

# Implementation of Optimal Schedules in Outsourcing with Identical Suppliers\*

Herbert Hamers<sup>†</sup>      Flip Klijn<sup>‡</sup>      Marco Slikker<sup>§</sup>

16 July 2018

## Abstract

This paper deals with decentralized decision-making situations in which firms outsource production orders to multiple identical suppliers. Each firm aims to minimize the sum of its completion times. We study whether a central authority can install a mechanism such that strategic interaction leads to a socially optimal schedule. For the case of single demand the shortest-first mechanism implements optimal schedules in Nash equilibrium. We show that for the general case there exists no anonymous mechanism that implements optimal schedules in correlated equilibrium.

*Keywords:* game theory, outsourcing, scheduling, efficiency, implementation, Nash equilibrium, correlated equilibrium, price of stability

*JEL-Numbers:* C72, D71, D82.

## 1 Introduction

In the last decade outsourcing has become increasingly important since advanced products are typically no longer completely built in-house. Hence, apart from managing their own production facilities, companies have an increasing need to tightly control outsourced operations. In this paper, we consider decentralized decision-making situations in which

---

\*We thank an associate editor and two anonymous reviewers for their comments and suggestions.

<sup>†</sup>CentER and Department of Econometrics and Operations Research, Tilburg University.

<sup>‡</sup>Corresponding author. Institute for Economic Analysis (CSIC), Barcelona GSE, and CentER; e-mail: [flip.klijn@iae.csic.es](mailto:flip.klijn@iae.csic.es). The first draft of the paper was written while F. Klijn was visiting CentER and the Department of Econometrics and Operations Research, Tilburg University. He gratefully acknowledges the hospitality of Tilburg University and an extramural fellowship from CentER. Financial support from AGAUR-Generalitat de Catalunya (2014-SGR-1064 and 2017-SGR-1359), the Spanish Ministry of Economy and Competitiveness through Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016 (ECO2014-59302-P and ECO2017-88130-P), and the Severo Ochoa Programme for Centres of Excellence in R&D (SEV-2015-0563) is also gratefully acknowledged.

<sup>§</sup>Industrial Engineering and Innovation Sciences, Eindhoven University of Technology.

companies or departments outsource production orders or jobs to multiple identical suppliers. We assume that each company can only freely decide to which supplier it outsources each of its jobs. In particular, outsourcing companies cannot decide on the order in which each supplier processes the assigned jobs. This assumption reflects the fact that most transactions in supply chains are governed by price-only contracts. A drawback of these contracts is that they do not coordinate the supply chain. That is, locally optimal decisions may lead to a joint outcome that can be improved upon (see, e.g., Perakis and Roels, 2007).

We assume that each outsourcing company is selfishly interested in minimizing the sum of completion times of its jobs. This may reflect for instance the fact that different jobs of the same outsourcing company are part of different final products, all of which the company aims to produce and sell as soon as possible. In our setting, individual agents interact to make decisions that affect them collectively. This is a common feature of many economic situations. Implementation theory is often used to design mechanisms such that the equilibrium outcomes satisfy a criterion of social optimality.<sup>1</sup> The question we are interested in is whether a central authority can install a mechanism such that strategic interaction leads to a socially optimal schedule. Here, a mechanism is a function that maps any collection of outsourcing decisions to a schedule. We assume that a mechanism respects the agents' decisions. In other words, if a job is sent to a particular machine by its owner, then the mechanism should process the job on that machine. We exclude from our analysis mechanisms that involve idle time. Finally, we take a normative approach by imposing two anonymity properties: the output of a mechanism should not depend on the names of the owners of the jobs nor on the names of the machines.

The solution concepts we focus on are Nash equilibria and correlated equilibria. Depending on the specification of the mechanism, a Nash equilibrium in pure strategies may not exist. We therefore define implementation of optimal schedules in correlated equilibrium. More precisely, a mechanism implements optimal schedules if for each outsourcing problem there exists some correlated equilibrium such that the schedules induced by its support are optimal. We find that the (anonymous) mechanism based on shortest-first policy (or Smith's (1956) rule) implements (in pure Nash equilibrium) optimal schedules provided that each agent has one job (Theorem 1). However, it is not possible to extend implementation to the case where some agent has more than one job. More precisely, we show that in the general case there exists no anonymous mechanism that implements optimal schedules in correlated equilibrium (Theorem 2). Hence, there exists no anonymous mechanism that implements optimal schedules in Nash equilibrium in mixed strategies.

Usually, given a social choice rule, if a mechanism has the property that, in each possible state of the world, the set of equilibrium outcomes coincides with the set of optimal outcomes identified by the social choice rule, then the social choice rule is said to be implemented by the mechanism (see, e.g., Maskin, 1985). We take a different approach by demanding the existence of some Nash or correlated equilibrium that guarantees optimal outcome(s). Since our definition is less demanding than the standard definition of

---

<sup>1</sup>Jackson (2001) reviews some of the fundamental results in implementation theory.

implementation, our impossibility result also holds for the standard definition of implementation. Our impossibility result implies that anonymity, even though often thought of as a weak desideratum, is incompatible with the implementation of optimal schedules. One possible interpretation is that it is a negative finding that may help the central authority in a reorientation towards other desiderata. Impossibility results exist for a wide variety of economic and social environments. For instance, Arrow’s (1950) famous result shows the impossibility of satisfactory electoral systems. Myerson and Satterthwaite’s (1983) classical implementation result for bilateral trading establishes the impossibility of efficient mechanisms without outside subsidies.

In order to appropriately discuss the rapidly growing literature that studies coordinating mechanisms for supply chains and competitive scheduling environments we note that our definition of implementation can also be phrased in terms of the price of stability.<sup>2</sup> Fixing a mechanism and an evaluation/objective criterion, the price of stability is the ratio between the value of the *best* Nash equilibrium and the optimal value. Then, a mechanism implements optimal schedules if for each outsourcing problem the price of stability equals one.

There are two classes of closely related papers that highlight the relevance of our approach.<sup>3</sup> First, there is a literature that similarly to our study focuses on the best Nash equilibria to determine the efficiency loss with respect to the social choice rule (defined by the optimal schedules). Bukchin and Hanany (2007) studied a two machine dispatching–sequencing model where each decision–maker owns a set of jobs and can choose between processing on an in–house machine or outsource to a slower machine. Each decision–maker aims to minimize the sum of completion times of its jobs (minsum objective). Bukchin and Hanany (2007) were the first to consider mixed–strategy Nash equilibria in the literature on scheduling and supply chains. They provided a scheduling–based mechanism to ensure that the centralized optimum is obtained at some equilibrium. Bukchin and Hanany (2011) considered a decentralized job shop scheduling system with two machines. The two machines are agents each of which has to decide in which order to process the jobs. Bukchin and Hanany (2011) proposed a mechanism based on penalization to ensure that a schedule with smallest total flowtime is obtained at some equilibrium. Agussurja and Lau (2009) studied the makespan for scheduling games with a common policy for all machines. They obtained bounds on the price of stability.

Second, there is a scheduling literature that focuses on the *worst* Nash equilibria to determine the efficiency loss with respect to the social choice rule (again defined by the optimal schedules). The ratio of the value of the *worst* Nash equilibrium and the optimal value is referred to as the price of anarchy.<sup>4</sup> Here we mention some of the recent

---

<sup>2</sup>The price of stability was first studied in Schulzan and Stier Moses (2003). The name was coined by Anshelevich *et al.* (2008).

<sup>3</sup>We refer to Li and Wang (2007) for a general review of coordination mechanisms for supply chain systems. Implementation theory and more generally game theory have become essential tools in the analysis of supply chains with multiple agents. Cachon and Netessine (2004) survey the applications of game theory to supply chain analysis. Li and Whang (2001) survey game theory models in operations management.

<sup>4</sup>The price of anarchy was first studied in Dubey (1986) and Koutsoupias and Papadimitriou (2009).

and mostly related works. Immorlica *et al.* (2009) studied the makespan for four prominent coordination mechanisms in four classes of multiple machine scheduling situations. They provided upper and lower bounds on the price of anarchy. Also focusing on the makespan, Caragiannis (2013) introduced three new coordination mechanisms and provided bounds on the price of anarchy. Cole *et al.* (2015) and Correa and Queyranne (2012) studied coordination mechanisms with the aim to minimize the weighted sum of completion times of jobs. In particular, they obtained an upper bound on the price of anarchy of Smith’s (1956) rule for the case of unrelated machines and restricted related machines, respectively. Abed *et al.* (2014) extended Smith’s (1956) rule by adding extra delays on the jobs accounting for the negative externality they impose on other agents. Using a weaker equilibrium concept than the Nash equilibrium, they provided a bound on the price of anarchy and showed that this ratio is best possible even if one allows for randomization or full information. Braat *et al.* (2014) and Hoeksma and Uetz (2012) studied the related machine scheduling problem with minsum objective. They provided bounds on the price of anarchy of Smith’s (1956) rule. Other related recent papers that study the weighted sum of completion times of jobs and price of anarchy include Angel *et al.* (2016), Cohen and Pascual (2015), and Lee *et al.* (2012). Finally, complementing the above mentioned literature, Aumann and Dombb (2010) and Epstein and Kleiman (2011) studied the Pareto–efficiency of the Nash equilibria of job scheduling games.

The remainder of the paper is organized as follows. In Section 2 we introduce the scheduling model and game. In Section 3 we state and prove our results. Section 4 concludes.

## 2 Preliminaries

### 2.1 Scheduling model

Let  $M$ ,  $|M| \geq 2$ , be the finite set of (identical) contract manufacturing companies, which we refer to as machines. Let  $N$  be the finite set of outsourcing companies, which we refer to as agents. Let  $J_i$  be the set of production orders, or shortly jobs, owned by agent  $i$ . Each job is owned by exactly one agent, i.e., for all  $i, i' \in N$  with  $i \neq i'$ ,  $J_i \cap J_{i'} = \emptyset$ . Let  $J = \cup_i J_i$  denote the set of jobs. For  $j \in J$ , let  $i(j)$  denote the agent that owns job  $j$ , i.e.,  $j \in J_{i(j)}$ . We assume each job has non-preemptive processing requirements, i.e., once it is started it cannot be interrupted. Each job  $j \in J$  has processing time  $p_j > 0$ . To avoid degenerate situations that require cumbersome notation we assume that for all  $j, j' \in J$  with  $j \neq j'$ ,  $p_j \neq p_{j'}$ . A (scheduling) problem is a quadruple  $\Lambda = (M, N, (J_i)_{i \in N}, (p_j)_{j \in J})$ .

A schedule is a sequenced assignment of the jobs to the machines. Formally, a schedule is a function  $\sigma : J \rightarrow M \times \{1, \dots, |J|\}$ , where  $\sigma(j) = (\sigma_1(j), \sigma_2(j)) = (m, k)$  indicates that job  $j$  is scheduled in position  $k$  of machine  $m$ . We assume that on each machine there is no idle time between jobs nor before the first job. Given a schedule  $\sigma$ , job  $j$ ’s predecessors are the jobs  $P(\sigma, j) = \{j' \in J : \sigma_1(j') = \sigma_1(j) \text{ and } \sigma_2(j') < \sigma_2(j)\}$ . Job  $j$ ’s completion

---

The name was coined by Papadimitriou (2001).

time is the sum of its processing time and the waiting time due to its predecessors, i.e.,

$$C_j(\sigma) = p_j + \left( \sum_{j' \in P(\sigma, j)} p_{j'} \right).$$

The central objective is to minimize the sum of completion times. Hence, a schedule  $\sigma^*$  is optimal if for any other schedule  $\sigma$ ,

$$\sum_{j \in J} C_j(\sigma^*) \leq \sum_{j \in J} C_j(\sigma).$$

Let  $\Sigma^*(\Lambda)$  denote the set of optimal schedules for  $\Lambda$ . Horowitz and Sahni (1976) introduced an algorithm to compute all optimal schedules in polynomial time. In Step I of their MFT algorithm, jobs are allocated one by one to machines. Each time the longest non-allocated job is allocated to a machine with smallest number of allocated jobs.

**Minimum Mean Flow Time<sup>5</sup> (MFT) algorithm.** (Horowitz and Sahni, 1976)

**Step I.** For each machine  $m$ , set  $l_m \equiv 0$ . Set  $J^* \equiv J$ . As long as  $J^* \neq \emptyset$ , do Procedure.

BEGIN PROCEDURE.

Let  $j^* \in J^*$  be such that  $p_{j^*} > p_j$  for all  $j \in J^*$ ,  $j \neq j^*$ . Let  $m \in M$  be a machine with lowest  $l_m$ .

Set  $a(j^*) \equiv m$  and update  $l_m \equiv l_m + 1$  as well as  $J^* \equiv J^* \setminus \{j^*\}$ .

END PROCEDURE.

**Step II.** Let  $\sigma^*$  be the schedule obtained by placing on machine  $m$  all jobs in  $a^{-1}(m)$  in order of increasing processing times.  $\diamond$

As will be mentioned in Remark 1, the following SFG algorithm generates optimal schedules. In Step I, jobs are allocated one by one to machines. Each time the shortest non-allocated job is allocated to a machine with smallest sum of processing times of allocated jobs.

**Shortest-First Greedy (SFG) algorithm.** (Ibarra and Kim, 1977)

**Step I.** For each machine  $m$ , set  $C_m \equiv 0$ . Set  $\bar{J} \equiv J$ . As long as  $\bar{J} \neq \emptyset$ , do Procedure.

BEGIN PROCEDURE.

Let  $\bar{j} \in \bar{J}$  be such that  $p_{\bar{j}} < p_j$  for all  $j \in \bar{J}$ ,  $j \neq \bar{j}$ . Let  $m \in M$  be a machine with lowest  $C_m$ .

Set  $a(\bar{j}) \equiv m$  and update  $C_m \equiv C_m + p_{\bar{j}}$  as well as  $\bar{J} \equiv \bar{J} \setminus \{\bar{j}\}$ .

END PROCEDURE.

**Step II.** Let  $\bar{\sigma}$  be the schedule obtained by placing on machine  $m$  all jobs in  $a^{-1}(m)$  in order of increasing processing times.  $\diamond$

In Example 1 we illustrate the shortest-first greedy algorithm.

---

<sup>5</sup>Minimum mean flow time and minimum sum of completion times are equivalent objectives.

**Example 1. (Illustration of the SFG algorithm.)** Let  $M = \{m_1, m_2, m_3\}$  and  $J = \{1, 2, \dots, 8\}$  with  $p_1 < p_2 < \dots < p_8$ . Then, one of the schedules  $\bar{\sigma}$  that can be obtained by the shortest–first greedy algorithm is depicted in Figure 1.  $\diamond$

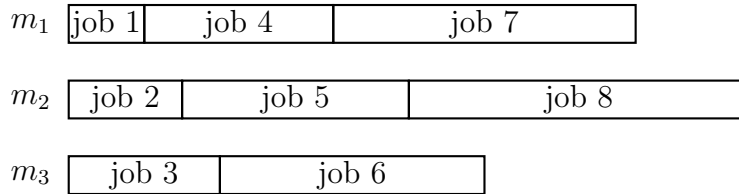


Figure 1: Output of the SFG algorithm in Example 1

The following remark will turn out to be useful in Section 3.

**Remark 1. (Optimality of the SFG algorithm.)**

Let  $\bar{\sigma}$  be a schedule obtained by the shortest–first greedy algorithm. Suppose without loss of generality that  $p_1 < p_2 < \dots < p_{|J|}$  and that  $\bar{\sigma}$  processes jobs  $1, \dots, |M|$  on machines  $m_1, \dots, m_{|M|}$ , respectively. Then,  $\bar{\sigma}$  processes job  $k$  on machine  $m_l$  where  $l = 1 + (k - 1) \bmod |M|$ . (See Example 1 for an illustration.) It can be easily verified that the  $|M|$  jobs with largest processing times are processed by  $\bar{\sigma}$  on different machines, the next  $|M|$  jobs with largest processing times are processed by  $\bar{\sigma}$  on different machines as well, etc. Hence,  $\bar{\sigma}$  can also be obtained by the MFT algorithm of Horowitz and Sahni (1976). So,  $\bar{\sigma}$  is optimal.<sup>6</sup>  $\diamond$

## 2.2 Scheduling game

Let  $\Lambda = (M, N, (J_i)_{i \in N}, (p_j)_{j \in J})$  be a scheduling problem. For any  $i \in N$ , we let  $\Pi_i^\Lambda$  denote the collection of functions  $\pi_i : J_i \rightarrow M$ . Here,  $\pi_i(j)$  indicates the machine to which player  $i$  allocates/sends his job  $j \in J_i$ . Hence, we will refer to  $\pi_i$  as an *allocation of player  $i$ 's jobs  $J_i$* . Let  $\Pi^\Lambda = (\Pi_i^\Lambda)_{i \in N}$ . Whenever no confusion is possible we omit the supindex  $\Lambda$  from  $\Pi_i^\Lambda$  and  $\Pi^\Lambda$ .

Let  $\mathcal{L}$  be a class of scheduling problems. A mechanism is a map  $\varphi$  that for each pair  $(\Lambda, (\pi_i)_{i \in N})$  with  $\Lambda \in \mathcal{L}$  and  $(\pi_i)_{i \in N} \in \Pi$  results in a schedule  $\varphi(\Lambda, (\pi_i)_{i \in N})$  for  $\Lambda$ . Let  $\varphi$  be a mechanism. Let  $\Lambda = (M, N, (J_i)_{i \in N}, (p_j)_{j \in J}) \in \mathcal{L}$ . We associate with  $\Lambda$  and  $\varphi$  a non-cooperative scheduling game of complete information  $\Gamma(\Lambda, \varphi) = (N, (\Pi_i)_{i \in N}, (c_i)_{i \in N})$ , which is explained next. The set of players is  $N$ . For each  $i \in N$ , the set of (pure) strategies of player  $i$  is given by the set  $\Pi_i$ . Each strategy–profile  $\pi \in \Pi$  induces a schedule  $\varphi(\Lambda, \pi) = (\varphi_1(\Lambda, \pi), \varphi_2(\Lambda, \pi))$ . In other words, for any  $i \in N$  and any  $j \in J_i$ , job

<sup>6</sup>Not all optimal schedules can be obtained by the SFG algorithm. For instance, let  $M = \{m_1, m_2\}$  and  $J = \{1, 2, 3\}$  with  $p_1 < p_2 < p_3$ . Consider the schedule that processes jobs 1 and 2 (in this order) on  $m_1$  and job 3 on  $m_2$ . The schedule is optimal (since it can be obtained by the MFT algorithm) but it cannot be obtained by the SFG algorithm (since jobs 1 and 2 are processed on the same machine).

$j$  is scheduled in position  $\varphi_{2j}(\Lambda, \pi)$  on machine  $\varphi_{1j}(\Lambda, \pi)$ . We will *assume throughout that mechanism  $\varphi$  respects the players' allocations*, i.e., for any job  $j \in J$ ,  $\varphi_{1j}(\Lambda, \pi) = \pi_{i(j)}(j)$ . Player  $i$ 's resulting "costs" are given by the sum of completion times of his jobs in  $\varphi(\Lambda, \pi)$ . In other words, player  $i$ 's cost function  $c_i$  is given by

$$c_i(\pi) = \sum_{j \in J_i} C_j(\varphi(\Lambda, \pi)), \text{ where } \pi \in \Pi.$$

**Remark 2. (No idle time.)** According to our definition, a mechanism yields schedules. Hence, we exclude from our analysis functions that yield sequenced assignments of jobs to machines that involve idle time between jobs or before the first job.  $\diamond$

Next, we give an example of a mechanism.

**Example 2. (Shortest–first mechanism  $\bar{\varphi}$ .)** Let  $\bar{\varphi}$  be the mechanism that for each scheduling problem and each allocation–profile yields the schedule obtained by putting on each machine the allocated jobs in order of increasing processing times (i.e., shortest–first policy or Smith's (1956) rule). Formally, for any scheduling problem  $\Lambda$ , any allocation–profile  $\pi \in \Pi$ , and any jobs  $j, j'$  such that  $\pi_{i(j)}(j) = \pi_{i(j')}(j')$ ,

- $\bar{\varphi}_{1j}(\Lambda, \pi) = \pi_{i(j)}(j)$ ;
- $\bar{\varphi}_{2j}(\Lambda, \pi) < \bar{\varphi}_{2j'}(\Lambda, \pi)$  if and only if  $p_j < p_{j'}$ .  $\diamond$

We complement Example 2 by showing how the shortest–first mechanism induces a scheduling game.

**Example 3. (A scheduling game induced by  $\bar{\varphi}$ .)**

Let  $\Lambda$  be the scheduling problem with  $M = \{m_1, m_2\}$ ,  $N = \{1, 2\}$ ,  $J_1 = \{\alpha, \gamma\}$ ,  $J_2 = \{\beta, \delta\}$  and  $(p_\alpha, p_\beta, p_\gamma, p_\delta) = (1, 2, 3, 4)$ . Consider the game  $\Gamma(\Lambda, \bar{\varphi})$ . Each player has 4 pure strategies regarding his 2 jobs: he can send both jobs to machine  $m_1$ , both jobs to machine  $m_2$ , or different jobs to different machines (two ways). Table 1 concisely depicts the scheduling game. Player 1 is the row player and each row indicates which jobs are sent to machine  $m_1$  (the complement is sent to machine  $m_2$ ). For instance,  $\{\alpha\}$  corresponds with player 1's strategy  $\pi_1$  with  $\pi_1(\alpha) = m_1$  and  $\pi_1(\gamma) = m_2$ . Similarly, player 2 is the column player and each column indicates which jobs are sent to machine  $m_1$ .

1\2	$\emptyset$	$\{\beta\}$	$\{\delta\}$	$\{\beta, \delta\}$
$\emptyset$	7,13	5,10	7, <b>7</b>	5,8
$\{\alpha\}$	6,11	<b>4</b> ,10	6, <b>7</b>	<b>4</b> ,10
$\{\gamma\}$	<b>4</b> ,10	6, <b>7</b>	<b>4</b> ,10	6,11
$\{\alpha, \gamma\}$	5,8	7, <b>7</b>	5,10	7,13

Table 1: Table of Example 3

Next, we illustrate that each pair of numbers indicates the costs induced by the corresponding strategy–profile. Consider, for instance, the pair  $(\{\alpha\}, \{\beta, \delta\})$ , which corresponds with profile  $\pi = (\pi_i)_{i=1,2}$  such that  $\pi_1(\alpha) = \pi_2(\beta) = \pi_2(\delta) = 1$  and  $\pi_1(\gamma) = 2$ .

Mechanism  $\bar{\varphi}$  from Example 2 applied to problem  $\Lambda$  and strategy-profile  $\pi$  places the jobs allocated to each machine in order of increasing processing times. Hence,  $\bar{\varphi}(\Lambda, \pi)$  is the schedule defined by  $\bar{\varphi}_\alpha(\Lambda, \pi) = (1, 1)$ ,  $\bar{\varphi}_\beta(\Lambda, \pi) = (1, 2)$ ,  $\bar{\varphi}_\delta(\Lambda, \pi) = (1, 3)$ , and  $\bar{\varphi}_\gamma(\Lambda, \pi) = (2, 1)$ . Recall that the first coordinate indicates the machine and the second coordinate indicates the position at that machine. So, machine  $m_1$  processes first  $\alpha$ , then  $\beta$ , and finally  $\delta$ ; machine  $m_2$  only processes job  $\gamma$ . Then, player 1's costs equal the sum of the completion times of his jobs  $\alpha$  and  $\gamma$ :  $C_\alpha(\bar{\varphi}(\Lambda, \pi)) + C_\gamma(\bar{\varphi}(\Lambda, \pi)) = 1 + 3 = 4$ . Similarly, player 2's costs equal the sum of the completion times of his jobs  $\beta$  and  $\delta$ :  $C_\beta(\bar{\varphi}(\Lambda, \pi)) + C_\delta(\bar{\varphi}(\Lambda, \pi)) = (1 + 2) + (1 + 2 + 4) = 10$ . Hence, in this case the costs of the two players are given by  $(4, 10)$ .

Finally, the boldfaced numbers in Table 1 indicate the best responses. For instance, if player 2 plays  $\{\beta\}$ , then the best response of player 1 is  $\{\alpha\}$ . One immediately verifies that there is no strategy-profile in which each player gives a best response to the other player's strategy. Hence, the game has no Nash equilibrium in pure strategies.  $\diamond$

The scheduling game in Example 3 has no Nash equilibrium in pure strategies. For this reason, it is natural to consider Nash equilibrium in mixed strategies.<sup>7</sup> But since our main result is an impossibility result and Nash equilibria in mixed strategies are particular correlated equilibria we will focus on correlated equilibria to obtain a stronger result. Let  $\Gamma(\Lambda, \varphi) = (N, (\Pi_i)_{i \in N}, (c_i)_{i \in N})$  be a scheduling game. A correlated equilibrium of  $\Gamma(\Lambda, \varphi)$  is a quadruple  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$  where

- $(\Omega, Pr)$  is a finite probability space, where w.l.o.g. for all  $\omega \in \Omega$ ,  $Pr(\omega) > 0$ ;
- for each player  $i \in N$ ,  $\mathcal{P}_i$  is a partition of  $\Omega$ ;
- for each player  $i \in N$ ,  $s_i$  is a function  $s_i : \Omega \rightarrow \Pi_i$  that is  $\mathcal{P}_i$ -measurable, i.e.,  $s_i(\omega) = s_i(\omega')$  whenever  $\omega, \omega' \in P_i$  for some  $P_i \in \mathcal{P}_i$

such that for all  $i \in N$  and all  $\mathcal{P}_i$ -measurable functions  $t_i : \Omega \rightarrow \Pi_i$  we have

$$\sum_{\omega \in \Omega} Pr(\omega) c_i([s_k(\omega)]_{k \in N}) \leq \sum_{\omega \in \Omega} Pr(\omega) c_i(t_i(\omega), [s_k(\omega)]_{k \in N \setminus \{i\}}). \quad (1)$$

Let  $\mathcal{E}(\Gamma(\Lambda, \varphi))$  be the set of correlated equilibria of game  $\Gamma(\Lambda, \varphi)$ . For any correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$ , let  $\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$  be the set of schedules obtained with positive probability, i.e.,

$$\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) = \{\varphi(\Lambda, [s_i(\omega)]_{i \in N}) : \omega \in \Omega\}.$$

It is well-known that for any Nash equilibrium in pure strategies, there is a correlated equilibrium that yields the same schedule. Similarly, for any Nash equilibrium in mixed strategies, there is a correlated equilibrium that yields the same distribution of schedules. Finally, for any correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$  and any subset  $\Omega' \subseteq \Omega$ , we denote  $Pr(\Omega') = \sum_{\omega \in \Omega'} Pr(\omega)$ .

---

<sup>7</sup>We omit the formal definition of Nash equilibrium in mixed strategies as it can be easily found in any standard text book on game theory.



### 3 Results

The question we are interested in is whether it is possible to design a mechanism such that strategic interaction leads to optimal schedules. Let  $\mathcal{L}$  be a class of scheduling problems. A mechanism  $\varphi$  **implements optimal schedules** (in correlated equilibrium) for  $\mathcal{L}$  if for each scheduling problem  $\Lambda \in \mathcal{L}$ , there exists some correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \in \mathcal{E}(\Gamma(\Lambda, \varphi))$  such that

$$\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \subseteq \Sigma^*(\Lambda). \quad (2)$$

**Remark 3. (Equivalent definition of implementation.)** Equivalently, a mechanism  $\varphi$  implements optimal schedules for a class of scheduling problems  $\mathcal{L}$  if for each scheduling problem  $\Lambda \in \mathcal{L}$ , there exists some correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \in \mathcal{E}(\Gamma(\Lambda, \varphi))$  such that for some (or equivalently, all)  $\sigma^* \in \Sigma^*(\Lambda)$  and for all  $\omega \in \Omega$ ,

$$\sum_{j \in J} C_j(\varphi(\Lambda, [s_i(\omega)]_{i \in N})) = \sum_{j \in J} C_j(\sigma^*). \quad \diamond$$

Next, we introduce two anonymity properties for mechanisms. First, since in any scheduling problem all machines are identical, it is natural to focus on mechanisms that leave the schedule “essentially” unchanged if only the names of the machines are permuted, i.e., mechanisms should satisfy “machine-anonymity.” Let  $\varphi$  be a mechanism for a class  $\mathcal{L}$ . Let  $\Lambda = (M, N, (J_i)_{i \in N}, (p_j)_{j \in J}) \in \mathcal{L}$  and consider the game  $\Gamma(\Lambda, \varphi)$ . Let  $\pi'$  be a strategy-profile that is obtained from some other strategy-profile  $\pi$  by permuting complete batches of jobs according to a permutation  $\rho : M \rightarrow M$ , i.e.,  $\pi' = \rho \circ \pi$  where  $\circ$  denotes composition. Then, since machines are identical, each job should be processed in the same position by  $\varphi(\Lambda, \pi)$  and  $\varphi(\Lambda, \pi')$  (but possibly on different machines). Formally, mechanism  $\varphi$  satisfies **machine-anonymity** on  $\mathcal{L}$  if for any  $\Lambda \in \mathcal{L}$ , any  $\pi \in \Pi$ , any permutation  $\rho : M \rightarrow M$ , and any  $j \in J$ ,

$$\varphi_{2j}(\Lambda, \rho \circ \pi) = \varphi_{2j}(\Lambda, \pi). \quad (3)$$

Second, we require that mechanisms do not yield a different schedule if only the ownership of the jobs is changed, i.e., mechanisms are to satisfy “owner-anonymity.” In other words, a mechanism is owner-anonymous if at two scheduling problems that only differ in ownership the same schedule is obtained when for each job its possibly distinct owners at the two problems allocate it to the same machine. Formally, a mechanism  $\varphi$  for a class  $\mathcal{L}$  satisfies **owner-anonymity** on  $\mathcal{L}$  if for any  $\Lambda' = (M', N', (J'_i)_{i \in N'}, (p'_j)_{j \in J'}) \in \mathcal{L}$  and any  $\Lambda^* = (M^*, N^*, (J^*_i)_{i \in N^*}, (p^*_j)_{j \in J^*}) \in \mathcal{L}$  with  $M' = M^*$ ,  $J' = J^*$  and for all  $j \in J'$ ,  $p'_j = p^*_j$ , it holds that  $\varphi(\Lambda', \pi') = \varphi(\Lambda^*, \pi^*)$  for any strategy-profile  $\pi'$  in  $\Gamma(\Lambda', \varphi)$ , and any strategy-profile  $\pi^*$  in  $\Gamma(\Lambda^*, \varphi)$  such that for all  $j \in J'$ ,  $\pi'_{i'(j)}(j) = \pi^*_{i^*(j)}(j)$ . Here,  $i'(j)$  and  $i^*(j)$  denote the owner of job  $j$  in problems  $\Lambda'$  and  $\Lambda^*$ , respectively.

Let  $\mathcal{L}^1$  be the class of single demand problems, i.e., each player has one job. Formally, let  $\mathcal{L}^1$  be the class of scheduling problems  $(M, N, (J_i)_{i \in N}, (p_j)_{j \in J})$  with  $|J_i| = 1$  for all  $i \in N$ .

$N$ . The next result shows that the shortest–first mechanism satisfies the two anonymity properties and implements optimal schedules for single demand problems and strongly builds upon Theorem 1 in Immorlica *et al.* (2009).

**Theorem 1. (Possibility of implementation for single demand.)**

*Shortest–first mechanism  $\bar{\varphi}$  satisfies machine–anonymity and owner–anonymity, and moreover implements optimal schedules in (pure strategy) Nash equilibrium for  $\mathcal{L}^1$ .*

*Proof.* Since  $\bar{\varphi}$  does not discriminate between machines, it follows that it satisfies machine–anonymity. Moreover, since  $\bar{\varphi}$  orders jobs based on their processing times (i.e., independently of the names of their owners), it satisfies owner–anonymity as well.

It remains to show that  $\bar{\varphi}$  implements optimal schedules in Nash equilibrium for  $\mathcal{L}^1$ . Let  $\Lambda \in \mathcal{L}^1$ . Let  $\bar{\sigma}$  be a schedule obtained by the SFG algorithm. Immorlica *et al.* (2009, Theorem 1) showed that  $\bar{\sigma}$  is the (unique) schedule induced by some Nash equilibrium  $\tilde{\pi}$  in pure strategies of the game  $\Gamma(\Lambda, \bar{\varphi})$ . By Remark 1,  $\bar{\sigma}$  is an optimal schedule. Hence, inclusion (2) holds.  $\square$

Next, we show that Theorem 1 cannot be extended to the case where players can have multiple jobs. In fact, we obtain an impossibility result, even if we relax the solution concept from Nash equilibrium to correlated equilibrium. In particular, there exists no machine–anonymous and owner–anonymous mechanism that implements optimal schedules in Nash equilibrium in mixed strategies.

**Theorem 2. (Impossibility of implementation.)**

*There is no mechanism that satisfies machine–anonymity and owner–anonymity and that implements optimal schedules in correlated equilibrium for the full class of scheduling problems.*

*Proof.* Let  $\varphi$  be a mechanism that satisfies machine–anonymity and owner–anonymity. Let

$$\Lambda_{J_1} = (M, N, (J_1, J_2), (p_j)_{j \in J})$$

be such that  $M = \{m_1, m_2\}$ ,  $N = \{1, 2\}$ ,  $J = \{a, b, c\}$  with processing times  $(p_a, p_b, p_c) = (1, 2, 3)$ , and  $J_1, J_2 \subseteq J$  such that  $J_1 \cup J_2 = J$  and  $J_1 \cap J_2 = \emptyset$ . We will later specify  $J_1$  (and thus  $J_2$  as well), i.e., choose the owner of each job in order to create a “convenient” scheduling game. Table 2 depicts the four optimal schedules, i.e., for all  $J_1 \subseteq J$ ,  $\Sigma^*(\Lambda_{J_1}) = \{\alpha, \beta, \gamma, \delta\}$ .

Since mechanism  $\varphi$  satisfies machine–anonymity and owner–anonymity, it follows that for any two jobs  $x, y \in \{a, b, c\}$ ,  $x \neq y$ , if they are the only jobs sent to some machine, then they are processed on that machine and their order is independent of the identity of the machine and the identity of the owners of any of the jobs. Therefore, we can write  $[x, y]$  (i.e., first  $x$  and then  $y$ ) or  $[y, x]$  (i.e., first  $y$  and then  $x$ ) to indicate that order. We distinguish among the following four cases. In each case we will specify  $J_1$  and prove that none of the correlated equilibria of the induced game  $\Lambda \equiv \Lambda_{J_1}$  satisfies (2).

	optimal schedule
$\alpha$	$m_1 : a, b$ $m_2 : c$
$\beta$	$m_1 : a, c$ $m_2 : b$
$\gamma$	$m_1 : c$ $m_2 : a, b$
$\delta$	$m_1 : b$ $m_2 : a, c$

Table 2: Optimal schedules for  $J = \{a, b, c\}$  with processing times  $(p_a, p_b, p_c) = (1, 2, 3)$ .

**Case I:**  $\varphi$  yields  $[b, c]$  and  $[a, b]$ .

Let  $J_1 = \{a, c\}$  and  $J_2 = \{b\}$ . Assume there exists a correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \in \mathcal{E}(\Gamma(\Lambda, \varphi))$  such that (2) holds.

For any set of schedules  $\Sigma$ , define

$$\Omega^\Sigma = \{\omega \in \Omega : \varphi(\Lambda, [s_i(\omega)]_{i \in N}) \in \Sigma\}.$$

Let  $i \in N$ . We qualify  $\Sigma$  to be “consistent” with  $\mathcal{P}_i$  if there are  $P_i^1, \dots, P_i^l \in \mathcal{P}_i$  such that  $\Omega^\Sigma = \cup_{r=1}^l P_i^r$ . If  $\Sigma$  is consistent with  $\mathcal{P}_i$ , then we can define a  $\mathcal{P}_i$ -measurable function  $t_i$  that coincides with  $s_i$  on  $\Omega \setminus \Omega^\Sigma$  and specify some “deviating allocation” on  $\Omega^\Sigma$ .

We prove that  $\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$  cannot be a subset of the set of optimal schedules  $\Sigma^*(\Lambda)$ , which contradicts (2). We proceed by considering all subsets of  $\Sigma^*(\Lambda)$ .

As an illustration, we prove first that  $\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \neq \{\alpha, \beta\}$ . Suppose that  $\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) = \{\alpha, \beta\}$ . The key observation is the following: Since player 2’s job is assigned to different machines at  $\alpha$  and  $\beta$ , it follows that  $\Sigma = \{\alpha\}$  is consistent with  $\mathcal{P}_2$ . Hence, we can consider the  $\mathcal{P}_2$ -measurable function  $t_2$  that coincides with  $s_2$  on  $\Omega \setminus \Omega^\Sigma$  and specify the deviating allocation “send job  $b$  to machine  $m_2$ ” on  $\Omega^\Sigma$ . Note that at profile  $(s_i)_{i \in N}$ , with probability  $Pr(\Omega^{\{\alpha\}})$  job  $b$  is processed and completed at time 3 (on machine  $m_1$ ) and with probability  $Pr(\Omega^{\{\beta\}}) = 1 - Pr(\Omega^{\{\alpha\}})$  job  $b$  is processed and completed at time 2 (on machine  $m_2$ ). However, at profile  $(s_1, t_2)$  job  $b$  is not only always processed on machine 2 it is also always completed at time 2 (due to the assumption that  $\varphi$  yields  $[b, c]$ ). Since this constitutes a violation of (1), it follows that  $\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \neq \{\alpha, \beta\}$ . The proof for this case is summarized in the 5th row (labeled  $\{\alpha, \beta\}$ ) in Table 3. All other subcases can be similarly discarded, as indicated in Table 3.

**Case II:**  $\varphi$  yields  $[b, c]$  and  $[b, a]$ .

Let  $J_1 = \{b, c\}$  and  $J_2 = \{a\}$ . Assume there exists a correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \in \mathcal{E}(\Gamma(\Lambda, \varphi))$  such that (2) holds. Since  $\varphi$  yields  $[b, a]$ , we have  $\alpha, \gamma \notin \mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$ . Hence,

$$\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \subseteq \{\beta, \delta\}.$$

The only three remaining cases of subsets of  $\Sigma^*(\Lambda)$  can be discarded as well. The arguments are similar to those in Case I and are presented in Table 4.

	player $i$	consistent $\Sigma$	dev. allocation on $\Omega^\Sigma$
$\{\alpha\}$	2	$\{\alpha\}$	$b \rightarrow m_2$
$\{\beta\}$	1	$\{\beta\}$	$a \rightarrow m_2$ and $c \rightarrow m_1$
$\{\gamma\}$	2	$\{\gamma\}$	$b \rightarrow m_1$
$\{\delta\}$	1	$\{\delta\}$	$a \rightarrow m_1$ and $c \rightarrow m_2$
$\{\alpha, \beta\}$	2	$\{\alpha\}$	$b \rightarrow m_2$
$\{\alpha, \gamma\}$	2	$\{\alpha\}$	$b \rightarrow m_2$
$\{\alpha, \delta\}$	1	$\{\delta\}$	$a \rightarrow m_1$ and $c \rightarrow m_2$
$\{\beta, \gamma\}$	1	$\{\beta\}$	$a \rightarrow m_2$ and $c \rightarrow m_1$
$\{\beta, \delta\}$	1	$\{\beta\}$	$a \rightarrow m_2$ and $c \rightarrow m_1$
$\{\gamma, \delta\}$	2	$\{\gamma\}$	$b \rightarrow m_1$
$\{\alpha, \beta, \gamma\}$	2	$\{\alpha\}$	$b \rightarrow m_2$
$\{\alpha, \beta, \delta\}$	1	$\{\delta\}$	$a \rightarrow m_1$ and $c \rightarrow m_2$
$\{\alpha, \gamma, \delta\}$	1	$\{\delta\}$	$a \rightarrow m_1$ and $c \rightarrow m_2$
$\{\beta, \gamma, \delta\}$	1	$\{\beta\}$	$a \rightarrow m_2$ and $c \rightarrow m_1$
$\{\alpha, \beta, \gamma, \delta\}$	1	$\{\beta\}$	$a \rightarrow m_2$ and $c \rightarrow m_1$

Table 3: Deviating allocations in Case I:  $[b, c]$ ,  $[a, b]$ , and  $J_1 = \{a, c\}$ .

	player $i$	consistent $\Sigma$	dev. allocation on $\Omega^\Sigma$
$\{\beta\}$	1	$\{\beta\}$	$b \rightarrow m_1$ and $c \rightarrow m_2$
$\{\delta\}$	1	$\{\delta\}$	$b \rightarrow m_2$ and $c \rightarrow m_1$
$\{\beta, \delta\}$	1	$\{\beta\}$	$b \rightarrow m_1$ and $c \rightarrow m_2$

Table 4: Deviating allocations in Case II:  $[b, c]$ ,  $[b, a]$ , and  $J_1 = \{b, c\}$ .

**Case III:**  $\varphi$  yields  $[c, b]$  and  $[a, c]$ .

Let  $J_1 = \{a, b\}$  and  $J_2 = \{c\}$ . Assume there exists a correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \in \mathcal{E}(\Gamma(\Lambda, \varphi))$  such that (2) holds.

We claim that  $\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$  cannot be a subset of the set of optimal schedules  $\Sigma^*(\Lambda)$ , which contradicts (2). The arguments are presented in Table 5.

**Case IV:**  $\varphi$  yields  $[c, b]$  and  $[c, a]$ .

Let  $J_1 = \{b, c\}$  and  $J_2 = \{a\}$ . Assume there exists a correlated equilibrium  $(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \in \mathcal{E}(\Gamma(\Lambda, \varphi))$  such that (2) holds. Since  $\varphi$  yields  $[c, a]$ , we have  $\beta, \delta \notin \mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N})$ . Hence,

$$\mathcal{O}(\Omega, Pr, (\mathcal{P}_i)_{i \in N}, (s_i)_{i \in N}) \subseteq \{\alpha, \gamma\}.$$

The only three remaining cases of subsets of  $\Sigma^*(\Lambda)$  can be discarded as well. The arguments are presented in Table 6.  $\square$

In the proof of Theorem 2 it is sufficient to consider a situation with only two machines. So, our impossibility results still holds if we weaken machine-anonymity to the property that only considers switches between two machines. However, the following remark shows that this property is in fact equivalent to machine-anonymity.

**Remark 4. (Equivalent definition of machine-anonymity.)** Machine-anonymity can equivalently be defined by imposing (3) for transpositions, i.e., permutations  $\rho : M \rightarrow$

	player $i$	consistent $\Sigma$	dev. allocation on $\Omega^\Sigma$
$\{\alpha\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\beta\}$	2	$\{\beta\}$	$c \rightarrow m_2$
$\{\gamma\}$	1	$\{\gamma\}$	$a \rightarrow m_1$ and $b \rightarrow m_2$
$\{\delta\}$	2	$\{\delta\}$	$c \rightarrow m_1$
$\{\alpha, \beta\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\alpha, \gamma\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\alpha, \delta\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\beta, \gamma\}$	1	$\{\gamma\}$	$a \rightarrow m_1$ and $b \rightarrow m_2$
$\{\beta, \delta\}$	2	$\{\beta\}$	$c \rightarrow m_2$
$\{\gamma, \delta\}$	1	$\{\gamma\}$	$a \rightarrow m_1$ and $b \rightarrow m_2$
$\{\alpha, \beta, \gamma\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\alpha, \beta, \delta\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\alpha, \gamma, \delta\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$
$\{\beta, \gamma, \delta\}$	1	$\{\gamma\}$	$a \rightarrow m_1$ and $b \rightarrow m_2$
$\{\alpha, \beta, \gamma, \delta\}$	1	$\{\alpha\}$	$a \rightarrow m_2$ and $b \rightarrow m_1$

Table 5: Deviating allocations in Case III:  $[c, b]$ ,  $[a, c]$ , and  $J_1 = \{a, b\}$ .

	player $i$	consistent $\Sigma$	dev. allocation on $\Omega^\Sigma$
$\{\alpha\}$	1	$\{\alpha\}$	$b \rightarrow m_2$ and $c \rightarrow m_1$
$\{\gamma\}$	1	$\{\gamma\}$	$b \rightarrow m_1$ and $c \rightarrow m_2$
$\{\alpha, \gamma\}$	1	$\{\alpha\}$	$b \rightarrow m_2$ and $c \rightarrow m_1$

Table 6: Deviating allocations in Case IV:  $[c, b]$ ,  $[c, a]$ , and  $J_1 = \{b, c\}$ .

$M$  that only switch two machines: for some  $m, m' \in M$ ,  $\rho(m) = m'$ ,  $\rho(m') = m$ , and  $\rho(m'') = m''$  for all  $m'' \in M \setminus \{m, m'\}$ . The equivalence follows from the fact that any permutation can be written as a product of transpositions.  $\diamond$

## 4 Outlook

In the context of outsourcing, we have considered the question whether a central authority can install a mechanism such that strategic interaction leads to a socially optimal schedule. Assuming that the mechanism has to satisfy two mild anonymity properties the answer is negative for the general domain, even if we consider correlated equilibria. On the other hand, as long as each company has a single job, the shortest-first mechanism fulfills all requirements.

There are at least three natural follow-up questions. First, it could be of interest to identify the family of mechanisms that satisfy the desiderata and provide implementability on the single demand domain, and then use other normative properties to select a single (or subfamily of) mechanism(s). Second, another possible research line would study the implementability of optimal schedules on the general demand domain if one or both of the two anonymity properties are weakened. Third, in the proof of Theorem 2 we distinguish among four different cases and construct in each case a specific scheduling problem such that no correlated equilibrium only induces optimal schedules. In particular, we cannot use the same scheduling problem to cover all cases. Each case deals with a particular class

of mechanisms that satisfy our anonymity properties. Hence, an interesting question is to determine the “problematic” scheduling problems in each case so that by excluding them a maximal domain for implementation can be obtained.

## References

- [1] Abed F., J.R. Correa, and CC. Huang (2014): “Optimal Coordination Mechanisms for Multi-job Scheduling Games,” In: Schulz A.S., Wagner D. (eds) Algorithms – ESA 2014. Lecture Notes in Computer Science, vol 8737. Springer, Berlin, Heidelberg.
- [2] Agussurja L., and H.C. Lau (2009): “The Price of Stability in Selfish Scheduling Games,” *Web Intelligence and Agent Systems*, 7(4), 305–311.
- [3] Angel E., E. Bampis, F. Pascual, and N. Thibault (2016): “Truthfulness for the Sum of Weighted Completion Times,” In: Dinh T., Thai M. (eds) Computing and Combinatorics. COCOON 2016. Lecture Notes in Computer Science, vol 9797. Springer, Cham.
- [4] Anshelevich E. A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden (2008): “The Price of Stability for Network Design with Fair Cost Allocation,” *SIAM J. Comput.*, 38(4), 1602–1623.
- [5] Arrow, K.J. (1950): “A Difficulty in the Concept of Social Welfare,” *Journal of Political Economy*, 58(4), 328–346.
- [6] Aumann, Y., and Y. Dombb (2010): “Pareto Efficiency and Approximate Pareto Efficiency in Routing and Load Balancing Games,” In: *Proceedings of the 3rd International Symposium on Algorithmic Game Theory*, pp. 66–77.
- [7] Braat, J., H. Hamers, F. Klijn, and M. Slikker (2016): “A Selfish Allocation Heuristic in Scheduling: Equilibrium and Inefficiency Bound Analysis,” Working Paper, Tilburg University.
- [8] Bukchin, Y., and E. Hanany (2007): “Decentralization Cost in Scheduling: A Game-Theoretic Approach,” *Manufacturing and Service Operations Management*, 9(3), 263–275.
- [9] Bukchin, Y., and E. Hanany (2011): “Decentralization Cost in Supply Chain Jobshop Scheduling with Minimum Flowtime Objective,” Working Paper, Tel Aviv University.
- [10] Cachon, G.P., and S. Netessine (2004): “Game Theory in Supply Chain Analysis.” In: Simchi-Levi, D., S.D. Wu, and Z.-J. Shen (Eds.), *Handbook of Quantitative Supply Chain Analysis*, pp. 13–65. New York: Springer.
- [11] Caragiannis, I. (2013): “Efficient Coordination Mechanisms for Unrelated Machine Scheduling,” *Algorithmica*, 66(3), 512–540.
- [12] Cohen J., and F. Pascual (2015): “Scheduling Tasks from Selfish Multi-tasks Agents,” In: Träff J., Hunold S., Versaci F. (eds) Euro-Par 2015: Parallel Processing. Euro-Par 2015. Lecture Notes in Computer Science, vol 9233. Springer, Berlin, Heidelberg.
- [13] Cole, R., J. Correa, V. Gkatzelis, V. Mirrokni, and N. Olver (2015): “Decentralized Utilitarian Mechanisms for Scheduling Games,” *Games and Economic Behavior*, 92, 306–326.
- [14] Correa, J.R., and M. Queyranne (2012): “Efficiency of Equilibria in Restricted Uniform Machine Scheduling with Total Weighted Completion Time as Social Cost,” *Naval Research Logistics*, 59(5), 384–395.
- [15] Dubey, P. (1986): “Inefficiency of Nash Equilibria,” *Mathematics of Operations Research*, 11(1), 1–8.

- [16] Epstein, L. and E. Kleiman (2011): “On the Quality and Complexity of Pareto Equilibria in the Job Scheduling Game,” In: *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 525–532.
- [17] Hoeksma, R., and M. Uetz (2012): “The Price of Anarchy for Minsum Related Machine Scheduling.” In: Solis–Oba, R., and G. Persiano (Eds.), 9th Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science Studies in Economic Theory, pp. 261–273. Berlin, Heidelberg: Springer.
- [18] Horowitz, E., and S. Sahni (1976): “Exact and Approximate Algorithms for Scheduling Nonidentical Processors,” *Journal of the ACM*, 23(2), 317–327.
- [19] Ibarra, O., and C. Kim (1977): “Heuristic Algorithms for Scheduling Independent Tasks on Non-identical Processors,” *Journal of the ACM*, 24(2) 280289.
- [20] Immorlica, N., L. Li, V. Mirrokni, and A. Schulz (2009): “Coordination Mechanisms for Selfish Scheduling,” *Theoretical Computer Science*, 410(17), 1589–1598.
- [21] Jackson, M.O. (2001): “A Crash Course in Implementation Theory,” *Social Choice and Welfare*, 18(4), 655–708.
- [22] Koutsoupias, E., and C. Papadimitriou (2009): “Worst-case Equilibria,” *Computer Science Review*, 3(2), 65–69.
- [23] Lee, K., J. Y-T. Leung, and M. L. Pinedo (2012): “Coordination Mechanisms for Parallel Machine Scheduling,” *European Journal of Operational Research*, 220(2), 305–313.
- [24] Li, X., and Q. Wang (2007): “Coordination Mechanisms of Supply Chain Systems,” *European Journal of Operational Research*, 179(1), 1–16.
- [25] Li, L., and S. Whang (2001): “Game Theory Models in Operations Management and Information Systems,” In: Chatterjee K. and W.F. Samuelson (Eds), *Game Theory and Business Applications*, pp. 95–131. New York: Kluwer Academic Publishers.
- [26] Maskin, E. (1985): “The Theory of Implementation in Nash Equilibrium: A Survey.” In: Hurwicz L., D. Schmeidler, and H. Sonnenschein (Eds.), *Social Goals and Social Organization: Volume in Memory of Elisha Pazner*, pp. 173–204. Cambridge: Cambridge University Press.
- [27] Myerson, R.B., and M.A. Satterthwaite (1983): “Efficient Mechanisms for Bilateral Trading,” *Journal of Economic Theory*, 29(2), 265–281.
- [28] Papadimitriou, C. (2001): “Algorithms, Games, and the Internet,” In: *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, pp. 749–753.
- [29] Perakis, G., and G. Roels (2007): “The Price of Anarchy in Supply Chains: Quantifying the Efficiency of Price-Only Contracts,” *Management Science*, 53(8), 1249–1268.
- [30] Schulzan, A., and N. Stier Moses (2003): “On the Performance of User Equilibria in Traffic Networks,” In: *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 86–87.
- [31] Smith, W. (1956): “Various Optimizers for Single-Stage Production,” *Naval Research Logistics Quarterly*, 3(1–2), 59–66.