

Inner Product Spaces for MinSum Coordination Mechanisms *

Richard Cole
Courant Institute
New York University
cole@cims.nyu.edu

José R. Correa
Dep. de Ingeniería Industrial
Universidad de Chile
jcorrea@dii.uchile.cl

Vasilis Gkatzelis
Courant Institute
New York University
gkatz@cims.nyu.edu

Vahab Mirrokni
Google Research
New York
mirrokni@google.com

Neil Olver
Department of Mathematics,
MIT
olver@math.mit.edu

ABSTRACT

We study coordination mechanisms aiming to minimize the weighted sum of completion times of jobs in the context of selfish scheduling problems. Our goal is to design local policies that achieve a good price of anarchy in the resulting equilibria for unrelated machine scheduling. To obtain these approximation bounds, we introduce a new technique that while conceptually simple, seems to be quite powerful. The method entails mapping strategy vectors into a carefully chosen inner product space; costs are shown to correspond to the norm in this space, and the Nash condition also has a simple description. With this structure in place, we are able to prove a number of results, as follows.

First, we consider Smith's Rule, which orders the jobs on a machine in ascending processing time to weight ratio, and show that it achieves an approximation ratio of 4. We also demonstrate that this is the best possible for deterministic non-preemptive strongly local policies. Since Smith's Rule is always optimal for a given fixed assignment, this may seem unsurprising, but we then show that better approximation ratios can be obtained if either preemption or randomization is allowed.

We prove that `ProportionalSharing`, a preemptive strongly local policy, achieves an approximation ratio of 2.618 for the weighted sum of completion times, and an approximation ratio of 2.5 in the unweighted case. We also observe that these bounds are tight. Next, we consider `Rand`, a natural non-preemptive but *randomized* policy. We show that it achieves an approximation ratio of at most 2.13; moreover, if the sum of the weighted completion times is negligible compared to the cost of the optimal solution, this improves to $\pi/2$.

*This work was supported in part by FONDECYT grant 1090050, by NSF grant CCF0830516, and by Andreas Mentzelopoulos Scholarships for the University of Patras.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

STOC'11, June 6–8, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0691-1/11/06 ...\$10.00.

Finally, we show that both `ProportionalSharing` and `Rand` induce potential games, and thus always have a pure Nash equilibrium (unlike Smith's Rule). This allows us to design the first *combinatorial* constant-factor approximation algorithm minimizing weighted completion time for unrelated machine scheduling. It achieves a factor of $2 + \epsilon$ for any $\epsilon > 0$, and involves imitating best response dynamics using a variant of `ProportionalSharing` as the policy.

Categories and Subject Descriptors

F.2.2 [Analysis of Algorithms and Problem Complexity]:
Nonnumerical Algorithms & Problems - Sequencing & Scheduling

General Terms

Algorithms, Theory

Keywords

Coordination Mechanisms, Scheduling, Price of Anarchy, Approximation Algorithms

1. INTRODUCTION

Traditionally, work in operations research has focused on finding globally optimal solutions for optimization problems. In tandem, computer scientists have long studied the effects of a lack of different kinds of resources, mainly the lack of computational resources in optimization. In designing massive decentralized systems, the *lack of coordination* among different participating agents has become an important consideration. This issue is typically addressed through distributed algorithms in which a central authority designs mechanisms (protocols) specifying the rules of the game, with the goal that the independent and selfish choices of the users result in a socially desirable outcome. To measure the performance of these algorithms, the global objective function (social cost) is evaluated at equilibrium points for selfish users. For games, probably the most accepted such measure is the *price of anarchy* [38], the worst case ratio of the social cost at a Nash equilibrium to that at a social optimum; the same measure can be used for coordination mechanisms; sometimes we call this their *approximation factor* to highlight that this is a distinct measure.

The by now standard approach to bound the price of anarchy (PoA) when social cost is taken to be the sum of individual costs works as follows [41]. First, the social cost is bounded by using the equilibrium conditions, noting that an individual is better off at equilibrium than she would be if she unilaterally changed her strategy to the one she would use in a centralized optimum. Second, the actual (weighted) sum of player costs is also upper bounded, using an appropriately chosen inequality, by a linear combination of the social cost of the equilibrium and the social cost of an optimal solution.

In this paper we establish a methodology to deal with the second step of this proof scheme. Our method interprets the sum in the second step as an inner product on a suitable space. Then, we apply the Cauchy-Schwartz inequality in the chosen inner product space, and go back to the original space by applying a minimum norm distortion inequality. Many of the existing results employ a special case of this approach in which the costs can be expressed in terms of quadratic polynomials to which the Cauchy-Schwartz inequality can be applied directly without the need for an intermediate inner product space. We apply our new method in the context of scheduling jobs on unrelated machines. Our method elucidates the hidden structure in the games we consider. Once the framework has been set up, our proofs become short and elegant, thus we anticipate that this method may prove useful elsewhere too.

Specifically, we consider the classic problem of scheduling n jobs on m unrelated machines from a game theoretic perspective. In this situation, job j takes time p_{ij} if processed on machine i , and also has an associated weight w_j . Although the central goal is to minimize the weighted sum of completion times of jobs, we consider the *scheduling game* in which each job is a fully informed player wanting to minimize its individual weighted completion time, while each machine announces a policy which it will follow in processing the jobs it is assigned. Our goal is to choose the policy so as to minimize the approximation ratio of the actual costs under this policy to the optimal costs obtainable under any policy. To this end, several approaches imposing incentives on self-interested agents have been proposed, including some using monetary transfers [7, 17, 26, 12], and others enforcing strategies on a fraction of users as a Stackelberg strategy [6, 37, 40, 49]. Ultimately one could also apply a VCG mechanism to achieve social efficiency. The main drawback of these methods is the need for global knowledge of the system. A different approach, and the focus of our paper, uses coordination mechanisms [15], which only require local computations.

More formally, a coordination mechanism [15, 35, 4, 10, 22] is a set of *local policies*, one per machine, specifying how the jobs assigned to that machine are scheduled. Here, *local* means that a machine's schedule must be a function of information related only to the jobs it is assigned, allowing the policy to be implemented in a distributed fashion. We actually study *strongly local* policies, meaning that the schedule of any machine i is a function only of the processing times p_{ij} , weights w_j and IDs of jobs assigned to it; for example, the schedule cannot depend on the processing times of those jobs on other machines ($p_{i'j}$). It will also be useful (especially when considering lowerbounds) for us to restrict attention to policies that always use the full capacity of a

machine, and release jobs immediately upon completion. We call such policies *prompt*.

Several local policies have been studied for scheduling problems in the context of both greedy and local search algorithms [34, 25, 42, 21, 1, 5, 8, 50], as well as coordination mechanisms [38, 20, 15, 35, 4, 10, 22]. Previous work mainly considered the makespan social cost as opposed to the weighted sum of completion times addressed here.

Our Results.

Employing our new technique, we develop the first constant-factor approximate coordination mechanisms for the selfish machine scheduling problem for unrelated machines. We start by studying Smith's Rule [48], a policy according to which machines process jobs in increasing order of their processing time to weight ratio. Here the space that appropriately fits our method is L^2 and a norm distortion inequality is in fact not needed. We prove that the approximation factor for this policy is exactly 4, improving upon a result by Correa and Queyranne [19]. We also show that this is the best possible among all deterministic and non-preemptive strongly local coordination mechanisms, assuming the prompt property.

The constant approximation ratio for the weighted sum of completion times is in sharp contrast to the known super-constant inapproximability results for coordination mechanisms for the makespan function [4, 27] (e.g., an $\Omega(m)$ lower bound for the shortest-first coordination mechanism). In fact, it is still open whether there is a coordination mechanism with a constant approximation ratio for the makespan function.

Next, we go beyond the approximation ratio of 4 using preemptive¹ and randomized mechanisms. First, we consider a preemptive policy, generalizing that of Dürr and Thang [22], in which each machine splits its processing capacity among its assigned jobs in proportion to their weights. We uncover a close connection of this policy to Smith's Rule, allowing us to apply a similar proof strategy, but yielding a significantly improved approximation factor of 2.618. On the other hand, we prove that with anonymous jobs, no set of deterministic prompt policies, be they preemptive or not, can achieve a factor better than 2.5. To break this new barrier we consider a policy in which jobs are randomly, but non-uniformly, ordered, based on their processing time to weight ratio. Under this policy the appropriate space has to be carefully chosen and uses a rather nonstandard inner product, induced by a Hilbert matrix, whose i, j entry equals $1/(i + j - 1)$. A norm distortion inequality is then needed to relate the norm in this space to the original cost of an optimal schedule, leading to yet another improvement in the approximation factor to 2.134. Moreover, we show a lower bound of $5/3$ for this policy.

Finally, inspired by our preemptive mechanism, along with the β -nice notion of [3], we design a new *combinatorial* $(2 + \epsilon)$ -approximation algorithm for optimizing the weighted sum of completion times on unrelated machines. This improves on the approximation factor of our mechanisms and complements the known non-combinatorial constant-factor approximation algorithms: a linear programming based $16/3$ -approximation algorithm [30], then an improvement to $\frac{3}{2} + \epsilon$, again based on linear programming [43], and finally the best

¹By preemption we mean that the computation of a job is suspended and, implicitly, resumed later.

currently known factor, a $\frac{3}{2}$ -approximation based on a convex quadratic relaxation [45, 46].

We obtain a number of other results, most of which are discussed in the full version of the paper. For the unit weight case, using Smith’s Rule, we obtain a constant upper bound on the price of anarchy by a reduction from the priority routing model of Farzad et al. [24], as shown in the full version; however, the resulting bound is not optimal. In the unit weight case, our preemptive mechanism simplifies to one, called **EqualSharing** [22], in which jobs share the processing capacity of a machine equally. Then the approximation ratio is 2.5, which follows either by a careful analysis based on local moves, or using our method with a modified Cauchy-Schwartz inequality. In addition, in the case where the weighted sum of processing times is negligible compared to the total cost, our randomized policy has an approximation ratio of $\pi/2$, which is tight. The latter follows by an interesting norm distortion inequality obtained by Chung et al. [16], for which we provide an alternative shorter proof. Furthermore, although for the Smith’s Rule policy pure equilibria may not exist [19], we show that all our preemptive and randomized mechanisms result in exact potential games. This implies that best-response dynamics of players converge to pure Nash Equilibria (PNE) and verifies that PNE always exist.

While we present our results for pure strategies and thus pure Nash equilibria, it is easy to observe that all the bounds can be stated within the smoothness framework of Roughgarden [41]. This means that all our bounds apply to the *robust* PoA and therefore to more general equilibrium concepts including mixed Nash equilibria, correlated equilibria and coarse correlated equilibria. In other words, our proofs also imply upper bounds for the average social cost value that would result in a repeated game setting only assuming that each player was using some sort of regret-minimization strategy.

It is important to stress that these bounds are on the price of anarchy (or approximation ratio) of *coordination mechanisms* and not that of games; thus these results do not follow from seemingly similar bounds for selfish routing [2]. The fact that our preemptive policy performs better than non-preemptive ones is in contrast to existing results for the makespan social cost function where the **EqualSharing** policy achieves an approximation ratio of $\Theta(m)$ [22], no better than **ShortestFirst**, which schedules jobs in increasing order of their processing times (i.e., Smith’s Rule in the unweighted case). In order to explain this counter-intuitive result, we show that both our preemptive policy and our randomized policy penalize each job with an extra charge beyond its cost under Smith’s Rule that is related to the externality that its scheduling causes. Throughout the paper we assume that jobs aim at minimizing their weighted completion time in the case of deterministic policies, and expected weighted completion time for randomized ones.

Other Related Work.

Scheduling problems have long been studied from a centralized optimization perspective. We adopt the standard three field notation $\alpha|\beta|\gamma$ [29]. The first parameter defines the machine model, the last specifies the objective function, while the second will not concern us in this paper.

Minimizing the (unweighted) sum of completion times is polynomial time solvable even for unrelated machines [33, 9]. For identical parallel machines ($P|\sum c_j$), the **ShortestFirst** policy leads to an optimal schedule at any pure Nash equilibrium² [18]. On the other hand, minimizing the weighted sum of completion times is NP-complete even for identical machines ($P|\sum w_j c_j$) [39]. Although the latter admits a PTAS [47], the general unrelated case ($R|\sum w_j c_j$) is APX-hard [32] and constant factor approximation algorithms have been proposed [30, 43, 45, 46].

Coordination mechanism design was introduced in the work of Christodoulou et al. [15]. They analyzed the **LongestFirst** policy w.r.t. makespan for identical machines ($P|C_{\max}$) and also studied a selfish routing game. Immorlica et al. [35] study four coordination mechanisms for different machine scheduling problems and survey the results for these problems. They further study the speed of convergence to equilibria and the existence of PNE for the **ShortestFirst** and **LongestFirst** policies. Azar, Jain, and Mirrokni [4] showed that the **ShortestFirst** policy and in fact any strongly local ordering policy (defined in Section 2) does not achieve an approximation ratio better than $\Omega(m)$. Additionally, they presented a non-preemptive local policy that achieves an approximation ratio of $O(\log m)$ and a policy that induces potential games and gives an approximation ratio of $O(\log^2 m)$. Caragiannis [10] subsequently showed an alternative $O(\log m)$ -approximate coordination mechanism that minimizes makespan for unrelated machine scheduling leading to potential games. Fleischer and Svitkina [27] show a lower bound of $\Omega(\log m)$ for all local ordering policies. It is still open whether there exists a coordination mechanism (even preemptive or randomized) achieving a constant approximation ratio for the makespan objective function.

More recently, Dürr and Thang proved that the **EqualSharing** policy results in potential games, and achieves a PoA of $\Theta(m)$ for $R|C_{\max}$. In the context of coordination mechanisms, an instance for which a preemptive policy has an advantage over non-preemptive ones was also shown by Caragiannis [10], who presented a local preemptive policy with approximation factor $O(\log m / \log \log m)$, beating the lower bound of $\Omega(\log m)$ for local ordering policies [27]. Correa and Queyranne [19] study the problem of minimizing the weighted sum of completion times, show that Smith’s Rule may induce games that do not have PNE, and that the price of anarchy under this policy is 4 in a more restricted environment than that considered here.

Coordination mechanisms are related to local search algorithms. Starting from a solution, a local search algorithm iteratively moves to a neighbor solution which improves the global objective. This is based on a neighborhood relation that is defined on the set of solutions. The local improvement moves in the local search algorithm correspond to the best-response moves of users in the game defined by the coordination mechanism. The speed of convergence and the approximation factor of local search algorithms for scheduling problems have been studied mainly for the makespan objective function [21, 23, 25, 34, 42, 44, 50, 1, 5]. Our combinatorial approximation algorithm for the weighted sum of completion times is the first local search algorithm for $R|\sum w_i C_i$ and is different from the previously studied algorithms for the makespan objective.

²In [35] it is shown that these equilibria are exactly the solutions generated by the shortest-first greedy algorithm.

2. PRELIMINARIES

Throughout this paper, let J be a set of n jobs to be scheduled on a set I of m machines. Let p_{ij} denote the processing time of job $j \in J$ on machine $i \in I$ and let w_j denote its weight (or importance or impatience). Our goal is to minimize the weighted sum of the completion times of the jobs, i.e. $\sum_{j \in J} w_j c_j$, where c_j is the completion time of job j . An assignment of jobs to machines is represented by a vector \mathbf{x} , where x_j denotes the machine to which job j is assigned.

The main scheduling model we study is *unrelated* machine scheduling ($R \mid \mid \sum w_j c_j$) in which the p_{ij} 's are arbitrary. Another model is the *restricted related* machines model in which each machine i has a speed q_i and each job j has a processing requirement p_j : job j can be scheduled only on a subset T_j of the machines, with processing time $p_{ij} = p_j/q_i$ for $i \in T_j$, and $p_{ij} = \infty$ otherwise. The *restricted identical* machines model is the special case of the restricted related machines model where all machines have the same speed.

A *coordination mechanism* is a set of local policies, one for each machine; the policy for a machine determines how to schedule the jobs assigned to it. The coordination mechanism thereby defines a game in which there are n agents (jobs) and each agent's strategy set is the set of machines I . Given an assignment \mathbf{x} , the disutility of job j is its weighted completion time $w_j c_j(\mathbf{x})$, as determined by the policy on the machine x_j . The goal of each job is to choose a strategy (i.e., a machine) that minimizes its disutility. A strategy profile \mathbf{x} is a *Nash equilibrium* if no player has an incentive to change strategy. Our goal is to design coordination mechanisms which give suitable incentives to the players, so that selfish behavior leads to equilibria with low social cost.

A game is a *potential game* if there exists a potential function over strategy profiles such that any player's deviation leads to a drop of the potential function if and only if its cost drops. A potential game is *exact* if after each move, the changes to the potential function and to the player's cost are equal. It is easy to see that a potential game always possesses a PNE.

We define a machine's policy to be *prompt* if the machine uses its full capacity and does not delay the release of any of its completed jobs. We say that a policy satisfies the *independence of irrelevant alternatives* or *IIA* property if for any pair of jobs, their relative ordering is independent of what other jobs are assigned to the machine. This property appears as an axiom in voting theory, bargaining theory and logic. Notice that deterministic non-preemptive policies with the IIA property can be described simply by a fixed ordering of all jobs; jobs are scheduled according to this order. Thus we call such policies *ordering* policies.

Here and throughout the paper, we use the shorthand notation ρ_{ij} for the ratio p_{ij}/w_j and $\phi = \frac{1+\sqrt{5}}{2}$ for the golden ratio. The coordination mechanisms we study in this paper use the same local policy on each machine, so henceforth we refer to a coordination mechanism using the name of the policy. The main policies we discuss are the following:

SmithRule [48]: Jobs on machine i are scheduled consecutively in increasing order of ρ_{ij} . In the unweighted case, this reduces to the *ShortestFirst* policy.

ProportionalSharing: Jobs are scheduled in parallel using time-multiplexing. At any moment in time, each uncompleted assigned job receives a fraction of the processor time

equal to its weight divided by the total weight of uncompleted jobs on the machine. In the unweighted case, this gives the *EqualSharing* policy.

Rand: This randomized policy has the property that for any two jobs j, j' assigned to machine i , the probability that job j is run before job j' is exactly $\frac{\rho_{ij'}}{\rho_{ij} + \rho_{ij'}}$. Thus larger (w.r.t. ρ_{ij}) jobs are more likely to appear later in the ordering. We show how to implement this policy in Section 4.2.

For any configuration \mathbf{x} , let $w_j c_j^\alpha(\mathbf{x})$ denote the cost for player j and $C^\alpha(\mathbf{x}) = \sum_{j \in J} w_j c_j^\alpha(\mathbf{x})$ denote the social cost, where $\alpha \in \{SR, PS, SF, ES, R\}$ will be the policy, namely *SmithRule*, *ProportionalSharing*, *ShortestFirst*, *EqualSharing* and *Rand*, respectively. Finally, slightly abusing notation, let $X_i = \{j \in J \mid x_j = i\}$ denote the set of jobs that have chosen machine i in configuration \mathbf{x} , and define X_i^* analogously for \mathbf{x}^* .

A local policy for machine i uses only the information about the jobs on the same machine i , but it can look at all the parameters of these jobs, including their processing times on other machines. By contrast, a *strongly local* policy may depend only on the processing time that these jobs have on this machine i . In order to quantify the inefficiency caused by the lack of coordination, we use the notion of *price of anarchy* [38], that is, the ratio between the social cost value of the worst Nash equilibrium and that of the social optimum. To be more precise, we are interested in upper bounds for the PoA of coordination mechanisms rather than the PoA of specific games. Applying [15] to the current context, the PoA of a coordination mechanism is defined to be the maximum ratio, taken over all the games G that the mechanism may induce, of the social cost of a Nash equilibrium of G divided by the optimum social cost achievable for the scheduling problem underlying G .

3. DETERMINISTIC NON-PREEMPTIVE COORDINATION MECHANISMS

It is known that given an assignment of jobs to machines, in order to minimize the weighted sum of completion times, *SmithRule* is optimal [48]. It is therefore only natural to consider this policy as a good first candidate. Our first theorem shows that using this rule will result in Nash equilibria with social cost at most a constant-factor of 4 away from the optimum.

Our analysis uses the map $\varphi : I^J \rightarrow L_2([0, \infty))^I$, which maps a configuration to a vector of functions as follows. We write $\mathbf{f} = \varphi(\mathbf{x})$, and define

$$f_i(y) = \sum_{j \in X_i: \rho_{ij} \geq y} w_j \quad (\text{recall that } \rho_{ij} = p_{ij}/w_j).$$

We assume for simplicity that all jobs have distinct ratios of processing time to weight. This assumption is just for simplicity; alternatively, we could introduce a tie breaking rule.

Let $\langle g, h \rangle := \int_0^\infty g(y)h(y)dy$ denote the usual inner product on L_2 , and in addition define $\langle \mathbf{f}, \mathbf{g} \rangle := \sum_{i \in I} \langle f_i, g_i \rangle$. In both cases, $\|\cdot\|$ refers to the induced norm. We also define

$$\eta(\mathbf{x}) = \sum_{j \in J} w_j p_{x_j j}.$$

We then have

LEMMA 3.1. For any configuration \mathbf{x} , the social cost of *SmithRule* is $C^{SR}(\mathbf{x}) = \frac{1}{2}\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}) \rangle + \frac{1}{2}\eta(\mathbf{x})$.

PROOF. Let $\mathbf{f} = \varphi(\mathbf{x})$. We have

$$\begin{aligned} \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}) \rangle &= \sum_{i \in I} \int_0^\infty f_i(y)^2 dy \\ &= \sum_{i \in I} \sum_{j \in X_i} \sum_{k \in X_i} w_j w_k \int_0^\infty \mathbf{1}_{\rho_{ij} \geq y} \mathbf{1}_{\rho_{ik} \geq y} dy \\ &= \sum_{i \in I} \sum_{j \in X_i} \sum_{k \in X_i} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} \\ &= \sum_{i \in I} \sum_{j \in X_i} w_j \left(2 \sum_{\substack{k \in X_i \\ \rho_{ik} \leq \rho_{ij}}} p_{ik} - p_{ij} \right) \\ &= 2C^{SR}(\mathbf{x}) - \eta(\mathbf{x}). \end{aligned}$$

The result follows. \square

THEOREM 3.2. The price of anarchy of *SmithRule* for unrelated machines ($R \parallel \sum w_j c_j$) is at most 4.

PROOF. Let \mathbf{x} and \mathbf{x}^* be two assignments, with \mathbf{x} a Nash equilibrium, and write $\mathbf{f} = \varphi(\mathbf{x})$, $\mathbf{f}^* = \varphi(\mathbf{x}^*)$. The completion time of job j if it deviates from \mathbf{x} according to \mathbf{x}^* is

$$c_j^{SR}(\mathbf{x}_{-j}, \mathbf{x}_j^*) = \sum_{\substack{k: x_k = x_j^* \\ \rho_{x_k k} < \rho_{x_j^* j}}} p_{x_k k} + p_{x_j^* j}.$$

Using the Nash condition, this yields

$$\begin{aligned} C^{SR}(\mathbf{x}) &= \sum_j w_j c_j^{SR}(\mathbf{x}) \leq \sum_j w_j c_j^{SR}(\mathbf{x}_{-j}, \mathbf{x}_j^*) \\ &= \sum_{i \in I} \sum_{j \in X_i^*} \left(\sum_{\substack{k \in X_i \\ \rho_{ik} < \rho_{ij}}} w_j w_k \frac{p_{ik}}{w_k} + w_j p_{ij} \right) \\ &\leq \sum_{i \in I} \sum_{j \in X_i^*} \sum_{k \in X_i} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} + \eta(\mathbf{x}^*) \\ &= \sum_{i \in I} \sum_{j \in X_i^*} \sum_{k \in X_i} w_j w_k \int_0^\infty \mathbf{1}_{\rho_{ij} \geq y} \mathbf{1}_{\rho_{ik} \geq y} dy + \eta(\mathbf{x}^*) \\ &= \langle \mathbf{f}^*, \mathbf{f} \rangle + \eta(\mathbf{x}^*). \end{aligned}$$

Now applying Cauchy-Schwartz, followed by the inequality $ab \leq a^2 + b^2/4$ for $a, b \geq 0$, we obtain

$$\begin{aligned} C^{SR}(\mathbf{x}) &\leq \|\mathbf{f}\| \|\mathbf{f}^*\| + \eta(\mathbf{x}^*) \\ &\leq \|\mathbf{f}^*\|^2 + \frac{1}{4} \|\mathbf{f}\|^2 + \eta(\mathbf{x}^*) \\ &\leq 2C^{SR}(\mathbf{x}^*) + \frac{1}{2} C^{SR}(\mathbf{x}) \quad \text{by Lemma 3.1.} \end{aligned}$$

Hence $C^{SR}(\mathbf{x}) \leq 4C^{SR}(\mathbf{x}^*)$. \square

The following result, proved in the full version, shows that (assuming promptness) no strongly local deterministic non-preemptive mechanism can do better than *SmithRule*. This also implies that the bound of Theorem 3.2 is tight.

THEOREM 3.3. The pure PoA of any strongly local deterministic non-preemptive prompt coordination mechanism is at least 4. This is true even for the case of restricted identical machines with unweighted jobs ($B \parallel \sum c_j$).

4. IMPROVEMENTS WITH PREEMPTION AND RANDOMIZATION

4.1 Preemptive Coordination Mechanism

In this section, we study the power of preemption and present *ProportionalSharing*, a preemptive mechanism that is strictly better w.r.t. the PoA than any strongly local deterministic non-preemptive prompt policy. These results create a clear dichotomy between such policies and *ProportionalSharing*. Additional advantages of this coordination mechanism are that, unlike *SmithRule*, it can handle anonymous jobs (jobs that do not have IDs), and the games it induces always possess PNE.

The improved PoA bounds may seem counter-intuitive at first since, given an assignment of jobs to machines, using *ProportionalSharing* instead of *SmithRule* only increases the social cost³ and doesn't decrease the cost of any player. A better understanding of this result can be obtained by observing that in our context, preemptive policies can be thought of (and also implemented⁴) as non-preemptive (non-prompt) policies. More specifically, jobs can be processed in an appropriate order non-preemptively, but possibly delayed past their completion time. From this perspective, *ProportionalSharing* can be implemented by using *SmithRule* to determine the processing order, but then holding each job back after it is completed by an amount exactly equal to the total delay it causes to the jobs that *SmithRule* schedules after it. This fact can be seen explicitly in the first equation of the upcoming Lemma 4.1. In this way, the interests of a player are aligned with those of the group by forcing it to "internalize its externalities", leading not only to better allocations but also to a better social cost, despite the extra charges. In our setting, one can think of using preemption as a natural way of implicitly enforcing taxes (delays) to the players.

LEMMA 4.1. Given an assignment \mathbf{x} , the weighted completion time of a job j on some machine i using *ProportionalSharing* (whether currently assigned there or not) is

$$\begin{aligned} w_j c_j^{PS}(\mathbf{x}) &= \sum_{\substack{k \in X_i \setminus \{j\} \\ \rho_{ik} \leq \rho_{ij}}} w_j p_{ik} + \sum_{\substack{k \in X_i \\ \rho_{ik} > \rho_{ij}}} w_k p_{ij} + w_j p_{ij} \\ &= \sum_{k \in X_i \setminus \{j\}} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} + w_j p_{ij}. \quad (1) \end{aligned}$$

PROOF. We notice that the completion time of job j is only affected by the amount of "work" that the processor has completed by that time and not by the way this processing time has been shared among the jobs. While job j is not complete, for each w_j units of processing time it receives, any other job k receives w_k units (assuming it is still being processed). Therefore, for job j and any job k such that $\rho_{ik} \leq \rho_{ij}$, we know that their whole processing demands, p_{ij} and p_{ik} respectively, have been served. Also, when job j is completed, the processing time spent on any job k with $\rho_{ik} > \rho_{ij}$ will be exactly $\frac{w_k p_{ij}}{w_j}$. Adding all these processing times and multiplying by player j 's weight, w_j gives the lemma. \square

³This is not the case for the makespan social cost function.

⁴Notice that this would technically disallow anonymous jobs.

THEOREM 4.2. *The price of anarchy of ProportionalSharing for unrelated machines ($R|\sum w_j c_j$) is at most $\phi + 1 = \frac{3+\sqrt{5}}{2} \approx 2.618$. This bound is tight even for the restricted related machines model.*

PROOF. By Lemma 4.1, we can see that for any assignment \mathbf{x} , we get $C^{PS}(\mathbf{x}) = 2C^{SR}(\mathbf{x}) - \eta(\mathbf{x}) = \|\varphi(\mathbf{x})\|^2$; note that the social cost for ProportionalSharing is less than twice that of SmithRule. Moreover, (1) is upper bounded by

$$\sum_{k \in X_i} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} + w_j p_{ij},$$

and so, using the Nash condition, for any equilibrium \mathbf{x} , and any other assignment \mathbf{x}^* ,

$$\begin{aligned} C^{PS}(\mathbf{x}) &= \sum_j w_j c_j^{PS}(\mathbf{x}) \leq \sum_j w_j c_j^{PS}(\mathbf{x}_{-j}, x_j^*) \\ &\leq \sum_j \left(\sum_{k: x_k = x_j^*} w_j w_k \min\{\rho_{x_j^* j}, \rho_{x_j^* k}\} + p_{x_j^* j} \right) \\ &= \langle \varphi(\mathbf{x}^*), \varphi(\mathbf{x}) \rangle + \eta(\mathbf{x}^*). \end{aligned}$$

This is identical to the equation we used (after an additional inequality) in the case of Smith's Rule.

Let \mathbf{x} be a Nash assignment, \mathbf{x}^* an optimal assignment w.r.t. Smith's Rule, and again define $\mathbf{f} = \varphi(\mathbf{x})$, $\mathbf{f}^* = \varphi(\mathbf{x}^*)$. Following the same method of analysis as for Smith's Rule, we obtain

$$\begin{aligned} C^{PS}(\mathbf{x}) &\leq \|\mathbf{f}\| \|\mathbf{f}^*\| + \eta(\mathbf{x}^*) \\ &\leq \alpha \|\mathbf{f}^*\|^2 + \frac{1}{4\alpha} \|\mathbf{f}\|^2 + \eta(\mathbf{x}^*) \\ &\leq 2\alpha C^{SR}(\mathbf{x}^*) + \frac{1}{4\alpha} C^{PS}(\mathbf{x}) + (1 - \alpha)\eta(\mathbf{x}^*) \\ &\leq (1 + \alpha)C^{SR}(\mathbf{x}^*) + \frac{1}{4\alpha} C^{PS}(\mathbf{x}), \end{aligned}$$

using that $\eta(\mathbf{x}^*) \leq C^{SR}(\mathbf{x}^*)$. Setting $\alpha = (1 + \sqrt{5})/4$ yields $C^{PS}(\mathbf{x}) \leq \frac{3+\sqrt{5}}{2} C^{SR}(\mathbf{x}^*)$.

The tightness of this bound follows from a construction in [11], where in fact they show that even if we consider the ratio $C^{PS}(\mathbf{x})/C^{PS}(\mathbf{x}^*)$, it can be arbitrarily close to $\phi + 1$. \square

For the case of equal weights, we obtain the following slightly improved bound. This result can be proved in our framework but using a variation of the Cauchy-Schwartz inequality derived from Lemma 4.4 below. However, we present here a different proof approach of independent interest.

THEOREM 4.3. *The price of anarchy of EqualSharing for unrelated machines ($R|\sum c_j$) is at most 2.5. This bound is tight even for the restricted related machines model.*

PROOF. We begin by proving the following lemma, which gives a tighter version of an inequality initially used by Christodoulou and Koutsoupias [14]:

LEMMA 4.4. *For any pair of non-negative integers k, k^* ,*

$$k^*(k+1) \leq \frac{5}{3} \frac{k^*(k^*+1)}{2} + \frac{1}{3} k^2.$$

PROOF. After some algebra, this translates to showing that for any pair of non-negative integers k and k^* ,

$$5k^{*2} + 2k^2 - 6k^*k - k^* \geq 0.$$

We start by taking the partial derivative of the LHS w.r.t. k , i.e. $4k - 6k^*$, from which we infer that for any given value of k^* , the LHS is minimized when $k = \frac{3}{2}k^*$. On substituting this into our inequality, we obtain

$$5k^{*2} + 2\left(\frac{3}{2}k^*\right)^2 - 6k^* \frac{3}{2}k^* - k^* \geq 0 \Rightarrow k^{*2} \geq 2k^*,$$

which is true for $k^* = 0$ and $k^* \geq 2$. For $k^* = 1$ our inequality becomes $k^2 - 3k + 2 \geq 0$ which is true for all non-negative integers k . \square

Now, using this lemma, we show that for any machine i

$$\sum_{j \in X_i^*} c_j^{ES}(\mathbf{x}_{-j}, x_j^*) \leq \frac{5}{3} \sum_{j \in X_i^*} c_j^{SF}(\mathbf{x}^*) + \frac{1}{3} \sum_{j \in X_i} c_j^{ES}(\mathbf{x}).$$

In order to show this, we first prove that this inequality only becomes tighter if for any two jobs $j, j' \in X_i \cup X_i^*$, their processing times on i are equal, i.e. $p_{ij} = p_{ij'}$. Assume that not all processing times are equal and let $Max_i = \{j \in X_i \cup X_i^* \mid \forall j' \in X_i \cup X_i^*, p_{ij} \geq p_{ij'}\}$ be the set of jobs of maximum processing time among the two sets. Also, let $k^* = |Max_i \cap X_i^*|$ and $k = |Max_i \cap X_i|$ be the number of maximum size jobs in sets X_i^* and X_i respectively.

For all the jobs $j \in Max_i$, we decrease p_{ij} by the minimum positive value Δ such that the cardinality of Max_i increases. After a change of this sort, the LHS drops by $(k^*(k+1))\Delta$ while the RHS drops by $(\frac{5}{3} \frac{k^*(k^*+1)}{2} + \frac{1}{3}k^2)\Delta$. Given Lemma 4.4 above, we conclude that the drop of the LHS is always less than or equal to the drop of the RHS. After repeatedly applying this same argument, we reduce all jobs to unit size while only making our initial inequality tighter. Finally, given Lemma 4.4 the inequality must be true for unit size jobs; summing up over all $i \in I$ yields

$$\sum_{j \in J} c_j^{ES}(\mathbf{x}_{-j}, x_j^*) \leq \frac{5}{3} C^{SF}(\mathbf{x}^*) + \frac{1}{3} C^{ES}(\mathbf{x}).$$

If we let \mathbf{x} be an equilibrium, the Nash condition gives $C^{ES}(\mathbf{x}) \leq \sum_{j \in J} c_j^{ES}(\mathbf{x}_{-j}, x_j^*)$, which in turn yields the inequality $C^{ES}(\mathbf{x}) \leq \frac{5}{2} C^{SF}(\mathbf{x}^*)$.

The tightness of this bound follows from Theorem 3 of [11]. The authors present a load balancing game lower bound, which is equivalent to assuming that all jobs have unit size and the machines are using EqualSharing; thus the same proof yields a (pure) PoA lower bound for restricted related machines and unweighted jobs. \square

On the negative side, we have the following (the proof of which can be found in the full version of this paper)

PROPOSITION 4.5. *When jobs are anonymous, the PoA of any deterministic prompt coordination mechanism is at least 2.5.*

4.2 Randomized Coordination Mechanism

In this section we examine the power of randomization and present Rand, which outperforms any strongly local deterministic prompt policy. Under Rand, for any pair of jobs on the same machine, the externalities they cause each other are shared in expectation. This is achieved with the following property: if two jobs j and j' are assigned to machine i , then

$$\mathbb{P}\{j \text{ precedes } j' \text{ in the ordering}\} = \frac{\rho_{ij'}}{\rho_{ij} + \rho_{ij'}}. \quad (2)$$

Recall $\rho_{ij} = p_{ij}/w_j$. A distribution over orderings with this property can be constructed as follows. Starting from the set of jobs X_i assigned to machine $i \in I$, select job $j \in X_i$ with probability $\rho_{ij}/\sum_{k \in X_i} \rho_{ik}$, and schedule j at the end. Then remove j from the list of jobs, and repeat this process. Note that this policy is different from a simple randomized policy that orders jobs uniformly at random. In fact, this simpler policy is known to give an $\Omega(m)$ PoA bound for the makespan function [35], and the same family of examples developed in [35] gives an $\Omega(m)$ lower bound for this policy in our setting. Nevertheless, we will prove the following bounds:

THEOREM 4.6. *The price of anarchy of Rand for unrelated machines $(R|\sum w_j c_j)$ is at most $32/15 = 2.133\cdots$. Moreover, if the sum of the processing times of the jobs is negligible compared to the social cost of the optimal solution, this bound improves to $\pi/2$, which is tight.*

The high level approach for obtaining the upper bound is just as in the previous section: find an appropriate mapping φ from an assignment into a convenient inner product space.

For simplicity, we assume in this section that the processing times have been scaled such that the ratios ρ_{ij} are all integral. This assumption is inessential and easily removed. We also take κ large enough so that, except for infinite processing times, $\rho_{ij} \leq \kappa$ for all $i \in I, j \in J$.

An inner product space.

The map φ we use gives the *signature* for each machine: in the unweighted case, this simply describes how many jobs of each size are assigned to the machine.

DEFINITION 4.7. *Given an assignment \mathbf{x} , its signature $\varphi(\mathbf{x}) \in \mathbb{R}_+^{m \times \kappa}$ is a vector indexed by a machine i and a processing time over weight ratio r ; we denote this component by $\varphi(\mathbf{x})_r^i$. Its value is then defined as*

$$\varphi(\mathbf{x})_r^i := \sum_{\substack{j \in X_i \\ \rho_{ij}=r}} w_j.$$

Also, let $\varphi(\mathbf{x})^i$ denote the vector $(\varphi(\mathbf{x})_0^i, \varphi(\mathbf{x})_1^i, \dots, \varphi(\mathbf{x})_\kappa^i)$.

Let M be the $\kappa \times \kappa$ matrix given by

$$M_{rs} = \frac{rs}{r+s}.$$

LEMMA 4.8. *Let \mathbf{x} be some assignment, and let $\mathbf{u} = \varphi(\mathbf{x})$.*

(i) *If job j is assigned to machine i , its expected completion time is given by*

$$c_j^R(\mathbf{x}) = (M\mathbf{u}^i)_{\rho_{ij}} + \frac{1}{2}p_{ij}.$$

(ii) *If job j is not assigned to machine i , then its expected completion time upon switching to i would be*

$$c_j^R(\mathbf{x}) = (M\mathbf{u}^i)_{\rho_{ij}} + p_{ij}.$$

PROOF. We consider case (i), so $x_j = i$; case (ii) is similar. The expected completion time of job j on machine i is

$$\begin{aligned} c_j^R(\mathbf{x}) &= \sum_{k \in X_i \setminus \{j\}} p_{ik} \mathbb{P}\{\text{job } k \text{ ahead of job } j\} + p_{ij} \\ &= \sum_{k \in X_i \setminus \{j\}} p_{ik} \frac{\rho_{ij}}{\rho_{ij} + \rho_{ik}} + p_{ij} \\ &= \sum_{k \in X_i} p_{ik} \frac{\rho_{ij}}{\rho_{ij} + \rho_{ik}} + \frac{1}{2}p_{ij}. \end{aligned}$$

We can rewrite this in terms of the signature as

$$c_j^R(\mathbf{x}) = \sum_s u_s^i M_{\rho_{ij}s} + \frac{1}{2}p_{ij} = (M\mathbf{u}^i)_{\rho_{ij}} + \frac{1}{2}p_{ij}. \quad \square$$

A crucial observation is the following:

LEMMA 4.9. *The matrix M is positive definite.*

PROOF. Let D be the diagonal matrix with $D_{rr} = r$. Then we have $M = DHD$, where the $\kappa \times \kappa$ matrix H is given by $H_{rs} = \frac{1}{r+s}$. This is a submatrix of the infinite Hilbert matrix $\left(\frac{1}{r+s-1}\right)_{r,s \in \mathbb{N}}$. The Hilbert matrix has the property that it is *totally positive* [13], meaning that the determinant of any submatrix is positive. It follows immediately that H is positive definite, and hence so is M . \square

Thus we may define an inner product by

$$\langle \mathbf{u}, \mathbf{v} \rangle := \sum_{i \in I} (\mathbf{u}^i)^T M \mathbf{v}^i, \quad (3)$$

with an associated norm $\|\cdot\|$. In addition, the social cost $C^R(\mathbf{x}) = \sum_j w_j c_j^R(\mathbf{x})$ of an assignment \mathbf{x} may be written in the convenient form

$$C^R(\mathbf{x}) = \|\varphi(\mathbf{x})\|^2 + \frac{1}{2}\eta(\mathbf{x}). \quad (4)$$

Competitiveness of Rand on a single machine.

How well Rand performs on a *single* machine, compared to the optimal SmithRule, turns out to play an important role. Suppose we have n jobs with size p_j and weight w_j , for $j \leq n$. The signature \mathbf{u} is given by just $u_r = \sum_{j:p_j/w_j=r} w_j$. Notice that the weighted sum of completion times according to SmithRule and Rand respectively are

$$\mathbf{u}^T S \mathbf{u} + \frac{1}{2} \sum_j w_j p_j \quad \text{and} \quad \mathbf{u}^T M \mathbf{u} + \frac{1}{2} \sum_j w_j p_j,$$

where $S_{rs} = \frac{1}{2} \min(r, s)$. The extra $\sum_j w_j p_j$ terms only help, and in fact turn out to be negligible in the worst case example; ignoring them, the goal is to determine the value $\max_{\mathbf{u} \geq 0} \frac{\mathbf{u}^T M \mathbf{u}}{\mathbf{u}^T S \mathbf{u}}$. The question is closely related to the worst-case distortion between two norms.

Interestingly, it turns out that this problem has been considered, and solved, in a different context. In [16], Chung, Hajela and Seymour consider the problem of *self-organizing sequential search*. In order to prove a tight bound on the performance of the “move-to-front” heuristic compared to the optimal ordering, they show:

THEOREM 4.10 ([16]). *For any sequence u_1, u_2, \dots, u_k with $u_r > 0$ for all r ,*

$$\sum_{r,s} u_r u_s \frac{rs}{r+s} < \frac{\pi}{4} \sum_{r,s} u_r u_s \min\{r, s\}. \quad (5)$$

Moreover, this is tight [28] (take $p_j = 1/j^2$, $w_j = 1$, and let $n \rightarrow \infty$). We also present a quite different proof of the theorem in the full version. All in all, we find that $\pi/2$ is a tight upper bound on the competitiveness of Rand on a single machine. The following lemma (which may also be cast as a norm distortion question), is much more easily demonstrated;

LEMMA 4.11. For any assignment \mathbf{x} , the social cost of Rand is $C^R(\mathbf{x}) \leq 2C^{SR}(\mathbf{x}) - \eta(\mathbf{x})$.

PROOF. Consider a particular machine i . We have

$$\begin{aligned} & \sum_{j,k \in X_i} w_j w_k \frac{\rho_{ij} \rho_{ik}}{\rho_{ij} + \rho_{ik}} \\ &= \sum_{j \neq k \in X_i} w_j w_k \frac{\rho_{ij} \rho_{ik}}{\rho_{ij} + \rho_{ik}} + \frac{1}{2} \sum_{j \in X_i} w_j p_{ij} \\ &\leq \sum_{j \neq k \in X_i} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} + \frac{1}{2} \sum_{j \in X_i} w_j p_{ij} \\ &= \sum_{j,k \in X_i} w_j w_k \min\{\rho_{ij}, \rho_{ik}\} - \frac{1}{2} \sum_{j \in X_i} w_j p_{ij}. \end{aligned}$$

Summing over all machines gives

$$C^R(\mathbf{x}) - \frac{1}{2}\eta(\mathbf{x}) \leq 2(C^{SR}(\mathbf{x}) - \frac{1}{2}\eta(\mathbf{x})) - \frac{1}{2}\eta(\mathbf{x})$$

from which the bound is immediate. \square

The upper bound.

We are now ready to prove the main theorem of this section.

PROOF OF THEOREM 4.6. Let \mathbf{x} be the assignment at a Nash equilibrium, and \mathbf{x}^* the assignment of the optimal solution, and let $\mathbf{u} = \varphi(\mathbf{x})$ and $\mathbf{u}^* = \varphi(\mathbf{x}^*)$.

From the Nash condition and Lemma 4.8, we obtain

$$\begin{aligned} C^R(\mathbf{x}) &\leq \sum_{j \in J} w_j c_j^R(\mathbf{x}_{-j}, \mathbf{x}_j^*) \\ &\leq \sum_{i \in I} \sum_{j \in X_i^*} w_j M(\mathbf{u}^i)_{\rho_{ij}} + \eta(\mathbf{x}^*) \\ &= \sum_{i \in I} (\mathbf{u}^*)^T M \mathbf{u}^i + \eta(\mathbf{x}^*) \\ &= \langle \mathbf{u}^*, \mathbf{u} \rangle + \eta(\mathbf{x}^*). \end{aligned}$$

Applying Cauchy-Schwartz

$$\begin{aligned} C^R(\mathbf{x}) &\leq \|\mathbf{u}^*\| \|\mathbf{u}\| + \eta(\mathbf{x}^*) \\ &\leq \frac{2}{3} \|\mathbf{u}^*\|^2 + \frac{3}{8} \|\mathbf{u}\|^2 + \eta(\mathbf{x}^*), \end{aligned} \quad (6)$$

Now recalling the definition of φ and applying Lemma 4.11, we obtain

$$\begin{aligned} C^R(\mathbf{x}) &\leq \frac{2}{3}(C^R(\mathbf{x}^*) - \frac{1}{2}\eta(\mathbf{x}^*)) + \frac{3}{8}(C^R(\mathbf{x}) - \frac{1}{2}\eta(\mathbf{x})) + \eta(\mathbf{x}^*) \\ &\leq \frac{2}{3}(2C^{SR}(\mathbf{x}^*) - \frac{3}{2}\eta(\mathbf{x}^*)) + \frac{3}{8}(C^R(\mathbf{x}) - \frac{1}{2}\eta(\mathbf{x})) + \eta(\mathbf{x}^*) \\ &\leq \frac{4}{3}C^{SR}(\mathbf{x}^*) + \frac{3}{8}C^R(\mathbf{x}). \end{aligned}$$

This gives a PoA of 32/15.

In the case where $\eta(\mathbf{x}^*)$ is negligible, we continue from (6):

$$\begin{aligned} C^R(\mathbf{x}) &\leq \|\mathbf{u}^*\| \|\mathbf{u}\| \\ &\leq \frac{1}{2} \|\mathbf{u}^*\|^2 + \frac{1}{2} \|\mathbf{u}\|^2 \\ &\leq \frac{\pi}{4} C^{SR}(\mathbf{x}^*) + \frac{1}{2} C^R(\mathbf{x}), \end{aligned}$$

by Theorem 4.10 and Equation 4. Therefore, we conclude that $C^R(\mathbf{x})/C^{SR}(\mathbf{x}^*) \leq \pi/2$. \square

As noted in the full version, a slight modification of the construction used to prove Proposition 4.5 can be used to show that the worst-case PoA of Rand is at least 5/3.

5. PNE EXISTENCE AND ALGORITHM

Existence of PNE.

Under SmithRule it may happen that no pure Nash equilibrium exists [19]. Here we show that ProportionalSharing and Rand both induce exact potential games, and hence always have PNE. For the case of ProportionalSharing, this generalizes [22, Theorem 3], which addresses EqualSharing.

THEOREM 5.1. The ProportionalSharing mechanism yields exact potential games, with potential

$$\Phi^{PS}(\mathbf{x}) = \frac{1}{2}C^{PS}(\mathbf{x}) + \frac{1}{2}\eta(\mathbf{x}). \quad (7)$$

Likewise, the Rand mechanism induces exact potential games with potential

$$\Phi^R(\mathbf{x}) = \frac{1}{2}C^R(\mathbf{x}) + \frac{1}{2}\eta(\mathbf{x}). \quad (8)$$

PROOF. We give the proof for ProportionalSharing; the proof for Rand is similar.

Consider an assignment \mathbf{x} and a job $j \in J$, and let $x_j = i$. Define $\mathbf{x}' := (\mathbf{x}_{-j}, i')$ as the assignment differing from \mathbf{x} only in that job j moves to some machine $i' \neq i$.

We may write the change in the potential function as

$$\Phi^{PS}(\mathbf{x}') - \Phi^{PS}(\mathbf{x}) = \sum_{k \in J} D_k + \frac{1}{2}w_j(p_{i'j} - p_{ij}), \quad (9)$$

where

$$D_k = \frac{1}{2}w_k (c_k^{PS}(\mathbf{x}') - c_k^{PS}(\mathbf{x})).$$

Consider a job $k \neq j$ on machine i . Since only job j left the machine, we have from Lemma 4.1 that

$$c_k^{PS}(\mathbf{x}') - c_k^{PS}(\mathbf{x}) = -w_j \min\{\rho_{ij}, \rho_{ik}\}.$$

Thus

$$\begin{aligned} \sum_{k \in X_i \setminus \{j\}} D_k &= -\frac{1}{2}w_j \sum_{k \in X_i \setminus \{j\}} w_k \min\{\rho_{ij}, \rho_{ik}\} \\ &= -\frac{1}{2}w_j (c_j^{PS}(\mathbf{x}) + p_{ij}). \end{aligned}$$

Similarly, considering jobs on i' yields

$$\begin{aligned} \sum_{k \in X_{i'}} D_k &= \frac{1}{2}w_j \sum_{k \in X_{i'}} w_k \min\{\rho_{i'j}, \rho_{i'k}\} \\ &= \frac{1}{2}w_j (c_j^{PS}(\mathbf{x}') - p_{i'j}). \end{aligned}$$

All other jobs are unaffected by the change, and so do not contribute to (9). Summing all terms (including D_j), we obtain

$$\Phi^{PS}(\mathbf{x}') - \Phi^{PS}(\mathbf{x}) = w_j (c_j^{PS}(\mathbf{x}') - c_j^{PS}(\mathbf{x})),$$

exactly the change in the cost of job j . \square

A combinatorial approximation algorithm.

Finally we define Approx, a deterministic strongly local policy that we will use in order to design a combinatorial constant factor approximation algorithm for the underlying optimization problem. The completion time of a job in this policy is exactly what its completion time would be if ProportionalSharing were being used plus its processing time, i.e. $c_j^A(\mathbf{x}) = c_j^{PS}(\mathbf{x}) + p_{x_j j}$.

Following the proof of Theorem 5.1 we can show that $\Phi^A(\mathbf{x}) = \frac{1}{2}C^A(\mathbf{x}) + \eta(\mathbf{x})$ is a potential function for the

games induced by this mechanism, and following the proof of Theorem 4.2, the PoA of the mechanism is at most 4. The advantage of this mechanism is that $C^A(\mathbf{x}) = 2C^{SR}(\mathbf{x})$ for any configuration \mathbf{x} and therefore, despite the larger PoA bound, computing an equilibrium allocation for the induced game yields a scheduling algorithm with approximation ratio 2 (because the scheduling algorithm, given the allocation, applies **SmithRule** and not **Approx**).

Computing such an allocation might be hard in general, but we show that imitating a natural best response dynamics gives rise to a simple polynomial time local search $(2 + \epsilon)$ -approximation algorithm. At each iteration, the scheduling algorithm reassigns the job which thereby obtains the largest possible improvement in the **Approx** costing (a best response move). In other words, we use **Approx** in order to find a good allocation and then switch to **SmithRule**.

In order to bound the running time of our local-search algorithm we will use the β -nice concept of [3]. Given some configuration \mathbf{x} , let

$$\Delta(\mathbf{x}) := \sum_j (w_j c_j(\mathbf{x}) - w_j c_j(\mathbf{x}_{-j}, x'_j)),$$

where x'_j is the best response to \mathbf{x}_{-j} for player j . Awerbuch et al. [3] define an exact potential game with potential function Φ and social cost function C to be β -nice if and only if, for any configuration \mathbf{x} , both $\Phi(\mathbf{x}) \leq C(\mathbf{x})$ and $C(\mathbf{x}) \leq \beta OPT + 2\Delta(\mathbf{x})$ hold⁵. Among other dynamics, they consider what they call *basic dynamics*; at each step, among all players that can uniquely deviate and improve their cost by some factor α , choose the one with the largest absolute improvement and allow that player to move. They subsequently show the following lemma, where \mathbf{x}^* is the configuration that minimizes the potential function.

LEMMA 5.2 ([3]). *Let $\frac{1}{8} > \epsilon > \alpha$. Consider an exact potential game that satisfies the β -nice property, and any initial state \mathbf{x}^0 . Then basic dynamics generates a profile \mathbf{x} with $C(\mathbf{x}) \leq \beta(1 + O(\epsilon))OPT$ in $O\left(\frac{n}{\epsilon} \log\left(\frac{\Phi(\mathbf{x}^0)}{\Phi(\mathbf{x}^*)}\right)\right)$ steps.*

We define a *coordination mechanism* to be β -nice if all the games that it induces are β -nice with OPT being the optimum social cost of the underlying machine scheduling problem, independent of the coordination mechanism. Our next lemma shows that **Approx** satisfies these conditions.

LEMMA 5.3. *The **Approx** coordination mechanism is β -nice with $\beta = 4$.*

PROOF. It is easy to verify that the potential function $\Phi^A(\mathbf{x}) = \frac{1}{2}C^A(\mathbf{x}) + \eta(\mathbf{x})$ satisfies $\Phi^A(\mathbf{x}) \leq C^A(\mathbf{x})$ for all configurations \mathbf{x} , since $\eta(\mathbf{x}) \leq \frac{1}{2}C^A(\mathbf{x})$. Therefore, what we need to show is that

$$C^A(\mathbf{x}) \leq \beta C^{SR}(\mathbf{x}^*) + 2\Delta(\mathbf{x}),$$

where

$$\Delta(\mathbf{x}) = \sum_{j \in J} (w_j c_j^A(\mathbf{x}) - w_j c_j^A(\mathbf{x}_{-j}, x'_j)),$$

and x'_j is the best response for player j in configuration \mathbf{x} . We note that since $c_j^A(\mathbf{x}_{-j}, x'_j) \leq c_j^A(\mathbf{x}_{-j}, x_j^*)$,

$$C^A(\mathbf{x}) - \sum_{j \in J} w_j c_j^A(\mathbf{x}_{-j}, x_j^*) \leq \Delta(\mathbf{x}). \quad (10)$$

⁵In their definition, unlike ours, OPT denotes the optimum social cost w.r.t. the game's cost functions.

Now following the same approach as in the proof of Theorem 4.2, we easily obtain that the PoA is at most 4. Also, based on this proof we can obtain the inequality

$$\sum_{j \in J} w_j c_j^A(\mathbf{x}_{-j}, x_j^*) \leq \frac{1}{4}C^A(\mathbf{x}) + 3C^{SR}(\mathbf{x}^*), \quad (11)$$

for any two assignments \mathbf{x} and \mathbf{x}^* . Summing inequalities (10) and (11), and simplifying we obtain

$$C^A(\mathbf{x}) \leq 4C^{SR}(\mathbf{x}^*) + \frac{4}{3}\Delta(\mathbf{x}),$$

proving the lemma. \square

If we consider that every machine uses **Approx**, then, as a result of Lemma 5.3 along with Lemma 5.2 and the fact that $C^A(\mathbf{x}) = 2C^{SR}(\mathbf{x})$ for any configuration \mathbf{x} , we get the following theorem bounding the running time of our algorithm; \mathbf{x}^* is the configuration that minimizes the potential function.

THEOREM 5.4. *Consider some initial job assignment \mathbf{x}^0 . Then, imitating basic dynamics generates a profile \mathbf{x} with $C^{SR}(\mathbf{x}) \leq (2 + O(\epsilon))OPT$ in $O\left(\frac{n}{\epsilon} \log\left(\frac{\Phi^A(\mathbf{x}^0)}{\Phi^A(\mathbf{x}^*)}\right)\right)$ steps.*

6. CONCLUDING REMARKS

On mapping machines to edges of a parallel link network, the machine scheduling problem for the case of related machines becomes a special case of general selfish routing games. In this context, the ordering policies on machines correspond to local queuing policies at the edges of the network. From this perspective, it would be interesting to generalize our results to network routing games. Designing such local queuing policies would be an important step toward more realistic models of selfish routing games when the routing happens over time [31, 24, 36]. We hope that our new technique along with the policies proposed in this paper could serve as a building block toward this challenging problem.

All the mechanisms discussed here are strongly local. For the case of the makespan objective, one can improve the approximation ratio from $\Theta(m)$ to $\Theta(\log m)$ by using local policies instead of just strongly local policies. It remains open whether there are local policies that perform even better than our strongly local ones.

Acknowledgements.

We thank Tammo Chakraborty for helpful discussions and Ioannis Caragiannis for pointing out related literature. The fourth author also thanks Yossi Azar for initial discussions about the subject of study of this paper. Part of this work was done while the second and last authors were visiting EPFL; we thank Fritz Eisenbrand for his hospitality.

7. REFERENCES

- [1] J. Aspnes, Y. Azar, A. Fiat, S.A. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *J. ACM*, 44(3):486–504, 1997.
- [2] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *STOC*, pages 57–66, 2005.
- [3] B. Awerbuch, Y. Azar, A. Epstein, V.S. Mirrokni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. In *ACM Conference on Electronic Commerce*, pages 264–273, 2008.

- [4] Y. Azar, K. Jain, and V.S. Mirrokni. (almost) optimal coordination mechanisms for unrelated machine scheduling. In *SODA*, pages 323–332, 2008.
- [5] Y. Azar, J. Naor, and R. Rom. The competitiveness of on-line assignments. *Journal of Algorithms*, 18:221–237, 1995.
- [6] A. Bagchi. Stackelberg differential games in economic models. *Springer-Verlag*, 1984.
- [7] M. Beckman, C.B. McGuire, and C.B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
- [8] A. Borodin, M. Nielsen, and C. Rackoff. (incremental) priority algorithms. In *SODA*, pages 752 – 761, 2002.
- [9] J. Bruno, E.G. Coffman, and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17:382–387, 1974.
- [10] I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *SODA*, pages 815–824, 2009.
- [11] I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. In *ICALP (1)*, pages 311–322, 2006.
- [12] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Taxes for linear atomic congestion games. *ACM Transactions on Algorithms (to appear)*.
- [13] M.-D. Choi. Tricks or treats with the hilbert matrix. *The American Mathematical Monthly*, 90(5):301–312, May 1983.
- [14] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *STOC*, pages 67–73, 2005.
- [15] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. *Theor. Comput. Sci.*, 410(36):3327–3336, 2009.
- [16] F. Chung, D. Hajela, and P. D. Seymour. Self-organizing sequential search and hilbert’s inequalities. *Journal of Computer and System Sciences*, 36(2):148–157, April 1988.
- [17] R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *J. Comput. Syst. Sci.*, 72(3):444–467, 2006.
- [18] R.W. Conway, W.L. Maxwell, and L.W. Miller. *Theory of Scheduling*. Addison-Wesley, Reading, MA, 1967.
- [19] J.R. Correa and M. Queyranne. Efficiency of equilibria in restricted uniform machine scheduling with minsum social cost. 2010 (manuscript).
- [20] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *SODA*, pages 413–420, 2002.
- [21] E. Davis and J.M. Jaffe. Algorithms for scheduling tasks on unrelated processors. *J. ACM*, 28(4):721–736, 1981.
- [22] C. Dürr and N.K. Thang. Non-clairvoyant scheduling games. In *SAGT*, pages 135–146, 2009.
- [23] E. Even-dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *ICALP*, pages 502–513, 2003.
- [24] B. Farzad, N. Olver, and A. Vetta. A priority-based model of routing. *Chicago Journal of Theoretical Computer Science*, 2008(1).
- [25] G. Finn and E. Horowitz. A linear time approximation algorithm for multiprocessor scheduling. *BIT*, 19:312–320, 1979.
- [26] L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *FOCS*, pages 277–285, 2004.
- [27] L. Fleischer and Z. Svitkina. Preference-constrained oriented matching. In *ANALCO*, pages 56–65, 2010.
- [28] G.H. Gonnet, J.I. Munro, and H. Suwanda. Exegesis of self-organizing linear search. *SIAM Journal on Computing*, 10(3):613–637, 1981.
- [29] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discrete Math.*, 5:287–326, 1979.
- [30] L.A. Hall, A.S. Schulz, D.B. Shmoys, and J. Wein. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Math. Oper. Res.*, 22(3):513–544, 1997.
- [31] M. Hoefer, V. S. Mirrokni, H. Röglin, and S.-H. Teng. Competitive routing over time. In *WINE*, pages 18–29, 2009.
- [32] H. Hoogeveen, P. Schuurman, and G.J. Woeginger. Non-approximability results for scheduling problems with minsum criteria. In *IPCO*, pages 353–366, 1998.
- [33] W.A. Horn. Minimizing average flow time with parallel machines. *Operations Research*, 21(3):846–847, 1973.
- [34] O.H. Ibarra and C.E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *J. ACM*, 24(2):280–289, 1977.
- [35] N. Immorlica, L. Li, V.S. Mirrokni, and A.S. Schulz. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410(17):1589–1598, 2009.
- [36] S. Koch and M. Skutella. Nash equilibria and the price of anarchy for flows over time. *Theory of Computing Systems*, to appear.
- [37] Y.A. Korilis, A.A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5(1):161–173, 1997.
- [38] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.
- [39] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Ann. Discrete Math.*, 4:281–300, 1977.
- [40] T. Roughgarden. Stackelberg scheduling strategies. In *STOC*, pages 104–113, 2001.
- [41] T. Roughgarden. Intrinsic robustness of the price of anarchy. In *STOC*, pages 513–522, 2009.
- [42] S. Sahni and Y. Cho. Bounds for list schedules on uniform processors. *Siam J. of Computing*, 9:91–103, 1980.
- [43] A.S. Schulz and M. Skutella. Scheduling unrelated machines by randomized rounding. *SIAM J. Discrete Math.*, 15(4):450–469, 2002.
- [44] P. Schuurman and T. Vredeveld. Performance guarantees of local search for multiprocessor scheduling. In *IPCO*, pages 370–382, 2001.
- [45] J. Sethuraman and M.S. Squillante. Optimal scheduling of multiclass parallel machines. In *SODA*, pages 963–964, 1999.
- [46] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *J. ACM*, 48(2):206–242, 2001.
- [47] M. Skutella and G.J. Woeginger. A ptas for minimizing the total weighted completion time on identical parallel machines. *Math. Oper. Res.*, 25(1):63–75, 2000.
- [48] W. Smith. Various optimizers for single stage production. *Naval Res. Logist. Quart.*, 3(1-2):59–66, 1956.
- [49] H. von Stackelberg. Marktform und Gleichgewicht. *Springer-Verlag*, 1934. English translation entitled *The Theory of the Market Economy*.
- [50] T. Vredeveld. *Combinatorial approximation algorithms. Guaranteed versus experimental performance*. 2002. Ph.D. thesis.