e(q+1)m^{\frac{1}{q+1}} \max_{j\in\mathcal{M}} L(O_j).$$

*Proof.* Let $x_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in assignment $N$; similarly let $y_{ij} = 1(0)$ if job $i$ is (not) assigned to machine $j$ in the optimal $O$. We make the following claim.

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{i-1})+x_{ij}p_{ij})^q \le \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} y_{ij}p_{ij}(L(N_j)+y_{ij}p_{ij})^q. \qquad (12)$$

By our index scheme and the fact that $C=1$, job $i$ is controlled by player $i$. Suppose job $i$ is in $j_1$ in $N$ and in $j_2$ in $O$. As player $i$ has no incentive to move his job from $j_1$ to $j_2$,

$$(p_{ij_1})^{1/q}(L(N_{j_1}^{i-1})+p_{ij_1}) \le (p_{ij_2})^{1/q}(L(N_{j_2}^{i-1})+p_{ij_2}) \le (p_{ij_2})^{1/q}(L(N_{j_2})+p_{ij_2}).$$

Raising the above inequality to the power of $q$ and summing it over all players, we have

$$\sum_{j\in\mathcal{M}}\sum_{i\in N_j} p_{ij}(L(N_j^{i-1})+p_{ij})^q \le \sum_{j\in\mathcal{M}}\sum_{i\in O_j} p_{ij}(L(N_j)+p_{ij})^q,$$

and Inequality (12) follows.

The rest of the analysis is completely the same as Caragiannis [7, Theorem 7]. He showed the following two inequalities:

$$\begin{aligned}(e-1)(q+1)\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} y_{ij}p_{ij}(L(N_j)+y_{ij}p_{ij})^q \le \\ ((\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} + e(\sum_{j\in\mathcal{M}} L(O_j)^{q+1})^{\frac{1}{q+1}})^{q+1} - \sum_{j\in\mathcal{M}} L(N_j)^{q+1}\end{aligned} \qquad (13)$$

$$(e-1)(\sum_{j\in\mathcal{M}} L(N_j))^{q+1} \le (e-1)(q+1)\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{M}} x_{ij}p_{ij}(L(N_j^{i-1})+x_{ij}p_{ij})^q \quad (14)$$

Combining (12), (13), and (14), we have

$$(\sum_{j\in\mathcal{M}} L(N_j)^{q+1})^{\frac{1}{q+1}} \le \frac{e}{e^{\frac{1}{q+1}}-1}(\sum_{j\in\mathcal{M}} L(O_j)^{q+1})^{\frac{1}{q+1}} \le e(q+1)m^{\frac{1}{q+1}}\max_{j\in\mathcal{M}} L(O_j),$$

where the second inequality is due to the inequality $e^z-1\ge z$ and Proposition 1. The proof follows. □

**Theorem 3.** *$A_3$-COORD guarantees a PNE. Moreover, by setting $q=1$, it guarantees the PoA of $O(\min\{W\sqrt{m}, \min_{\gamma\in\mathbb{Z}_{\ge 1}}\{m^{\frac{\gamma+1}{2\gamma}}+W^\gamma\}\})$. In case that $C=1$, by setting $q=\theta(\log mW)$, it guarantees the PoA of $O(\log m+\log W)$.*

*Proof.* The existence of PNE follows the same argument as in the proof of Theorem 1. The second part of the theorem follows from Lemmas 8, 9, and 10. □

Unfortunately, under $A_3$-COORD, it is NP-hard to decide the optimal strategy for a player. So we cannot use the same procedure as in the previous two mechanisms to build a PNE in polynomial time. The NP-hardness follows from the observation that when $q = 1$, a player controls all the jobs, and only two *identical* machines are given, finding an optimal strategy is equivalent to minimizing the weighted sum of completion times of jobs. (The latter problem is NP-hard by a reduction from the PARTITION problem [17].)

## References


1. Abed, F., Correa, J., and Huang, C.-C. Optimal coordination mechanisms for multi-job scheduling games. In *22nd Annual European Symposium on Algorithms (ESA)* (2014).
2. Abed, F., and Huang, C.-C. Preemptive coordination mechanisms for unrelated machines. In *20th Annual European Symposium on Algorithms (ESA)* (2012).
3. Anshelevich, E., Caskurlu, B., and Hate, A. Partition equilibrium always exists in resource selection games. In *SAGT* (2010), pp. 42–53.
4. Awerbuch, B., Azar, Y., Richter, Y., and Tsur, D. Tradeoffs in worst-case equilibria. *Theor. Comput. Sci. 361*, 2-3 (2006), 200–209.
5. Azar, Y., Jain, K., and Mirrokni, V. S. (almost) optimal coordination mechanisms for unrelated machine scheduling. In *SODA* (2008), pp. 323–332.
6. Bhaskar, U., Fleischer, L., Hoy, D., and Huang, C.-C. Equilibria of atomic flow games are not unique. In *SODA* (2009), pp. 748–757.
7. Caragiannis, I. Efficient coordination mechanisms for unrelated machine scheduling. In *SODA* (2009), pp. 815–824. Journal version to appear in Algorithmica.
8. Christodoulou, G., Koutsoupias, E., and Nanavati, A. Coordination mechanisms. In *ICALP* (2004), pp. 345–357.
9. Cohen, J., Dürr, C., and Thang, N. K. Non-clairvoyant scheduling games. *Theory Comput. Syst. 49*, 1 (2011), 3–23.
10. Cohen, J., Dürr, C., and Thang, N. K. Smooth inequalities and equilibrium inefficiency in scheduling games. In *WINE* (2012), pp. 350–363.
11. Cole, R., Correa, J. R., Gkatzelis, V., Mirrokni, V. S., and Olver, N. Inner product spaces for minsum coordination mechanisms. In *STOC* (2011), pp. 539–548.
12. Cominetti, R., Correa, J. R., and Stier-Moses, N. E. The impact of oligopolistic competition in networks. *Operations Research 57*, 6 (2009), 1421–1437.
13. Czumaj, A., and Vöcking, B. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms 3*, 1 (2007).
14. Feldman, M., and Tennenholtz, M. Partition equilibrium. In *SAGT* (2009), pp. 48–59.
15. Fleischer, L., and Svitkina, Z. Preference-constrained oriented matching. In *Proceedings of the Seventh Workshop on Analytic Algorithmics and Combinatorics (ANALCO)* (2010), pp. 66–73.
16. Fotakis, D., Kontogiannis, S., and Spirakis, P. Atomic congestion games among coalitions. *ACM Transactions on Algorithms 4*, 4 (2008). The conference version appeared in ICALP 2006.
17. Garey, M., and Johnson, D. *Computers and Intractablility.* Freeman, 1979.

18. Hayrapetyan, A., Tardos, E., and Wexler, T. The effect of collusion in congestion games. In *STOC* (New York, NY, USA, 2006), ACM Press, pp. 89–98.
19. Immorlica, N., Li, L. E., Mirrokni, V. S., and Schulz, A. S. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci. 410*, 17 (2009), 1589–1598.
20. Koutsoupias, E., and Papadimitriou, C. H. Worst-case equilibria. *Computer Science Review 3*, 2 (2009), 65–69.
21. Lenstra, J. K., Shmoys, D. B., and Tardos, É. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program. 46* (1990), 259–271.
22. Nash, J. Non-cooperative games. *The Annals of Mathematics 54(2)* (1951), 286–295.
23. Roughgarden, T. *Selfish Routing and the Price of Anarchy.* The MIT Press, 2005.

# A Counter-examples and Lower Bounds for Known Mechanisms

## A.1 PNE Existence for ShortestFirst, AJM-2, and BALANCE

**Theorem 4.** *ShortestFirst, AJM-2, and BALANCE do not induce PNE even for two identical machines.*

*Proof.* Consider the instance of two identical machines and two players. Player 1 controls a job of size 1 and a job of size 3. Player 2 controls a job of size 2 and a job of size 4. It is easy to verify that this instance has no PNE when using ShortestFirst. The theorem holds for AJM-2 as well because it is exactly equivalent to ShortestFirst in the case of two identical machines. The same example can be used as an evidence that BALANCE does not induce PNE. □

## A.2 PNE Existence for Makespan, BCOORD, and CCOORD

**Theorem 5.** *Makespan, BCOORD when p=1, and CCOORD when p=1, do not induce PNE even for two identical machines.*

*Proof.* Consider the instance of two identical machines and two players. Player 1 controls a job of size 1. Player 2 controls a job of size 2, a job of size 1, and a job of size 1/10. It is easy to verify that this instance has no PNE when using the Makespan policy. Notice that when p=1, both BCOORD and CCOORD are equivalent to Makespan in identical machines. □

## A.3 PoA for ACOORD, BCOORD, and CCOORD

In the following, if we do not specify the size $p_{ij}$ of job $i$ on machine $j$, we implicitly assume it is infinity. We first present a simple construction where each player controls just two jobs.

**Theorem 6.** *For all values of $p \geq 1$, when $C = 2$, ACOORD and BCOORD have PoA of at least $\Omega(m)$. The same bound holds for CCOORD when p=1.*

*Proof.* Consider an instance of $m$ machines and $n = m - 1$ players. Player $k$, for $1 \le k \le n$, controls two jobs, $i_k^0$ and $i_k^1$, both of which can be processed only on machine 0 and machine $k$. Job $i_k^0$ has size 1 on machine 0, and size 2 on machine $k$. Job $i_k^1$ has size $\delta$ on machine 0, and size $\delta \cdot n^p$ on machine $k$, for some small $\delta > 0$ where $\delta << 1/n^p$.

We first focus on ACOORD. Let the total order of all jobs be as follows $i_1^0, i_1^1, i_2^0, i_2^1, \cdots, i_n^0, i_n^1$. Let $N$ be the following assignment. Each player $k$ assigns job $i_k^0$ to machine 0 and job $i_k^1$ to machine $k$. To see that $N$ is a PNE, consider player $k$, whose current cost is $k + \delta \cdot n^p$. He has three other possible strategies:

1. Assign both of his jobs to machine 0. Then his cost is $k + k + \delta$.
2. Assign both of his jobs to machine $k$. Then his cost is $2 \cdot 2^{1/p} + (2 + \delta \cdot n^p) \cdot n$.
3. Assign $i_k^0$ to machine $k$ and $i_k^1$ to machine 0. Then his cost is $k - 1 + \delta + 2 \cdot 2^{1/p}$.

All these three strategies incur higher costs, so player $k$ has no reason to deviate. Thus, $N$ is a PNE with makespan of $m - 1$, while $OPT$ is 2.

Next we consider BCOORD (and notice that CCOORD is the same as BCOORD when p=1). We claim that the same assignment $N$ is also a PNE. Consider player $k$, whose current cost is $n + \delta \cdot n^p$. He has three other possible strategies:

1. Assign both of his jobs to machine 0. Then his cost is $2n + 2\delta$.
2. Assign both of his jobs to machine $k$. Then his cost is $(2 + \delta \cdot n^p) \cdot 2^{1/p} + (2 + \delta \cdot n^p) \cdot n$.
3. Assign $i_k^0$ to machine $k$ and $i_k^1$ to machine 0. Then his cost is $n - 1 + \delta + 2 \cdot 2^{1/p}$.

All these three strategies incur higher costs, so player $k$ has no reason to deviate. Thus, $N$ is a PNE with makespan of $m - 1$, while $OPT$ is 2.

□

We next expand on the same idea to show that the PoA of ACOORD/BCOORD/CCOORD can be much higher when C is large.

**Theorem 7.** *For all values of $p \ge 1$, ACOORD and BCOORD have PoA of at least $\Omega(C^{(1-\epsilon)(p+1)} m/p^2)$, for some small $\epsilon > 0$. The same bound holds for CCOORD when p=1.*

*Proof.* Consider an instance of $m = (n+1) \cdot (p^2 + \tau + 1)$ machines, where $\tau$ is some positive integer (the larger the $\tau$, the smaller the $\epsilon$ in the lower bound). Let $m_{uv}$ represent a machine for some $u$, $v$, where $0 \le u \le (p^2 + \tau)$, $0 \le v \le n$. Let $k_{st}$ denote a player for some $s, t$, where $0 \le s \le (p^2 + \tau)$, $1 \le t \le n$. We next specify the jobs controlled by each player and their sizes on the machines. Below we assume $\delta$ to be some small constant and $\delta << 1/(nC)^{6p^2}$.

1. For player $k_{0t}$, $1 \le t \le n$, he controls two jobs $i_{0t}^0$, $i_{0t}^1$. Job $i_{0t}^0$ has size 1 on machine $m_{00}$, and size 2 on machine $m_{0t}$; job $i_{0t}^1$ has size $\delta$ on machine $m_{00}$, and size $\delta \cdot n^p$ on machine $m_{0t}$.

2. For player $k_{st}$, with $s > 0$, $1 \le t \le n$, he controls $C+1$ jobs, $i_{st}^0, i_{st}^1, \cdots, i_{st}^C$. Job $i_{st}^0$ has size 1 on machine $m_{st}$, and size $C^{\sum_{a=1}^{s}(p/(p+1))^a}$ on machine $m_{s0}$; job $i_{st}^b$, for $1 \le b \le C$, has size $\delta$ on machine $m_{(s-1)0}$, and size $\delta(nC)^{2p^2}$ on machine $m_{st}$.

We first focus on ACOORD. Let the total order for all the jobs be as follows: $i_{01}^0, i_{01}^1, i_{02}^0, i_{02}^1, \cdots, i_{0n}^0, i_{0n}^1, i_{11}^0, i_{11}^1, \cdots, i_{11}^C, i_{12}^0, i_{12}^1, \cdots, i_{12}^C, \cdots, i_{1n}^0, i_{1n}^1, \cdots, i_{1n}^C, i_{21}^0, i_{21}^1, \cdots, i_{21}^C, i_{22}^0, i_{22}^1, \cdots, i_{22}^C, \cdots, i_{2n}^0, i_{2n}^1, \cdots, i_{2n}^C, i_{31}^0, i_{31}^1, \cdots, i_{31}^C, i_{32}^0, i_{32}^1, \cdots, i_{32}^C, \cdots, i_{3n}^0, i_{3n}^1, \cdots, i_{3n}^C, \cdots, i_{(p^2+\tau)1}^0, i_{(p^2+\tau)1}^1, \cdots, i_{(p^2+\tau)1}^C, i_{(p^2+\tau)2}^0, i_{(p^2+\tau)2}^1, \cdots, i_{(p^2+\tau)2}^C, \cdots, i_{(p^2+\tau)n}^0, i_{(p^2+\tau)n}^1, \cdots, i_{(p^2+\tau)n}^C$.

Let $N$ be the following assignment. Each player $k_{st}$ assigns job $i_{st}^0$ to machine $m_{s0}$ and the remaining job(s) to $m_{st}$. To see that $N$ is a PNE, consider player $k_{0t}$, $1 \le t \le n$. We can use the same argument as in the previous theorem to show that he has no reason to deviate. Next consider player $k_{st}$, with $s > 0$, $1 \le t \le n$, whose current cost is $t \cdot C^{((p+1)/p)\sum_{a=1}^{s}(p/(p+1))^a} + d$, where $d$ is the sum of the cost the jobs $i_{st}^1, i_{st}^2, \cdots, i_{st}^C$, thus $d << 1$. He has three other possible strategies:

1. Assign job $i_{st}^0$ to machine $m_{s0}$ and at least one of the remaining jobs to machine $m_{(s-1)0}$. Then his cost is at least $t \cdot C^{((p+1)/p)\sum_{a=1}^{s}(p/(p+1))^a} + n \cdot C^{\sum_{a=1}^{s-1}(p/(p+1))^a}$.
2. Assign job $i_{st}^0$ and at least one of the remaining jobs to machine $m_{st}$ . Then his cost is at least $1 + (1 + \delta(nC)^{2p^2}) \cdot (nC)^{2p}$.
3. Assign job $i_{st}^0$ to machine $m_{st}$ and the remaining jobs to machine $m_{(s-1)0}$. Then his cost is at least $1 + n \cdot C^{1+\sum_{a=1}^{s-1}(p/(p+1))^a} = 1 + n \cdot C^{((p+1)/p)\sum_{a=1}^{s}(p/(p+1))^a}$.

All these three strategies incur higher costs, so he has no reason to deviate. Thus, $N$ is a PNE with makespan of $n \cdot C^{((p+1)/p)\sum_{a=1}^{(p^2+\tau)}(p/(p+1))^a} = \Omega(C^{(1-\epsilon)(p+1)}m/p^2)$, for some small $\epsilon > 0$, while $OPT$ is 2.

Next we consider BCOORD (and notice that CCOORD is the same as BCOORD when p=1). We claim that the same assignment $N$ is also a PNE. Consider player $k_{0t}$, $1 \le t \le n$. We can use the same argument as in the previous theorem to show that he has no reason to deviate. Next consider player $k_{st}$, with $s > 0$, $1 \le t \le n$, whose current cost is $n \cdot C^{((p+1)/p)\sum_{a=1}^{s}(p/(p+1))^a} + d$, where $d$ is the sum of the cost the jobs $i_{st}^1, i_{st}^2, \cdots, i_{st}^C$, thus $d << 1$. He has three other possible strategies:

1. Assign job $i_{st}^0$ to machine $m_{s0}$ and at least one of the remaining jobs to machine $m_{(s-1)0}$. Then his cost is at least $n \cdot C^{((p+1)/p)\sum_{a=1}^{s}(p/(p+1))^a} + n \cdot C^{\sum_{a=1}^{s-1}(p/(p+1))^a}$ .
2. Assign job $i_{st}^0$ and at least one of the remaining jobs to machine $m_{st}$ . Then his cost is at least $1 + \epsilon(nC)^{2p^2} + (1 + \delta(nC)^{2p^2}) \cdot (nC)^{2p}$.
3. Assign job $i_{st}^0$ to machine $m_{st}$ and the remaining jobs to machine $m_{(s-1)0}$. Then his cost is at least $1 + n \cdot C^{1+\sum_{a=1}^{s-1}(p/(p+1))^a} = 1 + n \cdot C^{((p+1)/p)+\sum_{a=1}^{s}(p/(p+1))^a}$.

All these three strategies incur higher costs, so he has no reason to deviate. Thus, $N$ is a PNE with makespan of $n \cdot C^{((p+1)/p)\sum_{a=1}^{(p^2+\tau)}(p/(p+1))^a} = \Omega(C^{(1-\epsilon)(p+1)}m/p^2)$, for small $\epsilon > 0$, while $OPT$ is 2.

□