Optimal Coordination Mechanisms for Multi-Job Scheduling Games *

Fidaa Abed[1] and José R. Correa[2] and Chien-Chung Huang[3]

[1] Max-Planck-Institut für Informatik, fabed@mpi-inf.mpg.de
[2] Departamento de Ingenieria Industrial, Universidad de Chile, correa@uchile.cl
[3] Chalmers University of Technology, huangch@chalmers.se

Abstract. We consider the unrelated machine scheduling game in which players control subsets of jobs. Each player's objective is to minimize the weighted sum of completion time of her jobs, while the social cost is the sum of players' costs. The goal is to design simple processing policies in the machines with small coordination ratio, i.e., the implied equilibria are within a small factor of the optimal schedule. We work with a weaker equilibrium concept that includes that of Nash. We first prove that if machines order jobs according to their processing time to weight ratio, a.k.a. Smith-rule, then the coordination ratio is at most 4, moreover this is best possible among nonpreemptive policies. Then we establish our main result. We design a preemptive policy, *externality*, that extends Smith-rule by adding extra delays on the jobs accounting for the negative externality they impose on other players. For this policy we prove that the coordination ratio is $1 + \phi \approx 2.618$, and complement this result by proving that this ratio is best possible even if we allow for randomization or full information. Finally, we establish that this externality policy induces a potential game and that an ε-equilibrium can be found in polynomial time. An interesting consequence of our results is that an $\varepsilon-$local optima of $R| \,|\sum w_j C_j$ for the jump (a.k.a. move) neighborhood can be found in polynomial time and are within a factor of 2.618 of the optimal solution. The latter constitutes the first direct application of purely game-theoretic ideas to the analysis of a well studied local search heuristic.

1 Introduction

Machine scheduling originates in the optimization of manufacturing systems and their formal mathematical treatment dates back to at least the pioneering work of Smith [38]. In general, scheduling problems can be described as follows. Consider a set $\mathcal{J}$ of n jobs that have to be processed on a set $\mathcal{M}$ of m parallel machines. If processed on machine i, job j requires a certain processing time p_{ij} to be completed. Job j also has a weight w_j and, in addition, it may have other characteristics such as release dates, time windows, delays when switching a task from one machine to another, or precedence constraints. The goal is to find

* Research partially supported by the Millenium Nucleus Information and Coordination in Networks ICM/FIC P10-024F

an assignment of jobs to machines, and an ordering within each machine so that a certain objective functions is minimized. Denoting, for any such assignment and ordering, the *completion time* C_j of job j the time at which job j completes, we may write the two most widely studied objectives as $C_{\max} = \max_{j \in \mathcal{J}} C_j$ (the makespan) and $\sum_{j \in \mathcal{J}} w_j C_j$ (the sum of weighted completion times). In terms of the machine environment the most basic model is that of *identical* machines, where the processing times of jobs are the same on all machines. In the *related* machines environment each machine has a speed, and the processing time of a job on a machine is inversely proportional to the speed of that machine. Finally in *unrelated* machine scheduling the processing times are arbitrary, thus capturing all the above models as special cases. This latter machine environment, with the sum of weighted completion times objective, denoted by $R| \, | \sum w_j C_j$, is the focus of our paper.

Since the early work of Smith for the $\sum w_j C_j$ objective, a lot of work has been put in designing *centralized* algorithms providing reasonably close to optimal solutions with limited computational effort for these NP-hard problems [5, 15, 18, 21, 23, 24, 32–37]. The underlying assumption is that all information is gathered by a single entity which can enforce a particular schedule. However, as distributed environments emerge, understanding scheduling problems where jobs are managed by different selfish agents (players), who are interested in their own completion time, becomes a central question.

Coordination mechanisms. In recent times there has been quite some effort to understand these scheduling games in the special case in which agents control a single job in the system, which we call *single-job games*. In this context, there is a vast amount of work studying existence, uniqueness, the *price of anarchy* [26], and other characteristics of equilibrium when, given some processing rules, each agent seeks to minimize her own completion time. In the scheduling game each job is a fully informed player wanting to minimize its individual completion time, and its set of strategies correspond to the set of machines. Job j's completion time on a machine depends on the strategies chosen by other players, and on the *policy* (or processing rule) of the chosen machine. While the cost of a job is its weighted completion time, $w_j C_j$. A *coordination mechanism* is then a set of *local policies*, one per machine, specifying how the jobs assigned to that machine are scheduled. In a *local policy* the schedule on a machine depends on the full vector $(p_{1j}, p_{2j}, \ldots, p_{mj})$ and weights w_j of jobs assigned to that machine. In contrast, in a *strongly local policy* the schedule on machine i must be a function only of the processing times p_{ij} and weights w_j of the jobs assigned to i. In evaluating the efficiency of these policies, one needs a benchmark to compare this social cost against. The definition of the price of anarchy of the induced game considers a social optimum with respect to the costs specified by the chosen machine policies. However, to measure the quality of a coordination mechanism we consider the worst case ratio of the social cost at an equilibrium to the optimal social cost that could be achieved by the centralized optimization approach. We refer to this as the *coordination ratio* of a mechanism.

In this paper we take a step forward and study *multi-job games*, in which there is a set of agents $\mathcal{A}$ who control arbitrary sets of jobs. Specifically the set of jobs controlled by player $\alpha \in \mathcal{A}$ is denoted by $J(\alpha) \subset \mathcal{J}$ and its cost given a particular schedule is the sum of weighted completion times of its own jobs $\sum_{j \in J(\alpha)} w_j C_j$. As in single-job games, we concentrate on designing coordination mechanisms leading to small coordination ratios, when the social cost is the sum of weighted completion times of all jobs (or equivalently of all agents).

Machine policies. Throughout we assume that policies are prompt: they do not introduce deliberate idle time. In other words, if jobs $j_1, \ldots, j_k$ are assigned to machine i, then by time $\sum_{\ell=1}^{k} p_{ij_\ell}$ all jobs have been completed and released. Besides distinguishing between local and strongly local policies we distinguish between *nonpreemptive*, *preemptive*, and *randomized* policies. In nonpreemptive policies jobs are processed in some fixed deterministic order that may depend arbitrarily on the set of jobs assigned to the machine (processing time, weight, and ID), and once a job is completed it is released. On the other hand, preemptive policies may suspend a job before it completes in order to execute another job and the suspended job is resumed later. Interestingly, such policies can be considered as nonpreemptive policies, but where jobs may be held back after completion [11, 12]. Finally, randomized policies have the additional power that they can schedule jobs at random according to some distribution depending on the assigned jobs' characteristics. Another usual distinction is between policies that are *anonymous* and *non-anonymous*. In the former jobs with the same characteristics (except for IDs) must be treated equally and thus assigned the same completion time. In the latter, jobs may be distinguished using their IDs.

For instance consider the widely used policy known as Smith-rule (**SR**), which sorts jobs in nondecreasing order of their processing time to weight ratios. Formally **SR** processes jobs in nondecreasing order of $\rho_{ij} = p_{ij}/w_j$, and breaks ties using the job's IDs. This policy is strongly local, nonpreemptive, and non-anonymous.

Equilibrium concepts. For the single-job scheduling game the underlying concept of equilibrium is, quite naturally, that of Nash (NE)[28]. However, once we allow players to control many jobs and endow them with the weighted completion time cost, already computing a best response to a given situation may be NP-complete. Therefore, it is rather unlikely that such an equilibrium will be attained. To overcome this difficulty we consider a weaker equilibrium concept, which we call *weak equilibrium* (WE), namely, a schedule of all jobs is a WE if no player $\alpha \in \mathcal{A}$ can find a job $j \in J(\alpha)$ such that moving j to a different machine will strictly decrease her cost $\sum_{j \in J(\alpha)} w_j C_j$. We extend the WE concept to mixed (randomized) strategies by allowing player α to keep the distribution of all but one job $j \in \mathcal{J}(\alpha)$ and move job j to any machine. Observe that in the single-job game NE and WE coincide. Throughout, we provide bounds on the coordination ratio of policies for the weak equilibrium, and since NE are also WE our bounds hold for NE as well.

As the reader may have noticed, there is a close connection between WE and local optima of the *jump* (also called *move*) neighborhood (e.g. [39]). In a locally optimal solution of $R| \,|\sum w_j C_j$ for the jump neighborhood, no single job $j \in J$ may be moved to a different machine while decreasing the overall cost. Such solution is exactly a WE when a single player in the scheduling game controls all jobs and the machines use **SR**.

To illustrate the concept of weak equilibrium and the difference between the single-job and the multi-job games consider the following example on 4 machines, $m_1, \dots, m_4$, with the **SR** policy. There are 4 unit-weight jobs called a, b, c, d such that $p_{m_1,a} = 1 + \varepsilon$, $p_{m_1,b} = 1$, $p_{m_2,b} = 1.5$, $p_{m_2,c} = 2$, $p_{m_3,c} = 3$, $p_{m_3,d} = 2$, $p_{m_4,d} = 2$, and all other $p_{ij} = +\infty$. In this situation an equilibrium for the single-job game is that jobs a and b go to m_1, job c goes to m_2, and job d goes to m_3, leading to a total cost of $7+\varepsilon$. Consider now the multi-job game in which one player controls a, b and another player controls c, d. A NE is obtained when a goes to m_1, b goes to m_2, c goes to m_3, and d goes to m_4, and this has total cost $7.5 + \varepsilon$. A WE is obtained from instance when a goes to m_1, b goes to m_2, c goes to m_2, and d goes to m_3, having a total cost of $8 + \varepsilon$.

Related literature. The study of coordination mechanisms for single-job scheduling games, taking the makespan as social cost, was initiated by Christodoulou et al. [9]. However the implied bounds on the price of anarchy are constant only for simple environments such as when machines are identical. Indeed, Azar et al. [2], and Fleischer and Svitkina [19] show that, even for a restricted uniform machines environment "almost" no deterministic machine policies can achieve a constant price of anarchy. The result was finally established by Abed and Huang [1], who proved that no symmetric coordination mechanism satisfying the so-called "independence of irrelevant alternatives" property, even if preemption is allowed, can achieve a constant price of anarchy for the makespan objective. The existence of a randomized machine policy with such a desirable property is unknown. It is worth mentioning that there is a vast amount of related work considering the makespan social cost [6, 8, 14, 16, 25, 27].

The situation changes quite dramatically for the sum of weighted completion times objective. In this case Correa and Queyranne show that, for restricted related machines, smith rule induces a game with price of anarchy at most 4 [13], improving results implied by Farzad et al. [17] and Caragiannis et al [8] obtained in different contexts. Cole et al., extend this result to unrelated machines, and also design an improved preemptive policy, proportional sharing, achieving an approximation bound of 2.618 and an even better randomized policy [11, 12]. Further recent works include extensions and improvements by Bhattacharya et al. [3], Cohen et al. [10] and by Rahn and Schäfer [30], Hoeksma and Uetz [22].

Finally, performance guarantee results for the $\sum w_j C_j$ objective using natural local search heuristics are scarse, despite the vast amount of computational work [7, 29]. We are only aware of the results of Brueggemann et al. [4] who proved that for identical machines local optima for the jump neighborhood are within a factor of 3/2 of the optimal schedule.

Our results. We start by considering deterministic policies and prove that the coordination ratio of **SR** under WE is exactly 4. This generalizes the result for single-job games [12] and therefore it is the best possible coordination ratio that can be achieved nonpreemptively. We prove the upper bound of 4 for **SR** with mixed WE. This is relevant since a pure strategy NE may not exist in this setting [13]. Moreover, it is unclear whether the smoothness framework of Roughgarden [31] can be applied here: On the one hand our results hold for the more general framework of WE, while on the other hand having players that control multiple jobs makes it more difficult to prove the (λ, μ)-smoothness.

Before designing improved policies we observe that no anonymous policy may obtain a coordination ratio better than 4, and basically no policy, be it preemptive or randomized, local or strongly local, can achieve a coordination ratio better than 2.618. The latter is in sharp contrast with the case in which players control just one job where better ratios can be achieved with randomized policies [12]. Quite surprisingly we are able to design an "optimal" policy, which we call *externality* (**EX**), that guarantees a coordination ratio of 2.618 for WE. Under this **EX** policy, jobs are processed according to Smith rule but are held back (and not released) for some additional time after completion. This additional time basically equals the negative externality that this particular job imposes over other players. Additionally, we prove that **EX** defines a potential game, so that pure WE exists, and that the convergence time is polynomial. It is worth mentioning that in the single-job game **EX** coincides with the proportional-sharing (**PS**) policy [12], which in turn extends the EQUI policy of the unit-weight case [16]. On the other side, when a single player controls all jobs, **EX** coincides with **SR**. The idea of making jobs incorporate the externality they impose has also been used by Heydenreich et al. [20]. However their goal is different; they incorporate the externality in the form of payments to obtain truthful mechanisms rather than to improve efficiency.

Interestingly, our result for **EX** in case just one player controls all jobs implies a tight approximation guarantee of 2.618 for local optima under the jump neighborhood for $R| \,|\sum w_j C_j$. This tight guarantee also holds for the *swap* neighborhood, in which one is additionally allowed to swap jobs between machines so long as the objective function value decreases [39]. In addition, our fast convergence result for **EX** implies another new result, namely, that local search with the jump neighborhood, when only maximum gain steps are taken, converges in polynomial time. These facts appear to be quite surprising since, despite the very large amount of work on local search heuristics for scheduling problems [7, 29], performance guarantees, or polynomial time convergence results are are only known for identical machines [4].

Methodologically our work is based on the inner product framework of [12], but more is needed to deal with the multi-job environment. Our main contribution is however conceptual: On the one hand, we demonstrate that the natural economic idea of externalities leads to approximately optimal, and in a way best possible, outcomes even in decentralized systems with only partial information (in a full information and centralized setting one can easily design policies leading to optimal outcomes). On the other hand, we provide the first direct application of purely game-theoretic ideas to the analysis of natural and well studied local search heuristics that lead to the currently best known results.

Preliminaries. Recall that for a player $\alpha \in \mathcal{A}$, the set of job she controls is denoted by $J(\alpha) \subset \mathcal{J}$. Moreover, $\alpha(j)$ denotes the player controlling job j, so that $J(\alpha(j))$ is the set of jobs controlled by who is controlling j.

A *pure strategy profile* is a matrix $\mathbf{x} \in \{0,1\}^{\mathcal{M}\times\mathcal{J}}$ in which $\mathbf{x}_{ij} = 1$ if job j is assigned to machine i. By denoting $\mathbf{x}^\alpha$ the columns of $\mathbf{x}$ corresponding to jobs controlled by player α we say that $\mathbf{x}^\alpha$ is a pure strategy for this player. A mixed strategy for player α is a probability distribution over all $\mathbf{x}^\alpha \in \{0,1\}^{\mathcal{M}\times J(\alpha)}$. A set of mixed strategies for all players $\alpha \in \mathcal{A}$ leads to a (mixed) strategy profile $\mathbf{x} \in [0,1]^{\mathcal{M}\times\mathcal{J}}$ where $\mathbf{x}_{ij}$ is the probability of job j assigned to machine i. Note that the distributions of the different columns of $\mathbf{x}$ may not be independent. We denote by $\mathbf{x}_{-k}$ the matrix obtained by deleting the $k-$th column of $\mathbf{x}$. Observe that $\mathbf{x}_{-k}$ results from the joint probability distribution of all jobs $j' \neq k$ according to $\mathbf{x}$. More precisely $\mathbf{x}_{-k} \in [0,1]^{\mathcal{M}\times\mathcal{J}\setminus\{k\}}$ can be equivalently seen as the mixed strategy profile obtained when players different from $\alpha(k)$ continue using the same strategy, while player $\alpha(k)$ forgets job k and if she was playing the pure strategy $\mathbf{x}^{\alpha(k)} \in \{0,1\}^{\mathcal{M}\times J(\alpha)}$ with probability q, she plays the pure strategy for her jobs different from k, $\mathbf{x}_{-k}^{\alpha(k)} \in \{0,1\}^{\mathcal{M}\times J(\alpha)\setminus\{k\}}$ with probability q (these probabilities add up if she was playing with positive probability two strategies that were equal except for job k). We define $\mathbf{x}_{-K}$ analogously for a set of jobs $K \subseteq \mathcal{J}$.

Given a mechanism $\mathbf{M} \in \{\mathbf{SR}, \mathbf{EX}\}$ and a strategy profile $\mathbf{x}$, $\mathbb{E}[C_j^{\mathbf{M}}(\mathbf{x})]$ is the expected completion time of job j. The conditional expected completion time of job j on machine i when job k is assigned to machine i is denoted $\mathbb{E}[C_j^{\mathbf{M}}(\mathbf{x}_{-k}, k \to i)]$. The expected cost of the strategy profile $\mathbf{x}$ is $\mathbb{E}[C^{\mathbf{M}}(\mathbf{x})] = \sum_{j\in\mathcal{J}} w_j \mathbb{E}[C_j^{\mathbf{M}}(\mathbf{x})]$ and the expected cost of a player α under $\mathbf{x}$ is $\mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{x})] = \sum_{j\in J(\alpha)} w_j \mathbb{E}[C_j^{\mathbf{M}}(\mathbf{x})]$. For convenience we also define $\mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{x}_{-k}, k \to i)] = \sum_{j\in J(\alpha)} w_j \mathbb{E}[C_j^{\mathbf{M}}(\mathbf{x}_{-k}, k \to i)]$. Note that $\mathbb{E}[C^{\mathbf{M}}(\mathbf{x})] = \sum_{\alpha\in\mathcal{A}} \mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{x})]$.

A Nash equilibrium (NE) is therefore a strategy profile $\mathbf{x}$ such that for all player $\alpha \in \mathcal{A}$ and all strategy profiles $\mathbf{y}^\alpha$ for player α we have that:

$$\mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{x})] \leq \mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{y}^\alpha, \mathbf{x}_{-J(\alpha)})].$$

Similarly, a weak equilibrium (WE) is a strategy profile $\mathbf{x}$ such that for all player $\alpha \in \mathcal{A}$, all jobs $k \in J(\alpha)$, and all machines $i \in \mathcal{M}$, we have that:

$$\mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{x})] \leq \mathbb{E}[C_\alpha^{\mathbf{M}}(\mathbf{x}_{-k}, k \to i)].$$

The optimal assignement is the assignment in which the jobs are processed non-preemptively on the machines so that the cost is minimized. Throughout the paper, $\mathbf{x}^*$ denotes the optimal assignment (thus $\mathbf{x}^*$ is a pure strategy), and we define X_i^* as the set of jobs assigned to machine i under the optimal assignment. Given the assignment of jobs to machines, it is well-known that Smith Rule minimizes the total cost of jobs. Therefore $C^{\mathbf{SR}}(\mathbf{x}^*)$ is the optimal cost.

2 Nonpreemptive mechanisms

We now study nonpreemptive mechanisms (jobs have IDs, needed to break ties between identically looking jobs) and prove that SR has a coordination ratio of 4 for mixed WE. We work with mixed strategies since SR does not guarantee that existence of pure WE. As mentioned earlier, our result is best possible among nonpreemptive mechanisms [12].

Recall that under SR, each machine i schedules nonpreemptively its assigned jobs j in nondecreasing order of $\rho_{ij} = p_{ij}/w_j$, and ties are broken using the IDs. To simplify the presentation, we say that $\rho_{ik} < \rho_{ij}$ if k comes earlier than j in the SR order of machine i. Thus, given a strategy profile $\mathbf{x}$ we have $\mathbb{E}[C_j^{\text{SR}}(\mathbf{x}_{-j}, j \to i)] = p_{ij} + \sum_{k:\rho_{ik}<\rho_{ij}} \mathbf{x}_{ik} p_{ik}$ so that,

$$\begin{aligned}\mathbb{E}\left[C^{\text{SR}}(\mathbf{x})\right] &= \sum_{j\in\mathcal{J}} w_j \sum_{i\in\mathcal{M}} \mathbf{x}_{ij} \mathbb{E}[C_j^{\text{SR}}(\mathbf{x}_{-j}, j \to i)] \\ &= \sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{J}} \mathbf{x}_{ij} w_j (p_{ij} + \sum_{k:\rho_{ik}<\rho_{ij}} \mathbf{x}_{ik} p_{ik}).\end{aligned} \tag{1}$$

Extending the inner product space technique of Cole et al. [12], we let $\varphi : \mathbf{x} \to L_2([0,\infty])^{\mathcal{M}}$, which maps every strategy profile $\mathbf{x}$ to a vector of functions (one for each machine) as follows. If $\boldsymbol{f} = \varphi(\mathbf{x})$, then for each $i \in \mathcal{M}$, the i-th component of $\boldsymbol{f}$ is the function $f_i(y) := \sum_{j\in\mathcal{J},\rho_{ij}\geq y} \mathbf{x}_{ij} w_j$. Letting $\langle f_i, g_i\rangle = \int_0^\infty f_i(y) g_i(y) dy$ be the standard inner product on L_2 we get that $\langle \boldsymbol{f}, \mathbf{g}\rangle = \sum_{i\in\mathcal{M}} \langle f_i, g_i\rangle$. Additionally, we let $\eta_i(\mathbf{x}) = \sum_{j\in\mathcal{J}} w_j \mathbf{x}_{ij} p_{ij}$ and $\eta(\mathbf{x}) = \sum_{i\in\mathcal{M}} \eta_i(\mathbf{x})$.

The next lemma and expressions (2) and (3) follow easily from the derivations of Cole et al. [12]. The only difference is that here we need to prove the results for mixed strategies. We defer the proofs of this section to the full version.

Lemma 1. *For a strategy profile* $\mathbf{x}$ *and the optimal assignment* $\mathbf{x}^*$*, let* $\boldsymbol{f} = \varphi(\mathbf{x})$ *and* $\boldsymbol{f}^* = \varphi(\mathbf{x}^*)$*. Then* $\langle f_i, f_i^*\rangle = \sum_{j\in X_i^*} \sum_{k\in\mathcal{J}} w_j w_k \mathbf{x}_{ik} \min\{\rho_{ij}, \rho_{ik}\}$.

Similarly to Lemma 1, and using equation (1), we may evaluate

$$||\varphi(\mathbf{x})||^2 \leq 2\mathbb{E}\left[C^{\text{SR}}(\mathbf{x})\right]. \tag{2}$$

Additionally, when $\mathbf{x}$ is a pure strategy we have:

$$C^{\text{SR}}(\mathbf{x}) = \frac{1}{2}||\varphi(\mathbf{x})||^2 + \frac{1}{2}\eta(\mathbf{x}). \tag{3}$$

In what follows, let $\mathbf{x}$ denote a mixed weak equilibrium and $\mathbf{x}^*$ the optimal assignment. Let $\boldsymbol{f} = \varphi(\mathbf{x})$ and $\boldsymbol{f}^* = \varphi(\mathbf{x}^*)$.

Lemma 2. *Consider* $X_i^*(\alpha) = X_i^* \cap J(\alpha)$*, the jobs of player* α *assigned to machine* i *in the optimal solution. Then for each* $j \in X_i^*(\alpha)$ *we have:*

$$w_j \mathbb{E}\left[C_j^{\text{SR}}(\mathbf{x})\right] \leq w_j(p_{ij} + \sum_{k:\rho_{ik}<\rho_{ij}} \mathbf{x}_{ik} p_{ik}) + p_{ij} \sum_{k\in J(\alpha)\setminus\{j\},\rho_{ik}>\rho_{ij}} w_k \mathbf{x}_{ik}.$$

Lemma 3. *For a machine* $i \in \mathcal{M}$, $\sum_{j\in X_i^*} w_j \mathbb{E}[C_j^{\text{SR}}(\mathbf{x})] - \eta_i(\mathbf{x}^*) \leq \langle f_i, f_i^*\rangle$.

Theorem 1. $\mathbb{E}[C^{\text{SR}}(\mathbf{x})] \leq 4C^{\text{SR}}(\mathbf{x}^*)$.

3 Preemptive mechanisms

Finding policies that beat the coordination ratio of 4 for WE is impossible if we restrict to nonpreemptive ones. This holds even for the single-job game [12], where WE and NE coincide. Therefore we need to consider preemptive or randomized policies. We first observe that even with preemption, if we restrict to anonymous policies, beating the ratio of 4 is not possible. Furthermore, we prove that the absolute limit for basically any policy, be it preemptive or randomized, using even global information, and even if different machines use different policies, is $1 + \phi \approx 2.618$, where ϕ is the golden ratio. The precise set of policies for which this lower bound holds are those such that when machine $i \in M$ is assigned a single job, $j \in J$, then $C_j = p_{ij}$.

As the performance of **SR** coincides in the single-job and multi-job games one may wonder whether natural preemptive policies, that work well in the single-job game, also do in the multi-job game. Unfortunately this is not the case. Indeed we prove that the champion preemptive policy for the single-job game, Proportional-sharing [12, 16], has a coordination ratio of at least 5.848 for WE and at least 2.848 for NE. It is thus rather surprising that we can actually achieve this ratio with a fairly natural policy, externality (**EX**). A key ingredient of this policy is that it heavily relies on the ownership of the jobs, a feature that policies for the single-job game certainly do not share.

The results in this section are presented for pure strategy profiles. This is primarily done for simplicity and also because, as we will show later, our preemptive policy induces a potential game and therefore pure WE are guaranteed to exist. Thus, given a pure strategy profile $\mathbf{x}$, we may refer to $\mathbf{x}$ as an assignment, and we may let X_i denote the set of jobs assigned to machine i under $\mathbf{x}$, i.e., $j \in X_i$ if $\mathbf{x}_{ij} = 1$. Let also $X_i(\alpha) = X_i \cap J(\alpha)$ be the set of jobs controlled by player α on this machine i under $\mathbf{x}$.

Recall that in the proportional sharing policy (**PS**) [12], the machine processing power is split among the assigned jobs proportionally to their weight. Given an assignment $\mathbf{x}$, if job j is assigned to machine i, it can be observed that:

$$C_j^{\mathbf{PS}}(\mathbf{x}) = p_{ij} + \sum_{k \in X_i, \rho_{ik} < \rho_{ij}} p_{ik} + p_{ij} \sum_{k \in X_i \setminus \{j\}, \rho_{ik} > \rho_{ij}} \frac{w_k}{w_j}.$$

Proposition 1 ([12]). *Given an assignment* $\mathbf{x}$, $C^{\mathbf{PS}}(\mathbf{x}) = ||\varphi(\mathbf{x})||^2$.

In our externality policy, **EX**, given an assignment $\mathbf{x}$, the machine processes the jobs according to **SR** but once a job is completed, it is delayed for an amount of time accounting for the negative externality it is imposing on other players. Thus in **EX** the cost for the owner of job j due to this job will be

$$w_j C_j^{\mathbf{EX}}(\mathbf{x}) = w_j p_{ij} + w_j \sum_{k \in X_i, \rho_{ik} < \rho_{ij}} p_{ik} + p_{ij} \sum_{k \in X_i \setminus J(\alpha(j)), \rho_{ik} > \rho_{ij}} w_k.$$

The completion time of j is then defined by the previous equation. Observe that in the single-job game, **EX** reduces to **PS**, while if all jobs are controlled by a single player **EX** reduces to **SR**. Also, **EX** induces feasible schedules since no completion time can be smaller than that given by Smith-rule. Policy **EX** can be seen as a preemptive policy in which jobs are processed as in **SR**, except for an infinitesimal piece that is processed at the time defined by previous equation. Moreover **EX** is strongly local and nonanonymous. A consequence of the definitions of **SR**, **PS**, and **EX** is that for a fixed assignment $\mathbf{x}$ their costs satisfy:

$$\begin{aligned} C^{\mathbf{EX}}(\mathbf{x}) &= C^{\mathbf{SR}}(\mathbf{x}) + \sum_{i\in\mathcal{M}}\sum_{j\in X_i} p_{ij} \sum_{k\in X_i\setminus J(\alpha(j)),\rho_{ik}>\rho_{ij}} w_k \qquad (4)\\ &= C^{\mathbf{PS}}(\mathbf{x}) - \sum_{i\in\mathcal{M}}\sum_{j\in X_i} p_{ij} \sum_{k\in X_i(\alpha(j)),\rho_{ik}>\rho_{ij}} w_k. \end{aligned}$$

In the following, let $\mathbf{x}^*$ be an optimal assignment and $\mathbf{x}$ a WE. We also let $\varphi(\mathbf{x}) = \boldsymbol{f}$ and $\varphi(\mathbf{x}^*) = \boldsymbol{f}^*$ be as in the previous section.

Lemma 4. *Consider a job $j \in X_i^*$ and assume j is on i' under $\mathbf{x}$. Then*

$$w_j C_j^{\mathbf{EX}}(\mathbf{x}) \le w_j(p_{ij} + \sum_{\substack{k\in X_i,\\ \rho_{ik}<\rho_{ij}}} p_{ik}) + p_{ij}\sum_{\substack{k\in X_i,\\ \rho_{ik}>\rho_{ij}}} w_k - p_{i'j}\sum_{\substack{k\in X_{i'}(\alpha(j)),\\ \rho_{i'k}>\rho_{i'j}}} w_k.$$

Proof. The case $i' = i$ is immediate. For $i' \neq i$, consider the cost of jobs belonging player $\alpha(j)$ on machines i or i' under $\mathbf{x}$, which is,

$$w_j C_j^{\mathbf{EX}}(\mathbf{x}) + \sum_{k\in((X_i(\alpha)\cup X_{i'}(\alpha))\setminus\{j\}} w_k C_k^{\mathbf{EX}}(\mathbf{x}). \qquad (5)$$

Suppose that she moves j from machine i' to i, then the total cost of the same set of jobs is

$$\begin{aligned} &\sum_{k\in((X_i(\alpha)\cup X_{i'}(\alpha))\setminus\{j\}} w_k C_k^{\mathbf{EX}}(\mathbf{x}) - p_{i'j}\sum_{k\in X_{i'}(\alpha(j)),\rho_{i'k}>\rho_{i'j}} w_k + \\ &w_j(p_{ij} + \sum_{\substack{k\in X_i,\\ \rho_{ik}<\rho_{ij}}} p_{ik}) + p_{ij}\sum_{\substack{k\in X_i\setminus J(\alpha(j)),\\ \rho_{ik}>\rho_{ij}}} w_k + p_{ij}\sum_{\substack{k\in X_i(\alpha(j)),\\ \rho_{ik}>\rho_{ij}}} w_k. \qquad (6) \end{aligned}$$

Here the second term is the saving of the cost for those jobs $k \in \alpha(j)$ on machine i' that have larger ratios $\rho_{i'k}$ than $\rho_{i'j}$; the third and fourth terms are the cost of job j on machine i; and the fifth term is the increase of the cost of those jobs $k \in \alpha(j)$ on machine i that have larger ratios ρ_{ik} than ρ_{ij}. As $\mathbf{x}$ is a WE, the term (5) is upper bounded by (6). □

Lemma 5. $C^{\mathbf{EX}}(\mathbf{x}) \le \eta(\mathbf{x}^*) + \langle \boldsymbol{f}, \boldsymbol{f}^*\rangle - \sum_{i\in\mathcal{M}}\sum_{j\in X_i} p_{ij}\sum_{k\in X_i(\alpha(j)),\rho_{ik}>\rho_{ij}} w_k.$

Proof. By Lemma 4 and summing over all jobs in $\mathcal{J}$, we have that the total cost under **EX**, $\sum_{j\in\mathcal{J}} w_j C_j^{\mathbf{EX}}(\mathbf{x})$ is upper bounded by

$$\eta(\mathbf{x}^*) + \sum_{i\in\mathcal{M}}(\sum_{j\in X_i^*} w_j \sum_{\substack{k\in X_i,\\ \rho_{ik}<\rho_{ij}}} p_{ik} + \sum_{j\in X_i^*} p_{ij}\sum_{\substack{k\in X_i,\\ \rho_{ik}>\rho_{ij}}} w_k - \sum_{j\in X_i} p_{ij}\sum_{\substack{k\in X_i(\alpha(j)),\\ \rho_{ik}>\rho_{ij}}} w_k). \qquad (7)$$

By Lemma 1 and the fact that $\mathbf{x}$ is pure, we have

$$\langle f_i, f_i^* \rangle = \sum_{j \in X_i^*} \sum_{k \in X_i} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} = \sum_{j \in X_i^*} (w_j \sum_{\substack{k \in X_i, \\ \rho_{ik} \leq \rho_{ij}}} p_{ik} + p_{ij} \sum_{\substack{k \in X_i, \\ \rho_{ik} > \rho_{ij}}} w_k).$$

Summing over $i \in \mathcal{M}$ and subtracting the latter from (7)

$$C^{\mathbf{EX}}(\mathbf{x}) - \langle \boldsymbol{f}, \boldsymbol{f}^* \rangle \leq \eta(\mathbf{x}^*) - \sum_{i \in \mathcal{M}} \sum_{j \in X_i} p_{ij} \sum_{k \in X_i(\alpha(j)), \rho_{ik} > \rho_{ij}} w_k. \quad \square$$

Theorem 2. *Let ϕ be the golden ratio. Then $C^{\mathbf{EX}}(\mathbf{x}) \leq (1 + \phi) C^{\mathbf{SR}}(\mathbf{x}^*)$.*

Proof. Lemma 5 and Cauchy-Schwartz inequality imply that for $\beta > 1/4$

$$\begin{aligned} C^{\mathbf{EX}}(\mathbf{x}) &\leq \eta(\mathbf{x}^*) + \beta ||\boldsymbol{f}^*||^2 + \frac{1}{4\beta} ||\boldsymbol{f}||^2 - \sum_{i \in \mathcal{M}} \sum_{j \in X_i} p_{ij} \sum_{k \in X_i(\alpha(j)), \rho_{ik} > \rho_{ij}} w_k \\ &\leq \eta(\mathbf{x}^*) + \beta ||\boldsymbol{f}^*||^2 + \frac{1}{4\beta} ||\boldsymbol{f}||^2 - \frac{1}{4\beta} \sum_{i \in \mathcal{M}} \sum_{j \in X_i} p_{ij} \sum_{k \in X_i(\alpha(j)), \rho_{ik} > \rho_{ij}} w_k \\ &\leq \eta(\mathbf{x}^*) + 2\beta C^{\mathbf{SR}}(\mathbf{x}^*) - \beta \eta(\mathbf{x}^*) + \frac{1}{4\beta} C^{\mathbf{EX}}(\mathbf{x}) \\ &\leq (\beta + 1) C^{\mathbf{SR}}(\mathbf{x}^*) + \frac{1}{4\beta} C^{\mathbf{EX}}(\mathbf{x}), \end{aligned}$$

where the third inequality follows from equation (3), from Proposition 1 and from equation (4). By letting $\beta = \frac{1+\sqrt{5}}{4}$ the result follows. $\square$

As mentioned earlier, it turns out that **EX** is best possible. The proof of this fact is deferred to the full version.

Theorem 3. *The coordination ratio for weak equilibrium of any prompt mechanism is at least $1 + \phi$.*

4 Final remarks

We have proved that **SR** is the best possible nonpreemptive policy, and to beat its coordination ratio we have used **EX**, a policy that, as opposed to **SR**, importantly relies on who owns which job. We conjecture that if we restrict to policies that ignore the ownership of the jobs the ratio of 4 cannot be improved. This is indeed the case for nonpreemptive policies, and also for fully preemptive policies. Also, for natural policies with this property such as **PS** or the RAND policy [12] the technique in this paper only lead to larger bounds.

Our lower bound on general prompt seems to be the natural limit. Non-prompt policies that are allowed to use global information can certainly beat this as they can simply introduce very large delays for jobs that are not assigned to it in an optimal schedule. By doing this, such policies can easily achieve low coordination ratio (say optimal if they have unlimited computational power or 3/2 if they use the best known approximation algorithms. It would be interesting to explore what happens with this non prompt policies when they can only use local information.

Another interesting question refers to the quality of the actual NE of this game. Of course our upper bounds applies to that equilibrium concept, and furthermore we know that the coordination ratio of **EX** for NE is exactly 2.618 as in the single job case it coincides with **PS** [12]. However it may be possible that another deterministic policy has a better coordination ratio for NE.

Finally, we note that by mimicking the analysis in [12] we obtain a similar $2+\varepsilon$ approximation algorithm for $R||\sum w_j C_j$, independent of which jobs belong to which players. It is possible that by carefully choosing the game structure this can be beaten.

References

1. F. Abed, C.-C. Huang. Preemptive coordination mechanisms for unrelated machines. In *ESA* 2012.
2. Y. Azar, K. Jain, and V.S. Mirrokni. (Almost) Optimal coordination mechanisms for unrelated machine scheduling. In *SODA* 2008.
3. S. Bhattacharyay, S. Imz, J. Kulkarnix, K. Munagala. Coordination mechanisms from (almost) all scheduling policies. In *ITCS* 2014.
4. T. Brueggemann, J.L. Hurink, W. Kern. Quality of move-optimal schedules for minimizing total weighted completion time. *Oper. Res. Lett.*, 34(5):583-590, 2006.
5. J. Bruno, E.G. Coffman, and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Commun. ACM*, 17:382–387, 1974.
6. I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *SODA* 2009.
7. B. Chen, C.N. Potts, G.J. Woeginger. A review of machine scheduling: Complexity, algorithms and approximability. In Handbook of Combinatorial Optimization (volume 3), Kluwer Academic Publishers. 1998.
8. I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. *Algorithmica*, 61(3):606–637, 2011.
9. G. Christodoulou, E. Koutsoupias, A. Nanavati. Coordination mechanisms. *Theor. Comput. Sci.*, 410(36):3327–3336, 2009. Preliminary version in *ICALP* 2004.
10. J. Cohen, C. Dürr, N.K. Thang. Smooth inequalities and equilibrium inefficiency in scheduling games. In *WINE* 2012.
11. R. Cole, J.R. Correa, V. Gkatzelis, V.S. Mirrokni, N. Olver Inner product spaces for MinSum coordination mechanisms. In *STOC* 2011.
12. R. Cole, J.R. Correa, V. Gkatzelis, V. Mirrokni, N. Olver. Decentralized utilitarian mechanisms for scheduling games. *Game. Econ. Behav.*, to appear.
13. J.R. Correa, M. Queyranne. Efficiency of equilibria in restricted uniform machine scheduling with total weighted completion time as social cost. *Naval Res. Logist.*, 59:384–395, 2012.
14. A. Czumaj, B. Vöcking. Tight bounds for worst-case equilibria. *ACM T. Algo.*, 3, 2007.

15. E. Davis, J.M. Jaffe. Algorithms for scheduling tasks on unrelated processors. *J. ACM*, 28(4):721–736, 1981.
16. C. Dürr, N.K. Thang. Non-clairvoyant scheduling games. In *SAGT* 2009.
17. B. Farzad, N. Olver, A. Vetta. A priority-based model of routing. *Chic. J. Theor. Comput.*, 2008.
18. G. Finn, E. Horowitz. A linear time approximation algorithm for multiprocessor scheduling. *BIT*, 19:312–320, 1979.
19. L. Fleischer, Z. Svitkina. Preference-constrained oriented matching. In *ANALCO* 2010.
20. B. Heydenreich, R. Müller, M. Uetz. Mechanism Design for Decentralized Online Machine Scheduling. *Oper. Res.*, 58(2):445-457, 2010.
21. H. Hoogeveen, P. Schuurman, G.J. Woeginger. Non-approximability results for scheduling problems with minsum criteria. In *IPCO* 1998.
22. R. Hoeksma, M. Uetz. The Price of Anarchy for Minsum related machine scheduling. In *WAOA* 2011.
23. W.A. Horn. Minimizing average flow time with parallel machines. *Oper. Res.*, 21(3):846–847, 1973.
24. O.H. Ibarra, C.E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *J. ACM*, 24(2):280–289, 1977.
25. N. Immorlica, L. Li, V.S. Mirrokni, and A.S. Schulz. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410(17):1589–1598, 2009.
26. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS* 1999.
27. Lu, P., Yu, C.: Worst-case Nash equilibria in restricted routing. In *WINE*, 231–238, 2008.
28. J. Nash. Equilibrium points in N-person games. *PNAS*, 36, 48-49, 1950.
29. C.N. Potts, V. Strusevich. Fifty years of scheduling: a survey of milestones. *J Oper. Res. Society*, 60(1):41-68, 2009.
30. M. Rahn, G. Schäfer. Bounding the inefficiency of altruism through social contribution games. Manuscript 2013.
31. T. Roughgarden. Intrinsic robustness of the price of anarchy. In *STOC* 2009.
32. S. Sahni, Y. Cho. Bounds for list schedules on uniform processors. *SIAM J. Comput.*, 9:91–103, 1980.
33. A.S. Schulz, M. Skutella. Scheduling unrelated machines by randomized rounding. *SIAM J. Discrete Math.*, 15(4):450–469, 2002.
34. P. Schuurman, T. Vredeveld. Performance guarantees of local search for multiprocessor scheduling. In *IPCO* 2001.
35. J. Sethuraman, M.S. Squillante. Optimal scheduling of multiclass parallel machines. In *SODA* 1999.
36. M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *J. ACM*, 48(2):206–242, 2001.
37. M. Skutella, G.J. Woeginger. A ptas for minimizing the total weighted completion time on identical parallel machines. *Math. Oper. Res.*, 25(1):63–75, 2000.
38. W. Smith. Various optimizers for single stage production. *Naval Res. Logist. Quart.*, 3(1-2):59–66, 1956.
39. T. Vredeveld, C. Hurkens. Experimental comparison of approximation algorithms for scheduling unrelated parallel machines. *INFORMS J. Comput.*, 14:175–189, 2002.