European Journal of Operational Research 295 (2021) 12–21

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

# Open shop scheduling games

Ata Atay[a,*], Pedro Calleja[b], Sergio Soteras[c]

[a] Departamento de Análisis Económico, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain
[b] Departament de Matemàtica Econòmica, Financera i Actuarial, and BEAT, Universitat de Barcelona, Barcelona, Spain
[c] Universitat Oberta de Catalunya, Barcelona, Spain

ARTICLE INFO

*Article history:*
Received 30 July 2019
Accepted 10 February 2021
Available online 20 February 2021

*Keywords:*
Scheduling
Open shop
Cooperative game theory
Core allocation rule
Balancedness

ABSTRACT

This paper takes a game theoretical approach to open shop scheduling problems to minimize the sum of completion times. We assume that there is an initial schedule to process the jobs (consisting of a number of operations) on the machines and that each job is owned by a different player. Thus, we can associate a cooperative TU-game to any open shop scheduling problem, assigning to each coalition the maximal cost savings it can obtain through admissible rearrangements of jobs' operations. A number of different approaches to admissible schedules for a coalition are introduced and, in the main result of the paper, a core allocation rule is provided for games arising from unit (execution times and weights) open shop scheduling problems for the most of these approaches. To sharpen the bounds of the set of open shop scheduling problems that result in games that are balanced, we provide two counterexamples: one for general open shop problems and another for further relaxations of the definition of admissible rearrangements for a coalition.



## 1. Introduction

In a scheduling problem a set of jobs have to be executed by a number of machines. Such a general formulation arises in many real life situations, like manufacturing processes, computer science, logistics, etc. In this paper we consider *open shop scheduling problems* introduced by Gonzalez and Sahni (1976) in which $n$ jobs consisting of $m$ operations have to be processed on $m$ machines, each operation on a different machine. We do not allow preemptions, the order in which the jobs' operations are processed is immaterial but two operations of the same job cannot be processed simultaneously (for a survey see Chapter 8 in Pinedo, 2012 or Chapter 6 in Leung, 2004).

In this paper we additionally assume that there is an initial schedule ("first come, first served" for instance) to process the jobs' operations on the machines and that each job is owned by a different player. Moreover, for any admissible schedule, each player incurs some waiting cost until all her job's operations has been processed and she can leave the system. Concretely, each player's cost is assumed to be a weight of the completion time of her job. Thus, we can take a game theoretical approach to address the question on how to distribute the cost savings the players can obtain by cooperation, whenever they rearrange their jobs to be processed in an optimal way, minimizing total weighted completion times. The computational complexity of finding optimal schedules for open shop problems to minimize the sum of weighted completion times has been well-established in the literature since Achugbue and Chin (1982). However, Adiri and Amit (1984) provide two neat algorithms to obtain optimal schedules for unit open shop scheduling problems, where all processing times of all operations are equal as well as all players' weights for their completion times.

Curiel, Pederzoli, and Tijs (1989) are the first to study one-machine situations with weighted completion time as the cost-criterion from such game theoretical point of view. Other multiple machine problems have been studied from this perspective as parallel machines (Hamers, Klijn, & Suijs, 1999; Calleja, Borm, Hamers, Klijn, & Slikker, 2002) or flow shop problems (van den Nouweland, Krabbenborg, & Potters, 1992; Estévez-Fernández, Mosquera, Borm, & Hamers, 2008). Curiel, Hamers, and Klijn (2002) provide an extensive review of scheduling problems that have been treated from this point of view.

In case of cooperation, and in order to obtain stable allocations of the total cost savings, where no coalition of players receives less that the cost savings it can generate by itself, we need to determine first what rearrangements of their jobs' operations are allowed for the coalition. An accepted and broadly used definition of admissible rearrangement for a coalition (Curiel et al., 1989) imposes that the set of predecessors of a player not in the

* Corresponding author.
*E-mail addresses:* aatay@cee.uned.es (A. Atay), calleja@ub.edu (P. Calleja), ssoterass@uoc.com (S. Soteras).

https://doi.org/10.1016/j.ejor.2021.02.030
0377-2217/

coalition on a machine should be the same as initially. For open shop problems, and due to the fact that machines may incur idle time, such condition is not sufficiently strong to protect players outside the coalition of being hurt. This forces us to think of additional conditions on what should be admissible for a coalition. At this point, and inspired by Curiel, Prasad, Tijs, and Veltman (1993), we use two different types of requisites. On one hand, we establish conditions guaranteeing that waiting costs of players outside the coalition do not increase by preventing either the starting time of the operations of jobs outside the coalition or the completion time of such jobs to increase. On the other hand, we study requisites that take into account the position of the operations in the queue on each machine. Here, we tolerate that the set of predecessors of players not in the coalition is contained in, rather than equal to, the initial set of predecessors; or similarly, the position in the queue for players outside the coalition does not change (does not increase). It is worth to point out that all these relaxations of the first condition on the set of predecessors allow for the operations of jobs in a coalition to jump over operations of jobs not belonging to the coalition. Combining these different types of conditions, in total we consider nineteen definitions of admissible rearrangements for a coalition.

For one machine scheduling problems, Curiel et al. (1989) provide a constructive proof of the existence of stable allocations, by proposing an allocation rule in the core of the associated game, for the first of the definitions. Slikker (2006) shows the existence of stable allocations indirectly for another four of the approaches. Van Velzen and Hamers (2003) and Musegaas, Borm, and Quant (2015) study the existence of core elements considering slightly different definitions of admissibility. In the main result of the paper, we provide a stable allocation rule of the total cost savings obtained by cooperation for unit open shop scheduling problems and for all but three of the definitions we propose of admissible schedules for a coalition. Unfortunately, except for balancedness, the games do not present any further structure as convexity or $\sigma-$ component additivity, classes of games widely studied in the literature of scheduling games. For the other three definitions of admissible rearrangements we provide a counterexample to show that stable allocations may not exist. Moreover, the core of the associated games might be empty for general open shop problems for the most of the proposed approaches to admissible rearrangements.

The rest of the paper is organized as follows. In Section 2, we introduce open shop scheduling problems and present some optimal schedules for unit open shop problems when considering the weighted completion time criterion. Section 3 introduces the coalitional game associated with an open shop scheduling problem with initial schedule and discusses which rearrangements should be admissible for a coalition. In Section 4, we show our main result, that is the core of a unit open shop scheduling game is nonempty for a wide set of definitions of admissible rearrangements for a coalition. We additionally show that such games neither need to be convex nor $\sigma-$ component additive. Finally, in Section 5, we study to what extend balancedness still holds when we consider general open shop problems.

## 2. Open shop scheduling problems

An *open shop scheduling problem* consists of $n$ jobs, $N = \{1, \dots, n\}$, each of them consisting of $m$ operations, each one to be processed on a different machine, being $M = \{1, 2, \dots, m\}$ the set of machines. When no confusion arises we denote by $|N| = n$ the cardinality of the set of jobs and by $|M| = m$ the cardinality of the set of machines. Alternatively, we can think of a set of players, any of them needing to finish a job that consists of $m$ operations, each of these operations to be processed on a different machine. Players and jobs are used interchangeably throughout this paper.

The operation of job $i \in N$ on machine $j \in M$ is denoted by the pair $(i, j)$, $p_i^j > 0$ denotes the processing time of $(i, j)$, and by $p := \left(p_i^j\right)_{i \in N, j \in M}$ we denote the matrix of processing times. We assume that all operations have to be processed uninterrupted, that is, preemptions are not allowed. Moreover, in an open shop scheduling problem the process order of a job's operations is immaterial, but two operations of the same job cannot be processed simultaneously. Also, a machine cannot process more than one job at a time.

A *schedule* is a mapping $s : N \times M \to \mathbb{R}_+$ that assigns to every operation a starting time. The set of all feasible schedules, according to the open shop specifications, is denoted by $\mathcal{S}$. Let $s \in \mathcal{S}$, $s(i, j)$ denotes the starting time of operation $(i, j)$ according to the schedule $s$. Since no preemption is allowed, $C_i^j(s) = s(i, j) + p_i^j$ is the completion time of the operation $(i, j)$ according to $s$. We denote the completion time of job $i$ according to $s$ by $C_i(s) = \max_{j \in M} C_i^j(s)$.

A *scheme* $\sigma = (\sigma^j)_{j \in M}$ is a collection of $m$ bijections, $\sigma^j : N \to \{1, \dots, n\}$, one for each machine $j \in M$, where $\sigma^j(i) = k$ interprets the operation of job (player) $i$ on machine $j$ is at position $k$ according to scheme $\sigma$. In other words, player $i$ has the right to process her operation on machine $j$ before her $n - k$ followers according to $\sigma^j$. The set of all possible schemes is denoted by $\Sigma$. A feasible schedule $s \in \mathcal{S}$ is *compatible* with the scheme $\sigma \in \Sigma$ if and only if for all $j \in M$ and $i, i' \in N$ it holds

$$s(i, j) < s(i', j) \iff \sigma^j(i) < \sigma^j(i').$$

In the next example, we illustrate that a given scheme $\sigma \in \Sigma$ admits a number of different compatible feasible schedules. On the other hand, a given schedule $s \in \mathcal{S}$ is clearly compatible with a unique scheme.

**Example 1.** Consider the open shop scheduling problem with $N = \{1, 2\}$, $M = \{1, 2\}$, $p_i^j = 1$ for all $i \in N$ and all $j \in M$, and consider the scheme $\sigma^1 = \sigma^2 = (1, 2)$. Then, $\sigma$ admits, among others, the following two feasible schedules $s_1$ and $s_2$:

| | | | | |
|---|---|---|---|---|
| $m_1$ | 1 | 2 | | $s_1$, |
| $m_2$ | | 1 | 2 | |

| | | | | |
|---|---|---|---|---|
| $m_1$ | | 1 | 2 | $s_2$. |
| $m_2$ | 1 | 2 | | |

In the first schedule machine 2 incurs idle time, while in the second schedule machine 1 incurs idle time.

A *semi-active schedule* is such that there does not exist an operation which could be started earlier without altering the processing scheme or violating the restrictions on the processing of operations, according to the open shop specifications. So, all machines start processing all operations as soon as it is possible without violating the fact that two operations of the same job cannot be processed at the same time. In Example 1, there are no more semi-active schedules than $s_1$ and $s_2$. Observe that there is no one-to-one correspondence between schemes and semi-active schedules for open shop problems.

Every job (player) $i \in N$ has a waiting cost that is linear with respect to the moment it can leave the system, i.e. the cost function of a job $i \in N$ for a given $s \in \mathcal{S}$ is of type $c_i(s) = \alpha_i C_i(s)$ where $\alpha_i > 0$ is the weight or waiting cost per unit time of player $i$. By $\alpha := (\alpha_i)_{i \in N}$ we refer to the vector of weights. An open shop scheduling problem with these specifications is a 4−tuple $(M, N, p, \alpha)$. Our first aim is to find an optimal schedule $s^* \in \mathcal{S}$ that minimizes the weighted sum of completion times. Note that since the waiting

costs are non-decreasing with respect to the completion time for all $i \in N$, then, we only need to look at semi-active schedules.

Finding optimal schedules for such open shop situations is a difficult computational problem and has been proved to be NP-hard even if there are only two machines (see for instance Achugbue & Chin, 1982). Thus, we will pay special attention to unit open shop problems. In a unit open shop problem, $p_i^j = p$ for all $i \in N$ and all $j \in M$, and $\alpha_i = \alpha$ for all $i \in N$. Without loss of generality we assume that $p_i^j = 1$ for all $i \in N$ and all $j \in M$, and $\alpha_i = 1$ for all $i \in N$. A unit open shop scheduling problem is a pair $(N, M)$.

Adiri and Amit (1984) provide two different optimal schedules for unit open shop problems minimizing the weighted sum of completion times. For our purpose, here we introduce one of them as Algorithm 2 (Algorithm 1 in Adiri & Amit (1984)):

**Algorithm 2.**

> Schedule operations of player $i \in N$ continuously, starting at the earliest possible time (respecting operations processing restrictions) on machine $m$ if there exists $k \in \mathbb{N}$ such that $i = km$, and on machine $i \mod (m)$ otherwise,[1] until machine $m$. Then, move to the earliest possible time (respecting operations processing restrictions) on machine 1 and schedule continuously the remaining operations of player $i$.

Next, we provide a unit open shop scheduling problem with six players and four machines to illustrate Algorithm 2.

**Example 3.** Consider $(N, M)$ with $N = \{1, 2, 3, 4, 5, 6\}$ and $M = \{1, 2, 3, 4\}$. Then, an optimal schedule $s^* \in \mathcal{S}$ according to Algorithm 2 is:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $m_1$ | 1 | 4 | 3 | 2 | 5 | | | 6 |
| $m_2$ | 2 | 1 | 4 | 3 | 6 | 5 | | |
| $m_3$ | 3 | 2 | 1 | 4 | | 6 | 5 | |
| $m_4$ | 4 | 3 | 2 | 1 | | | 6 | 5 |

which is only compatible with the associated scheme $\sigma$:

| | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma^1$ | 1 | 4 | 3 | 2 | 5 | 6 |
| $\sigma^2$ | 2 | 1 | 4 | 3 | 6 | 5 |
| $\sigma^3$ | 3 | 2 | 1 | 4 | 6 | 5 |
| $\sigma^4$ | 4 | 3 | 2 | 1 | 6 | 5 |

As noted in Adiri and Amit (1984), if $n = mk + l$ with $k = \lfloor \frac{n}{m} \rfloor$ and $l \geq 0$, this algorithm constructs $k$ compact blocks where machines do not stop between operations. In block $1 \leq r \leq k$, $m$ jobs start processing at time $(r-1)m$ and finish at time $rm$. In case $l > 0$, in the last block $k+1$, the last $l$ jobs start at $km$ and finish at $(k+1)m$, but in this block the machines incur some idle interval.

Note that there is a machine ($m_2$ in Example 3) that processes all operations continuously, and $C_i(s^*) = \left\lceil \frac{\sigma^2(i)}{m} \right\rceil m$ for all $i \in N$,[1] indeed. Note also that, in fact, there are many optimal schedules which can be obtained by just switching the names of the players.

## 3. Open shop scheduling games

Under the assumption that there is an initial feasible schedule $s_0 \in \mathcal{S}$ that describes the initial processing of the operations on all machines, an *open shop scheduling problem with initial schedule*, $s_0$, is a 5−tuple $(N, M, p, \alpha, s_0)$, while a *unit open shop scheduling problem with initial schedule*, $s_0$, is a triplet $(N, M, s_0)$.

[1] For all $x \in \mathbb{R}$, $\lceil x \rceil := \min\{k \in \mathbb{Z} \mid x \leq k\}$.

A *cooperative transferable utility (TU) game* is defined by a pair $(N, v)$ where $N$ is the (finite) player set and the characteristic function $v$ assigns a real number $v(T)$ to each coalition $T \subseteq N$, with $v(\emptyset) = 0$.

For any coalition $\emptyset \neq T \subseteq N$ and any feasible schedule $s \in \mathcal{S}$, by $c_T(s) = \sum_{i \in T} c_i(s)$, we denote the waiting cost of the coalition $T$ according to $s$. Then, given an open shop scheduling problem with initial schedule $(N, M, p, \alpha, s_0)$, we define the open shop scheduling game $(N, v)$ where the characteristic function assigns to every coalition the maximal cost savings it can obtain by means of admissible rearrangements (or admissible schedules). That is, if $\mathcal{AS}(T) \subseteq \mathcal{S}$ denotes the set of admissible schedules for coalition $\emptyset \neq T \subseteq N$,

$$v(T) = c_T(s_0) - c_T(s_T^*),$$

where $s_T^*$ is an optimal admissible schedule for coalition $T$, that is $c_T(s_T^*) = \min_{s \in \mathcal{AS}(T)} c_T(s)$. In the following, $\mathcal{AS}_*(T)$ stands for the set of optimal admissible schedules for coalition $T$.

Observe that for $(N, M, s_0)$, the corresponding unit open shop scheduling game $(N, v)$ can be easily defined by means of the completion times of the players (jobs). Let $C_T(s) = \sum_{i \in T} C_i(s)$ for all $s \in \mathcal{S}$, then $v(T) = C_T(s_0) - C_T(s_T^*)$ for all $\emptyset \neq T \subseteq N$, where $s_T^* \in \mathcal{AS}_*(T)$.

Clearly, $\mathcal{AS}(N)$ should coincide with $\mathcal{S}$ under any definition of admissible rearrangement. Curiel et al. (1993) impose two principles that should be considered when defining which rearrangements are admissible for a coalition:

(i) The rearrangement should not hurt the interests of the players outside the coalition.
(ii) The rearrangement should be possible without an active cooperation of players outside the coalition.

Following most of the literature on one or multiple parallel machines (see for instance Curiel et al., 2002) we say that a schedule $s$, which is compatible with the unique scheme $\sigma$, will be admissible for a coalition $\emptyset \neq T \subset N$ if for each machine, no player outside the coalition $T$ has a different set of predecessors as initially. That is, if by $\sigma_0$ we denote the unique scheme compatible with $s_0$, for all $i \in N \setminus T$ and all $j \in M$, it holds

$$\{k \in N : \sigma^j(k) < \sigma^j(i)\} = \{k \in N : \sigma_0^j(k) < \sigma_0^j(i)\}. \tag{1}$$

Hence, for a given $j \in M$, switches are only allowed among players from connected coalitions. A coalition $T \subseteq N$ is called connected with respect to $\sigma_0^j$ if for all $i, i' \in T$ and $k$ such that $\sigma_0^j(i) < \sigma_0^j(k) < \sigma_0^j(i')$, it holds that $k \in T$. We denote by $T/\sigma_0^j$ the set of maximally connected components of $T$ according to $\sigma_0^j$. We denote the set of admissible schedules for coalition $T$ that satisfies (1) by $\mathcal{AS}^1(T)$. Unfortunately, as shown in Example 4, given $T \subseteq N$, $\mathcal{AS}_*^1(T)$ might include admissible rearrangements that hurt players outside the coalition $T$.

**Example 4.** Consider $(N, M, s_0)$ with $N = \{1, 2, 3, 4, 5\}$, $M = \{1, 2\}$, and the initial schedule, $s_0$, as follows:

| | | | | | | |
|---|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | | 3 | 4 | 5 |
| $m_2$ | 5 | 1 | 3 | 4 | 2 | |

Let $T = \{3, 5\}$. The optimal schedule $s_T^* \in \mathcal{AS}_*^1(T)$ is:

| | | | | | | |
|---|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | **3** | 4 | **5** | |
| $m_2$ | **5** | 1 | | **3** | 4 | 2 |

Then, $C_3(s_0) = 4$, $C_5(s_0) = 6$ while $C_3(s_T^*) = 4$, $C_5(s_T^*) = 5$, and $v(\{3, 5\}) = 1$. However, $s_T^*$ hurts the interests of player 2, who does not take part of the coalition $T$, since $C_2(s_0) = 5 < 6 = C_2(s_T^*)$. It is worth mentioning that although one may argue that to obtain

$s_T^*$, from $s_0$, requires the active cooperation of players 2 and 4 as operations (2,2) and (4,2) are delayed, we can also interpret that is enough that player 3 decides to process the operation (3,1) first instead of the operation (3,2).

The following example shows that, given $T \subseteq N$, $\mathcal{AS}_*^1(T)$ might include an admissible rearrangement that unambiguously requires the active cooperation of players outside coalition $T$.

**Example 5.** Consider $(N, M, s_0)$ with $N = \{1, 2, 3\}$, $M = \{1, 2\}$, and the initial schedule, $s_0$, as follows:

| | | | | | |
|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | | |
| $m_2$ | | 1 | | 3 | 2 |

Let $T = \{2\}$. An optimal schedule $s_T^* \in \mathcal{AS}_*^1(T)$ is:

| | | | | |
|---|---|---|---|---|
| $m_1$ | 1 | **2** | | 3 |
| $m_2$ | | 1 | 3 | **2** |

Then, $C_2(s_0) = 5$, $C_2(s_T^*) = 4$, and hence $v(\{2\}) = 1$. Even though $s_T^*$ delays the operation (3,1), $C_3(s_0) = C_3(s_T^*)$, and hence player 3 is not hurt. However, changing from schedule $s_0$ to $s_T^*$ requires the active cooperation of player 3. Observe that, contrary to $s_0$, in $s_T^*$ player 3 decides to process first (3,2) instead of (3,1).

In view of Examples 4 and 5, it is clear that due to the fact that coalitions can make use of idle times on machines, condition (1) is not enough to guarantee the two principles required for the definition of admissible rearrangements of a coalition.

In Curiel et al. (1993), a number of different approaches to admissible arrangements are studied. They combine two different ideas. In the first one, players in a coalition are allowed to jump over players outside the coalition. We will address this approach later. In the second one, they simply focus on the starting time (completion time) of operations of players outside the coalition. If those times do not increase, they will not be worse off. We present three proposals, inspired by those in Curiel et al. (1993). The first one is based on the starting time of operations. That is, a schedule $s \in \mathcal{S}$, with corresponding compatible scheme $\sigma$, is admissible for coalition $\emptyset \neq T \subset N$ if it satisfies (1) and the starting time of operations of players outside $T$ remains unchanged:

for all $i \in N \setminus T$ and all $j \in M$, it holds that, $s(i, j) = s_0(i, j)$. (a)

We denote the set of admissible schedules for coalition $T$ that satisfy (1) and (a) by $\mathcal{AS}^{1a}(T)$.

A second approach considers that a schedule $s \in \mathcal{S}$ with corresponding compatible scheme $\sigma$ is admissible for a coalition $\emptyset \neq T \subset N$ if it satisfies (1) and the starting time of operations of players outside $T$ does not increase:

for all $i \in N \setminus T$ and all $j \in M$, it holds that, $s(i, j) \leq s_0(i, j)$. (b)

We denote the set of admissible schedules for coalition $T$ that satisfy (1) and (b) by $\mathcal{AS}^{1b}(T)$.

In Curiel et al. (1993), only one machine problems are studied, hence condition (a) is equivalent to enforcing non-increasing completion times for all $i \in N \setminus T$. Following that spirit, we introduce a new definition of admissible rearrangements. A schedule $s \in \mathcal{S}$ with corresponding compatible scheme $\sigma$ is admissible for a coalition $\emptyset \neq T \subset N$ if it satisfies (1) and the completion time of players outside $T$ does not increase:

for all $i \in N \setminus T$, it holds that, $C_i(s) \leq C_i(s_0)$. (c)

We denote the set of admissible schedules for coalition $T$ that satisfy (1) and (c) by $\mathcal{AS}^{1c}(T)$.

Clearly, for a given $\emptyset \neq T \subset N$, we have $\mathcal{AS}^{1a}(T) \subseteq \mathcal{AS}^{1b}(T) \subseteq \mathcal{AS}^{1c}(T) \subseteq \mathcal{AS}^{1}(T)$. Moreover, conditions (a), (b), and (c) ensure that admissible rearrangements will not hurt the interests of players outside the coalition.

On the other hand, in Example 5, for $T = \{2\}$, $s_T^* \in \mathcal{AS}_*^{1c}(T)$ and, as noted, changing from $s_0$ to $s_T^*$ requires the active cooperation of player 3. Observe that this is possible because player 3 "makes use" of the idle times on machines. Given $s_0 \in \mathcal{S}$ and $\emptyset \neq T \subset N$, a player $i \in N \setminus T$ can only "make use" of idle times in profit of coalition $T$ to reach a rearrangement $s \in \mathcal{AS}^{1c}(T)$, as player 3 in Example 5, if there is a machine $j \in M$ such that the initial starting time of operation $(i, j)$, $s_0(i, j)$, is smaller than the starting time of $(i, j)$ according to $s$;

$$s_0(i, j) < s(i, j).$$

In Example 5, such machine is $m_1$. Hence, obviously, $s \notin \mathcal{AS}^{1b}(T)$. Then, admissible rearrangements in $\mathcal{AS}^{1a}(T)$ and $\mathcal{AS}^{1b}(T)$ do not allow for the active cooperation of players outside $T$.

Allowing players in a coalition to jump over players outside the coalition, that is relaxing condition (1), give rise to larger sets of admissible rearrangements. If we follow the same approach introduced in Curiel et al. (1993) and later used by Slikker (2006), we would like to change condition (1) by the following weaker condition on the schemes. Let $\sigma_0$ be the unique scheme compatible with $s_0$, we say that a schedule $s$ is admissible for $\emptyset \neq T \subset N$ if for all $i \in N \setminus T$ and all $j \in M$, it holds

$$\sigma_0^j(i) = \sigma^j(i), \tag{2}$$

where $\sigma$ is the unique scheme compatible with $s$. Condition (2) requires that players outside the coalition $T$ maintain the same position as initially on every machine. From the perspective of the members of coalition $T$ and for an arbitrary machine $j \in M$, now, they are allowed to switch their positions, independently if they belong to the same connected component according to $\sigma_0^j$. The set of admissible schedules for coalition $T$ that satisfy (2) is denoted by $\mathcal{AS}^2(T)$.

Further relaxations of condition (1) can be obtained by imposing that a schedule $s$ is admissible for $\emptyset \neq T \subset N$, if for all $i \in N \setminus T$ and all $j \in M$ it holds

$$\{k \in N : \sigma^j(k) < \sigma^j(i)\} \subseteq \{k \in N : \sigma_0^j(k) < \sigma_0^j(i)\}, \tag{3}$$

with $\sigma_0$ and $\sigma$ being the schemes corresponding to $s_0$ and $s$, respectively. Clearly (3) weakens condition (1) since each player outside coalition $T$ do not get any new predecessor on each machine. This approach is similar to the one taken in Musegaas et al. (2015). By $\mathcal{AS}^3(T)$ we denote the set of admissible schedules for coalition $T$ that satisfy (3). Interestingly, condition (1) implies both, conditions (2) and (3), but conditions (2) and (3) do not imply each other.

Finally, we introduce the weakest relaxation of condition (1): a schedule $s$ is admissible for $\emptyset \neq T \subset N$, if for all $i \in N \setminus T$ and all $j \in M$, it holds

$$\sigma_0^j(i) \geq \sigma^j(i). \tag{4}$$

with $\sigma_0$ and $\sigma$ being the schemes corresponding to $s_0$ and $s$, respectively. Condition (4) imposes that the number of predecessors for a player outside coalition $T$ do not increase on each machine. By $\mathcal{AS}^4(T)$ we denote the set of admissible schedules for coalition $T$ that satisfy (4). Notice that (4) is implied by conditions (1)–(3).

Of course, if we want to prevent hurting players in $N \setminus T$ we should combine (2)–(4) with either (a) or (b) or (c). We would then obtain some new domains of admissible schedules for $T$: $\mathcal{AS}^{2a}(T)$, $\mathcal{AS}^{2b}(T)$, $\mathcal{AS}^{2c}(T)$, $\mathcal{AS}^{3a}(T)$, $\mathcal{AS}^{3b}(T)$, $\mathcal{AS}^{3c}(T)$, $\mathcal{AS}^{4a}(T)$, $\mathcal{AS}^{4b}(T)$ and $\mathcal{AS}^{4c}(T)$.

For every different set of admissible rearrangements, we can associate the corresponding cooperative TU-game. Let $(N, v^k)$ denote the game where the set of admissible rearrangements for a coalition $\emptyset \neq T \subseteq N$ is $\mathcal{AS}^k(T)$, with $k \in \{1, 2, 3, 4\}$; and let $(N, v^{kl})$ denote the game where the set of admissible rearrangements for a

coalition $\emptyset \neq T \subseteq N$ is $\mathcal{AS}^{kl}(T)$, with $k \in \{1,2,3,4\}$ and $l \in \{a,b,c\}$. Proposition 6 provides the relation among these games.

**Proposition 6.** *Let $(N, M, p, \alpha, s_0)$ be an open shop scheduling problem with initial schedule. Then, it holds*

$$\begin{aligned} &v^1(N) = v^2(N) = v^3(N) = v^4(N), \\ &v^k(N) = v^{ka}(N) = v^{kb}(N) = v^{kc}(N) && \text{for all } k \in \{1,2,3,4\}, \\ &v^1(T) \leq v^2(T) \leq v^4(T) && \text{for all } T \subset N, \\ &v^1(T) \leq v^3(T) \leq v^4(T) && \text{for all } T \subset N, \text{ and} \\ &v^{ka}(T) \leq v^{kb}(T) \leq v^{kc}(T) \leq v^k(T) && \text{for all } T \subset N \text{ and all } k \in \{1,2,3,4\}. \end{aligned}$$

**Proof.** It follows from the observation that for an arbitrary $\emptyset \neq T \subset N$, it holds

$$\begin{aligned} &\mathcal{AS}^1(T) \subseteq \mathcal{AS}^2(T) \subseteq \mathcal{AS}^4(T), \\ &\mathcal{AS}^1(T) \subseteq \mathcal{AS}^3(T) \subseteq \mathcal{AS}^4(T), && \text{and} \\ &\mathcal{AS}^{ka}(T) \subseteq \mathcal{AS}^{kb}(T) \subseteq \mathcal{AS}^{kc}(T) \subseteq \mathcal{AS}^k(T) && \text{for all } k \in \{1,2,3,4\}. \end{aligned}$$

In case $T = N$ all the sets of admissible schedules are equal and equal to $\mathcal{S}$, i.e. $\mathcal{S} = \mathcal{AS}^1(N) = \mathcal{AS}^2(N) = \mathcal{AS}^3(N) = \mathcal{AS}^4(N)$ and $\mathcal{S} = \mathcal{AS}^{ka}(N) = \mathcal{AS}^{kb}(N) = \mathcal{AS}^{kc}(N) = \mathcal{AS}^k(N)$ for all $k \in \{1,2,3,4\}$. □

## 4. Non-emptiness of the core for unit open shop games

Given a cooperative game $(N, v)$, a *payoff vector* $x \in \mathbb{R}^N$ represents the payoffs to the players. Each component $x_i$ is interpreted as the allotment to player $i \in N$. The total payoff to a coalition $S \subseteq N$ is denoted by $x(S) = \sum_{i \in S} x_i$ with $x(\emptyset) = 0$. In order to study the set of stable allocations of the total cost savings $N$ can obtain, we introduce the *core* of a cooperative game $(N, v)$ that consists of those payoff vectors that satisfy efficiency and every coalition $S \subset N$ receives at least its worth: $x(S) \geq v(S)$ (Gillies, 1959). Formally, the core of a cooperative game $(N, v)$ is:

$$C(v) = \{x \in \mathbb{R}^N \mid x(N) = v(N), \quad x(S) \geq v(S) \quad \text{for all} \quad S \subset N\}.$$

A game is *balanced* if it has a non-empty core. Convexity (Shapley, 1971) and $\sigma$-component additivity (Curiel, Potters, Prasad, Tijs, & Veltman, 1994) are conditions that have been extensively studied to prove balancedness of scheduling games associated with different scheduling problems (see for instance Curiel et al., 1994; Hamers, Borm, & Tijs, 1995; Borm, Fiestras-Janeiro, Hamers, Sánchez, & Voorneveld, 2002; Musegaas, Borm, & Quant, 2018). A game $(N, v)$ is said to be *convex* if $v(T \cup \{i\}) - v(T) \geq v(S \cup \{i\}) - v(S)$ for all $i \in N$ and all $S \subseteq T \subseteq N \backslash \{i\}$. Given an open shop scheduling problem with initial schedule, the next remark, which is shown by means of counterexamples, stresses that none of the sixteen classes of open shop games introduced in Section 3 needs to be convex, not even for unit open shop scheduling problems.

**Remark 7.** Let $(N, M, s_0)$ be a unit open shop scheduling problem with initial schedule. Then, the associated games $(N, v^k)$ for all $k \in \{1,2,3,4\}$ and $(N, v^{kl})$ for all $k \in \{1,2,3,4\}$ and all $l \in \{a,b,c\}$ need not be convex.

We first show, in Example 8, that $(N, v^1)$ and $(N, v^{1l})$ with $l \in \{a,b,c\}$ need not be convex.

**Example 8.** Consider $(N, M, s_0)$ with $N = \{1,2,3,4,5,6,7\}$, $M = \{1,2\}$, and the initial schedule, $s_0$, as follows:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $m_2$ | 6 | 5 | 7 | 1 | 2 | 3 | 4 |

Take $S = \{3,5\}$, $T = \{1,2,3,5,7\}$ and $i = 4$. The optimal schedule for coalition $S$ under condition (1), $s^*_S \in \mathcal{AS}^1_*(S)$, is $s_0$, since the operations of players 3 and 5 are disconnected on both machines and machines process all operations continuously according to $s_0$.

The schedule $s^*_{S\cup\{i\}} \in \mathcal{AS}^1_*(S \cup \{i\})$ is optimal for coalition $S \cup \{i\}$ :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | **5** | **3** | **4** | 6 | 7 |
| $m_2$ | 6 | **5** | 7 | 1 | 2 | **3** | **4** |

The schedule $s^*_T \in \mathcal{AS}^1_*(T)$ is optimal for coalition $T$:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | **1** | **2** | **3** | 4 | **5** | 6 | **7** |
| $m_2$ | 6 | **1** | **2** | **5** | **3** | **7** | 4 |

The schedule $s^*_{T\cup\{i\}} \in \mathcal{AS}^1_*(T \cup \{i\})$ is optimal for coalition $T \cup \{i\}$:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | **1** | **2** | **3** | **5** | **4** | 6 | **7** |
| $m_2$ | 6 | **1** | **2** | **3** | **5** | **7** | **4** |

Therefore, $v^1(T \cup \{i\}) - v^1(T) = 6 - 5 < 2 - 0 = v^1(S \cup \{i\}) - v^1(S)$. Hence, the game $(N, v^1)$ is not convex.

Moreover, since every optimal schedule satisfies (a) and, thus, (a) and (c), the coalitional worth for $S$, $S \cup \{i\}$, $T$, and $T \cup \{i\}$ is the same in the game $(N, v^1)$ as in the game $(N, v^{1l})$ for all $l \in \{a,b,c\}$. Hence neither $(N, v^1)$ nor $(N, v^{1l})$ for all $l \in \{a,b,c\}$ are convex games.

Second, we show, in Example 9, that $(N, v^2)$, $(N, v^4)$ and $(N, v^{2l})$, $(N, v^{4l})$ with $l \in \{a,b,c\}$ need not be convex.

**Example 9.** Consider $(N, M, s_0)$ with $N = \{1,2,3,4,5,6,7\}$, $M = \{1,2\}$, and the initial schedule, $s_0$, as follows:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $m_2$ | 6 | 4 | 5 | 2 | 3 | 7 | 1 |

Take $S = \{4,5,6\}$, $T = \{2,3,4,5,6\}$ and $i = 1$. The optimal schedule for coalition $S$ under condition (4), $s^*_S \in \mathcal{AS}^4_*(S)$, is $s_0$. Coalition $S$ can not obtain any cost saving by switching the positions of the operations of jobs 4, 5, and 6 on the machines, neither by diminishing the position of the operations of jobs outside the coalition. The schedule $s^*_{S\cup\{i\}} \in \mathcal{AS}^4_*(S \cup \{i\})$ is optimal for coalition $S \cup \{i\}$:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | **4** | 2 | 3 | **5** | **6** | **1** | 7 |
| $m_2$ | **6** | **4** | **5** | 2 | 3 | 7 | **1** |

The schedule $s^*_T \in \mathcal{AS}^4_*(T)$ is optimal for coalition $T$:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | 1 | **2** | **3** | **4** | **5** | **6** | 7 |
| $m_2$ | **2** | **3** | **4** | **5** | **6** | 7 | 1 |

The schedule $s^*_{T\cup\{i\}} \in \mathcal{AS}^4_*(T \cup \{i\})$ is optimal for coalition $T \cup \{i\}$:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | **2** | **3** | **4** | **5** | **1** | **6** | 7 |
| $m_2$ | **3** | **2** | **5** | **4** | **6** | 7 | **1** |

Therefore, $v^4(T \cup \{i\}) - v^4(T) = 6 - 4 < 4 - 0 = v^4(S \cup \{i\}) - v^4(S)$. Hence, the game $(N, v^4)$ is not convex.

Furthermore, every optimal schedule holds condition (2) and, additionally, satisfies (a) and, thus, (b) and (c). Then, the coalitional worth for $S$, $S \cup \{i\}$, $T$, and $T \cup \{i\}$ is the same in the game $(N, v^4)$ as in the games $(N, v^2)$, $(N, v^{2l})$, $(N, v^{4l})$ for all $l \in \{a,b,c\}$. Hence, neither $(N, v^2)$ and $(N, v^4)$, nor $(N, v^{2l})$ and $(N, v^{4l})$ for all $l \in \{a,b,c\}$ are convex games.

Finally, we show that $(N, v^3)$ and $(N, v^{3l})$ with $l \in \{a,b,c\}$ need not be convex. Here, we will consider both, Examples 8 and 9. First, consider Example 8 to conclude that $(N, v^{3a})$ need not be convex.

Example 8 (revisited): *The optimal schedules for coalitions $S$, $S \cup \{i\}$, $T$ and $T \cup \{i\}$ under condition (1) are also optimal if we impose (3) and (a) together. Observe that, although (3) is weaker than*

(1), *condition (a) is very strong and moreover, machines operate continuously according to* $s_0$ *leaving no room to obtain additional cost savings to the coalitions. Therefore, the worth for these coalitions is the same in the game* $(N, v^1)$ *as in the game* $(N, v^{3a})$ *which does neither need to be convex.*

To finish, consider Example 9 again.

Example 9 (revisited): *For coalitions* $S$, $T$ *and* $T \cup \{i\}$, *the schedules* $s^*_S$, $s^*_T$ *and* $s^*_{T\cup\{i\}}$ *satisfy condition (3), and, additionally conditions (b) and (c). Consequently, the coalitional worth for these coalitions is the same in the game* $(N, v^4)$ *than in the games* $(N, v^3)$ *and* $(N, v^{3l})$ *with* $l \in \{b, c\}$. *However, the schedule* $s^*_{S\cup\{i\}}$ *for coalition* $S \cup \{i\}$ *does not satisfy (3), since, compared to* $s_0$, *the operations of jobs 2 and 3 get a new predecessor on the first machine. An optimal schedule for coalition* $S \cup \{i\}$ *under condition (3) is:*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $m_1$ | 2 | 3 | **4** | **5** | **6** | **1** | 7 |
| $m_2$ | **6** | **4** | **5** | 2 | 3 | 7 | **1** |

*Therefore,* $v^3(T \cup \{i\}) - v^3(T) = 6 - 4 < 3 - 0 = v^3(S \cup \{i\}) - v^3(S)$. *Hence, the game* $(N, v^3)$ *is not convex. Notice that the optimal schedule for coalition* $S \cup \{i\}$ *under condition (3) also satisfies (b) and (c). Then, the worth of* $S \cup \{i\}$ *is the same in the game* $(N, v^3)$ *as in the games* $(N, v^{3b})$ *and* $(N, v^{3c})$, *and, hence, none of these three games is convex.*

We have then provided the necessary arguments, by means of counterexamples, to show Remark 7.

$\sigma$-component additivity, as well as convexity, implies balancedness. A game $(N, v)$ is said to be *superadditive* if $v(S \cup T) \geq v(S) + v(T)$ for all $S, T \subseteq N$, $S \cap T = \emptyset$. Let $\sigma$ be an order on the player set $N$, a game $(N, v)$ is said to be *$\sigma$-component additivity* if it satisfies the following three conditions:

- $v(\{i\}) = 0$ for all $i \in N$,
- $(N, v)$ is superadditive,
- $v(T) = \sum_{S\in T/\sigma} v(S)$ for all $T \subseteq N$.

In Example 5, and for $T = \{2\}$, we obtained $v^1(T) = 1$. So, the game $(N, v^1)$ is not $\sigma$-component additive. In the following remark we emphasize, by a single counterexample, that none of the sixteen games introduced in Section 3 is $\sigma$-component additive, not even for unit open shop scheduling problems.

**Remark 10.** Let $(N, M, s_0)$ be a unit open shop scheduling problem with initial schedule. Then, the associated games $(N, v^k)$ for all $k \in \{1, 2, 3, 4\}$ and $(N, v^{kl})$ for all $k \in \{1, 2, 3, 4\}$ and all $l \in \{a, b, c\}$ need not be $\sigma$-component additive.

Example 11 provides the necessary arguments to show Remark 10.

**Example 11.** Consider $(N, M, s_0)$ with $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $M = \{1, 2, 3, 4\}$, and the initial schedule, $s_0$, as follows:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $m_1$ | **1** | **2** | 3 | **4** | **5** | 6 | 7 | 8 | 9 | 10 |
| $m_2$ | **5** | 3 | **2** | 9 | 6 | **4** | **1** | 7 | 10 | 8 |
| $m_3$ | 3 | **1** | **5** | 7 | 9 | 10 | 8 | **4** | **2** | 6 |
| $m_4$ | **2** | **5** | **1** | 10 | 3 | 8 | 6 | 9 | 7 | **4** |

We first show that the game $(N, v^{1a})$ need not be $\sigma$-component additive. Observe that $v^{1a}(\{i\}) = 0$ for all $i \in N$. Moreover $v^{1a}(\{1, 4\}) = v^{1a}(\{2, 4\}) = v^{1a}(\{4, 5\}) = 1$. To reach the optimal schedules for coalitions $\{1, 4\}$, $\{2, 4\}$, and $\{4, 5\}$, players 1 and 4 switch their operations on second machine, players 2 and 4 switch their operations on third machine, and players 4 and 5 switch their operations on first machine, respectively. Hence, $\{1, 4\}$, $\{2, 4\}$, and $\{4, 5\}$ have to be connected with respect to $\sigma$, but there is no ordering of the ten players that makes that possible. Consequently, $(N, v^{1a})$ is not $\sigma$-component additive.

Moreover, from Proposition 6, given an arbitrary coalition $T \subset N$, it holds that $v^{1a}(T)$ is smaller than or equal to the coalitional worth of $T$ in any other of the fifteen games introduced in Section 3. Then, the coalitional worth of coalitions $\{1, 4\}$, $\{2, 4\}$, and $\{4, 5\}$ is strictly positive in any of them. So, $\{1, 4\}$, $\{2, 4\}$, and $\{4, 5\}$ have to be connected with respect to $\sigma$, but this is not possible. We conclude that none of the games $(N, v^k)$ with $k \in \{1, 2, 3, 4\}$ and $(N, v^{kl})$ with $k \in \{1, 2, 3, 4\}$ and $l \in \{a, b, c\}$ is $\sigma$-component additive.

Given an open shop scheduling problem with initial schedule $(N, M, p, \alpha, s_0)$, and in view of Proposition 6, if we show that the associated game $(N, v^4)$ is balanced, we can conclude that any of the rest of games introduced in Section 3 is balanced as well. Unfortunately, balancedness does not need to hold for general open shop problems (see Remark 16 in Section 5) and, henceforth, we restrict our attention to unit open shop scheduling problems with initial schedule $(N, M, s_0)$. As a consequence of the lack of structure of the game $(N, v^4)$, displayed in Remarks 7 and 10, we explore the possibility to obtain an allocation rule of the total cost savings laying in the core. This strategy has been used from Curiel, Pederzoli and Tijs (1989), who introduce the equal gain splitting rule, *EGS*, for 1-machine situations. In our setting, a 1-machine situation corresponds to an open shop scheduling problem with initial schedule $(N, M, p, \alpha, s_0)$ with $M = \{1\}$. Concretely, they show that $EGS(N, M, p, \alpha, s_0) \in C(N, v^1)$ for every 1-machine situation. An interpretation of the *EGS* is as follows. For 1-machine situations, we can always describe a procedure to obtain an optimal schedule (the higher the urgency of the job, $\frac{\alpha_i}{p_i}$, the sooner it is processed) from the initial one by switching jobs of neighbours involving some strictly positive cost savings, consecutively. Roughly speaking the *EGS* assigns to each player the half of the cost savings obtained by every switch of her job with the job of a neighbour in such a procedure. This idea is difficult to extend to open shop problems with more than a single machine. Not only because of finding optimal schedules is an NP-hard problem, but also due to obtaining a procedure to construct the optimal schedule from the initial one by means of switches involving some positive cost savings, if possible, might be hard as well.

However, given a unit open shop scheduling problem with initial schedule $(N, M, s_0)$ and in view of Algorithm 2 (see also Example 3), for all $j \in M$ there exists an optimal schedule, that we call $s^*_j$, for $N$ such that its unique compatible scheme $\sigma^* \in \Sigma$ satisfies $(\sigma^*)^j = \sigma^j_0$ and, moreover, machine $j$ does not incur any idle time (operations on machine $j$ are processed continuously) according to $s^*_j$. For any $j \in M$, we introduce the $j$-based allocation $\mu^j(N, M, s_0) \in \mathbb{R}^N$ by:

$$\mu^j_i(N, M, s_0) = C_i(s_0) - C_i(s^*_j) \quad \text{for all } i \in N.$$

Given $j \in M$, this allocation assigns to each player the difference between her initial waiting cost and her cost associated with the optimal schedule $s^*_j$ for $N$. We can interpret that according to $\mu^j$ the processing order rights acquired by the players according to $\sigma^j_0$ are dominant. Observe that urgencies of players are all equal and there are as many optimal schedules as possible orders on the jobs, according to Algorithm 2. So, apparently, it could be natural to respect one of the initial orders on the machines when designing the optimal schedule. This allocation is efficient, since

$$\sum_{i\in N} \mu^j_i(N, M, s_0) = \sum_{i\in N} \left(C_i(s_0) - C_i(s^*_j)\right) = C_N(s_0) - C_N(s^*_j) = v^4(N),$$

where the last equality follows from the fact that $s^*_j$ is optimal for $N$. However, as the next example shows, $\mu^j(N, M, s_0)$ does not need to satisfy $\mu^j_i(N, M, s_0) \geq v^4(\{i\})$ for all $i \in N$, and hence need not be a core element.

**Example 12.** Consider $(N, M, s_0)$ with $N = \{1, 2, 3, 4\}$, $M = \{1, 2\}$, and the initial schedule, $s_0$, as follows:

| | | | | |
|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | 4 |
| $m_2$ | 3 | 4 | 1 | 2 |

Let $j = 1$. Then, $s_1^*$ is:

| | | | | |
|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | 4 |
| $m_2$ | 2 | 1 | 4 | 3 |

Now, consider $i = 3$. Then,

$$\mu_3^1(N, M, s_0) = C_3(s_0) - C_3(s_1^*) = 3 - 4 = -1 < v^4(\{3\}) = 0.$$

Hence, the allocation $\mu^1(N, M, s_0)$ is not a core element of the game $(N, v^4)$.

Given $(N, M, s_0)$, in Theorem 14 we will show that although the $j$ -based allocation does not need to be a core element, surprisingly, the average of all $\mu^j(N, M, s_0)$ always belongs to the core. For any $(N, M, s_0)$, the average machine-based allocation rule $\bar{\mu}(N, M, s_0) \in \mathbb{R}^N$ is defined by

$$\bar{\mu}(N, M, s_0) = \frac{1}{m} \sum_{j \in M} \mu^j(N, M, s_0).$$

The interpretation of $\bar{\mu}$ is very natural, since it makes all possible optimal schedules for $N$ (obtained from Algorithm 2) that maintains the initial order in one of the machines, $s_j^*$ for all $j \in M$, equally likely. Then, a player receives her expectation of cost savings. In order to show our main result, let us first prove a technical lemma that establishes an upper bound for the difference between the completion times of a job $i \in N$, according to an initial schedule $s_0$ and to any other feasible schedule $s$.

**Lemma 13.** *Let $(N, M, s_0)$ be a unit open shop scheduling problem with initial schedule. Then, for all feasible schedule $s \in \mathcal{S}$, and $i \in N$, it holds*

$$\frac{1}{m} \sum_{j \in M} \left( C_i(s_0) - \left\lceil \frac{C_i^j(s)}{m} \right\rceil m \right) \geq C_i(s_0) - C_i(s). \tag{5}$$

**Proof.** Let $(N, M, s_0)$, $s \in \mathcal{S}$, and $i \in N$. Let $j^* \in M$ be such that $C_i(s) = C_i^{j^*}(s)$. Then, clearly $C_i^j(s) \leq C_i^{j^*}(s)$ for all $j \in M$. Moreover,

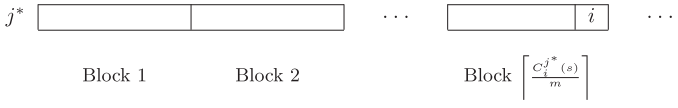
$$C_i^{j^*}(s) \leq \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil m.$$

Here, if according to $s$ we make consecutive blocks of $m$ units of time from the moment at which the system starts processing, $\left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$ stands for the block in which the operation of player $i$ is processed on machine $j^*$.

We distinguish between two cases:

*Case 1:* $C_i^{j^*}(s) = \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil m$.

In Case 1, the operation of job $i$ on machine $j^*$ is exactly the last operation to be processed in block $\left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$ according to $s$. In the graph we represent only machine $j^*$:




From the definition of $j^* \in M$ it follows that

$$C_i(s_0) - \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil m = C_i(s_0) - C_i(s), \tag{6}$$

and for any other machine $j \in M$

$$\begin{aligned} C_i(s_0) - \left\lceil \frac{C_i^j(s)}{m} \right\rceil m &\geq C_i(s_0) - \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil m \\ &= C_i(s_0) - C_i(s). \end{aligned} \tag{7}$$

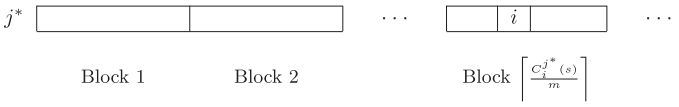
Then, from (6) and (7)

$$\begin{aligned} \frac{1}{m} \sum_{j \in M} \left( C_i(s_0) - \left\lceil \frac{C_i^j(s)}{m} \right\rceil m \right) &\geq \frac{1}{m} \sum_{j \in M} (C_i(s_0) - C_i(s)) \\ &= C_i(s_0) - C_i(s), \end{aligned}$$

which finishes Case 1.

*Case 2:* $C_i^{j^*}(s) < \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil m$.

In Case 2, the operation of job $i$ on machine $j^*$ is not the last operation to be processed in block $\left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$ according to $s$. In the graph we represent only machine $j^*$:




By definition of $j^*$,

$$\left\lceil \frac{C_i^j(s)}{m} \right\rceil \leq \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$$

for all $j \in M$.

Let $J^* = \left\{ j \in M : \left\lceil \frac{C_i^j(s)}{m} \right\rceil = \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil \right\}$. Then, according to $s$, $J^*$ is the set of machines for which the operation of player $i$ according to $s$ is processed in the same block as the operation of player $i$ is processed on machine $j^*$, that is, in block $\left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$. Note that $J^* \neq \emptyset$ since $j^* \in J^*$. On the other hand, if $j \in M \setminus J^*$, then

$$\left\lceil \frac{C_i^j(s)}{m} \right\rceil < \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$$

or equivalently

$$\left\lceil \frac{C_i^j(s)}{m} \right\rceil \leq \left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil - 1 \tag{8}$$

To establish an upper bound for $|J^*|$, notice that all $j \in J^*$, $j \neq j^*$, process the operation of job $i$ in the same block as $j^*$, i.e. in block $\left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$, that contains exactly $m$ units of time. Additionally, the operation of job $i$ is processed at the unit of time $C_i^{j^*}(s)$ on machine $j^*$ and we have $C_i^j(s) < C_i^{j^*}(s)$ for all $j \in J^*$, $j \neq j^*$. Hence, as block $\left\lceil \frac{C_i^{j^*}(s)}{m} \right\rceil$ starts processing at $\left\lfloor \frac{C_i^{j^*}(s)}{m} \right\rfloor m$ and $s \in \mathcal{S}$ is a feasible schedule, and thus, two operations of the same job are not allowed to be processed simultaneously on two different machines, there are only $C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m} \right\rfloor m$ positions available in the block to process the operation of job $i$ on different machines. So, there are as much as $C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m} \right\rfloor m$ different machines in $J^*$, i.e.

$$|J^*| \leq C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m} \right\rfloor m. \tag{9}$$

Consequently,

$$|M \setminus J^*| \geq m - \left(C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m} \right\rfloor m\right). \tag{10}$$

Then,

$$\begin{aligned}
&\sum_{j\in M}\left(C_i(s_0) - \left\lceil \frac{C_i^j(s)}{m}\right\rceil m\right)\\
&= \sum_{j\in J^*}\left(C_i(s_0) - \left\lceil \frac{C_i^j(s)}{m}\right\rceil m\right) + \sum_{j\in M\setminus J^*}\left(C_i(s_0) - \left\lceil \frac{C_i^j(s)}{m}\right\rceil m\right)\\
&\geq \sum_{j\in J^*}\left(C_i(s_0) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right) + \sum_{j\in M\setminus J^*}\left(C_i(s_0) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m\right)\\
&= |J^*|\left(C_i(s_0) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right) + |M\setminus J^*|\left(C_i(s_0) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m\right)\\
&= |J^*|\left(C_i(s_0) - C_i(s) + C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right)\\
&\quad + |M\setminus J^*|\left(C_i(s_0) - C_i(s) + C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m\right)\\
&= m(C_i(s_0) - C_i(s)) + |J^*|\left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right)\\
&\quad + |M\setminus J^*|\left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m\right)\\
&\geq m(C_i(s_0) - C_i(s)) + \left(C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m}\right\rfloor m\right)\left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right)\\
&\quad + |M\setminus J^*|\left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m\right)\\
&\geq m(C_i(s_0) - C_i(s)) + \left(C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m}\right\rfloor m\right)\left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right)\\
&\quad + \left(m - \left(C_i^{j^*}(s) - \left\lfloor \frac{C_i^{j^*}(s)}{m}\right\rfloor m\right)\right)\left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m\right)\\
&= m(C_i(s_0) - C_i(s)) + \left(C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m\right)m + m\left(m - C_i^{j^*}(s) + \left\lfloor \frac{C_i^{j^*}(s)}{m}\right\rfloor m\right)\\
&= m(C_i(s_0) - C_i(s)) + m\left(m\left(1 + \left\lfloor \frac{C_i^{j^*}(s)}{m}\right\rfloor - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil\right)\right)\\
&= m(C_i(s_0) - C_i(s)).
\end{aligned}$$

Here, the first inequality follows from the definition of $J^*$ and (8). The second inequality follows from (9) and the fact that $C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m < 0$, due to in Case 2 we assume $C_i(s) = C_i^{j^*}(s) < \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m$. The third inequality holds from (10), and the fact that $C_i(s) = C_i^{j^*}(s)$, and then

$$C_i(s) - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil m + m = m\left(\frac{C_i^{j^*}(s)}{m} + 1 - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil\right) > 0.$$

The last but one equality follows from $C_i(s) = C_i^{j^*}(s)$ by definition of $j^*$. Finally, the last equality follows from the assumption that in Case 2 the operation of job $i$ on machine $j^*$ is not the last operation to be processed in block $\left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil$, and consequently, $\frac{C_i^{j^*}(s)}{m} \notin \mathbb{Z}$ and $1 + \left\lfloor \frac{C_i^{j^*}(s)}{m}\right\rfloor - \left\lceil \frac{C_i^{j^*}(s)}{m}\right\rceil = 0$.

Consequently, (5) holds and this finishes Case 2. □

Remarkably, Lemma 13 holds for an arbitrary player $i \in N$ and an arbitrary feasible schedule $s \in \mathcal{S}$. In particular, for any player belonging to a given coalition $T \subseteq N$ and an optimal schedule, $s_T^* \in \mathcal{AS}_*^4(T) \subset \mathcal{S}$, for such coalition, which will be helpful to prove the next theorem. Now, we can state the main result of the paper.

**Theorem 14.** *Let $(N, M, s_0)$ be a unit open shop scheduling problem with initial schedule. Then, $\bar{\mu}(N, M, s_0) \in C(v^4)$.*

**Proof.** Let $(N, M, s_0)$, $\mu^j(N, M, s_0) = \mu^j$, and $\bar{\mu}(N, M, s_0) = \bar{\mu}$. First, we show that the allocation rule $\bar{\mu} \in \mathbb{R}^N$ is efficient:

$$\begin{aligned}
\bar{\mu}(N) = \sum_{i\in N}\bar{\mu}_i &= \sum_{i\in N}\frac{1}{m}\sum_{j\in M}\left(C_i(s_0) - C_i(s_j^*)\right)\\
&= \frac{1}{m}\sum_{j\in M}\sum_{i\in N}\left(C_i(s_0) - C_i(s_j^*)\right)\\
&= \frac{1}{m}\sum_{j\in M} C_N(s_0) - C_N(s_j^*)\\
&= v^4(N),
\end{aligned}$$

where the fifth equality follows from the fact that $s_j^*$ is an optimal schedule for $N$ for all $j \in M$ and hence, $C_N(s_j^*) = C_N(s_{j'}^*)$ for all $j, j' \in M$, $j \neq j'$. It remains to prove $\bar{\mu}(T) \geq v^4(T)$ for all $T \subset N$. Let $\emptyset \neq T \subset N$ be an arbitrary proper coalition, $s_T^* \in \mathcal{AS}_*^4(T)$ an optimal schedule for coalition $T$ and $\sigma_T^*$ the unique scheme compatible with $s_T^*$. Then,

$$\begin{aligned}
\bar{\mu}(T) = \sum_{i\in T}\bar{\mu}_i &= \sum_{i\in T}\frac{1}{m}\sum_{j\in M}\mu_i^j\\
&= \sum_{i\in T}\frac{1}{m}\sum_{j\in M}\left(C_i(s_0) - C_i(s_j^*)\right)\\
&= \sum_{i\in T}\frac{1}{m}\sum_{j\in M}\left(C_i(s_0) - \left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil m\right)\\
&= \frac{1}{m}\sum_{j\in M}\left(\sum_{i\in T}C_i(s_0) - \sum_{i\in T}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil m\right)\\
&\geq \frac{1}{m}\sum_{j\in M}\left(\sum_{i\in T}C_i(s_0) - \sum_{i\in T}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil m\right)\\
&\geq \frac{1}{m}\sum_{j\in M}\left(\sum_{i\in T}C_i(s_0) - \sum_{i\in T}\left\lceil \frac{C_i^j(s_T^*)}{m}\right\rceil m\right)\\
&= \sum_{i\in T}\frac{1}{m}\sum_{j\in M}\left(C_i(s_0) - \left\lceil \frac{C_i^j(s_T^*)}{m}\right\rceil m\right)\\
&\geq \sum_{i\in T}C_i(s_0) - C_i(s_T^*) = v^4(T).
\end{aligned}$$

The fourth equality follows from the fact that the unique compatible scheme $\sigma^*$ with $s_j^*$ satisfies $(\sigma^*)^j = \sigma_0^j$ and moreover, according to the schedule $s_j^*$, operations are processed continuously on machine $j$. So, in view of Algorithm 2 (see also Example 3), $C_i(s_j^*) = \left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil m$.

To show the first inequality it is enough to prove that $\sum_{i\in T}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil \geq \sum_{i\in T}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil$ for all $j \in M$. Let $j \in M$, observe that, with respect to the orders $(\sigma_T^*)^j$ and $\sigma_0^j$, it holds that $\sum_{i\in N}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil = \sum_{i\in N}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil = \left\lceil \frac{1}{m}\right\rceil + \left\lceil \frac{2}{m}\right\rceil + \cdots + \left\lceil \frac{n}{m}\right\rceil$. Moreover, since $s_T^* \in \mathcal{AS}_*^4(T)$ satisfies condition (4), we have $(\sigma_T^*)^j(i) \leq \sigma_0^j(i)$ for all $i \in N\setminus T$ and,

consequently, $\sum_{i\in N\setminus T}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil \leq \sum_{i\in N\setminus T}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil$. Then, $\sum_{i\in T}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil = \sum_{i\in N}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil - \sum_{i\in N\setminus T}\left\lceil \frac{(\sigma_T^*)^j(i)}{m}\right\rceil \geq \sum_{i\in N}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil - \sum_{i\in N\setminus T}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil = \sum_{i\in T}\left\lceil \frac{\sigma_0^j(i)}{m}\right\rceil$ and we conclude that the first inequality holds.

The second inequality follows from $C_i^j(s_T^*) \geq \left(\sigma_T^*\right)^j(i)$ for all $i \in T$ and all $j \in M$. Observe that the completion time of the operation of job $i$ on machine $j$ according to the schedule $s_T^*$ can not be strictly smaller than the position of such operation on machine $j$ with respect to the compatible scheme $\sigma_T^*$, since all operations last one unit of time and, moreover, machine $j$ may incur idle time. The last inequality follows from Lemma 13 applied to the players belonging to coalition $T \subseteq N$ and the feasible schedule, $s_T^* \in \mathcal{AS}_*^4(T) \subset \mathcal{S}$. □

In the next corollary, we summarize that an immediate consequence of Theorem 14 is that the sixteen classes of unit open shop games are balanced.

**Corollary 15.** *Let $(N, M, s_0)$ be a unit open shop scheduling problem with initial schedule. Then, $\tilde{\mu}(N, M, s_0) \in C(v^k)$ for all $k \in \{1, 2, 3, 4\}$ and $\tilde{\mu}(N, M, s_0) \in C(v^{kl})$ for all $k \in \{1, 2, 3, 4\}$ and all $l \in \{a, b, c\}$.*

## 5. Non-balancedness results

In Section 4, we prove balancedness for unit open shop games. In this section, we investigate whether, in general, open shop scheduling games are balanced. Here, we provide a counterexample. To be more precise, to show the following remark we present an example with 5 jobs and 3 machines that results in a cooperative game with an empty core for eight of the sixteen approaches to admissible schedules for a coalition introduced in Section 3. Moreover, we show that those games are not even superadditive.

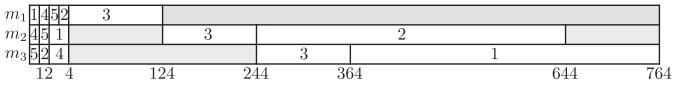
**Remark 16.** Let $(N, M, p, \alpha, s_0)$ be an open shop scheduling problem with initial schedule. Then, the associated games $(N, v^k)$ and $(N, v^{kc})$ for all $k \in \{1, 2, 3, 4\}$ neither need to be balanced, nor superadditive.

The necessary arguments to see that Remark 16 holds are provided in Example 17.

**Example 17.** Consider $(N, M, p, \alpha, s_0)$ with $N = \{1, 2, 3, 4, 5\}$, $M = \{1, 2, 3\}$, $\alpha_1 = 1$, $\alpha_2 = 5$, $\alpha_3 = 50$, $\alpha_4 = \alpha_5 = 2000$, and processing times as follows:

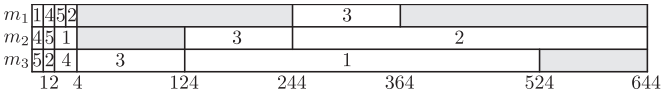
$$p_i^j = \begin{cases} 1 & if \quad (i,j) \in \{(1,1),(2,1),(4,1),(5,1),(4,2),(5,2),(2,3),(5,3)\} \\ 2 & if \quad (i,j) \in \{(1,2),(4,3)\} \\ 120 & if \quad (i,j) \in \{(3,1),(3,2),(3,3)\} \\ 400 & if \quad (i,j) \in \{(1,3),(2,2)\} \end{cases}$$

In order to provide a clear illustration of the schedules in this example, and due to the big differences in the processing times of the operations, in the figures, the operations of the jobs do not respect the appropriate proportions, as well as the time line at the bottom should be read carefully. The initial schedule, $s_0$, is:



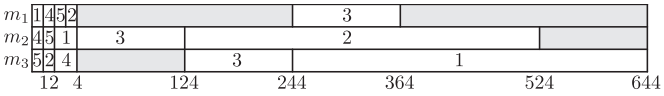

We first show that the game $(N, v^{1c})$ need not be balanced. Notice that $C_1(s_0) = 764$, $C_2(s_0) = 644$, $C_3(s_0) = 364$, $C_4(s_0) = 4$ and $C_5(s_0) = 3$.

Let $T = \{1\}$. The optimal schedule for coalition $T$, $s_T^* \in \mathcal{AS}_*^{1c}(T)$ is:



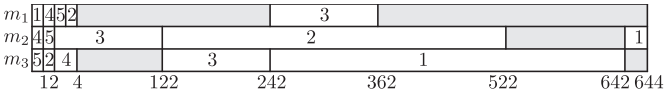

Here, according to $s_T^*$, operations are processed continuously on machine 3 and, consequently, $C_1(s_T^*) = 524$. On the other hand, the completion times of the rest of players are the same as initially and hence, condition (c) holds. Moreover, the unique scheme compatible with $s_T^*$ is like the initial scheme and condition (1) holds as well. Then, together with $\alpha_1 = 1$, we obtain $v^{1c}(\{1\}) = 764 - 524 = 240$.

Now, let $S = \{2, 3, 4, 5\}$. An optimal schedule for coalition $S$, $s_S^* \in \mathcal{AS}_*^{1c}(S)$, is:




Observe that the best alternative for $S$ is to process all operations continuously on machine 2 diminishing the completion time of player 2. Notice that $\alpha_3$, $\alpha_4$ and $\alpha_5$ are very high compared to $\alpha_2$ but, however, the completion times of jobs 4 and 5 can not be reduced, while the completion time of 3 can hardly be reduced. Here, $C_2(s_S^*) = 524$ and the completion times for the rest of members of coalition $S$ are equal to the initial ones. Furthermore, the completion time of player 1 diminishes and then condition (c) holds, and the unique scheme compatible with $s_S^*$ is as the initial one satisfying condition (1). Then, together with $\alpha_2 = 5$, we obtain $v^{1c}(\{2, 3, 4, 5\}) = 5(644 - 524) = 600$.

Finally, an optimal schedule for the grand coalition $N$, $s^*$, is:




Roughly speaking, the high values of $\alpha_3$, $\alpha_4$ and $\alpha_5$, together with the fact that there is a small room for, if possible, reducing the completion times of jobs 3, 4 and 5, enforces the grand coalition to choose between processing the operation (3,2) on machine 2 first (as in $s_S^*$), diminishing the completion time of player 2, or processing the operation (3,3) on machine 3 first (as in $s_T^*$), diminishing the completion time of player 1. Since $\alpha_2$ is five times $\alpha_1$, the first of these two alternatives is better by means of cost savings. Here, $C_1(s^*) = 644$, $C_2(s^*) = 522$, $C_3(s^*) = 362$, while $C_4(s^*) = C_4(s_0)$ and $C_5(s^*) = C_5(s_0)$. Therefore, and since $\alpha_1 = 1$, $\alpha_2 = 5$, and $\alpha_3 = 50$ we obtain $v^{1c}(N) = 120 + 610 + 100 = 830$.

Hence, there does not exist an allocation $x \in \mathbb{R}^5$ that satisfies $x(N) = 830$, $x_1 \geq 240$, and $x_2 + x_3 + x_4 + x_5 \geq 600$, and $C(v^{1c}) = \emptyset$. Furthermore, in view of Proposition 6, given an arbitrary proper coalition $T \subset N$, it holds that $v^{1c}(T)$ is smaller than or equal to the worth of this coalition in the games $(N, v^k)$ and $(N, v^{kc})$ for all $k \in \{1, 2, 3, 4\}$, while, the worth of the grand coalition is the same for these eight games. Then, we conclude that none of the games $(N, v^k)$ and $(N, v^{kc})$ with $k \in \{1, 2, 3, 4\}$ is balanced.

To finish, observe that $v^{1c}(T) + v^{1c}(S) > v^{1c}(T \cup S) = v^{1c}(N)$ and $S \cap T = \emptyset$. Thus, the game $(N, v^{1c})$ is neither superadditive, nor are the games $(N, v^k)$ and $(N, v^{kc})$ for all $k \in \{1, 2, 3, 4\}$.

Unfortunately, we can not use Example 17 to extend the non-balancedness result to the games $(N, v^{kl})$ for $k \in \{1, 2, 3, 4\}$ and $l \in \{a, b\}$ that can also be associated to an open shop scheduling problem with initial schedule. It is important to point out that the worth of the coalition $T = \{1\}$ is no longer positive with these approaches. Then, whether or not the non-balancedness result in

Remark 16 can be extended to these eight different games, remains an open question.

To finish, we provide a counterexample to illustrate that further relaxations on the admissible rearrangements lead to games that also violate balancedness. To be precise, we wonder if for unit open shop scheduling games, balancedness holds if we no longer impose any condition on the associated schemes to obtain admissible schedules for a coalition. Let $T \subset N$, we now admit any schedule for $T$ except if it hurts players in $N \setminus T$. Let $\mathcal{AS}^l(T)$ for $l \in \{a, b, c\}$ be the set of admissible rearrangements for a coalition $T \subset N$ that satisfies only condition $(l)$, but not necessarily (1)–(4). For $l \in \{a, b, c\}$, by $(N, v^l)$ we denote the game associated with an open shop scheduling problem with initial schedule, $(N, M, p, \alpha, s_0)$, where the set of admissible rearrangements is $\mathcal{AS}^l(T)$ for any $\emptyset \neq T \subset N$. The relation among such games is stated in the next proposition.

**Proposition 18.** *Let $(N, M, p, \alpha, s_0)$ be an open shop scheduling problem with initial schedule. Then, it holds*

$$v^a(N) = v^b(N) = v^c(N)$$
$$v^a(T) \leq v^b(T) \leq v^c(T) \qquad \text{for all } T \subset N.$$

Next, we remark the non-balancedness result.

**Remark 19.** Let $(N, M, s_0)$ be a unit open shop scheduling problem with initial schedule. Then, the associated games $(N, v^l)$ for all $l \in \{a, b, c\}$ need not be balanced.

In view of Proposition 18, it is enough to show that there is $(N, M, s_0)$ such that $(N, v^a)$ is not balanced, as depicted in Example 20.

**Example 20.** Consider $(N, M, s_0)$ with $N = \{1, 2, 3\}$, $M = \{1, 2\}$, and the initial schedule, $s_0$, as follows:

| | | | | | |
|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | | |
| $m_2$ | | 1 | | 3 | 2 |

Let $T = \{2\}$. Then, the optimal schedule for $T$, $s^*_T \in \mathcal{AS}^a_*(T)$, is:

| | | | | |
|---|---|---|---|---|
| $m_1$ | 1 | **2** | 3 | |
| $m_2$ | **2** | 1 | | 3 |

Hence, $v^a(\{2\}) = 3$. Let $S = \{3\}$, the optimal schedule for $S$, $s^*_S \in \mathcal{AS}^a_*(S)$, is:

| | | | | | |
|---|---|---|---|---|---|
| $m_1$ | 1 | 2 | **3** | | |
| $m_2$ | **3** | 1 | | | 2 |

and hence, $v^a(\{3\}) = 1$. Finally, the optimal schedule for the grand coalition $N$, $s^*$, is:

| | | | | |
|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | |
| $m_2$ | 2 | 1 | | 3 |

Hence, $v^a(N) = 3$, and there does not exist an allocation $x \in \mathbb{R}^3$ that satisfies $x_1 + x_2 + x_3 = 3$, $x_2 \geq 3$, $x_3 \geq 1$, and $x_1 \geq v^a(\{1\})$ since, obviously $v^a(\{1\}) = 0$. Therefore, $C(v^a) = \emptyset$. □


## Acknowledgements

We acknowledge three anonymous referees for their suggestions and helpful comments. Ata Atay gratefully acknowledges financial support from the Spanish Ministry of Science and Innovation through grant PGC2018-096977-B-I00. Pedro Calleja acknowledges the support from research grant ECO2017-86481-P (Agencia Estatal de Investigación (AEI) y Fondo Europeo de Desarrollo Regional (FEDER)) and 2017SGR778 (Generalitat de Catalunya). This work has been partly supported by COST Action CA16228 European Network for Game Theory. During the earlier stages of this project, Ata Atay received support from the Hungarian National Research, Development and Innovation Office via the grant PD-128348, the Hungarian Academy of Sciences via the Cooperation of Excellences Grant (KEP-6/2019), and from the Belgian French speaking community ARC project no. 15/20-072 of Saint-Louis University - Brussels.


## References


Achugbue, J. O., & Chin, F. Y. (1982). Scheduling the open shop to minimize mean flow time. *SIAM Journal on Computing, 11*, 709–720.

Adiri, I., & Amit, N. (1984). Openshop and flowshop scheduling to minimize sum of completion times. *Computers & Operations Research, 11*, 275–284.

Borm, P., Fiestras-Janeiro, G., Hamers, H., Sánchez, E., & Voorneveld, M. (2002). On the convexity of games corresponding to sequencing situations with due dates. *European Journal of Operational Research, 136*, 616–634.

Calleja, P., Borm, P., Hamers, H., Klijn, F., & Slikker, M. (2002). On a new class of parallel sequencing situations and related games. *Annals of Operations Research, 109*, 265–277.

Curiel, I., Hamers, H., & Klijn, F. (2002). Sequencing games: A survey. In P. Borm, & H. Peters (Eds.), *Chapters in game theory. theory and decision library C* (pp. 27–50). Boston, MA: Springer.

Curiel, I., Pederzoli, G., & Tijs, S. (1989). Sequencing games. *European Journal of Operational Research, 40*, 344–351.

Curiel, I., Potters, J., Prasad, R., Tijs, S., & Veltman, B. (1994). Sequencing and cooperation. *Operations Research, 42*, 566–568.

Curiel, I. P. J., Prasad, R., Tijs, S., & Veltman, B. (1993). Cooperation in one machine scheduling. *Zeitschrift für Operations Research, 38*, 113–129.

Estévez-Fernández, A., Mosquera, M. A., Borm, P., & Hamers, H. (2008). Proportionate flow shop games. *Journal of Scheduling, 11*, 433–447.

Gillies, D. B. (1959). Solutions to general non-zero-sum games. In A. W. Tucker, & R. D. Luce (Eds.), *Contributions to the theory of games IV* (pp. 47–85). Princeton University Press.

Gonzalez, T. F., & Sahni, S. (1976). Open shop scheduling to minimize finish time. *Journal of the ACM (JACM), 23*, 665–679.

Hamers, H., Borm, P., & Tijs, S. (1995). On games corresponding to sequencing situations with ready times. *Mathematical Programming, 69*, 471–483.

Hamers, H., Klijn, F., & Suijs, J. (1999). On the balancedness of multiple machine sequencing games. *European Journal of Operational Research, 119*, 678–691.

Leung, J. Y.-T. (2004). *Handbook of scheduling: Algorithms, models, and performance analysis*. Boca Raton, Florida: Chapman & Hall/CRC.

Musegaas, M., Borm, P., & Quant, M. (2015). Step out–step in sequencing games. *European Journal of Operational Research, 246*, 894–906.

Musegaas, M., Borm, P., & Quant, M. (2018). On the convexity of step out–step in sequencing games. *TOP, 26*, 68–109.

van den Nouweland, A., Krabbenborg, M., & Potters, J. (1992). Flow-shops with a dominant machine. *European Journal of Operational Research, 62*, 38–46.

Pinedo, M. (2012). *Scheduling: Theory, algorithms, and systems*. New York: Springer.

Shapley, L. S. (1971). Cores of convex games. *International Journal of Game Theory, 1*, 11–26.

Slikker, M. (2006). Relaxed sequencing games have a nonempty core. *Naval Research Logistics, 53*, 235–242.

Van Velzen, B., & Hamers, H. (2003). On the balancedness of relaxed sequencing games. *Mathematical Methods of Operations Research, 57*, 287–297.