

Selfish, Local and Online Scheduling via Vector Fitting *

Danish Kashaev †

Abstract. We provide a dual fitting technique on a semidefinite program yielding simple proofs of tight bounds for the robust price of anarchy of several congestion and scheduling games under the sum of weighted completion times objective. The same approach also allows to bound the approximation ratio of local search algorithms and the competitive ratio of online algorithms for the scheduling problem $R \parallel \sum w_j C_j$. All of our results are obtained through a simple unified dual fitting argument on the same semidefinite programming relaxation, which can essentially be obtained through the first round of the Lasserre/Sum of Squares hierarchy.

As our main application, we show that the known coordination ratio bounds of respectively 4 , $(3 + \sqrt{5})/2 \approx 2.618$, and $32/15 \approx 2.133$ for the scheduling game $R \parallel \sum w_j C_j$ under the coordination mechanisms Smith's Rule, Proportional Sharing and Rand (STOC 2011) can be extended to congestion games and obtained through this approach. For the natural restriction where the weight of each player is proportional to its processing time on every resource, we show that the last bound can be improved from 2.133 to 2 . This improvement can also be made for general instances when considering the price of anarchy of the game, rather than the coordination ratio. As a further application of this technique in a game theoretic setting, we show that it recovers the tight bound of $(3 + \sqrt{5})/2$ for the price of anarchy of weighted affine congestion games and the Kawaguchi-Kyan bound of $(1 + \sqrt{2})/2$ for the pure price of anarchy of $P \parallel \sum w_j C_j$.

Moreover, we show that this approach recovers the known tight approximation ratio of $(3 + \sqrt{5})/2$ for a natural local search algorithm for $R \parallel \sum w_j C_j$, as well as the best currently known combinatorial approximation algorithm for this problem achieving an approximation ratio of $(5 + \sqrt{5})/4 + \varepsilon \approx 1.809 + \varepsilon$, and provide an almost matching lower bound.

Finally, we show that this technique also extends to online algorithms by analyzing a randomized algorithm for $R \parallel \sum w_j C_j$ achieving a competitive ratio of 4 in an online setting where the arrival order of the jobs is adversarial and the ordering on each machine is optimal.

1 Introduction. A standard way of quantifying inefficiency of selfish behaviour in algorithmic game theory is the price of anarchy, introduced in [49]. It is defined as the ratio between the cost of a worst-case Nash equilibrium and the cost of a social optimum. This definition can be used to understand inefficiency of pure or mixed Nash equilibria, and can also be extended to more general notions, such as correlated or coarse-correlated equilibria.

Developing tools to bound the price of anarchy is a central question, and several approaches have been proposed in the literature to tackle this problem. One technique that has been very successful for a variety of games is the smoothness framework, introduced in [55]. One advantage of this approach is that it automatically bounds the price of anarchy for all the different notions of equilibria mentioned above, yielding bounds on the robust price of anarchy of a game [55].

Another possible avenue is to use convex relaxations to help bound the price of anarchy, as done in [50]. The high-level approach is to formulate a convex relaxation of the underlying optimization problem of a given game, and to construct a feasible solution to the dual of that relaxation, whose cost can then be compared to the cost of an equilibrium. Bounding the ratio between the cost of the equilibrium and of the feasible dual solution then yields an upper bound on the price of anarchy by weak duality.

In this paper, we build on this approach and show that a single convex semidefinite programming relaxation can be used to obtain tight (robust) price of anarchy bounds for several different congestion and scheduling games. This relaxation can in fact be obtained using the first round of the Lasserre hierarchy [51], and the proofs

*The ArXiv version of the paper can be accessed at <https://arxiv.org/abs/2505.10082>

†danish.kashaev@cwi.nl, Centrum Wiskunde & Informatica, Amsterdam

bounding the price of anarchy through the dual of that relaxation are surprisingly simple and essentially follow the same template for all the games considered. In addition to bounding the price of anarchy, it turns out that the same approach also allows to bound the approximation ratio of local search algorithms and the competitive ratio of online algorithms for machine scheduling.

As a main illustration of this technique, we consider the following model of congestion games. We are given a set of players N and a set of resources E . The strategy set for each player $j \in N$ is a collection of subsets of resources and is denoted by $\mathcal{S}_j \subseteq 2^E$. Each player has a resource-dependent processing time $p_{ej} \geq 0$ and a weight $w_j \geq 0$. Once each player chooses a strategy, if a given resource $e \in E$ is shared by several players, then e uses a *coordination mechanism*, defined as a local policy for each resource, in order to process the players using it. One natural example of such a coordination mechanism is to order the players by increasing Smith ratios, defined as the ratio between the processing time on a resource and the weight of a given player [60].

This model is a generalization of the unrelated machine scheduling game $R \parallel \sum w_j C_j$, where each job needs to selfishly pick a machine to minimize its own weighted completion time, while knowing that each machine uses a coordination mechanism to process the jobs assigned to it. In our model, the set of resources E is the set of machines, and the strategy set of each player is a subset of the machines. An important special case of our model, which generalizes $R \parallel \sum w_j C_j$, is the following selfish routing game. We are given a directed graph $G = (V, E)$ and a set of players N . Each player j wants to pick a path between a source node $s_j \in V$ and a sink node $t_j \in V$. The strategy set \mathcal{S}_j for player $j \in N$ is the set of all paths between s_j and t_j . A parallel link network where each player has the same source and sink node exactly corresponds to the $R \parallel \sum w_j C_j$ scheduling problem.

The work of [25] considers three different coordination mechanisms for $R \parallel \sum w_j C_j$. Their main results are that *Smith's Rule* leads to a tight price of anarchy of 4, and this can be improved to $(3 + \sqrt{5})/2 \approx 2.618$ and $32/15 \approx 2.133$ by respectively considering a preemptive mechanism called *Proportional Sharing*, as well as a randomized one named *Rand*. The latter two results in fact bound the *coordination ratio* of the coordination mechanism, meaning that the cost of a worst-case Nash equilibrium is compared to the cost of an optimal solution under *Smith's Rule*, since this is always how an optimal solution processes the jobs once an assignment is given [60]. The proof technique they use to obtain their results is based on the smoothness framework [55]. In order to exploit the structure of the problem, they map strategy vectors into a carefully chosen inner product space, where the social cost is closely related to a squared norm in that space. Generalizing their results to selfish routing games was mentioned as an open question.

The inner product space structure developed in [25] turns out to have a natural connection to semidefinite programming, since the latter can be seen as optimizing over inner products of vectors. In this work, we study this connection and show that it leads to simple dual fitting proofs that allow to tightly bound the price of anarchy for several different congestion and scheduling games in a unified way. Moreover, it also allows to bound the approximation ratio of local search algorithms, as well as the competitive ratio of online algorithms for such problems. We hope that this new approach might turn out to be useful in other contexts as well.

Our contributions. Our main contribution is a unified dual fitting technique on a single semidefinite program to bound the price of anarchy of games, the approximation ratio of local search algorithms, and the competitive ratio of online algorithms for problems whose underlying optimization problem can be cast as a binary quadratic program. We illustrate the applicability of this approach for different scheduling and congestion problems. The semidefinite program used can be obtained by applying one round of the Lasserre/Sum of Squares hierarchy to the exact binary quadratic program.

We show that the three coordination ratio bounds of respectively 4, $(3 + \sqrt{5})/2 \approx 2.618$ and $32/15 \approx 2.133$ for the policies *Smith's Rule*, *Proportional Sharing* and *Rand* can be obtained through our approach in the above congestion game model. This yields alternative and simple proofs of these results in a more general model, which avoid the use of minimum norm distortion inequalities, as done in [25]. We moreover show that the last bound can be improved from 2.133 to 2 for the natural special case where the processing times are proportional to the weight of a given player on every feasible resource. This means that every resource has a real-value $\lambda_e \geq 0$, interpreted as the processing power, and the processing time of every player satisfies $p_{ej} \in \{\lambda_e w_j, \infty\}$ for every $e \in E, j \in N$. The importance of this model in a scheduling setting has been mentioned in [44]. This improvement from 2.133 to 2 can also be obtained for general instances if one considers the price of anarchy of the game, rather than the coordination ratio. This means that the cost of a worst-case Nash equilibrium is now compared against an optimal solution using the *Rand* policy, rather than *Smith's Rule*.

Moreover, we show that the same approach (on the same relaxation) can be used to bound the approximation

ratio of local search algorithms for machine scheduling under the sum of weighted completion times objective. We first consider a natural algorithm whose local optima simply ensure that no job can decrease the global objective function by switching to a different machine. Observe the analogy with Nash equilibria, which ensure that no job can improve its own objective (or completion time) by switching machines. We recover the approximation ratios of $(3 + \sqrt{5})/2 \approx 2.618$ and $(5 + \sqrt{5})/4 \approx 1.809$ for the scheduling problems $R \parallel \sum w_j C_j$ and $P \mid \mathcal{M}_j \mid \sum w_j C_j$ given in [27]. In addition, we also analyze an improved local search algorithm for $R \parallel \sum w_j C_j$ attaining a bound of $(5 + \sqrt{5})/4 + \varepsilon \approx 1.809 + \varepsilon$ [18], and show an almost matching lower bound of 1.791. To the best of our knowledge, this is the currently best known *combinatorial* approximation algorithm for this problem.

As a further illustration of the technique in a game theoretic setting, we apply it to two classical games and show that it yields simple proofs of known tight price of anarchy bounds. We first show how to get the tight bound of $(3 + \sqrt{5})/2$ for the price of anarchy of weighted affine congestion games. While a dual fitting proof through a convex relaxation of this bound is already provided in [50], this result showcases the versatility of our SDP relaxation and of the fitting strategy. In addition, a dual fitting proof of the Kawaguchi-Kyan bound of $(1 + \sqrt{2})/2$ for the pure price of anarchy of the scheduling game $P \parallel \sum w_j C_j$ is also provided through the same relaxation. We note that the dual fitting strategy used for this result uses a reduction to worst-case instances of [56].

Finally, we also study the $R \parallel \sum w_j C_j$ scheduling problem in the following online setting. A set of jobs arrives online in an adversarial order. Whenever a job arrives, an online algorithm needs to irrevocably assign it to a machine, at which point the job enters the schedule on that machine in the correct position with respect to the Smith ratio. This can equivalently be seen as the arrival order of the jobs as being online, but the ordering on each machine as being an offline decision. It is known that the greedy algorithm achieves a competitive ratio of 4 [35]. We provide a different 4-competitive randomized algorithm for this model analyzed using our SDP dual fitting approach, illustrating how the technique can be adapted to online algorithms.¹

Further related work. There is a vast literature on exact or approximation algorithms for scheduling problems under the (weighted) sum of completion times objective. We adopt the standard three-field notation $\alpha|\beta|\gamma$ of [33]. The problem with unweighted completion times $R \parallel \sum C_j$ is polynomial time solvable [39, 13]. For $P \parallel \sum C_j$ on parallel machines, the shortest first policy gives an optimal solution which also turns out to be a Nash equilibrium [26]. On the other hand, the weighted completion times objective is NP-hard even for $P \parallel \sum w_j C_j$ [52]. A PTAS is known for $P \parallel \sum w_j C_j$ [59], while $R \parallel \sum w_j C_j$ is APX-hard [38]. Constant factor approximation algorithms are however possible, with major results being a simple 3/2-approximation by rounding a convex relaxation [58, 57] and the first algorithm breaking the 3/2-approximation using a semidefinite relaxation [9]. We note that the primal semidefinite program used in our paper is very similar to their relaxation. Building on this, subsequent improvements have been made [42, 40, 36] with the current best (to the best of our knowledge) approximation algorithm for this problem obtaining a ratio of $1.36 + \varepsilon$ [53]. In the special case where Smith ratios are uniform, an improved bound of $(1 + \sqrt{2})/2 + \varepsilon$ has been obtained [44].

Scheduling problems have also been vastly studied from a game theoretic perspective. For $P \parallel \sum w_j C_j$, the pure price of anarchy of *Smith's Rule* coincides with the approximation ratio of a simple greedy algorithm and was shown to be $(1 + \sqrt{2})/2 \approx 1.207$ in a classic result of [46]. A much simpler proof of this result is shown in [56]. Interestingly, the mixed price of anarchy of this game is higher, with a tight bound of 3/2 even for $P \parallel \sum C_j$ [54]. For the unweighted version, *Smith's Rule* in fact reduces to the shortest processing time first policy, under which [37] shows an upper and lower bound of respectively 2 for the robust price of anarchy and $e/(e-1) \approx 1.58$ for the pure price of anarchy of $Q \parallel \sum C_j$. For related machines, it is still an interesting open question whether the upper bounds of respectively 2 and 4 for $Q \parallel \sum C_j$ and $Q \parallel \sum w_j C_j$ can be improved.

Coordination mechanisms were introduced in the work of [22] for $P \parallel C_{max}$ and a selfish routing/congestion game. Four different scheduling games under four different policies were analyzed in [43] under the makespan objective. Upper and lower bounds for different coordination mechanisms for $R \parallel C_{max}$ can be found in [7, 15, 31, 24, 2]. Further work on coordination mechanisms for the makespan objective has been done in [10, 16, 48].

The literature for the sum of completion times objective is somewhat sparser. The work of [25] considers $R \parallel \sum w_j C_j$ and shows that the policies *Smith's Rule*, *Proportional Sharing* and *Rand* respectively give bounds of 4, 2.618 and 2.133 on the robust price of anarchy. The first two bounds are tight, with matching lower bounds

¹In the ArXiv version of the paper, we also show how to analyze the greedy algorithm in the more general congestion model.

given in [28] and [17]. The latter two coordination mechanisms can in fact be interpreted as a cost-sharing protocol [18]. Using similar techniques, [1] extend some of the previous results to multi-job scheduling games. Coordination mechanisms for a more general model with release dates and assignment costs have been studied in [10].

The study of the price of anarchy for weighted congestion games was initiated in [49] for parallel links under the maximum load (or makespan) objective. Tight bounds for parallel links have been shown in [29]. For general networks under the MinSum objective with affine latency functions, the works of [5, 21] establish that the price of anarchy is $5/2$ for the unweighted version and $(3 + \sqrt{5})/2$ in the weighted case. Other models have been studied in [3, 17, 12, 61, 30]. To the best of our knowledge, the literature on coordination mechanisms for congestion/selfish routing games is relatively sparse [22, 23, 11].

The problem of online scheduling of jobs on machines has also been well studied. One natural model consists of jobs having a release date and arriving online at that point in time, where the objective is to minimize an objective function depending on the weighted flow time of jobs. In this model, strong lower bounds are known, even for preemptive algorithms [47, 32, 20]. Given these lower bounds, such scheduling problems have been considered in the speed augmentation model, where each machine is allowed to run at a ε -fraction faster speed than the offline optimum [19, 41, 45, 8]. Dual fitting approaches on LPs and convex programs for such scheduling problems have been developed in [4, 34]. Another related line of work concerns online load balancing problems for different L_p norms [6, 17, 14].

2 Preliminaries

Game format. All the games/problems considered in this paper are of the following form. A set of players N is given. Each player $j \in N$ has a strategy set \mathcal{S}_j , and we denote by $x_{ij} \in \{0, 1\}$ the binary value indicating whether the player chooses strategy $i \in \mathcal{S}_j$. If $x_{ij} \in [0, 1]$, then this corresponds to the probability of player j independently choosing strategy i . The (expected) cost incurred by the player is denoted by $C_j(x)$ and is a *quadratic* (possibly non-convex) function of x . Given weights $w_j \geq 0$ for every player $j \in N$, the total social cost is the weighted sum of costs incurred by every player, and we denote it by $C(x) = \sum_j w_j C_j(x)$.

Scheduling. One example falling in this class are scheduling games. Given is a set of jobs $J = N$, which are the players, and a set of machines M . The strategy set of every player is a subset of the machines $\mathcal{S}_j \subseteq M$. We adopt the standard three-field notation $\alpha|\beta|\gamma$ of [33], with α denoting the machine environment, β denoting the constraints, and γ denoting the objective function. The most general such problem we consider is $R \parallel \sum w_j C_j$, where each job $j \in N$ has unrelated processing times $p_{ij} \in \mathbb{R}_+ \cup \{\infty\}$ for each $i \in M$. If $p_{ij} = \infty$, we will without loss of generality assume that $i \notin \mathcal{S}_j$. Once an assignment x is fixed, the optimal way to process the jobs for each machine is to order them by increasing Smith ratios, which we denote as $\delta_{ij} := p_{ij}/w_j$. We denote $k \prec_i j$ if k precedes j in the ordering of machine i , meaning that $\delta_{ik} \leq \delta_{ij}$. We assume ties are broken in a consistent way. The completion time of every job is then

$$C_j(x) = \sum_{i \in M} x_{ij} \left(p_{ij} + \sum_{k \prec_i j} p_{ik} x_{ik} \right).$$

Observe that this is indeed a quadratic function in x . If every job has the same processing time $p_{ij} = p_j$ on every machine, this model is denoted by $P \parallel \sum w_j C_j$. If $p_{ij} \in \{p_j, \infty\}$, then the model is denoted as $P|\mathcal{M}_j| \sum w_j C_j$.

Congestion model. We consider the following model of congestion games, which generalize the scheduling games described above. We are given a set of players N and a set of resources E . The strategy set for each player $j \in N$ is denoted by $\mathcal{S}_j \subseteq 2^E$ and is a collection of subsets of resources. Each player has a resource-dependent processing time $p_{ej} \geq 0$ and a weight $w_j \geq 0$. Without loss of generality, we assume that for every feasible strategy $i \in \mathcal{S}_j$ of a player $j \in N$, we have that $p_{ej} < \infty$ for every $e \in i$ (otherwise simply remove i from \mathcal{S}_j since it is not a valid strategy). The Smith ratio is defined as $\delta_{ej} = p_{ej}/w_j$ for every $e \in E, j \in N$. We denote $k \prec_e j$ if $\delta_{ek} \leq \delta_{ej}$, meaning that k has a smaller Smith ratio than j on the resource $e \in E$, where ties are broken in a consistent manner. For a given assignment $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$, we denote

$$z_{ej} := \sum_{i \in \mathcal{S}_j: e \in i} x_{ij}.$$

We invite the reader to think about pure assignments. In that case, $x_{ij} \in \{0, 1\}$ is binary and indicates whether or not player j chooses strategy $i \in \mathcal{S}_j$, whereas $z_{ej} \in \{0, 1\}$ takes value one if j uses the resource $e \in E$, i.e.

chooses a strategy $i \in \mathcal{S}_j$ containing resource $e \in E$. In the case of mixed assignments, $x_{ij} \in [0, 1]$ represents the probability of player j independently choosing strategy i , whereas $z_{ej} \in [0, 1]$ represents the probability of player j using resource e . Once an assignment x is fixed, Smith's Rule is again the optimal way for every resource to process the jobs, and the cost incurred by a player $j \in N$ is given by:

$$C_j(x) = \sum_{i \in \mathcal{S}_j} x_{ij} \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_e j} p_{ek} z_{ek} \right).$$

Nash equilibria. An assignment x is a Nash equilibrium if no player can get a lower cost by changing his/her strategy. The price of anarchy is defined as the ratio between the cost of a worst-case Nash equilibrium and the cost of an optimal solution. Unless explicitly stated otherwise, we consider mixed Nash equilibria, meaning that the following set of constraints is satisfied:

$$(2.1) \quad C_j(x) \leq C_j(x_{-j}, i) \quad \forall j \in N, \forall i \in \mathcal{S}_j$$

where x_{-j} refers to the assignment of all players other than j . In Appendix E, we show how to extend our results to a more general equilibrium concept, namely a *coarse-correlated* equilibrium.

Coordination mechanisms. In the scheduling setting, a coordination mechanism is a set of local policies, one for each machine, deciding on how the jobs assigned to it should be processed. Smith's Rule is one example of such a policy, which is in fact optimal in terms of the social cost. However, picking a different policy may help improve the price of anarchy. One policy considered in this paper is a preemptive mechanism called *Proportional Sharing*, where the jobs are scheduled in parallel, with each uncompleted job receiving a fraction of the processor time proportional to its weight. Another mechanism is *Rand*, which orders the jobs randomly by ensuring that the probability of job j being scheduled before k is $\delta_{ik}/(\delta_{ij} + \delta_{ik})$ for every pair of jobs j, k . The reader is referred to [25] for details. In our congestion model, each resource uses one of these coordination mechanisms to process the players using that resource. Note that this modifies the cost $C_j(x)$ incurred by every player, and thus also the social cost $C(x)$.

Coordination ratio and price of anarchy. We make a distinction between the coordination ratio of a coordination mechanism and the price of anarchy of the game for our congestion model. The coordination ratio measures the ratio between a worst-case Nash equilibrium and an optimal solution if every resource uses Smith's Rule to process the players. In contrast, the price of anarchy of the game compares to a weaker optimal solution where each resource uses the chosen mechanism to process the players.

Online model. We also study the $R \parallel \sum w_j C_j$ scheduling problem in the following online setting. A set of jobs N arrives online in an adversarial order. Each time a job $j \in N$ arrives, it reveals a weight $w_j \geq 0$, and a subset $\mathcal{S}_j \subset M$ of machines it can be assigned to with unrelated processing times $p_{ij} \geq 0$ for every $i \in \mathcal{S}_j$, at which point an online algorithm needs to irrevocably assign that job to a machine. Once every job has arrived, every machine can reorder the jobs optimally, i.e. by increasing Smith ratios $\delta_{ij} := p_{ij}/w_j$. Equivalently, whenever a job j is assigned to a machine i , it enters the schedule at the correct position with respect to the Smith ratio. The quality of an online algorithm is measured by the competitive ratio, defined as the worst-case, over all instances, of the ratio between the cost of the online algorithm and that of the optimal offline solution.

Outline of the paper. The semidefinite programming relaxation and a high-level view of the approach is presented in Section 3. The analysis of the coordination ratio and the price of anarchy of *Smith's Rule*, *Proportional Sharing* and *Rand* for our congestion model are presented in Section 4. The analysis of the price of anarchy of weighted affine congestion games is shown in Section 5. The analysis of local optima for machine scheduling is done in Section 6. The online algorithm for $R \parallel \sum w_j C_j$ is presented in Section 7. The pure price of anarchy of $P \parallel \sum w_j C_j$ is presented in Appendix A.

3 The semidefinite programming relaxation.

3.1 The primal-dual pair. We assume in this section some basics on semidefinite programs (SDPs), which can be found in Appendix C. Let N be a set of players, with each player $j \in N$ having a strategy set \mathcal{S}_j . An exact quadratic program to compute the social optimum is given by

$$\begin{aligned}
 & \min C(x) \\
 & \sum_{i \in \mathcal{S}_j} x_{ij} = 1 \quad \forall j \in N \\
 & x_{ij} \in \{0, 1\} \quad \forall j \in N, \forall i \in \mathcal{S}_j.
 \end{aligned}$$

Since we assume $C(x)$ to be quadratic in x , the social cost can be written as

$$(3.1) \quad C(x) = C_{\{0,0\}} + 2 \sum_{j \in N, i \in \mathcal{S}_j} C_{\{0,ij\}} x_{ij} + \sum_{\substack{j, k \in N \\ i \in \mathcal{S}_j, i' \in \mathcal{S}_k}} C_{\{ij, i'k\}} x_{ij} x_{i'k}$$

for some symmetric matrix C of dimension $1 + \sum_{j \in N} |\mathcal{S}_j|$, which has one row/column corresponding to each x_{ij} , in addition to one extra row/column that we index by 0. The above equation (3.1) can be written in a compact way as $C(x) = \langle C, X \rangle := \text{Tr}(C^T X)$, where X is the rank one matrix $X = (1, x)(1, x)^T$, where the notation $(1, x)$ refers to a vector in dimension $1 + \sum_{j \in N} |\mathcal{S}_j|$ obtained by appending a coordinate with value 1 to x .

We now consider a semidefinite convex relaxation of the above quadratic program, which can essentially be obtained through the Lasserre/Sum of Squares hierarchy [51]. The variable of the program is a positive semidefinite matrix X of dimension $1 + \sum_{j \in N} |\mathcal{S}_j|$, which has one row/column corresponding to each x_{ij} , in addition to one extra row/column that we index by 0.

$$\begin{aligned}
 & \min \langle C, X \rangle \\
 & \sum_{i \in \mathcal{S}_j} X_{\{ij, ij\}} = 1 \quad \forall j \in N \\
 & X_{\{0,0\}} = 1 \\
 & X_{\{0, ij\}} = X_{\{ij, ij\}} \quad \forall j \in N, i \in \mathcal{S}_j \\
 & X_{\{ij, i'k\}} \geq 0 \quad \forall (i, j), (i', k) \text{ with } j, k \in N. \\
 & X \succeq 0
 \end{aligned}$$

To see that this is in fact a relaxation to the previous quadratic program, note that for any binary feasible assignment x , the rank-one matrix $X = (1, x)(1, x)^T$ is a feasible solution to the SDP with the same objective value. The key observation that makes this work is the fact that $x_{ij}^2 = x_{ij}$ for $x_{ij} \in \{0, 1\}$, leading to $X_{\{ij, ij\}} = x_{ij}^2 = x_{ij} = X_{\{0, ij\}}$. The dual to this relaxation, written in vector form, is the following. The computation of the dual is shown in Appendix D.1. We call this relaxation (SDP-C).

$$\begin{aligned}
 (3.2) \quad & \max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \\
 & y_j \leq C_{\{ij, ij\}} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \quad \forall j \in N, i \in \mathcal{S}_j \\
 & \langle v_{ij}, v_{i'k} \rangle \leq 2 C_{\{ij, i'k\}} \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N.
 \end{aligned}$$

The variables of this program are real-valued $y_j \in \mathbb{R}$ for every $j \in N$, as well as vectors $v_0 \in \mathbb{R}^d$ and $v_{ij} \in \mathbb{R}^d$ for every $j \in N, i \in \mathcal{S}_j$ in some dimension $d \in \mathbb{N}$. This will be the relaxation used for every dual fitting argument in this paper. Depending on the problem we are considering, the matrix C , which only depends on the total social cost, is then picked accordingly. The computation of this matrix for every problem considered is shown in Appendix D.2.

3.2 High-level view of the approach and intuition of the dual. We describe here a high-level view of the dual fitting approach and of its main ideas. We also provide some intuition in how the dual program (3.2) is used. For clarity, we illustrate the concepts on a simple concrete toy example: a weighted load balancing game, which is a special case of an affine weighted congestion game later analyzed in full detail in Section 5.

Example: load balancing. We are given a set of players N and a set of resources E . The strategy set of every player $j \in N$ is a subset of resources $\mathcal{S}_j \subseteq E$ with unrelated weights $w_{ij} \geq 0$ associated for every $i \in \mathcal{S}_j$. Consider a pure assignment x , the *load* of a resource $i \in E$ is defined as the total weight of players assigned to it and is formally defined as $\ell_i(x) = \sum_{j \in N} w_{ij} x_{ij}$. The cost of a player j is then defined as $C_j(x) = \sum_{i \in E} \ell_i(x) w_{ij} x_{ij}$, meaning that it is the weight multiplied by the load of the resource picked. The social cost can be written as follows

$$(3.3) \quad C(x) = \sum_{j \in N} C_j(x) = \sum_{i \in E} \sum_{j, k \in N} w_{ij} w_{ik} x_{ij} x_{ik} = \sum_{\substack{j, k \in N \\ i \in \mathcal{S}_j, i' \in \mathcal{S}_k}} w_{ij} w_{ik} x_{ij} x_{i'k} \mathbf{1}_{\{i=i'\}}.$$

Note that the social cost can also be written in a simple way as $C(x) = \sum_{i \in E} \ell_i(x)^2$. The above equation is however written in the form (3.1).

Specializing the dual SDP. After understanding what the social cost looks like as a quadratic function in the form (3.1), we are able to write down the dual program (3.2) for a considered game. In our example, the above equation tells us that the matrix C has diagonal entries $C_{\{ij, ij\}} = w_{ij}^2$ and non-diagonal entries $C_{\{ij, i'k\}} = w_{ij} w_{ik} \mathbf{1}_{\{i=i'\}}$, meaning that we can write down the dual as:

$$(3.4) \quad \max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2$$

$$(3.5) \quad y_j \leq w_{ij}^2 - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \quad \forall j \in N, \forall i \in \mathcal{S}_j$$

$$(3.6) \quad \langle v_{ij}, v_{i'k} \rangle \leq 2 w_{ij} w_{ik} \mathbf{1}_{\{i=i'\}} \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N.$$

Given any Nash equilibrium x , the goal is to use this dual program to construct a feasible solution with objective value at least $\rho C(x)$ for some $\rho \in [0, 1]$. By weak duality, this would directly imply an upper bound of $1/\rho$ for the price of anarchy.

Correspondence between the SDP constraints and the Nash conditions. The key insight is that the first set of constraints (3.5) of the SDP has the same structure as that of the Nash equilibria inequalities (2.1). Our goal is to pick a fitting which will ensure that this set of constraints corresponds to (or is implied by) these equilibrium conditions. Fix a Nash equilibrium x and let us write down what the Nash conditions imply for our toy example:

$$C_j(x) \leq w_{ij} (\ell_i(x) + w_{ij}) = w_{ij}^2 + w_{ij} \ell_i(x) \quad \forall j \in N, \forall i \in \mathcal{S}_j.$$

A natural way to achieve the desired correspondence is to have the following:

$$(3.7) \quad y_j \sim C_j(x) \quad , \quad w_{ij}^2 - \frac{1}{2} \|v_{ij}\|^2 \sim w_{ij}^2 \quad , \quad \langle v_0, v_{ij} \rangle \sim w_{ij} \ell_i(x)$$

where the \sim notation indicates that both quantities are within a fixed constant (which should be the same for all three cases above) of each other. For local search algorithms, the Nash inequalities get replaced by local optima conditions. For online algorithms, the inequalities become an equilibrium condition satisfied by the online algorithm at every time step.

Picking the vector fitting. Observe that the second correspondence above implies that $\|v_{ij}\|^2 \sim w_{ij}^2$. The second set of SDP constraints (3.6) tells us that for $i \neq i'$, one should have $\langle v_{ij}, v_{i'k} \rangle \leq 0$. We will in fact ensure tightness of this constraint by fitting such two vectors to be *orthogonal*. A very natural candidate for the fitting of v_{ij} in our example thus becomes the following choice:

$$v_{ij} \in \mathbb{R}^E \quad \text{defined as} \quad v_{ij}(e) = \alpha w_{ij} \mathbf{1}_{\{i=e\}}$$

for some constant $\alpha \in [0, \sqrt{2}]$ to be determined. The upper bound on α follows again from the second set of SDP constraints (3.6), since we now get $\langle v_{ij}, v_{i'k} \rangle = \alpha^2 w_{ij} w_{i'k} \mathbf{1}_{\{i=i'\}}$ under our fitting.

How should v_0 now be picked? There are two desirable properties to be satisfied: we want $\langle v_0, v_{ij} \rangle \sim w_{ij} \ell_i(x)$ as mentioned above, in addition to relating $\|v_0\|^2$ to the social cost $C(x)$, since it appears in the objective function of the SDP. A very natural candidate becomes the following:

$$v_0 \in \mathbb{R}^E \quad \text{defined as} \quad v_0(e) = \beta \ell_e(x)$$

where $\beta \geq 0$ is to be determined. Note that we now indeed get $\langle v_0, v_{ij} \rangle = \alpha \beta w_{ij} \ell_i(x)$ and $\|v_0\|^2 = \beta^2 C(x)$, since (3.3) can be rewritten as $C(x) = \sum_{i \in E} \ell_i(x)^2$.

Optimizing the constants. How should α and β be picked? We have seen that $\alpha \in [0, \sqrt{2}]$ and $\beta \geq 0$. Observe that under our fitting, constraints (3.5) now become $y_j \leq (1 - \alpha^2/2) w_{ij}^2 + \alpha \beta w_{ij} \ell_i(x)$. Correspondence (3.7) then tells us to set $1 - \alpha^2/2 = \alpha \beta$ and $y_j = \alpha \beta C_j(x)$. The objective value (3.4) of the SDP then becomes $(\alpha \beta - \beta^2/2) C(x)$. Since we want to pick α and β to maximize the dual objective in order to get the best possible bound on the price of anarchy, we would want to solve the following optimization problem:

$$\max\{\alpha \beta - \beta^2/2 : 1 - \alpha^2/2 = \alpha \beta, \alpha \in [0, \sqrt{2}], \beta \geq 0\}.$$

Solving this optimization problem would give a price of anarchy bound of $(3 + \sqrt{5})/2$, which is tight in this setting by a lower bound construction of [17]. At the high-level, this is the approach used to derive the results in this paper. We invite the reader to keep this intuition even for more complex games.

3.3 Different inner product spaces. In order to construct a feasible solution to this SDP, one needs to construct vectors v_0 and $\{v_{ij}\}_{j \in N, i \in \mathcal{S}_j}$ living in a Euclidean space \mathbb{R}^d for some $d > 0$, in addition to real values $\{y_j\}_{j \in N}$ such that both sets of constraints of the SDP are satisfied. Note that the inner product is the standard Euclidean one where, for given $f, g \in \mathbb{R}^d$, it is defined as $\langle f, g \rangle := \sum_{i=1}^d f_i g_i$. For some games, it will be more convenient to work in a different inner product space, as done in [25]. Let us fix a finite set E , where each $e \in E$ induces a finite set of positive real values $0 = \delta_0^{(e)} \leq \delta_1^{(e)} \leq \dots \leq \delta_n^{(e)}$. We define the following inner product space:

$$\mathcal{F}(E) := \left\{ f : E \times [0, \infty) \rightarrow \mathbb{R}_+ : f(e, t) = \sum_{j=1}^n \alpha_{ej} \mathbf{1}_{\{\delta_{j-1}^{(e)} \leq t \leq \delta_j^{(e)}\}}; \alpha_{ej} \in \mathbb{R} \right\}.$$

In words, each element satisfies the fact that $f(e, \cdot)$ is a step-function with breakpoints at $\delta_1^{(e)} \leq \dots \leq \delta_n^{(e)}$ for every $e \in E$. A valid inner product for two $f, g \in \mathcal{F}(E)$ is then given by:

$$(3.8) \quad \langle f, g \rangle := \sum_{e \in E} \int_0^\infty f(e, t) g(e, t) dt.$$

Another inner product space we will consider is the following. Let us fix E to be a finite set and $K \in \mathbb{N}$. For any positive-definite matrix $M \in \mathbb{R}^{K \times K}$, we can consider the space $\mathcal{G}(E, M) := \mathbb{R}^{E \times [K]}$ where the inner product for two $f, g \in \mathcal{G}(E, M)$ is given by:

$$(3.9) \quad \langle f, g \rangle := \sum_{e \in E} f(e, \cdot)^T M g(e, \cdot).$$

We now argue that we can work in these spaces without loss of generality, the proof can be found in Appendix B.

LEMMA 3.1. *Any feasible dual fitting to (SDP-C) obtained in the inner product spaces $\mathcal{F}(E)$ and $\mathcal{G}(E, M)$ can be converted into a feasible dual fitting with the same objective value in a standard Euclidean space \mathbb{R}^d for some $d > 0$ endowed with the standard inner product.*

4 Congestion games with coordination mechanisms.

4.1 Smith's Rule. The first coordination mechanism we consider is Smith's Rule. If x is a mixed assignment, each player first independently picks a strategy according to his/her distribution specified by x to get an assignment. Once an assignment is set, each resource orders the players using it by increasing Smith

ratios and processes them in that order. The cost incurred by a player j on a resource e that he/she is using is then $p_{ej} + \sum_{k \prec_e j} p_{ek} z_{ek}$. The total cost incurred by a player is the sum of the costs incurred on all the resources used. More formally, the completion time of player $j \in N$ under Smith's Rule is defined to be:

$$(4.1) \quad C_j(x) = \sum_{i \in S_j} x_{ij} \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_e j} p_{ek} z_{ek} \right).$$

The outer sum only has one term for a binary assignment. For a mixed assignment x , the expression above is the expected cost, by the law of total probability and independence. The social cost is the sum of weighted completion times:

$$(4.2) \quad C(x) := \sum_{j \in N} w_j C_j(x) = \sum_{e \in E} \sum_{j \in N} w_j p_{ej} z_{ej} + \sum_{e \in E} \sum_{j \in N, k \prec_e j} w_j p_{ek} z_{ek} z_{ej},$$

where the second equality follows by using the definition of $C_j(x)$ and changing the order of summation. Moreover, if x is a Nash equilibrium, the following inequalities are satisfied:

$$(4.3) \quad C_j(x) \leq \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_e j} p_{ek} z_{ek} \right) \quad \forall j \in N, \forall i \in \mathcal{S}_j.$$

The dual semidefinite relaxation (3.2) then becomes the following, we call it *(SDP-SR)*. The computation of the cost matrix C in this setting is shown in Appendix D.2.

$$(4.4) \quad \max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2$$

$$(4.5) \quad y_j \leq \sum_{e \in i} w_j p_{ej} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \quad \forall j \in N, \forall i \in \mathcal{S}_j$$

$$(4.6) \quad \langle v_{ij}, v_{i'k} \rangle \leq \sum_{e \in i \cap i'} w_j w_k \min \{ \delta_{ej}, \delta_{ek} \} \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N.$$

We note that, in order to bound the coordination ratio of a coordination mechanism, one needs to construct a feasible dual solution to this relaxation, since it gives a valid lower bound on the optimal solution. Indeed, once an assignment is fixed, the optimal ordering on every resource is to schedule the players according to Smith's Rule [60].

THEOREM 4.1. *For any Nash equilibrium x of the above congestion game, where each resource uses the Smith's Rule policy, there exists a feasible (SDP-SR) solution with value at least $1/4 C(x)$. This implies that the price of anarchy and the coordination ratio is at most 4.*

Remark 4.2. This bound is tight, with a matching lower bound given in [28] even for scheduling on restricted identical machines with unit processing times.

Proof. We assume that the SDP vectors live in the inner product space $\mathcal{F}(E)$. By Lemma 3.1, this is without loss of generality. Let us fix $\beta = 1/2$, we now state the dual fitting for (SDP-SR):

- $v_0(e, t) := \beta \sum_{k \in N} w_k z_{ek} \mathbb{1}_{\{t \leq \delta_{ek}\}}$
- $v_{ij}(e, t) := w_j \mathbb{1}_{\{e \in i\}} \mathbb{1}_{\{t \leq \delta_{ej}\}} \quad \forall j \in N, \forall i \in \mathcal{S}_j$
- $y_j := \beta w_j C_j(x) \quad \forall j \in N.$

Let us now compute the different inner products and norms that we need. For a job $j \in N$ and a strategy $i \in \mathcal{S}_j$, we have

$$\|v_{ij}\|^2 = \sum_{e \in i} w_j^2 \delta_{ej} = \sum_{e \in i} w_j p_{ej}.$$

For v_0 , we give an upper bound with respect to $C(x)$:

$$\begin{aligned}
 \frac{1}{\beta^2} \|v_0\|^2 &= \sum_{e \in E} \sum_{j, k \in N} w_j w_k z_{ej} z_{ek} \int_0^\infty \mathbb{1}_{\{t \leq \delta_{ej}\}} \mathbb{1}_{\{t \leq \delta_{ek}\}} dt = \sum_{e \in E} \sum_{j, k \in N} w_j w_k z_{ej} z_{ek} \min\{\delta_{ej}, \delta_{ek}\} \\
 &= \sum_{e \in E} \sum_{j \in N} w_j^2 z_{ej}^2 \delta_{ej} + 2 \sum_{e \in E} \sum_{j \in N, k \prec_{ej}} w_j w_k z_{ej} z_{ek} \delta_{ek} \\
 &= \sum_{e \in E} \sum_{j \in N} w_j p_{ej} z_{ej}^2 + 2 \sum_{e \in E} \sum_{j \in N, k \prec_{ej}} w_j p_{ek} z_{ej} z_{ek} \\
 (4.7) \quad &\leq 2 C(x).
 \end{aligned}$$

The last equality uses the definition of the Smith Ratio $\delta_{ej} = p_{ej}/w_j$, whereas the last inequality follows from the fact that $z_{ej}^2 \leq z_{ej}$ (since $z_{ej} \in [0, 1]$) as well as the definition of the social cost (4.2). In addition, for any $(i, j) \neq (i', k)$ with $j, k \in N$, we have

$$(4.8) \quad \langle v_{ij}, v_{i'k} \rangle = \sum_{e \in E} w_j w_k \mathbb{1}_{\{e \in i\}} \mathbb{1}_{\{e \in i'\}} \int_0^\infty \mathbb{1}_{\{t \leq \delta_{ej}\}} \mathbb{1}_{\{t \leq \delta_{ek}\}} dt = \sum_{e \in i \cap i'} w_j w_k \min\{\delta_{ej}, \delta_{ek}\}$$

and observe that this tightly satisfies the second set of SDP constraints (4.6). Finally,

$$\langle v_0, v_{ij} \rangle = \beta \sum_{e \in i} \sum_{k \in N} w_j w_k z_{ek} \min\{\delta_{ej}, \delta_{ek}\}.$$

Let us now check that this is a feasible solution to *(SDP-SR)*. The second set of constraints is satisfied due to (4.8). The first set of constraints (4.5) under the above fitting becomes:

$$\begin{aligned}
 y_j &\leq \sum_{e \in i} w_j p_{ej} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \\
 \iff \beta w_j C_j(x) &\leq \frac{1}{2} \sum_{e \in i} w_j p_{ej} + \beta \sum_{e \in i} \sum_{k \in N} w_j w_k z_{ek} \min\{\delta_{ej}, \delta_{ek}\} \\
 \iff C_j(x) &\leq \sum_{e \in i} \left(p_{ej} + \sum_{k \in N} w_k z_{ek} \min\{\delta_{ej}, \delta_{ek}\} \right) \\
 \iff C_j(x) &\leq \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_{ej}} p_{ek} z_{ek} + \sum_{k \succeq_{ej}} w_k z_{ek} \delta_{ej} \right).
 \end{aligned}$$

We have simplified both sides by $\beta w_j = 1/2 w_j$ in the third line, which holds by definition of $\beta := 1/2$. We have also used the definition of the Smith ratio $\delta_{ek} = p_{ek}/w_k$ in the last line. This set of constraints is now clearly satisfied by the Nash conditions (4.3). The objective function of this SDP can now be lower bounded using (4.7):

$$\sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \geq \beta \sum_{j \in N} w_j C_j(x) - \beta^2 C(x) = (\beta - \beta^2) C(x) = \frac{1}{4} C(x). \quad \square$$

4.2 The Proportional Sharing policy. In this section, we consider a preemptive policy for every resource named *Proportional Sharing*. Once an assignment is fixed, each resource splits its processing capacity among the uncompleted jobs proportionally to their weights. Given an assignment x , the completion time of player $j \in N$ is defined to be:

$$C_j(x) = \sum_{i \in S_j} x_{ij} \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_{ej}} p_{ek} z_{ek} + \sum_{k \succ_{ej}} w_k z_{ek} \delta_{ej} \right).$$

For this policy, it is in fact more intuitive to understand the definition by looking at the weighted completion time:

$$w_j C_j(x) = \sum_{i \in S_j} x_{ij} \sum_{e \in i} \left(w_j p_{ej} + \sum_{k \in N \setminus \{j\}} w_j w_k \min\{\delta_{ej}, \delta_{ek}\} z_{ek} \right).$$

The social cost is the sum of weighted completion times:

$$(4.9) \quad C(x) := \sum_{j \in N} w_j C_j(x) = \sum_{e \in E} \sum_{j \in N} w_j p_{ej} z_{ej} + 2 \sum_{e \in E} \sum_{j \in N, k \prec_e j} w_j p_{ek} z_{ek} z_{ej}.$$

Observe that there is now a factor 2 in front of the second term if one compares it to the *Smith Rule* policy. Moreover, if x is a Nash equilibrium, the following inequalities are satisfied:

$$(4.10) \quad C_j(x) \leq \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_e j} p_{ek} z_{ek} + \sum_{k \succ_e j} w_k z_{ek} \delta_{ej} \right) \quad \forall j \in N, \forall i \in \mathcal{S}_j.$$

We first need a small lemma about two parameters that will play a key role in the dual fitting. The first property will ensure feasibility of the solution, whereas the second one will be the constant in front of the objective function.

LEMMA 4.3. *Let $\alpha, \beta \geq 0$ be defined as $\alpha^2 := 2/\sqrt{5}$ and $\beta := 1/\alpha - \alpha/2$. The following two properties hold:*

- $1 - \alpha^2/2 = \alpha\beta$
- $\alpha\beta - \beta^2/2 = 2/(3 + \sqrt{5})$

Proof. The first property is immediate by definition of β . For the second property, we get

$$\alpha\beta - \frac{\beta^2}{2} = 1 - \frac{\alpha^2}{2} - \frac{1}{2} \left(\frac{1}{\alpha} - \frac{\alpha}{2} \right)^2 = \frac{3}{2} - \frac{5\alpha^2}{8} - \frac{1}{2\alpha^2} = \frac{3}{2} - \frac{5}{4\sqrt{5}} - \frac{\sqrt{5}}{4} = \frac{2}{3 + \sqrt{5}}. \quad \square$$

THEOREM 4.4. *For any Nash equilibrium x of the above congestion game, where each resource uses the Proportional Sharing policy, there exists a feasible (SDP-SR) solution with value at least $2/(3 + \sqrt{5})C(x)$, implying that the coordination ratio is at most $(3 + \sqrt{5})/2$.*

Remark 4.5. This bound is tight, with a matching lower bound given in [17] even for the price of anarchy of the game.

Proof. The proof is very similar to the one of Theorem 4.1, but with the modified constants $\alpha^2 := 2/\sqrt{5}$ and $\beta := 1/\alpha - \alpha/2$ stated in Lemma 4.3. We now state the dual fitting.

- $v_0(e, t) := \beta \sum_{k \in N} w_k z_{ek} \mathbb{1}_{\{t \leq \delta_{ek}\}}$
- $v_{ij}(e, t) := \alpha w_j \mathbb{1}_{\{e \in i\}} \mathbb{1}_{\{t \leq \delta_{ej}\}} \quad \forall j \in N, \forall i \in \mathcal{S}_j$
- $y_j := \alpha\beta w_j C_j(x) \quad \forall j \in N.$

Using the same computations as in Theorem 4.1, we compute the different inner products and norms that we need.

- $\|v_0\|^2 = \beta^2 \left(\sum_{e \in E} \sum_{j \in N} w_j p_{ej} z_{ej}^2 + 2 \sum_{e \in E} \sum_{j \in N, k \prec_e j} w_j p_{ek} z_{ej} z_{ek} \right) \leq \beta^2 C(x)$
- $\|v_{ij}\|^2 = \alpha^2 \sum_{e \in i} w_j p_{ej}$
- $\langle v_0, v_{ij} \rangle = \alpha\beta \sum_{e \in i} \sum_{k \in N} w_j w_k z_{ek} \min\{\delta_{ej}, \delta_{ek}\}$
- $\langle v_{ij}, v_{i'k} \rangle = \alpha^2 \sum_{e \in i \cap i'} w_j w_k \min\{\delta_{ej}, \delta_{ek}\}$

The main difference with *Smith's Rule* which allows us to get an improved bound is the fact that the upper bound on the squared norm of v_0 is a factor 2 stronger in this case (see (4.7)), due to the new definition of the social cost $C(x)$ given in (4.9). To see that this solution is feasible, note that the second set of SDP constraints (4.6) is

satisfied due to the last computation above and the fact that $\alpha^2 \leq 1$. The first set of constraints (4.5) under the above fitting reads:

$$\begin{aligned}
 y_j &\leq \sum_{e \in i} w_j p_{ej} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \\
 \iff \alpha \beta w_j C_j(x) &\leq \left(1 - \frac{\alpha^2}{2}\right) \sum_{e \in i} w_j p_{ej} + \alpha \beta \sum_{e \in i} \sum_{k \in N} w_j w_k z_{ek} \min\{\delta_{ej}, \delta_{ek}\} \\
 \iff C_j(x) &\leq \sum_{e \in i} \left(p_{ej} + \sum_{k \in N} w_k z_{ek} \min\{\delta_{ej}, \delta_{ek}\} \right) \\
 \iff C_j(x) &\leq \sum_{e \in i} \left(p_{ej} + \sum_{k \prec_{ej}} p_{ek} z_{ek} + \sum_{k \succeq_{ej}} w_k z_{ek} \delta_{ej} \right).
 \end{aligned}$$

The third equivalence follows from the first property of Lemma 4.3. We see that this is satisfied due to the Nash conditions (4.10). The objective value of the solution can now be lower bounded as follows:

$$\sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \geq \alpha \beta \sum_{j \in N} w_j C_j(x) - \frac{\beta^2}{2} C(x) = \left(\alpha \beta - \frac{\beta^2}{2} \right) C(x) = \frac{2}{3 + \sqrt{5}} C(x)$$

where the last equality follows by the second property of Lemma 4.3. \square

4.3 The Rand policy. In this section, we consider a randomized policy named *Rand*. If x is a mixed assignment, each player first independently picks a strategy according to his/her distribution specified by x . We denote by $N(e) \subseteq N$ the (possibly random) subset of players using resource $e \in E$. Each resource then orders the players using it randomly in a way ensuring that for any pair $j, k \in N(e)$, the probability that j comes after k in the ordering is exactly equal to $\delta_{ej}/(\delta_{ej} + \delta_{ek})$. Such a distribution can be achieved by sampling one player $j \in N(e)$ at random with probability $\delta_{ej}/\sum_{k \in N(e)} \delta_{ek}$, putting that player at the end of the ordering, and repeating this process. The expected completion time of every player is thus given by:

$$C_j(x) = \sum_{i \in S_j} x_{ij} \sum_{e \in i} \left(p_{ej} + \sum_{k \neq j} \frac{\delta_{ej}}{\delta_{ej} + \delta_{ek}} p_{ek} z_{ek} \right).$$

The social cost is the sum of weighted completion times:

$$(4.11) \quad C(x) := \sum_{j \in N} w_j C_j(x) = \sum_{e \in E} \sum_{j \in N} w_j p_{ej} z_{ej} + \sum_{e \in E} \sum_{j \in N, k \neq j} \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}} w_j w_k z_{ej} z_{ek}.$$

Moreover, if x is a Nash equilibrium, the following inequalities are satisfied:

$$(4.12) \quad C_j(x) \leq \sum_{e \in i} \left(p_{ej} + \sum_{k \neq j} \frac{\delta_{ej}}{\delta_{ej} + \delta_{ek}} p_{ek} z_{ek} \right) \quad \forall j \in N, i \in \mathcal{S}_j.$$

We now state a small lemma about some constants that will be important for the fitting. The first property will ensure that our dual fitting is feasible, whereas the second property will be the constant in front of the objective value of our SDP solution, thus determining the coordination ratio.

LEMMA 4.6. *Let $\alpha, \beta \geq 0$ be defined as $\alpha := 1$ and $\beta := 3/4$. The following two properties hold:*

- $1 - \alpha^2/4 = \alpha \beta$
- $\alpha \beta - \beta^2/2 = 15/32$

Proof. The proof is immediate. \square

THEOREM 4.7. *For any instance of the above congestion game under the Rand policy, and for any Nash equilibrium x , there exists a feasible (SDP-SR) solution with value at least $15/32 C(x)$. This implies that the coordination ratio is at most $32/15 \approx 2.133$.*

Proof. For simplicity of presentation, let us assume without loss of generality that the processing times are scaled such that the Smith ratios $\delta_{ej} = p_{ej}/w_j$ with $p_{ej} < \infty$ are all integral. Moreover, let us take $K \in \mathbb{N}$ large enough such that $\delta_{ej} \leq K$ for all pairs $(e, j) \in E \times N$ such that $p_{ej} < \infty$. Consider the matrix $M \in \mathbb{R}^{K \times K}$ given by

$$M_{r,s} := \frac{r \ s}{r + s} \quad \forall r, s \in \{1, \dots, K\}.$$

A key insight shown in [25] is that this matrix is positive-definite. By Lemma 3.1, we can thus assume that the SDP vectors live in the space $\mathcal{G}(E, M)$. Let α, β be defined as in Lemma 4.6, we now state the dual fitting:

- $v_0(e, r) := \beta \sum_{k \in N} w_k z_{ek} \mathbb{1}_{\{\delta_{ek}=r\}}$
- $v_{ij}(e, r) := \alpha w_j \mathbb{1}_{\{e \in i\}} \mathbb{1}_{\{\delta_{ej}=r\}} \quad \forall j \in N, i \in \mathcal{S}_j$
- $y_j := \alpha \beta w_j C_j(x) \quad \forall j \in N.$

Let us now compute the different inner products and norms that we need. For every $j \in N, i \in \mathcal{S}_j$:

$$\frac{1}{\alpha^2} \|v_{ij}\|^2 = \sum_{e \in i} M_{\{\delta_{ej}, \delta_{ej}\}} w_j^2 = \sum_{e \in i} \frac{\delta_{ej}}{2} w_j^2 = \frac{1}{2} \sum_{e \in i} w_j p_{ej}.$$

For the squared norm of v_0 , we give an upper bound with respect to $C(x)$:

$$\begin{aligned} \frac{1}{\beta^2} \|v_0\|^2 &= \sum_{e \in E} \sum_{r,s=1}^K M_{r,s} v_0(e, r) v_0(e, s) = \sum_{e \in E} \sum_{j,k \in N} w_j w_k z_{ej} z_{ek} M_{\{\delta_{ej}, \delta_{ek}\}} \\ (4.13) \quad &= \sum_{e \in E} \sum_{j,k \in N} \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}} w_j w_k z_{ej} z_{ek} \leq C(x) \end{aligned}$$

where the last inequality holds by (4.11) and $z_{ej}^2 \leq z_{ej}$. For any pair $(i, j) \neq (i', k)$ with $j, k \geq 1$:

$$(4.14) \quad \frac{1}{\alpha^2} \langle v_{ij}, v_{i'k} \rangle = \sum_{e \in i \cap i'} M_{\{\delta_{ej}, \delta_{ek}\}} w_j w_k = \sum_{e \in i \cap i'} w_j w_k \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}}.$$

Finally, we have:

$$\begin{aligned} \frac{1}{\alpha \beta} \langle v_0, v_{ij} \rangle &= \sum_{e \in i} \sum_{k \in N} w_j w_k z_{ek} M_{\{\delta_{ej}, \delta_{ek}\}} = \sum_{e \in i} \sum_{k \in N} \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}} w_j w_k z_{ek} \\ &= w_j \sum_{e \in i} \sum_{k \in N} \frac{\delta_{ej}}{\delta_{ej} + \delta_{ek}} p_{ek} z_{ek} \end{aligned}$$

where the last equality follows by plugging in the definition of $\delta_{ek} = p_{ek}/w_k$.

Let us now check that this solution is indeed feasible for (SDP-SR). The second set of constraints (4.6) is satisfied due to (4.14), the fact that $\alpha = 1$, as well as observing that $rs/(r+s) \leq \min\{r, s\}$ for all $r, s \geq 0$. The first set of constraints (4.5) under the above fitting gives:

$$\begin{aligned} y_j &\leq \sum_{e \in i} w_j p_{ej} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \\ &\iff \alpha \beta w_j C_j(x) \leq \left(1 - \frac{\alpha^2}{4}\right) \sum_{e \in i} w_j p_{ej} + \alpha \beta w_j \sum_{e \in i} \sum_{k \in N} \frac{\delta_{ej}}{\delta_{ej} + \delta_{ek}} p_{ek} z_{ek} \\ &\iff C_j(x) \leq \sum_{e \in i} \left(p_{ej} + \sum_{k \in N} \frac{\delta_{ej}}{\delta_{ej} + \delta_{ek}} p_{ek} z_{ek} \right). \end{aligned}$$

We have simplified both sides by $\alpha\beta w_j = (1 - \alpha^2/4)w_j$ in the last equivalence, which holds by the first property of Lemma 4.6. These inequalities are now clearly satisfied by the Nash conditions (4.12), implying that our fitted solution is in fact feasible. The objective value of our solution can be lower bounded as:

$$\sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \geq \alpha\beta \sum_{j \in N} w_j C_j(x) - \frac{\beta^2}{2} C(x) = \left(\alpha\beta - \frac{\beta^2}{2} \right) C(x) = \frac{15}{32} C(x)$$

where the last equality follows from the second property of Lemma 4.6. \square

We now show that this bound can be improved if we consider the natural special case where the processing time of each player is proportional to its weight for every resource. This means that every resource has a real-value $\lambda_e \geq 0$, and the processing time of every player satisfies $p_{ej} \in \{\lambda_e w_j, \infty\}$ for every $e \in E, j \in N$. Observe that this means that the Smith ratios are uniform for the jobs assigned to a resource: $\delta_{ej} = p_{ej}/w_j = \lambda_e$. The only difference with respect to the previous proof will be a change of constants α, β .

LEMMA 4.8. *Let $\alpha, \beta \geq 0$ be defined as $\alpha := 2/\sqrt{3}$ and $\beta := 1/\sqrt{3}$. The following two properties hold:*

- $1 - \alpha^2/4 = \alpha\beta$
- $\alpha\beta - \beta^2/2 = 1/2$

Proof. The proof is immediate. \square

THEOREM 4.9. *If the Smith ratios are uniform for every resource, for any instance of the above game and any Nash equilibrium x , there exists a feasible (SDP-SR) solution with value at least $1/2 C(x)$. This implies that the coordination ratio of the game is at most 2.*

Proof. Let α, β be as in Lemma 4.8. The only part of the proof of Theorem 4.7 which breaks down under these new constants is the fact that the second set of constraints (4.6) of the SDP might be violated, since we now have $\alpha^2 = 4/3 > 1$. Indeed (4.14) states that:

$$\langle v_{ij}, v_{i'k} \rangle = \alpha^2 \sum_{e \in i \cap i'} w_j w_k \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}}.$$

The proof of Theorem 4.7 used the easy observation that $rs/(r + s) \leq \min\{r, s\}$ for every $r, s \geq 0$ to argue feasibility of the solution. Observe that this bound is very close to tight when $s \gg r$ (or vice versa). In the case of uniform Smith ratios, we can get an improved bound since $\delta_{ej} = \delta_{ek} = \lambda_e$:

$$\langle v_{ij}, v_{i'k} \rangle = \alpha^2 \sum_{e \in i \cap i'} w_j w_k \frac{\lambda_e}{2} \leq \sum_{e \in i \cap i'} w_j w_k \lambda_e = \sum_{e \in i \cap i'} w_j w_k \min\{\delta_{ej}, \delta_{ek}\}$$

where the inequality follows since $\alpha^2/2 = 4/6 \leq 1$. By the second property of Lemma 4.8, the objective value can now be lower bounded as

$$\sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \geq \alpha\beta \sum_{j \in N} w_j C_j(x) - \frac{\beta^2}{2} C(x) = \left(\alpha\beta - \frac{\beta^2}{2} \right) C(x) = \frac{1}{2} C(x). \quad \square$$

We now show that this bound of 2 can also be attained for arbitrary instances if we consider the price of anarchy of the game, rather than the coordination ratio, meaning that we now compare against the optimal solution under the *Rand* policy. More precisely, we compare against the best possible assignment x , whose expected cost is measured if every resource uses the *Rand* policy to process the players. Note that this cost is always higher than the cost if every resource were to use *Smith's Rule*. In that case, a relaxation giving a valid lower bound on the social optimum is the following, we call it (SDP-RAND). The computation of the cost matrix C to plug-in in (3.2) in this setting is once again left to Appendix D.2.

$$\begin{aligned}
\max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \\
y_j \leq \sum_{e \in i} w_j p_{ej} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \quad \forall j \in N, \forall i \in \mathcal{S}_j \\
\langle v_{ij}, v_{i'k} \rangle \leq 2 \sum_{e \in i \cap i'} w_j w_k \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}} \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N.
\end{aligned}$$

THEOREM 4.10. *For any instance of the above game under the Rand policy, and for any Nash equilibrium x , there exists a feasible (SDP-RAND) solution with value at least $1/2 C(x)$. This implies that the price of anarchy of the game is at most 2.*

Proof. The proof is identical to the one of Theorem 4.7, but with the modified constants α, β stated in Lemma 4.8. This new choice of constants is not valid for (SDP-SR), due to the fact that $\alpha^2 > 1$. Indeed, equation (4.14) means that the second set of constraints (4.6) of (SDP-SR) might now be violated. However, the second set of constraints of (SDP-RAND) is always satisfied, since $\alpha^2 \leq 2$. The objective function guarantee follows from the second property of Lemma 4.8. \square

5 Weighted affine congestion games. In this section, we consider the classic weighted affine congestion game. The price of anarchy of this game was settled in [5, 21] with a tight bound of $(3 + \sqrt{5})/2$ and this bound can also be obtained through a dual fitting argument on a convex program [50]. We show here how to recover this bound in a simple way through our approach. For simplicity of presentation, we assume in this section that the Nash equilibria considered are pure, extensions to more general equilibrium notions can be found in Appendix E.

The setting is the following. There is a set N of players and a set E of resources. The strategy set for each player $j \in N$ is denoted by $\mathcal{S}_j \subseteq 2^E$ and is a collection of subsets of resources. Let us also assume that we have unrelated weights $w_{ej} \geq 0$ for every $j \in N, e \in E$. Given a strategy profile x , the *load* of a resource is given by:

$$\ell_e(x) := \sum_{j \in N} w_{ej} \sum_{i \in \mathcal{S}_j: e \in i} x_{ij}.$$

The cost incurred by a player j for a pure assignment x is then given by

$$C_j(x) := \sum_{i \in \mathcal{S}_j} x_{ij} \sum_{e \in i} w_{ej} (a_e \ell_e(x) + b_e)$$

where $a_e, b_e \in \mathbb{R}_{\geq 0}$ for every $e \in E$. The social cost then becomes:

$$(5.1) \quad C(x) := \sum_{j \in N} C_j(x) = \sum_{e \in E} a_e \ell_e(x)^2 + b_e \ell_e(x)$$

where the last equality holds by changing the order of summation and using the definition of $\ell_e(x)$.

The Nash equilibrium conditions imply the following constraints for every $j \in N, i \in \mathcal{S}_j$:

$$(5.2) \quad C_j(x) \leq \sum_{e \in i} w_{ej} (a_e (\ell_e(x) + w_{ej}) + b_e) = \sum_{e \in i} w_{ej} (a_e w_{ej} + b_e) + \sum_{e \in i} w_{ej} a_e \ell_e(x).$$

Indeed, if a player $j \in N$ decides to switch to a strategy $i \in \mathcal{S}_j$, then the load on every resource $e \in i$ can go up by at most w_{ej} . The semidefinite relaxation (3.2) in this special case becomes the following, we call it (SDP-CG).

$$\begin{aligned}
\max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \\
y_j \leq \sum_{e \in i} w_{ej} (a_e w_{ej} + b_e) - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \quad \forall j \in N, \forall i \in \mathcal{S}_j \\
\langle v_{ij}, v_{i'k} \rangle \leq 2 \sum_{e \in i \cap i'} a_e w_{ej} w_{ek} \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N.
\end{aligned}$$

THEOREM 5.1. *For any instance of the above game, and any Nash equilibrium x , there exists a feasible (SDP-CG) solution with objective value at least $2/(3 + \sqrt{5})C(x)$.*

Proof. The vectors of the SDP will live in the space \mathbb{R}^E . Let $\alpha, \beta \geq 0$ be defined as in Lemma 4.3. We now state the dual fitting:

- $v_0(e) := \beta \sqrt{a_e} \ell_e(x)$
- $v_{ij}(e) := \alpha \sqrt{a_e} w_{ej} \mathbb{1}_{\{e \in i\}} \quad \forall j \in N, i \in \mathcal{S}_j$
- $y_j = \alpha \beta C_j(x) \quad \forall j \in N.$

Let us now compute the different inner products and norms that we need.

- $\|v_0\|^2 = \beta^2 \sum_{e \in E} a_e \ell_e(x)^2 \leq \beta^2 C(x)$
- $\|v_{ij}\|^2 = \alpha^2 \sum_{e \in i} a_e w_{ej}^2$
- $\langle v_0, v_{ij} \rangle = \alpha \beta \sum_{e \in i} a_e w_{ej} \ell_e(x)$
- $\langle v_{ij}, v_{i'k} \rangle = \alpha^2 \sum_{e \in i \cap i'} a_e w_{ej} w_{ek}$

Let us now check feasibility of the solution. The second set of constraints is satisfied by the fourth computation above and the fact that $\alpha^2 = 2/\sqrt{5} \leq 2$. The first set of constraints is satisfied due to the Nash conditions (5.2). Indeed, under the above fitting, for every $j \in N, i \in \mathcal{S}_j$, the first set of SDP constraints reads:

$$\alpha \beta C_j(x) \leq (1 - \alpha^2/2) \sum_{e \in i} a_e w_{ej}^2 + \sum_{e \in i} w_{ej} b_e + \alpha \beta \sum_{e \in i} a_e w_{ej} \ell_e(x).$$

If there was a factor of $(1 - \alpha^2/2) \leq 1$ multiplying the term $\sum_{e \in i} w_{ej} b_e$, then this would be equivalent to (5.2) because of the first condition of Lemma 4.3. Not having this term only increases the right hand side and thus ensures that this set of constraints is satisfied, implying that the SDP solution is feasible. The objective function can now be lower bounded as:

$$\sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \geq \alpha \beta \sum_{j \in N} C_j(x) - \frac{\beta^2}{2} C(x) = \left(\alpha \beta - \frac{\beta^2}{2} \right) C(x) = \frac{2}{3 + \sqrt{5}} C(x)$$

where the last equality follows by the second property of Lemma 4.3. \square

6 Analyzing local search algorithms for scheduling. We now show that this approach can also be useful to bound the approximation ratio of local search algorithms. We focus on the $R||\sum w_j C_j$ scheduling problem, for which the jobs are denoted as $J = N$, and for which we rewrite the (SDP-SR) relaxation (4.4):

$$\begin{aligned} \max_{j \in J} y_j - \frac{1}{2} \|v_0\|^2 \\ y_j &\leq w_j p_{ij} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \quad \forall j \in J, \forall i \in \mathcal{S}_j \\ \langle v_{ij}, v_{i'k} \rangle &\leq w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \mathbb{1}_{\{i=i'\}} \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in J. \end{aligned}$$

Given an assignment $x \in \{0, 1\}^{M \times J}$, the completion time of every job $j \in J$ is:

$$C_j(x) = \sum_{i \in M} x_{ij} \left(p_{ij} + \sum_{k \prec_i j} p_{ik} x_{ik} \right).$$

Let us also define the following quantity for every $j \in J$:

$$(6.1) \quad D_j(x) = \sum_{i \in M} \sum_{k \succ_i j} w_k p_{ij} x_{ij} x_{ik}$$

and let us denote the weighted sum of processing times as:

$$(6.2) \quad \eta(x) = \sum_{i \in M} \sum_{j \in J} w_j p_{ij} x_{ij}.$$

The total cost can then be written in the following ways:

$$(6.3) \quad C(x) = \sum_{j \in J} w_j C_j(x) = \eta(x) + \sum_{i \in M} \sum_{j \in J} \sum_{k \prec_i j} w_j p_{ik} x_{ij} x_{ik}$$

$$(6.4) \quad C(x) = \eta(x) + \sum_{j \in J} D_j(x) = \eta(x) + \sum_{i \in M} \sum_{j \in J} \sum_{k \succ_i j} w_k p_{ij} x_{ij} x_{ik}.$$

6.1 A simple and natural local search algorithm. A natural and simple local search algorithm for this problem is to move a job from one machine to another if that improves the objective function. If such an improvement is not possible, then a local optimum $x \in \{0, 1\}^{M \times J}$ has been reached. Such a local optimum is called a *JumpOpt* in [27], and it is shown that the local optimality implies the following constraints. We provide a proof for the sake of completeness.

LEMMA 6.1. *For any local optimum JumpOpt solution x of the scheduling problem $R \parallel \sum w_j C_j$, the following constraints are satisfied:*

$$w_j C_j(x) + D_j(x) \leq w_j p_{ij} + \sum_{k \in J \setminus \{j\}} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \quad \forall j \in J, \forall i \in M.$$

Proof. Fix a job j assigned to machine $i^* \in M$ in the local optimum x and let us assume that this job switches to machine $i \in M$. The difference of weighted completion times for job j is

$$w_j \left(p_{ij} + \sum_{k \prec_i j} p_{ik} x_{ik} \right) - w_j \left(p_{i^* j} + \sum_{k \prec_{i^*} j} p_{i^* k} x_{i^* k} \right).$$

Moreover, the only other jobs for which the completion time is modified are the jobs assigned to i^* and i coming after j in the ordering of the respective machine. Due to the switch of j , these jobs assigned to i^* have their completion time decreased, whereas the ones assigned to i have their completion time increased. The total difference in cost for these jobs is then

$$\sum_{k \succ_i j} w_k p_{ij} x_{ik} - \sum_{k \succ_{i^*} j} w_k p_{i^* j} x_{i^* k}.$$

Since x is a local optimum for the global objective function, the total difference in cost (i.e. the sum of the two expressions above) should be non-negative. After rearranging the terms, this is equivalent to

$$w_j \left(p_{i^* j} + \sum_{k \prec_{i^*} j} p_{i^* k} x_{i^* k} \right) + \sum_{k \succ_{i^*} j} w_k p_{i^* j} x_{i^* k} \leq w_j \left(p_{ij} + \sum_{k \prec_i j} p_{ik} x_{ik} \right) + \sum_{k \succ_i j} w_k p_{ij} x_{ik}.$$

Observe that this is exactly the statement of the lemma, finishing the proof. \square

We now show that we can recover the tight approximation ratio of $(3 + \sqrt{5})/2$ given in [27] using our dual fitting approach. Observe the analogy with the proof strategy for the price of anarchy in the previous section. The main difference is that the Nash conditions are replaced by the local optimality conditions of Lemma 6.1, and the y variables are fitted differently.

THEOREM 6.2. *For any JumpOpt local optimum x of the scheduling problem $R \parallel \sum w_j C_j$, there exists a feasible (SDP-SR) solution with value at least $2/(3 + \sqrt{5}) C(x)$.*

Proof. We assume that the SDP vectors belong to the space $\mathcal{F}(M)$, which is without loss generality by Lemma 3.1. Let us fix α, β as in Lemma 4.3, i.e. $\alpha^2 := 2/\sqrt{5}$ and $\beta := 1/\alpha - \alpha/2$. We now state the dual fitting:

- $v_0(i, t) := \beta \sum_{k \in J} w_k x_{ik} \mathbb{1}_{\{t \leq \delta_{ik}\}}$

- $v_{ij}(i', t) := \alpha w_j \mathbb{1}_{\{t \leq \delta_{ij}\}} \mathbb{1}_{\{i=i'\}}$ $\forall j \in J, \forall i \in \mathcal{S}_j$
- $y_j := \alpha\beta \left(w_j C_j(x) + D_j(x) \right)$ $\forall j \in J$.

The desired inner products and norms can be computed to be the following, using essentially the same computations as in the proof of Theorem 4.1:

$$\begin{aligned} \frac{1}{\beta^2} \|v_0\|^2 &= 2C(x) - \eta(x) & \frac{1}{\alpha\beta} \langle v_0, v_{ij} \rangle &= \sum_{k \in J} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \\ \frac{1}{\alpha^2} \|v_{ij}\|^2 &= w_j p_{ij} & \frac{1}{\alpha^2} \langle v_{ij}, v_{i'k} \rangle &= w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \mathbb{1}_{\{i=i'\}}. \end{aligned}$$

The second set of SDP constraints is satisfied due to the last computation above and the fact that $\alpha^2 \leq 1$. The first set of constraints under this fitting gives:

$$\begin{aligned} y_j &\leq w_j p_{ij} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \\ \iff \alpha\beta \left(w_j C_j(x) + D_j(x) \right) &\leq \left(1 - \frac{\alpha^2}{2} \right) w_j p_{ij} + \alpha\beta \sum_{k \in J} w_j w_k x_{ik} \min\{\delta_{ij}, \delta_{ik}\}. \end{aligned}$$

These are satisfied by Lemma 4.3, which states that $1 - \alpha^2/2 = \alpha\beta$, as well as the local optimality conditions of Lemma 6.1. The objective function can now be lower bounded as:

$$\begin{aligned} \sum_{j \in J} y_j - \frac{1}{2} \|v_0\|^2 &= \alpha\beta \left(2C(x) - \eta(x) \right) - \frac{\beta^2}{2} \left(2C(x) - \eta(x) \right) = \frac{2}{3 + \sqrt{5}} \left(2C(x) - \eta(x) \right) \\ (6.5) \quad &\geq \frac{2}{3 + \sqrt{5}} C(x) \end{aligned}$$

where the first equality follows from (6.3) and (6.4), the second equality follows from the second property of Lemma 4.3 and the inequality follows from $\eta(x) \leq C(x)$. \square

We now show as in [27] that one can get an improved bound for the restricted identical machines setting, denoted by $P|\mathcal{M}_j| \sum w_j C_j$. The improvement comes from the fact that for a *JumpOpt* solution x and an optimal solution x^* , we have $\eta(x) = \eta(x^*) = \sum_{j \in J} w_j p_j$ in this setting. This means that, instead of bounding $\eta(x) \leq C(x)$ in the last step of (6.5), we can now use the stronger upper bound $\eta(x) \leq C(x^*)$.

THEOREM 6.3. *For any *JumpOpt* local optimum x of the scheduling problem $P|\mathcal{M}_j| \sum w_j C_j$, there exists a feasible (SDP-SR) solution with value at least $2/(3 + \sqrt{5}) (2C(x) - C(x^*))$. By weak duality, this implies that the approximation ratio of x is at most $(5 + \sqrt{5})/4 \approx 1.809$.*

Proof. By upper bounding $\eta(x) \leq C(x^*)$ in the last step of (6.5), we get the first statement of the theorem. By weak duality, and since the dual solution constructed is feasible, we get that

$$\frac{2}{3 + \sqrt{5}} \left(2C(x) - C(x^*) \right) \leq C(x^*) \iff \frac{C(x)}{C(x^*)} \leq \frac{5 + \sqrt{5}}{4}. \quad \square$$

6.2 An improved local search algorithm. In this subsection, we show how our approach allows to analyze an improved local search algorithm for $R||\sum w_j C_j$ by [18] achieving an approximation ratio of $(5 + \sqrt{5})/4 + \varepsilon \approx 1.809 + \varepsilon$ for every $\varepsilon > 0$. To the best of our knowledge, this is the best currently known combinatorial approximation algorithm for this problem. We ignore here the issue of the running time and simply analyze the quality of a local optimum, referring the reader to [18] for further details. Let us fix the constant $\gamma := (9 + \sqrt{5})/19 \approx 0.591$. For each job $j \in J$ and an assignment x , we keep a potential function

$$f_j(x) = \sum_{i \in M} x_{ij} \left(w_j p_{ij} + \gamma \sum_{k \neq j} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \right) \quad \forall j \in J.$$

If a job $j \in J$ can pick a different machine than the one it is currently on and decrease its potential function $f_j(x)$, then this constitutes an improving move for the local search algorithm. If several improving moves exist, the algorithm picks the one giving the largest decrease in $f_j(x)$. For a local optimum x , we get the following constraints:

$$(6.6) \quad f_j(x) \leq w_j p_{ij} + \gamma \sum_{k \neq j} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \quad \forall j \in J, \forall i \in M.$$

As usual with this approach, we first need a small lemma about important constants.

LEMMA 6.4. *Let $\alpha, \beta, \gamma \geq 0$ be defined as $\alpha^2 = (\sqrt{5} + 1)/5$, $\beta^2 = (\sqrt{5} - 1)/5$ and $\gamma = (9 + \sqrt{5})/19$. The following properties hold:*

- $\alpha\beta/\gamma = 1 - \alpha^2/2$
- $\alpha\beta(2\gamma - 1)/\gamma = \beta^2/2$
- $2\alpha\beta - \beta^2 = 4/(5 + \sqrt{5})$

Proof. The proof consists of simple computations and is omitted. These equations can also be checked on a computer. \square

THEOREM 6.5. *For any local optimum x of the above local search algorithm for $R||\sum w_j C_j$, there exists a feasible (SDP-SR) solution with value at least $4/(5 + \sqrt{5}) C(x)$.*

Proof. We assume that the SDP vectors belong to the space $\mathcal{F}(M)$, which is without loss of generality by Lemma 3.1. Let us fix α, β, γ as in Lemma 6.4. We now state the dual fitting:

- $v_0(i, t) := \beta \sum_{k \in J} w_k x_{ik} \mathbb{1}_{\{t \leq \delta_{ik}\}}$
- $v_{ij}(i', t) := \alpha w_j \mathbb{1}_{\{t \leq \delta_{ij}\}} \mathbb{1}_{\{i = i'\}}$ $\forall j \in J, \forall i \in \mathcal{S}_j$
- $y_j := \frac{\alpha\beta}{\gamma} f_j(x)$ $\forall j \in J$.

The desired inner products and norms can be computed to be the following, using essentially the same computations as in the proof of Theorem 4.1:

$$\begin{aligned} \frac{1}{\beta^2} \|v_0\|^2 &= 2C(x) - \eta(x) & \frac{1}{\alpha\beta} \langle v_0, v_{ij} \rangle &= \sum_{k \in J} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \\ \frac{1}{\alpha^2} \|v_{ij}\|^2 &= w_j p_{ij} & \frac{1}{\alpha^2} \langle v_{ij}, v_{i'k} \rangle &= w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \mathbb{1}_{\{i = i'\}}. \end{aligned}$$

The second set of SDP constraints is satisfied due to the last computation above and the fact that $\alpha^2 \leq 1$. The first set of constraints under this fitting gives:

$$\begin{aligned} y_j &\leq w_j p_{ij} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \\ \iff \frac{\alpha\beta}{\gamma} f_j(x) &\leq \left(1 - \frac{\alpha^2}{2}\right) w_j p_{ij} + \frac{\alpha\beta}{\gamma} \gamma \sum_{k \in J} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik}. \end{aligned}$$

These are satisfied by Lemma 6.4, as well as the local optimality conditions (6.6). To argue about the objective, it can be checked (through a simple computation that we omit) that:

$$\sum_{j \in J} f_j(x) = 2\gamma C(x) - (2\gamma - 1) \eta(x).$$

The objective function then becomes:

$$\begin{aligned} \sum_{j \in J} y_j - \frac{1}{2} \|v_0\|^2 &= \frac{\alpha\beta}{\gamma} \left(2\gamma C(x) - (2\gamma - 1) \eta(x) \right) - \frac{\beta^2}{2} \left(2C(x) - \eta(x) \right) \\ &= \left(2\alpha\beta - \beta^2 \right) C(x) - \left(\frac{\alpha\beta (2\gamma - 1)}{\gamma} - \frac{\beta^2}{2} \right) \eta(x) \\ &= \left(2\alpha\beta - \beta^2 \right) C(x) = \frac{4}{5 + \sqrt{5}} C(x) \end{aligned}$$

where the two last equalities follow from Lemma 6.4. \square

We now provide an almost matching lower bound instance, inspired by constructions in [17, 27]. We believe that the upper bound of $(5 + \sqrt{5})/4 \approx 1.809$ is tight.

THEOREM 6.6. *There exists an instance of $R \parallel \sum w_j C_j$ with a local optimum to the above local search algorithm with approximation ratio at least 1.791.*

Proof. Let $\lambda \approx 1.33849$ be the positive solution to the equation $\lambda^2 = 1 + \gamma \lambda$. We consider an instance with jobs $J = [n]$ and machines $M = [n+1]$. The weights of the jobs are defined as $w_1 = \lambda$ and $w_j = 1/\lambda^{j-1}$ for every $j \geq 2$. The feasible machines are $\mathcal{S}_j = \{j, j+1\}$ for every $j \in J$ with processing times $p_{1,1}, p_{2,1} = \lambda$ for the first job and $p_{j,j} = \lambda^{j-1}, p_{j+1,j} = \lambda^{j+1}$ for every $j \geq 2$.

The feasible solution where each job j gets assigned to machine j has cost $\sum_{j \in J} w_j p_{j,j} = \lambda^2 + (n-1)$, showing that the optimum solution x^* satisfies $C(x^*) \leq n - 1 + \lambda^2$.

We now claim that the solution x where each job j gets assigned to machine $j+1$ is a local optimum. To see this, observe that the first job clearly cannot decrease his potential function $f_1(x)$ since $p_{1,1} = p_{2,1}$ and no other job is assigned to machine 1 or 2. For $j \geq 2$, we have $f_j(x) = w_j p_{j+1,j} = \lambda^2$. If job j were to be reassigned to machine j , then

$$f_j(x_{-j}, j) = w_j p_{j,j} + \gamma w_{j-1} w_j \min\{\delta_{j,j-1}, \delta_{j,j}\} = 1 + \gamma \lambda,$$

which shows that x is a local optimum, by definition of λ . The cost of this solution is then $\sum_{j \in N} w_j p_{j+1,j} = n\lambda^2$. The approximation ratio of this solution now satisfies

$$\frac{C(x)}{C(x^*)} \geq \frac{n\lambda^2}{n - 1 + \lambda^2} \xrightarrow{n \rightarrow \infty} \lambda^2 \approx 1.79154.$$

Picking n large enough thus finishes the proof. \square

7 An online randomized algorithm. In this section, we show that our technique extends to online algorithms by analyzing a 4-competitive randomized algorithm for the problem $R \parallel \sum w_j C_j$.² As a reminder, jobs arrive online and the ordering on each machine is optimal, i.e. when a job j is assigned to a machine i , it enters the schedule in the right position with respect to the Smith ratio $\delta_{ij} = p_{ij}/w_j$. The (SDP-SR) relaxation (4.4) for $R \parallel \sum w_j C_j$ becomes the following:

$$\begin{aligned} \max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \\ y_j \leq w_j p_{ij} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle & \quad \forall j \in N, \forall i \in \mathcal{S}_j \\ \langle v_{ij}, v_{i'k} \rangle \leq w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \mathbb{1}_{\{i=i'\}} & \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N. \end{aligned}$$

Since we are now considering a randomized algorithm, we will refer to $X_{ij} \in \{0, 1\}$ as the binary random variable indicating whether $j \in N$ is assigned to $i \in M$. Moreover, we will refer to $x_{ij} := \mathbb{E}[X_{ij}] \in [0, 1]$ as the probabilities defined by the algorithm. The random completion time of a job $j \in N$ is thus

$$C_j(X) = \sum_{i \in M} \sum_{k \preceq i} p_{ik} X_{ij} X_{ik}.$$

²In the ArXiv version of this paper, we show how to analyze the greedy algorithm in the congestion model achieving the same bound.

The total cost is the sum of weighted completion times:

$$(7.1) \quad C(X) := \sum_{j \in N} w_j C_j(X) = \sum_{i \in M} \sum_{j \in N, k \preceq i j} w_j p_{ik} X_{ij} X_{ik}.$$

We can split this cost on a machine by machine basis by defining:

$$(7.2) \quad L_i(X) = \sum_{j \in N, k \preceq i j} w_j p_{ik} X_{ij} X_{ik} \quad \forall i \in M.$$

Clearly, we have that $C(X) = \sum_{i \in M} L_i(X)$. Since the jobs arrive online, we denote them as $N = \{1, \dots, n\}$, where job k arrives before job j if $k < j$.

7.1 The algorithm. Let us now describe the algorithm. Whenever $j \in N$ arrives, we consider the following potential functions for every $i \in \mathcal{S}_j$:

$$(7.3) \quad f_{ij}(t) := w_j p_{ij} + 2 \left(t w_j p_{ij} + \sum_{k < j} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \right).$$

We now consider a waterfilling type of algorithm, described in Algorithm 7.1, which defines a probability distribution $(x_{ej})_{e \in \mathcal{S}_j}$ ensuring that

$$x_{ej} > 0 \implies f_{ej}(x_{ej}) \leq f_{ij}(x_{ij}) \quad \forall i \in \mathcal{S}_j.$$

For convenience of notation, we also let $x_{ej} = 0$ if $e \notin \mathcal{S}_j$. Observe that this means that for every $e \in \mathcal{S}_j$ with $x_{ej} > 0$, we get that $f_{ej}(x_{ej}) = \lambda$ for some constant λ , whereas $f_{ej}(x_{ej}) \geq \lambda$ if $x_{ej} = 0$. In particular, since $\sum_{e \in \mathcal{S}_j} x_{ej} = 1$, we also get the following equilibrium inequality:

$$(7.4) \quad \sum_{e \in M} x_{ej} f_{ej}(x_{ej}) \leq f_{ij}(x_{ij}) \quad \forall i \in \mathcal{S}_j.$$

Once the algorithm has constructed the distribution $(x_{ij})_{i \in \mathcal{S}_j}$, it simply randomly samples a machine from this distribution to assign the job to. Note that the random decisions are independent for any two jobs $j \neq k$.

Algorithm 7.1 Waterfilling and randomized rounding

when $j \in N$ arrives:

Compute $(x_{ij})_{i \in \mathcal{S}_j}$ such that $\sum_i x_{ij} = 1$ and $\sum_{e \in M} x_{ej} f_{ej}(x_{ej}) \leq f_{ij}(x_{ij}) \quad \forall i \in \mathcal{S}_j$

Assign j to $i \in \mathcal{S}_j$, i.e. set $X_{ij} = 1$ with probability x_{ij}

return X

7.2 Analysis of the algorithm. By a slight abuse of notation, let us denote the expected cost per machine $i \in M$ as

$$(7.5) \quad L_i(x) := \mathbb{E}[L_i(X)] = \sum_{j \in N, k \preceq i j} w_j p_{ik} \mathbb{E}[X_{ij} X_{ik}]$$

and let us also define the following quantity for every $i \in M$:

$$(7.6) \quad F_i(x) := \sum_{j \in N, k \preceq i j} w_j p_{ik} x_{ij} x_{ik}.$$

Since the algorithm makes independent choices for any two jobs $j \neq k$, meaning that $\mathbb{E}[X_{ij} X_{ik}] = \mathbb{E}[X_{ij}] \mathbb{E}[X_{ik}] = x_{ij} x_{ik}$, the only difference between $L_i(x)$ and $F_i(x)$ occurs when $j = k$, in which case we respectively get a contribution of $\mathbb{E}[X_{ij}^2] = \mathbb{E}[X_{ij}] = x_{ij}$ and x_{ij}^2 .

Let us now define $L_i^{(j)}(x)$ and $F_i^{(j)}(x)$ to be the expressions (7.5) and (7.6) for the assignment x after an online job j has arrived (i.e. where $x_{ik} = 0$ for every $k > j$ and every $i \in M$). We now define the increments over time of these two quantities as:

$$\delta_i^{(j)}(x) := F_i^{(j)}(x) - F_i^{(j-1)}(x) \quad \text{and} \quad \Delta_i^{(j)}(x) := L_i^{(j)}(x) - L_i^{(j-1)}(x).$$

We are now ready to state key inequalities satisfied by our algorithm which will be needed for the dual fitting. These inequalities are analogous to the Nash equilibrium inequalities or local optima inequalities needed in the previous sections.

LEMMA 7.1. *For any solution $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$ constructed by Algorithm 7.1, the following inequalities are satisfied for every $j \in N$:*

$$\sum_{e \in M} \left(\delta_e^{(j)}(x) + \Delta_e^{(j)}(x) \right) \leq w_j p_{ij} + 2 \sum_{k \in N} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \quad \forall i \in \mathcal{S}_j.$$

Proof. Let us first understand how the cost increases when $j \in N$ is assigned to $i \in M$ by the algorithm. The weighted completion time of j increases the cost by:

$$w_j \left(p_{ij} + \sum_{k < j: k \succ_{ij}} p_{ik} X_{ik} \right).$$

Moreover, the completion time of the already arrived jobs assigned to that machine coming after j in the ordering of that machine (i.e. with a higher Smith ratio) is increased by p_{ij} , leading to an increase in the objective of

$$\sum_{k < j: k \succ_{ij}} w_k p_{ij} X_{ik}.$$

The total increase in cost (i.e. the sum of the two above quantities) can be written succinctly as follows:

$$w_j p_{ij} + \sum_{k < j} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} X_{ik}.$$

Using that, we can see that the increments become:

$$\begin{aligned} \Delta_e^{(j)}(x) &= x_{ej} \left(w_j p_{ej} + \sum_{k < j} w_j w_k \min\{\delta_{ej}, \delta_{ek}\} x_{ek} \right), \\ \delta_e^{(j)}(x) &= x_{ej} \left(w_j p_{ej} x_{ej} + \sum_{k < j} w_j w_k \min\{\delta_{ej}, \delta_{ek}\} x_{ek} \right) = x_{ej} \left(\sum_{k \leq j} w_j w_k \min\{\delta_{ej}, \delta_{ek}\} x_{ek} \right) \end{aligned}$$

where the last equality follows from $w_j p_{ej} = w_j^2 \delta_{ej}$. From the above two equations, we see that $\delta_e^{(j)}(x) + \Delta_e^{(j)}(x) \leq x_{ej} f_{ej}(x_{ej})$, by definition of (7.3). The equilibrium condition (7.4) then tells us that

$$\sum_{e \in M} \left(\delta_e^{(j)}(x) + \Delta_e^{(j)}(x) \right) \leq \sum_{e \in M} x_{ej} f_{ej}(x_{ej}) \leq f_{ij}(x_{ij}) \quad \forall i \in \mathcal{S}_j.$$

Looking again at the definition (7.3) completes the proof of the lemma. \square

We are now ready to analyze the competitive ratio of the algorithm in this model.

THEOREM 7.2. *For any online instance and any solution $(x_{ij})_{j \in N, i \in \mathcal{S}_j}$ obtained by Algorithm 7.1, there exists a feasible (SDP-SR) solution with objective value at least $\mathbb{E}[C(X)]/4$. By weak duality, this implies that the competitive ratio of the algorithm is at most 4.*

Proof. We assume that the SDP vectors belong to the space $\mathcal{F}(M)$, which is without loss of generality by Lemma 3.1. We now state the dual fitting:

- $v_0(i, t) := \frac{1}{2} \sum_{k \in N} w_k x_{ik} \mathbf{1}_{\{t \leq \delta_{ik}\}}$
- $v_{ij}(i', t) := w_j \mathbf{1}_{\{t \leq \delta_{ij}\}} \mathbf{1}_{\{i=i'\}} \quad \forall j \in N, \forall i \in \mathcal{S}_j$
- $y_j := \frac{1}{4} \sum_{e \in M} (\delta_e^{(j)}(x) + \Delta_e^{(j)}(x)) \quad \forall j \in N.$

The desired inner products can be computed to be the following, using essentially the same computations as in Theorem 4.1:

$$\begin{aligned} \langle v_0, v_{ij} \rangle &= \frac{1}{2} \sum_{k \in N} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik} \quad \|v_{ij}\|^2 = w_j p_{ij} \\ \|v_0\|^2 &= \frac{1}{4} \sum_{j, k \in N} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ij} x_{ik} \quad \langle v_{ij}, v_{i'k} \rangle = w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \mathbf{1}_{\{i=i'\}}. \end{aligned}$$

The second set of SDP constraints is tightly satisfied due to the last computation above. The first set of constraints under this fitting gives:

$$\begin{aligned} y_j &\leq w_j p_{ij} - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle \\ \iff \frac{1}{4} \sum_{e \in M} (\delta_e^{(j)}(x) + \Delta_e^{(j)}(x)) &\leq \frac{1}{2} w_j p_{ij} + \frac{1}{2} \sum_{k \in N} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ik}. \end{aligned}$$

These are now satisfied by Lemma 7.1. To argue about the objective, observe that

$$\sum_{j \in N} y_j = \frac{1}{4} \sum_{e \in M} \sum_{j \in N} (\delta_e^{(j)}(x) + \Delta_e^{(j)}(x)) = \frac{1}{4} \sum_{e \in M} (L_e(x) + F_e(x))$$

by using the definition of the increments and the fact that the sum is telescoping. In addition,

$$\|v_0\|^2 = \frac{1}{4} \sum_{j, k \in N} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} x_{ij} x_{ik} \leq \frac{1}{2} \sum_{e \in M} F_e(x).$$

Hence, the objective can be lower bounded as:

$$\sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 \geq \frac{1}{4} \sum_{e \in M} (L_e(x) + F_e(x)) - \frac{1}{4} \sum_{e \in M} F_e(x) = \frac{1}{4} \sum_{e \in M} L_e(x) = \frac{1}{4} \mathbb{E}[C(X)]. \quad \square$$

8 Concluding remarks. In this paper, we built on the work of [50] which showed a way to use convex programming duality to prove price of anarchy bounds for different games. We showed that a unique convex program turns out to be surprisingly powerful and allows to get tight upper bounds for a large class of congestion and scheduling games. Moreover, it can also be used to bound the approximation ratio of local search algorithms and the competitive ratio of online algorithms for such problems. The dual program has a simple structure with the first set of constraints being similar to equilibrium inequalities, guiding the dual fitting approach. This program also has a natural connection to the inner product space structure developed in [25]. It would be interesting if this approach can be extended to new games where price of anarchy bounds are not yet settled. Moreover, all the problems we considered had a quadratic (possibly non-convex) objective function, which made the first round of the Lasserre/Sum of Squares SDP hierarchy powerful enough to write a tractable convex relaxation. It would be interesting if a similar technique can work for problems with a higher degree polynomial objective by considering later rounds of the hierarchy. To the best of our knowledge, such dual fitting arguments on semidefinite programs have not been explored much: we hope and believe that there may be additional applications to such an approach.

Acknowledgements. The author would like to thank Zhuan Khye Koh for useful discussions. The author would also like to thank Guido Schäfer for useful discussions, as well as nice feedback on an earlier version of this document. The author also thanks an anonymous reviewer who pointed out that the online greedy algorithm has been analyzed in [35].

References

- [1] F. Abed, J. R. Correa, and C.-C. Huang. Optimal coordination mechanisms for multi-job scheduling games. In *Algorithms-ESA 2014: 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings 21*, pages 13–24. Springer, 2014.
- [2] F. Abed and C.-C. Huang. Preemptive coordination mechanisms for unrelated machines. In *European Symposium on Algorithms*, pages 12–23. Springer, 2012.
- [3] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. *SIAM Journal on Computing*, 40(5):1211–1233, 2011.
- [4] S. Anand, N. Garg, and A. Kumar. Resource augmentation for weighted flow-time explained by dual fitting. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1228–1241. SIAM, 2012.
- [5] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 57–66, 2005.
- [6] B. Awerbuch, Y. Azar, E. F. Grove, M.-Y. Kao, P. Krishnan, and J. S. Vitter. Load balancing in the $l_1/\text{sub } p/\text{norm}$. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 383–391. IEEE, 1995.
- [7] Y. Azar, K. Jain, and V. Mirrokni. (almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 323–332, 2008.
- [8] N. Bansal and K. Pruhs. Server scheduling in the l_p norm: a rising tide lifts all boat. In *Proceedings of the thirty-fifth annual ACM symposium on theory of computing*, pages 242–250, 2003.
- [9] N. Bansal, A. Srinivasan, and O. Svensson. Lift-and-round to improve weighted completion time on unrelated machines. In *Proceedings of the forty-eighth annual ACM symposium on theory of computing*, pages 156–167, 2016.
- [10] S. Bhattacharya, S. Im, J. Kulkarni, and K. Munagala. Coordination mechanisms from (almost) all scheduling policies. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 121–134, 2014.
- [11] S. Bhattacharya, J. Kulkarni, and V. Mirrokni. Coordination mechanisms for selfish routing over time on a tree. In *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I 41*, pages 186–197. Springer, 2014.
- [12] K. Bhawalkar, M. Gairing, and T. Roughgarden. Weighted congestion games: the price of anarchy, universal worst-case examples, and tightness. *ACM Transactions on Economics and Computation (TEAC)*, 2(4):1–23, 2014.
- [13] J. Bruno, E. G. Coffman Jr, and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17(7):382–387, 1974.
- [14] I. Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 972–981. Citeseer, 2008.
- [15] I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. *Algorithmica*, 66(3):512–540, 2013.
- [16] I. Caragiannis and A. Fanelli. An almost ideal coordination mechanism for unrelated machine scheduling. *Theory of Computing Systems*, 63:114–127, 2019.
- [17] I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I 33*, pages 311–322. Springer, 2006.

- [18] I. Caragiannis, V. Gkatzelis, and C. Vinci. Coordination mechanisms, cost-sharing, and approximation algorithms for scheduling. In *Web and Internet Economics: 13th International Conference, WINE 2017, Bangalore, India, December 17–20, 2017, Proceedings 13*, pages 74–87. Springer, 2017.
- [19] J. S. Chadha, N. Garg, A. Kumar, and V. Muralidhara. A competitive algorithm for minimizing weighted flow time on unrelated machines with speed augmentation. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 679–684, 2009.
- [20] C. Chekuri, S. Khanna, and A. Zhu. Algorithms for minimizing weighted flow time. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 84–93, 2001.
- [21] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *European Symposium on Algorithms*, pages 59–70. Springer, 2005.
- [22] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. In *International Colloquium on Automata, Languages, and Programming*, pages 345–357. Springer, 2004.
- [23] G. Christodoulou, K. Mehlhorn, and E. Pyrga. Improving the price of anarchy for selfish routing via coordination mechanisms. *Algorithmica*, 69(3):619–640, 2014.
- [24] J. Cohen, C. Dürr, and T. Nguyen Kim. Non-clairvoyant scheduling games. *Theory of Computing Systems*, 49:3–23, 2011.
- [25] R. Cole, J. R. Correa, V. Gkatzelis, V. Mirrokni, and N. Olver. Inner product spaces for minsum coordination mechanisms. In *Proceedings of the forty-third annual ACM symposium on theory of computing*, pages 539–548, 2011.
- [26] M. Conway and W. Maxwell. Miller, theory of scheduling. *Reading: Addison Wesley*, 1967.
- [27] J. R. Correa and F. T. Muñoz. Performance guarantees of local search for minsum scheduling problems. *Mathematical Programming*, 191(2):847–869, 2022.
- [28] J. R. Correa and M. Queyranne. Efficiency of equilibria in restricted uniform machine scheduling with total weighted completion time as social cost. *Naval Research Logistics (NRL)*, 59(5):384–395, 2012.
- [29] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions On Algorithms (TALG)*, 3(1):1–17, 2007.
- [30] B. Farzad, N. Olver, and A. Vetta. A priority-based model of routing. *Chicago Journal of Theoretical Computer Science*, 1, 2008.
- [31] L. Fleischer and Z. Svitkina. Preference-constrained oriented matching. In *2010 Proceedings of the Seventh Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 66–73. SIAM, 2010.
- [32] N. Garg and A. Kumar. Minimizing average flow-time: Upper and lower bounds. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 603–613. IEEE, 2007.
- [33] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, volume 5, pages 287–326. Elsevier, 1979.
- [34] A. Gupta, R. Krishnaswamy, and K. Pruhs. Online primal-dual for non-linear optimization with applications to speed scaling. In *International Workshop on Approximation and Online Algorithms*, pages 173–186. Springer, 2012.
- [35] V. Gupta, B. Moseley, M. Uetz, and Q. Xie. Greed works—online algorithms for unrelated machine stochastic scheduling. *Mathematics of operations research*, 45(2):497–516, 2020.

- [36] D. G. Harris. Dependent rounding with strong negative-correlation, and scheduling on unrelated machines to minimize completion time. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2275–2304. SIAM, 2024.
- [37] R. Hoeksma and M. Uetz. The price of anarchy for minsum related machine scheduling. In *International workshop on approximation and online algorithms*, pages 261–273. Springer, 2011.
- [38] H. Hoogeveen, P. Schuurman, and G. J. Woeginger. Non-approximability results for scheduling problems with minsum criteria. In *International conference on integer programming and combinatorial optimization*, pages 353–366. Springer, 1998.
- [39] W. Horn. Minimizing average flow time with parallel machines. *Operations research*, 21(3):846–847, 1973.
- [40] S. Im and S. Li. Improved approximations for unrelated machine scheduling. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2917–2946. SIAM, 2023.
- [41] S. Im and B. Moseley. An online scalable algorithm for minimizing lk-norms of weighted flow time on unrelated machines. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 95–108. SIAM, 2011.
- [42] S. Im and M. Shadloo. Weighted completion time minimization for unrelated machines via iterative fair contention resolution*. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2790–2809. SIAM, 2020.
- [43] N. Immorlica, L. E. Li, V. S. Mirrokni, and A. S. Schulz. Coordination mechanisms for selfish scheduling. *Theoretical computer science*, 410(17):1589–1598, 2009.
- [44] C. Kalaitzis, O. Svensson, and J. Tarnawski. Unrelated machine scheduling of jobs with uniform smith ratios. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2654–2669. SIAM, 2017.
- [45] B. Kalyanasundaram and K. Pruhs. Speed is as powerful as clairvoyance. *Journal of the ACM (JACM)*, 47(4):617–643, 2000.
- [46] T. Kawaguchi and S. Kyan. Worst case bound of an lrf schedule for the mean weighted flow-time problem. *SIAM Journal on Computing*, 15(4):1119–1129, 1986.
- [47] H. Kellerer, T. Tautenhahn, and G. J. Woeginger. Approximability and nonapproximability results for minimizing total flow time on a single machine. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pages 418–426, 1996.
- [48] K. Kollias. Nonpreemptive coordination mechanisms for identical machines. *Theory of Computing Systems*, 53:424–440, 2013.
- [49] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Annual symposium on theoretical aspects of computer science*, pages 404–413. Springer, 1999.
- [50] J. Kulkarni and V. Mirrokni. Robust price of anarchy bounds via lp and fenchel duality. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1030–1049. SIAM, 2014.
- [51] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.
- [52] J. K. Lenstra, A. R. Kan, and P. Brucker. Complexity of machine scheduling problems. In *Annals of discrete mathematics*, volume 1, pages 343–362. Elsevier, 1977.
- [53] S. Li. Approximating unrelated machine weighted completion time using iterative rounding and computer assisted proofs. *arXiv preprint arXiv:2404.04773*, 2024.

- [54] M. Rahn and G. Schäfer. Bounding the inefficiency of altruism through social contribution games. In *International Conference on Web and Internet Economics*, pages 391–404. Springer, 2013.
- [55] T. Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5):1–42, 2015.
- [56] U. Schwiegelshohn. An alternative proof of the kawaguchi–kyan bound for the largest-ratio-first rule. *Operations Research Letters*, 39(4):255–259, 2011.
- [57] J. Sethuraman and M. S. Squillante. Optimal scheduling of multiclass parallel machines. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 963–964, 1999.
- [58] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *Journal of the ACM (JACM)*, 48(2):206–242, 2001.
- [59] M. Skutella and G. J. Woeginger. A ptas for minimizing the total weighted completion time on identical parallel machines. *Mathematics of Operations Research*, 25(1):63–75, 2000.
- [60] W. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.
- [61] S. Suri, C. D. Tóth, and Y. Zhou. Selfish load balancing and atomic congestion games. In *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 188–195, 2004.

A Recovering the Kawaguchi-Kyan bound for $P \parallel \sum w_j C_j$. In this section, we show that we can recover the optimal bound of $(1 + \sqrt{2})/2$ for the pure price of anarchy of the scheduling game on parallel machines $P \parallel \sum w_j C_j$, where each machine uses increasing Smith ratios to schedule the jobs. To do so, we make use of a sequence of reductions to worst-case instances provided in [56]. The first assumption that we can make is that $w_j = p_j$ for every job j . The (SDP-SR) dual semidefinite program shown in (4.4) and used in Section 6 for $R \parallel \sum w_j C_j$ in this special case becomes the following. We denote the set of jobs by J and the set of machines by M .

$$\begin{aligned} \max \sum_{j \in J} y_j - \frac{1}{2} \|v_0\|^2 \\ y_j \leq p_j^2 - \frac{1}{2} \|v_{ij}\|^2 + \langle v_0, v_{ij} \rangle & \quad \forall j \in J, \forall i \in M \\ \langle v_{ij}, v_{i'k} \rangle \leq p_j p_k \mathbf{1}_{\{i=i'\}} & \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in J. \end{aligned}$$

Moreover, the reduction in [56] states that we may assume the instance only has two different processing times $\varepsilon, p > 0$, where ε is an arbitrarily small constant. Jobs with processing time ε are called small jobs, and the total workload of these jobs is $|M|$, i.e. the total number of small jobs is $|M|/\varepsilon$. Jobs with processing times p are called large jobs and the total number of large jobs is $k < |M|$, i.e. strictly less than the number of machines. In addition, in a pure Nash equilibrium x :

- All small jobs are started and completed in the interval $[0, 1]$.
- All large jobs are started at 1.

In this reduced instance, it is also possible to get an exact expression for the optimum solution. In particular, define $\alpha := m/(m - k)$ and $\beta := (m + pk)/m$, an optimal solution x^* then has cost:

$$C(x^*) = \begin{cases} kp^2 + \frac{m-k}{2}\alpha^2 & \text{if } p \geq \alpha \\ \frac{1}{2}(kp^2 + m\beta^2) & \text{if } p \leq \alpha \end{cases}$$

It can then be shown that in both cases $C(x)/C(x^*) \leq (1 + \sqrt{2})/2$ through a simple calculus analysis. The reader is referred to [56] for details.

We show here that we can construct a feasible dual solution to the SDP matching the objective value of $C(x^*)$, showing that the SDP does not have an integrality gap on such a reduced instance and thus implying that the price of anarchy is at most $(1 + \sqrt{2})/2$ by a dual fitting proof.

THEOREM A.1. *For any instance of the above game on the reduced instance, there exists a feasible (SDP-SR) solution with objective value $C(x^*)$, implying that the price of anarchy is at most $(1 + \sqrt{2})/2$.*

Proof. The vectors in our dual fitting will live in the space \mathbb{R}^M . Let us denote the total number of machines by $m = |M|$, and let us set $\alpha := m/(m - k)$. We denote by $\mathbf{1}$ the all ones vector and by e_i the i^{th} standard basis vector.

We now state the dual fitting for the case where $p \geq \alpha$:

- $v_0 = \alpha \mathbf{1}$
- If j is a large job, then set $v_{ij} = \alpha e_i$ and $y_j = p^2 + \alpha^2/2$
- If j is a small job, then set $v_{ij} = \varepsilon e_i$ and $y_j = \varepsilon\alpha$

Let us check that this solution is indeed feasible. Clearly, if $i \neq i'$, then $\langle v_{ij}, v_{i'k} \rangle = 0$ by orthogonality of e_i and $e_{i'}$. For two jobs $j \neq k$, we have that $\langle v_{ij}, v_{ik} \rangle \leq \|v_{ij}\| \|v_{ik}\| \leq p_j p_k$ where we use Cauchy-Schwarz for the first inequality and the fact that $\alpha \leq p$ if some job is large for the second inequality. This shows that the second set of SDP constraints is satisfied.

Moreover, the first set of constraints is satisfied as well, as the SDP inequalities yield $y_j \leq p^2 + \alpha^2/2$ for large jobs and $y_j \leq \varepsilon^2/2 + \varepsilon\alpha$ for small jobs, which is satisfied by our choice of y_j . The objective value of this dual solution is then:

$$\sum_{j \in J} y_j - \frac{1}{2} \|v_0\|^2 = k \left(p^2 + \frac{\alpha^2}{2} \right) + \frac{m}{\varepsilon} \varepsilon \alpha - \frac{1}{2} m \alpha^2 = kp^2 + \frac{m - k}{2} \alpha^2$$

where the first equality follows since the number of small jobs is m/ε and the last equality follows by observing that $m\alpha = (m - k)\alpha^2$ by definition of α .

For the case where $p \leq \alpha$, we define $\beta := (m + pk)/m$. We now state the dual fitting:

- $v_0 = \beta \mathbf{1}$
- If j is a large job, then set $v_{ij} = p e_i$ and $y_j = p^2/2 + \beta p$
- If j is a small job, then set $v_{ij} = \varepsilon e_i$ and $y_j = \varepsilon\beta$

Similarly to before, the second set of constraints is satisfied by orthogonality of the standard basis vectors and the fact that $\|v_{ij}\| = p_j$ for all jobs (either small or large). The first set of constraints yields $y_j \leq p^2/2 + \beta p$ for large jobs and $y_j \leq \varepsilon^2/2 + \varepsilon\beta$ for small jobs, which is clearly satisfied by our choice of y_j . The objective value of this dual solution is then:

$$\sum_{j \in J} y_j - \frac{1}{2} \|v_0\|^2 = k \left(\frac{p^2}{2} + \beta p \right) + \frac{m}{\varepsilon} \varepsilon \beta - \frac{m\beta^2}{2} = \frac{kp^2}{2} + \beta(kp + m) - \frac{m\beta^2}{2} = \frac{1}{2}(kp^2 + m\beta^2)$$

where the last equality follows by observing that $m\beta^2 = (m + pk)\beta$ by the definition of β . \square

B Missing proofs.

B.1 Proof of Lemma 3.1.

Proof. For both spaces, we argue that a collection of elements in it can be mapped to a collection of vectors in a standard Euclidean space while preserving all the pairwise inner products (and thus also preserving the norms). This then clearly implies the claim.

We first show the statement for $\mathcal{F}(E)$. Let us denote the difference between two breakpoints as $\Delta_j^{(e)} := \delta_j^{(e)} - \delta_{j-1}^{(e)}$. For each element $f \in \mathcal{F}(E)$, define $f' \in \mathbb{R}^{E \times [n]}$ as $f'(e, j) := f\left(e, \delta_{j-1}^{(e)} + \Delta_j^{(e)}/2\right) \sqrt{\Delta_j^{(e)}}$. By computing the integral of a step function, we clearly have that for $f, g \in \mathcal{F}(E)$,

$$\langle f, g \rangle = \sum_{e \in E} \sum_{j=1}^n \Delta_j^{(e)} f\left(e, \delta_{j-1}^{(e)} + \Delta_j^{(e)}/2\right) g\left(e, \delta_{j-1}^{(e)} + \Delta_j^{(e)}/2\right) = \langle f', g' \rangle.$$

Note that the last inner product is the standard Euclidean one, thus showing the claim for $\mathcal{F}(E)$.

We now show the claim for $\mathcal{G}(E, M)$. Let us write a rank one decomposition of the positive definite matrix $M = \sum_{j=1}^K u_j u_j^T$, which can for instance be done through the spectral decomposition. For each $f \in \mathcal{G}(E, M)$, we define a modified $f' \in \mathbb{R}^{E \times [K]}$ as $f'(e, j) := f(e, \cdot)^T u_j$. Clearly, we then have that, for $f, g \in \mathcal{G}(E, M)$:

$$\langle f, g \rangle = \sum_{e \in E} \sum_{j=1}^K f(e, \cdot)^T u_j u_j^T g(e, \cdot) = \sum_{e \in E} \sum_{j=1}^K f'(e, j) g'(e, j) = \langle f', g' \rangle.$$

□

C Preliminaries on SDPs. A symmetric matrix $X \in \mathbb{R}^{n \times n}$ is *positive semidefinite*, denoted as $X \succeq 0$, if the following equivalent conditions hold:

1. $x^T X x \geq 0$ for all $x \in \mathbb{R}^n$
2. All the eigenvalues of X are non-negative
3. There exist vectors $v_1, \dots, v_n \in \mathbb{R}^d$ for some $d > 0$ such that $X_{ij} = \langle v_i, v_j \rangle$ for all $i, j \in [n]$.

For $A, B \in \mathbb{R}^{n \times n}$, the *trace inner product* is defined as:

$$\langle A, B \rangle := \text{Tr}(A^T B) = \sum_{i,j=1}^n A_{ij} B_{ij}.$$

Given symmetric matrices $A_1, \dots, A_m \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^m$, a *semidefinite program (SDP)* in standard form is the following optimization problem:

$$p^* = \sup_X \{ \langle C, X \rangle : \langle A_k, X \rangle = b_k \ (k \in [m]), X \succeq 0 \}.$$

Each SDP of that form admits a dual SDP program:

$$d^* = \inf_y \left\{ b^T y : Y = \sum_{k=1}^m y_k A_k - C, Y \succeq 0 \right\}.$$

Weak duality holds, meaning that $p^* \leq d^*$. By Property 3 described above, in order to come-up with a feasible dual solution, it is enough to construct $y \in \mathbb{R}^m$, as well as vectors $v_1, \dots, v_n \in \mathbb{R}^d$ in some dimension $d > 0$ such that $Y_{ij} = (\sum_{k=1}^m y_k A_k - C)_{ij} = \langle v_i, v_j \rangle$ for every $i, j \in [n]$.

D Computation of the dual SDPs.

D.1 Taking the dual.

Recall that our primal semidefinite programming relaxation is the following.

$$\begin{aligned} & \min \langle C, X \rangle \\ & \sum_{i \in \mathcal{S}_j} X_{\{ij, ij\}} = 1 & \forall j \in N \\ & X_{\{0,0\}} = 1 \\ & X_{\{0, ij\}} = X_{\{ij, ij\}} & \forall j \in N, i \in \mathcal{S}_j \\ & X_{\{ij, i'k\}} \geq 0 & \forall (i, j), (i', k) \text{ with } j, k > 0. \\ & X \succeq 0 \end{aligned}$$

It can be easily checked that the following form of semidefinite programs is a primal-dual pair. The dual variables $(\lambda_i)_i$ and $(\mu_j)_j$ respectively correspond to the equality and inequality constraints, whereas the matrix variable Y corresponds to the semidefinite constraint.

$$\begin{array}{ll}
 \min \langle C, X \rangle & \max \sum_i b_i \lambda_i \\
 \langle A_i, X \rangle = b_i \quad \forall i & Y = C - \sum_i \lambda_i A_i - \sum_j \mu_j B_j \\
 \langle B_j, X \rangle \geq 0 \quad \forall j & Y \succeq 0, \quad \mu \geq 0. \\
 X \succeq 0 &
 \end{array}$$

Observe that our above primal SDP is in fact of that form. Let us denote by $(y_j)_{j \in N}, z$ and $(\sigma_{ij})_{j \in N, i \in \mathcal{S}_j}$ the dual variables respectively corresponding to the three sets of equality constraints. Let us denote by $\mu_{\{ij, i'k\}} \geq 0$ the dual variables corresponding to the inequality (or non-negativity) constraints. The dual objective then becomes $\sum_{j \in N} y_j + z$.

All the games considered will satisfy the fact that the objective matrix is all zeros in the first row and column: $C_{\{0,0\}} = 0$ and $C_{\{0,ij\}} = 0$ for every $j \in N$ and $i \in \mathcal{S}_j$. The dual matrix equality then becomes:

$$\begin{array}{ll}
 Y_{\{0,0\}} = -z & \\
 Y_{\{0, ij\}} = \frac{\sigma_{ij}}{2} & \forall j \in N, i \in \mathcal{S}_j \\
 Y_{\{ij, ij\}} = C_{\{ij, ij\}} - y_j - \sigma_{ij} - \mu_{\{i, \}} & \forall j \in N, i \in \mathcal{S}_j \\
 Y_{\{ij, i'k\}} = C_{\{ij, i'k\}} - \mu_{\{ij, i'k\}} & \forall (i, j) \neq (i', k) \text{ with } j, k > 0.
 \end{array}$$

Note that we can now eliminate the dual variables z and σ by the first two equalities. Moreover, we can eliminate the $\mu \geq 0$ variables by replacing the last two equalities by inequalities. Let us now do the change of variable $Y' = 2Y$ and let the vectors of the Cholesky decomposition of Y' be v_0 and $(v_{ij})_{j \in N, i \in \mathcal{S}_j}$, meaning that $Y'_{a, b} = \langle v_a, v_b \rangle$ holds for all the entries of Y' . The dual SDP in vector form can then be rewritten as:

$$\begin{array}{ll}
 \max \sum_{j \in N} y_j - \frac{1}{2} \|v_0\|^2 & \\
 y_j \leq C_{\{ij, ij\}} - \frac{1}{2} \|v_{ij}\|^2 - \langle v_0, v_{ij} \rangle & \forall j \in N, i \in \mathcal{S}_j \\
 \langle v_{ij}, v_{i'k} \rangle \leq 2 C_{\{ij, i'k\}} & \forall (i, j) \neq (i', k) \text{ with } j, k > 0.
 \end{array}$$

Clearly, we can also do a change of variable $v'_0 := -v_0$ to get the relaxation (3.2).

D.2 Specializing it to the different games considered. Let us now describe how the objective matrix C looks like for the different games that we need. Recall from Section 3.1 that we need to pick a symmetric matrix C such that $C(x) = \langle C, X \rangle = \text{Tr}(C^T X)$ where $X = (1, x)(1, x)^T$ is a binary rank one matrix and $C(x)$ is the social cost. By definition of the trace inner product, this is equivalent to:

$$C(x) = C_{\{0,0\}} + 2 \sum_{j \in N, i \in \mathcal{S}_j} C_{\{0,ij\}} x_{ij} + \sum_{\substack{j, k \in N \\ i \in \mathcal{S}_j, i' \in \mathcal{S}_k}} C_{\{ij, i'k\}} x_{ij} x_{i'k}.$$

Recall also that $x_{ij}^2 = x_{ij}$ since $x_{ij} \in \{0, 1\}$. Hence, if the social cost does not have constant terms, we will always be able to pick C such that $C_{\{0,0\}} = 0$ and $C_{\{0,ij\}} = 0$ for every $j \in N, i \in \mathcal{S}_j$, which we do for all the games below.

For the congestion game under the *Smith Rule* policy, the social cost in (4.2) can be written as:

$$C(x) = \sum_{j \in N} w_j C_j(x) = \sum_{\substack{j \in N \\ i \in \mathcal{S}_j \\ e \in i}} w_j p_{ej} x_{ij} + \frac{1}{2} \sum_{\substack{j \in N, k \neq j \\ i \in \mathcal{S}_j, i' \in \mathcal{S}_k \\ e \in i \cap i'}} w_j w_k \min\{\delta_{ej}, \delta_{ek}\} x_{ij} x_{i'k}.$$

Therefore, the objective matrix C is the following:

$$C_{\{ij, ij\}} = \sum_{e \in i} w_j p_{ej}, \quad C_{\{ij, i'k\}} = \frac{1}{2} \sum_{e \in i \cap i'} w_j w_k \min\{\delta_{ej}, \delta_{ek}\}.$$

If one considers the scheduling problem $R||\sum w_j C_j$ under *Smith's Rule*, which is a special case of the previous setting, then

$$C_{\{ij, ij\}} = w_j p_{ij} \quad , \quad C_{\{ij, i'k\}} = \frac{1}{2} w_j w_k \min\{\delta_{ij}, \delta_{ik}\} \mathbf{1}_{\{i=i'\}}.$$

For the congestion game under the *Rand* policy, the social cost in (4.11) gives

$$C_{\{ij, ij\}} = \sum_{e \in i} w_j p_{ej} \quad , \quad C_{\{ij, i'k\}} = \sum_{e \in i \cap i'} w_j w_k \frac{\delta_{ej} \delta_{ek}}{\delta_{ej} + \delta_{ek}}.$$

For the weighted affine congestion game, we have seen that

$$C(x) := \sum_{j \in N} C_j(x) = \sum_{e \in E} a_e \ell_e(x)^2 + b_e \ell_e(x)$$

where $\ell_e(x) = \sum_{j \in N} w_{ej} \sum_{i \in \mathcal{S}_j: e \in i} x_{ij}$. The objective matrix in that case is

$$C_{\{ij, ij\}} = \sum_{e \in i} w_{ej} (a_e w_{ej} + b_e) \quad , \quad C_{\{ij, i'k\}} = \sum_{e \in i \cap i'} a_e w_{ej} w_{ek}.$$

E Robust price of anarchy. In this section, we describe how our proofs can be adapted to give bounds on the *coarse-correlated price of anarchy*, meaning that we can now generalize our results by considering *coarse-correlated equilibria*, instead of *mixed* (or *pure*) Nash equilibria.

Let N be a game with a strategy set \mathcal{S}_j and payoff function C_j for every player $j \in N$. A distribution σ over $\mathcal{S}_1 \times \dots \times \mathcal{S}_n$ is a coarse correlated equilibrium if

$$(E.1) \quad \mathbb{E}_{X \sim \sigma}[C_j(X)] \leq \mathbb{E}_{X \sim \sigma}[C_j(X_{-j}, i)] \quad \forall j \in N, i \in \mathcal{S}_j.$$

Note that this generalizes a mixed Nash equilibrium. In that case, σ is a product distribution, i.e. every player j picks a random strategy independently from its own distribution, which we denoted by $(x_{ij})_{i \in \mathcal{S}_j}$ previously in the paper. We note that our formulas for $C_j(x)$ - see for instance (4.1) - for non-binary x (i.e. interpreting x as a collection of probability distributions rather than an integer assignment) implicitly use this independence assumption, meaning that the current proofs do not directly go through for coarse-correlated equilibria. Let us first rewrite *(SDP-C)* (3.2) in a more convenient matrix form for this argument.

$$\begin{aligned} \max \sum_{j \in N} \varphi_j - \frac{1}{2} Y_{\{0,0\}} \\ \varphi_j \leq C_{\{ij, ij\}} - \frac{1}{2} Y_{\{ij, ij\}} + Y_{\{0, ij\}} & \quad \forall j \in N, i \in \mathcal{S}_j \\ Y_{\{ij, i'k\}} \leq 2 C_{\{ij, i'k\}} & \quad \forall (i, j) \neq (i', k) \text{ with } j, k \in N \\ Y \succeq 0 \end{aligned}$$

One way to generalize our results is to consider random dual *(SDP-C)* solutions, i.e. doing a dual fitting on a realization $X \sim \sigma$, which induces binary random variables $\{X_{ij}\}_{j \in N, i \in \mathcal{S}_j}$ and $\{Z_{ej}\}_{j \in N, e \in E}$. For any price of anarchy dual fitting argument in this paper, first replace every occurrence of respectively x_{ij} and z_{ej} by X_{ij} and Z_{ej} , in which case v_0 and every y_j become random variables (note that every v_{ij} is always deterministic). To get a feasible dual solution, we now set $Y_{a,b} := \mathbb{E}_{X \sim \sigma}[\langle v_a, v_b \rangle]$ for every indices a, b as well as $\varphi_j := \mathbb{E}_{X \sim \sigma}[\langle y_j \rangle]$.

The second set of constraints of *(SDP-C)* is always satisfied deterministically in our fittings, while the first set of constraints is satisfied by considering expectations, due to inequality (E.1). Moreover, Y is positive semidefinite since it is a convex combination of positive semidefinite matrices.

We thus get a feasible solution with objective value V satisfying $V \geq \rho \mathbb{E}_{X \sim \sigma}[C(X)]$ for some desired bound $\rho \in [0, 1]$. Since the dual solution is feasible, we have $V \leq C(x^*)$, where x^* is the social optimum. Combining these two equations gives:

$$\mathbb{E}_{X \sim \sigma}[C(X)] \leq \frac{1}{\rho} C(x^*)$$

hence yielding a bound on the coarse-correlated price of anarchy.