

Balancedness of sequencing games with multiple parallel machines

Citation for published version (APA):

Slikker, M. (2003). *Balancedness of sequencing games with multiple parallel machines*. (BETA publicatie : working papers; Vol. 95). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2003

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Balancedness of sequencing games with multiple parallel machines*

Marco Slikker[†]

April 25, 2003

Abstract

We provide simple constructive proofs of balancedness of classes of m -PS games, which arise from sequencing situations with m parallel machines. This includes the setting that is studied by Calleja et al. (2001) and Calleja et al. (2002), who provided a complex constructive proof and a simple non-constructive proof of balancedness of a restricted class of 2-PS games, respectively. Furthermore, we provide two counterexamples to illustrate that our balancedness results cannot be extended to a general setting.

Keywords: COOPERATIVE GAME THEORY, SCHEDULING, BALANCEDNESS.

JEL classification: C71

1 Introduction

The study of sequencing situations with m parallel machines and n products that each have to be processed on every machine already dates back to Pritsker et al. (1971). They assume that the different processes on different products, called jobs, can be executed simultaneously. For example one can think of a plumber, electrician, and painter who each have to do a job in a set of houses. Pritsker et al. (1971) consider regular measures of performance associated with the products and conclude that there always exist an optimal schedule with the ‘no passing property’, i.e., products are served in the same order on all machines. As an example of such a regular measure one can think of the costs of a product being proportional to the last finishing time of its jobs. This situation

*The author thanks Herbert Hamers for useful discussions, suggestions, and comments.

[†]Department of Technology Management, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven. E-mail: M.Slikker@tm.tue.nl.

is reconsidered by Calleja et al. (2002) who study cooperative games arising from such a class of parallel sequencing situations. One of their main result deals with games arising from parallel sequencing situations in which all jobs have the same processing time and all products have the same weight. They show that the games associated with these simple 2-PS situations are balanced. Their proof is simple, but nonconstructive. On the other hand, Calleja et al. (2001) provide a constructive proof of balancedness of games associated with simple 2-PS situations. This proof, however, is quite complex.

The analysis of sequencing situations from a cooperative game-theoretical point of view dates back to Curiel et al. (1989). They consider one-machine sequencing situations with weighted completion times as their cost-criterion. Convexity of the associated sequencing games was proven. A recent review on sequencing games can be found in Curiel et al. (2002). Several additional features to the basic setting are covered, like ready times, due dates, more admissible rearrangements, and multiple machines. With respect to the last feature, we mention van den Nouweland et al. (1992), who concentrate on flow shops with a dominant machine, and Hamers et al. (1999) and Slikker (2002), who concentrate on multiple identical machines and jobs that need to be served by one machine only.

In the current paper we focus on balancedness of games associated with parallel sequencing situations (PS-situations for short). First, we reestablish the result of Calleja et al. (2001) and Calleja et al. (2002) by providing a straightforward balancedness-proof for games associated with simple PS situations. This combines the advantages of the complex constructive proof of Calleja et al. (2001) and the simple nonconstructive proof of Calleja et al. (2002). Furthermore, the result is valid for simple PS situations in general and is not restricted to situations with two machines only. Subsequently, we will provide two examples to show that balancedness of games associated with PS situations does not hold in general. Allowing for arbitrary differences in processing times appears to be too much to be sure of balancedness of the associated games. Notwithstanding this negative result we will prove balancedness of a class of games associated with parallel sequencing situations in which weights of the products are completely arbitrary. Though production times are not completely free, a sufficient condition on these production times is the existence of a dominant machine, which is defined as a machine at which each product has a job with maximum production time.¹

The setup of the remainder of this paper is as follows. In section 2 we provide some preliminaries. Subsequently, we show in section 3 that cooperative games arising from simple m -PS situations are balanced. In section 4 we provide two counterexamples for

¹We remark that our definition of a dominant machine differs from the definition employed by van den Nouweland et al. (1992).

balancedness of cooperative games arising from m -PS situations. In section 5, we prove balancedness for situations with arbitrary weights and a dominant machine. We conclude in section 6.

2 Preliminaries

In this section we introduce the notation that will be adopted throughout this paper and include some related results. Our notation is largely in line with Calleja et al. (2002).

In a parallel sequencing situation (PS-situation) $M = \{1, \dots, m\}$ denotes the set of machines and $N = \{1, \dots, n\}$ the set of players. Each player has a product that has to be processed on each machine. Player and product will be used interchangeably throughout this work. The processing of a product on a specific machine is called a job. Several jobs of one player/product can be performed simultaneously. On each machine, the jobs have some initial position described by $\rho = (\rho^j)_{j \in M}$. This initial set of orders is a specific element of the set of rearrangements. In a rearrangement $\sigma = (\sigma^j)_{j \in N}$ the positions of the jobs are described for each $j \in M$ by a bijection $\sigma^j : N \rightarrow \{1, \dots, n\}$, where $\sigma^j(i) = s$ means that the job of product i on machine j is in position s . The production time of the job of product i on machine j is denoted by p_i^j . Usually, it is assumed that $p_i^j > 0$, but a situation in which product i does not have to be processed on machine j could be modeled by setting $p_i^j = 0$ and placing this ‘job’ in first position, setting $\rho^j(i) = 1$.² For notational convenience, we denote $p = (p_i^j)_{j \in M; i \in N}$. Finally, concerning the costs of spending time in the system, every player has a linear cost function $c_i : [0, \infty) \rightarrow \mathbb{R}$ defined by $c_i(t) = \alpha_i t$. Hence, since the completion times of product i on machine j according to σ^j will be denoted by $C_i^j(\sigma^j) = \sum_{k: \sigma^j(k) \leq \sigma^j(i)} p_k^j$ we have that the costs of product i according to $\sigma = (\sigma^j)_{j \in N}$ can be represented by

$$c_i(\sigma) = \alpha_i \max_{j \in M} C_i^j(\sigma^j).$$

The total costs of all products according to σ is denoted by $c_N(\sigma) = \sum_{i \in N} c_i(\sigma)$. For notational convenience, we denote $\alpha = (\alpha_i)_{i \in N}$. Combining these definitions we can formally define a *parallel sequencing situation* by a tuple (M, N, ρ, α, p) , with M , N , ρ , α , and p as described above.

A PS-situation is called an m -PS situation if it deals with m machines. Furthermore, a PS-situation is called *simple* if it deals with a situation in which all weights are equal and all processing times are equal.

²Of course, not all jobs can be in first position if several products have empty jobs on the same machine. However, straightforward adaptations can then be made.

The set of rearrangements of products in N on machines in M is denoted by $\Pi(M, N)$. The optimal scheme for N , or in case of multiple optimal schemes an arbitrary optimal scheme, is denoted by $\sigma_* = (\sigma_*^j)_{j \in N}$ and satisfies

$$c_N(\sigma_*) = \min_{\sigma \in \Pi(M, N)} c_N(\sigma).$$

A rearrangement $\sigma \in \Pi(M, N)$ is called admissible for coalition $S \subseteq N$ if for each machine, no job of a player outside S has a different set of predecessors as originally. Formally, for all $j \in M$ and all $i \in N$ we require that

$$\{k \in N \mid \sigma^j(k) < \sigma^j(i)\} = \{k \in N \mid \rho^j(k) < \rho^j(i)\}.$$

The set of rearrangements σ that satisfy this condition is denoted by $\Pi^S(M, N)$. Pritsker et al. (1971) showed that for the grand coalition there always exists an optimal set of rearrangements that satisfy the *no-passing property*, i.e., all machines process the products in the same order. Calleja et al. (2002) show that such a property does not need to hold for a coalition $S \subset N$.

A cooperative game with transferable utilities, *TU-game*, is a pair (N, v) with N a set of players and $v : 2^N \rightarrow \mathbb{R}$ the characteristic function, which assigns to every coalition $S \subseteq N$ its value $v(S)$ with $v(\emptyset) = 0$. Then, given a PS-situation (M, N, ρ, α, p) we can define the associated PS-game (N, v) by setting for any coalition $S \subseteq N$ the worth of this coalition equal to the maximal costs savings the coalition can obtain by admissible rearrangements. Formally,

$$v(S) = \sum_{i \in S} c_i(\rho) - \min_{\sigma \in \Pi^S(M, N)} \sum_{i \in S} c_i(\sigma).$$

Along the lines of Calleja et al. (2002) it is a straightforward exercise to prove that games associated with PS-situations are *monotonic* ($v(S) \geq v(T)$ if $S \supseteq T$) and *super-additive* ($v(S \cup T) \geq v(S) + v(T)$ for all disjoint $S, T \subseteq N$). Furthermore, if the initial order on all machines is the same, $\rho^j = \bar{\rho}$ for all $j \in M$ then the associated PS-situation is $\bar{\rho}$ -component additive, i.e., it satisfies the following three conditions, (1): $v(\{i\}) = 0$ for all $i \in N$; (2): $(N, v,)$ is superadditive; (3): $v(S) = \sum_{T \in S/\bar{\rho}} v(T)$, where $T \in S/\bar{\rho}$ is the set of all maximally connected components of S . The set $S/\bar{\rho}$ is a partition of S in which two players i and j with $\bar{\rho}(i) < \bar{\rho}(j)$ belong to the same maximally connected component of S iff for all $k \in N$ with $\bar{\rho}(i) < \bar{\rho}(k) < \bar{\rho}(j)$ we have that $k \in S$.

Curiel et al. (1994) proved that $\bar{\rho}$ -component additive games have a nonempty core. The *core* $C(N, v)$ of a game (N, v) consists of the payoff vectors $x \in \mathbb{R}^N$ that satisfy condition $\sum_{i \in S} x_i \geq v(S)$ for all $S \subseteq N$ and $\sum_{i \in N} x_i = v(N)$. Bondareva (1963) and Shapley (1967) independently identified the class of games that have non-empty cores as

the class of balanced games. To describe this class, we define for all $S \subseteq N$ the vector e^S by $e_i^S = 1$ for all $i \in S$ and $e_i^S = 0$ for all $i \in N \setminus S$. A map $\kappa : 2^N \setminus \{\emptyset\} \rightarrow [0, 1]$ is called a *balanced map* if $\sum_{S \in 2^N \setminus \{\emptyset\}} \kappa(S) e^S = e^N$. Further, a game (N, v) is called *balanced* if for every balanced map $\kappa : 2^N \setminus \{\emptyset\} \rightarrow [0, 1]$ it holds that $\sum_{S \in 2^N \setminus \{\emptyset\}} \kappa(S) v(S) \leq v(N)$. The following theorem is due to Bondareva (1963) and Shapley (1967).

Theorem 2.1 Let (N, v) be a coalitional game. Then $C(N, v) \neq \emptyset$ if and only if (N, v) is balanced.

Recall that a PS-situation is called simple if $\alpha_i = \alpha_k$ for all $i, k \in N$ and $p_i^j = p_k^l$ for all $i, k \in N$ and all $j, l \in M$. The following theorem is one of the main results of Calleja et al. (2001) and Calleja et al. (2002).

Theorem 2.2 Any simple 2-PS game is balanced.

Calleja et al. (2001) provide a constructive proof that is quite complex. On the other hand, Calleja et al. (2002) provide a simple nonconstructive proof.

3 Balancedness

In this section we will provide a simple constructive proof of balancedness of simple m -PS games. This combines the advantages of similar results of Calleja et al. (2001) and Calleja et al. (2002). Moreover, it extends their result from a setting with two machines to a setting with an arbitrary number of machines.

Without loss of generality we will throughout this section restrict our analysis of simple m -PS games to simple m -PS games with all weights and all production times equal to 1. This implies that it suffices to describe a simple m -PS game by the triple (M, N, ρ) . The following lemma provides an upper bound for the values of coalitions in simple m -PS games.

Lemma 3.1 Let (M, N, ρ) be a simple m -PS situation and let (N, v) be the associated cooperative game. For all $S \subseteq N$ it holds that

$$v(S) \leq \sum_{i \in S} \left[\max_{j \in M} \{C_i^j(\rho^j)\} - \text{average}_{j \in M} \{C_i^j(\rho^j)\} \right], \quad (1)$$

with equality for $S = N$.

Proof: First, we will prove the equality for the grand coalition. Let $\sigma_* = (\sigma_*^j)_{j \in M}$ be an optimal rearrangement for the grand coalition with $\sigma_*^j = \sigma_*^1$ for all $j \in M$, which exists

because, due to Pritsker et al. (1971), there exists an optimal rearrangement with the 'no passing property'. Then

$$\begin{aligned} c_N(\sigma_*) &= 1 + 2 + \dots + |N| \\ &= \sum_{i \in N} \left[\text{average}_{j \in M} \{C_i^j(\rho^j)\} \right]. \end{aligned}$$

Since

$$c_N(\rho) = \sum_{i \in S} \max_{j \in M} \{C_i^j(\rho^j)\}$$

we find that

$$v(N) = \sum_{i \in N} \left[\max_{j \in M} \{C_i^j(\rho^j)\} - \text{average}_{j \in M} \{C_i^j(\rho^j)\} \right].$$

Secondly, consider $S \subset N$. Let $c_S(\sigma)$ denote the costs for coalition S associated with rearrangement σ . Then

$$c_S((\rho^j)_{j \in M}) = \sum_{i \in S} \max_{j \in M} C_i^j(\rho^j).$$

Let $i_1^{S,j}, \dots, i_s^{S,j}$, with $s = |S|$ denote the available positions for coalition S on machine j in increasing order. Hence, $k_r^S = \max_{j \in M} \{i_r^{S,j}\}$, $r \in \{1, \dots, s\}$ denote optimal finishing times of the players in S if on any machine the jobs could freely switch. Allowing for these switches obviously enlarges the possibilities of coalition S . Hence, with $\sigma_{S,*} = (\sigma_{S,*}^j)_{j \in M}$ an optimal rearrangement for S , we have

$$\begin{aligned} c_S(\sigma_{S,*}) &\geq \sum_{r=1}^s k_r^S \\ &\geq \frac{1}{|M|} \sum_{r=1}^s \sum_{j \in M} i_r^{S,j} \\ &= \sum_{i \in S} \text{average}_{j \in M} \{C_i^j(\rho^j)\}. \end{aligned}$$

Hence,

$$\begin{aligned} v(S) &\leq \sum_{i \in S} \max_{j \in M} C_i^j(\rho^j) - \sum_{i \in S} \text{average}_{j \in M} \{C_i^j(\rho^j)\} \\ &= \sum_{i \in S} \left[\max_{j \in M} \{C_i^j(\rho^j)\} - \text{average}_{j \in M} \{C_i^j(\rho^j)\} \right]. \end{aligned}$$

This completes the proof. □

Using this lemma we can easily prove the first main result of this paper.

Theorem 3.1 Let (M, N, ρ) be a simple m -PS situation and let (N, v) be the associated m -PS game. Then $(x_i)_{i \in N}$ with

$$x_i = \max_{j \in M} \{C_i^j(\rho^j)\} - \text{average}_{j \in M} \{C_i^j(\rho^j)\}$$

for all $i \in N$ belongs to the core of (N, v) .

Proof: By lemma 3.1 we conclude directly that $\sum_{i \in N} x_i = v(N)$ and that $\sum_{i \in S} x_i \geq v(S)$ for all $S \subset N$. \square

4 Two counterexamples

In the previous section we proved balancedness of m -PS games in case all processing times on all machines are equal, and, additionally, all weights are equal. A question that comes immediately to the fore is whether all m -PS games are balanced. In this section we provide two counterexamples.

First, we present an example of a 3-PS situation with 3 products that results in a cooperative game with an empty core. All weights of the players are equal.

Example 4.1 Let (M, N, ρ, α, p) be a 3-PS situation with 3 players. So, $M = \{1, 2, 3\}$ and $N = \{1, 2, 3\}$. Let the initial rearrangement ρ be such that the order of the jobs on the three machines are 1-2-3, 2-3-1, and 3-1-2, respectively. Furthermore, let all weights be equal to 1 and let

$$p_i^j = \begin{cases} 1 & \text{if } j = i; \\ 50 & \text{if } j \neq i. \end{cases}$$

A schematic representation is given in figure 1.

Using this figure, we will determine the associated 3-PS game. Obviously, all 1-player coalitions can obtain no cost-savings. Consider coalition $\{1, 2\}$. The best this coalition can do, is to switch places on the third machine, which will decrease the completion time of player 2 by 50, while the completion time of player 1 remains unchanged since his job at machine 2 finishes at time 101 already. Hence, $v(\{1, 2\}) = 50$. Similarly, we have $v(\{1, 3\}) = v(\{2, 3\}) = 50$. Finally, the best the grand coalition can do is to use the same order on each machine, e.g., 1-2-3. Hence, the costs savings equal $v(N) = 101 + 101 + 101 - 50 - 100 - 101 = 52$. Summarizing, we have

$$v(S) = \begin{cases} 0 & \text{if } |S| \leq 1; \\ 50 & \text{if } |S| = 2; \\ 52 & \text{if } S = N. \end{cases}$$

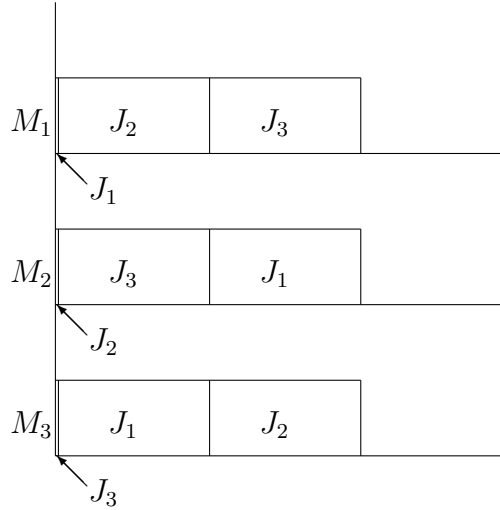


Figure 1: The initial situation.

Since, $v(\{1, 2\}) + v(\{1, 3\}) + v(\{2, 3\}) = 150 > 104 = 2v(N)$ a balancedness condition is not satisfied. By theorem 2.1 we conclude that this 3-PS game has an empty core.³ \diamond

It follows straightforwardly that the example can be extended to include more than three machines and/or more than three products. Since balancedness in case of m machines and two products follows immediately from superadditivity, the next point of interest would be whether a general balancedness-result can be derived for games associated with 2-PS situations. In our second example we show that such a general result does not hold. In this example we present a 2-PS situation with 4 products that results in a cooperative game with an empty core as well. Once again, all weights of the players are equal.

Example 4.2 Let (M, N, ρ, α, p) be a 2-PS situation with 4 players. So, $M = \{1, 2\}$ and $N = \{1, 2, 3, 4\}$. Let the initial rearrangement ρ be such that the order of the jobs on the two machines are 4-1-2-3, and 2-4-3-1, respectively. Furthermore, let all weights be equal to 1 and let

$$p_i^j = \begin{cases} 1 & \text{if } (j, i) \in \{(1, 4), (2, 2)\}; \\ 100 & \text{if } (j, i) = (1, 2); \\ 50 & \text{otherwise.} \end{cases}$$

A schematic representation is given in figure 2.

³A 3-person game with $v(S) = 0$ if $|S| \leq 1$ and $v(S) = 1$ if $|S| \geq 1$ can be achieved in a similar setting by choosing $p_i^j = 0$ if $j = i$ and $p_i^j = 1$ otherwise. Processing times equal to zero, however, might be less appealing.

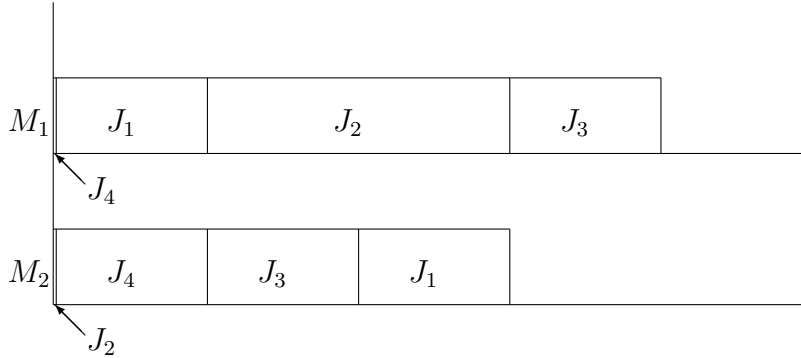


Figure 2: The initial situation.

Using this figure, we will determine (part of) the associated 2-PS game. Obviously, all 1-player coalitions can obtain no cost-savings. Consider coalition $\{1, 2\}$. The best this coalition can do, is to switch places on the first machine, which will decrease the completion time of player 2 by 50, while the completion time of player 1 remains unchanged since his job at machine 2 finishes at time 151 already. Hence, $v(\{1, 2\}) = 50$. By switching on machines 2 and 1, respectively, we have $v(\{1, 3\}) = v(\{2, 3\}) = 50$ as well.

Finally, by the 'no-passing property', the best the grand coalition can do is to use the same order on each machine. By checking all 24 possible orders it follows that, for example, 4-1-3-2 is optimal with associated costs equal to $50+100+150+201=501$. Originally, costs equal $150+151+201+51=553$. Hence, the costs savings equal $v(N) = 553 - 501 = 52$. Since $v(\{1, 2\}) + v(\{1, 3\}) + v(\{2, 3\}) + 2v(\{4\}) = 150 > 104 = 2v(N)$ a balancedness condition is not satisfied. By theorem 2.1 we conclude that (N, v) has an empty core. \diamond

5 Dominant machines

Given the positive results in section 3 and the negative results in section 4 we wonder whether we can sharpen the bound of the set of m -PS situations that result in m -PS games that are balanced.

Before we can introduce a new set of allocation rules we have to introduce some additional notation. Let $(N, \sigma, (p_i)_{i \in N}, (\alpha_i)_{i \in N})$ be a 1-machine sequencing situation. In our notation, this situation would have been represented by $(M, N, \sigma, p, \alpha)$ with $|M| = 1$. Define $g_{ik} = \max\{0, \alpha_k p_i - \alpha_i p_k\}$. Curiel et al. (1989) prove that the equal gain splitting

rule, defined by

$$\text{EGS}_i(N, \sigma, (p_i)_{i \in N}, (\alpha_i)_{i \in N}) = \frac{1}{2} \sum_{k: \sigma(k) < \sigma(i)} g_{ki} + \frac{1}{2} \sum_{k: \sigma(k) > \sigma(i)} g_{ik}$$

for all $i \in N$, always belongs to the core of the sequencing game associated with this situation.

We define the j -based allocation rule x^j in the m -PS game associated with situation (M, N, ρ, α, p) by

$$x_i^j(M, N, \rho, \alpha, p) = \alpha_i \left[\max_{k \in M} \{C_i^k(\rho^k)\} - C_i^j(\rho^j) \right] + \text{EGS}_i(N, \rho^j, p, (\alpha_i)_{i \in N})$$

for all $i \in N$. This allocation rule consists of two parts. First it attributes to a player its weight times the difference between the maximum completion time of its jobs and the completion time on machine j . Secondly, it attributes the amount that the EGS-rule applied on the one-machine situation with machine j only would attribute to the players. We remark that in general, this allocation rule need not be efficient.

Furthermore, we introduce the notion of a dominant machine. A machine $j^* \in M$ is called dominant if for each player the job on this machine has a production time that is at least as much as on any other machine. Formally, $j^* \in M$ is called *dominant* if $p_i^{j^*} \geq p_i^j$ for all $j \in M$ and all $i \in N$.

The following theorem shows that x^{j^*} belongs to the core if j^* is a dominant machine.

Theorem 5.1 Let (M, N, ρ, α, p) be an m -PS situation with $j^* \in M$ a dominant machine. Then $x^{j^*}(M, N, \rho, \alpha, p)$ belongs to the core of the associated m -PS game.

Proof: For convenience we write x^{j^*} instead of $x^{j^*}(M, N, \rho, \alpha, p)$. First, we will show that $\sum_{i \in N} x_i^{j^*} = v(N)$. Let $\sigma_{*,1}^{j^*}$ be an optimal order for machine j^* in one-machine sequencing situation $(N, \rho^{j^*}, \alpha, (p_i^{j^*})_{i \in N})$, i.e., if machine j^* would be the only machine. Define rearrangement τ_* by $\tau_*^j = \sigma_{*,1}^{j^*}$ for all $j \in M$. Then for any rearrangement σ we have

$$\begin{aligned} \sum_{i \in N} \alpha_i \max_{j \in M} C_i^j(\sigma^j) &\geq \sum_{i \in N} \alpha_i C_i^{j^*}(\sigma^{j^*}) \\ &\geq \sum_{i \in N} \alpha_i C_i^{j^*}(\sigma_{*,1}^{j^*}) \\ &= \sum_{i \in N} \alpha_i \max_{j \in M} C_i^j(\tau_*^j) \\ &= \sum_{i \in N} c_i(\tau_*) \end{aligned}$$

The first inequality follows since on the right-hand-side we consider j^* only rather than the maximum over all $j \in M$. The second inequality follows by definition of $\sigma_{*,1}^{j^*}$. Finally, the first equality holds since j^* is a dominant machine which implies that the costs do not increase if we go from the one-machine situation to the current situation in which the order on *all* machines would be $\sigma_{*,1}^{j^*}$. Hence, any rearrangement results in at least the same costs as τ_* , so τ_* is an optimal rearrangement for the grand coalition.

Using this we have

$$\begin{aligned}
\sum_{i \in N} x_i^{j^*} &= \sum_{i \in N} \left[\alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\rho^{j^*}) \right] + \text{EGS}_i(N, \rho^{j^*}, p, (\alpha_i)_{i \in N}) \right] \\
&= \sum_{i \in N} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\rho^{j^*}) \right] + \sum_{i \in N} \alpha_i \left[C_i^{j^*}(\rho^{j^*}) - C_i^{j^*}(\sigma_{*,1}^{j^*}) \right] \\
&= \sum_{i \in N} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\sigma_{*,1}^{j^*}) \right] \\
&= \sum_{i \in N} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - \max_{j \in M} \{C_i^j(\tau_*)\} \right] \\
&= v(N),
\end{aligned}$$

where the first and second equalities follow by definition, the third by rewriting, and the fourth by noting that if all machines have order $\sigma_{*,1}^{j^*}$ then all jobs have their maximal completion time on machine j^* . The fifth equality follows since τ_* is optimal.

Secondly, we will show that $\sum_{i \in S} x_i^{j^*} \geq v(S)$ for all $S \subseteq N$. Therefore, let $S \subseteq N$. Let $(\sigma_S^j)_{j \in M}$ be optimal orders of coalition S . Let (N, w) be the sequencing game associated with 1-machine sequencing situation $(N, \rho^{j^*}, (p_i^{j^*})_{i \in N}, (\alpha_i)_{i \in N})$. Denote the optimal order of coalition S in this 1-machine sequencing situation by τ_S . Then

$$\begin{aligned}
\sum_{i \in S} x_i^{j^*} &= \sum_{i \in S} \left[\alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\rho^{j^*}) \right] + \text{EGS}_i(N, \rho^{j^*}, p, (\alpha_i)_{i \in N}) \right] \\
&\geq \sum_{i \in S} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\rho^{j^*}) \right] + w(S) \\
&= \sum_{i \in S} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\rho^{j^*}) \right] + \sum_{i \in S} \alpha_i \left[C_i^{j^*}(\rho^{j^*}) - C_i^{j^*}(\tau_S) \right] \\
&\geq \sum_{i \in S} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\rho^{j^*}) \right] + \sum_{i \in S} \alpha_i \left[C_i^{j^*}(\rho^{j^*}) - C_i^{j^*}(\sigma_S^{j^*}) \right] \\
&= \sum_{i \in S} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - C_i^{j^*}(\sigma_S^{j^*}) \right] \\
&\geq \sum_{i \in S} \alpha_i \left[\max_{j \in M} \{C_i^j(\rho^j)\} - \max_{j \in M} C_i^j(\sigma_S^j) \right] \\
&= v(S),
\end{aligned}$$

where the first, second, and last equalities follow by definition, the first inequality holds since the equal gain splitting rule belongs to the core of a one-machine sequencing game, the second inequality since τ_S is optimal in $(N, \rho^{j^*}, (p_i^{j^*})_{i \in N}, (\alpha_i)_{i \in N})$ and hence, at least as good as $\sigma_S^{j^*}$, the third equality follows by rearranging terms, and the last inequality since for all $i \in S$ we have that $C_i^{j^*}(\sigma_S^{j^*})$ is at most equal to $\max_{j \in M} C_i^j(\sigma_S^j)$. \square

6 Concluding remarks

In the current paper we have made serious progress in the analysis of balancedness of m -PS games. Up to now, only simple 2-PS games had been analyzed. In this paper we have shown balancedness for m -PS games associated with m -PS situations with a dominant machine and arbitrary weights for the players. Two counterexamples prove that balancedness is not guaranteed for equal weights and arbitrary production times in case the number of products and machines are both at least three and in case the number of machines is at least two and the number of products at least four.⁴ Since balancedness in case of m machines and two products follows immediately from superadditivity, the only issue that remains is balancedness of situations with two machines, three products, arbitrary weights, and arbitrary production times. The same issue for the subclass with equal weights for the players is still unresolved as well, but restricting to equal production times or production times that allow for a dominant machine results in balanced games, even for arbitrary weights, as was shown in the current paper.

The allocation rule that was used in the balancedness proof in section 5 depends on the presence of a dominant machine. If all jobs of a product have the same processing time on all machines then all machines are dominant. Consequently, all j -based allocation rules x^j belong to the core, as does its average. This average can be rewritten as

$$x_i = \alpha_i \left[\max_{j \in M} \{C_i^j(\sigma^j)\} - \text{average}_{j \in M} \{C_i^j(\sigma^j)\} \right] + \sum_{j \in M} \frac{1}{m} \text{EGS}_i(N, \sigma^j, p, (\alpha_i)_{i \in N}),$$

This relates the result of section 3 with the results of section 5.⁵

Finally, we would like to remark that the results in section 3 are implied by the results in section 5. However, we believe that the proofs in section 3 provide some insights that

⁴The counterexamples in section 4 can easily be extended to include more products and or more machines.

⁵The second part of this allocation rule can be seen as a probabilistic equal gain splitting rule (PEGS-rule) as studied by Hamers and Slikker (1995) with equal weight to the initial orders of the different machines.

are not provided by the proof in section 5. Therefore, we have chosen to present both of them in a natural order.

References

- Bondareva, O. (1963). Certain applications of the methods of linear programming to the theory of cooperative games (In Russian). *Problemy Kibernetiki*, 10:119–139.
- Calleja, P., Borm, P., Hamers, H., and Klijn, F. (2001). On a new class of parallel sequencing situations and related games. CentER Discussion Paper 2001-03, Tilburg University, Tilburg, The Netherlands.
- Calleja, P., Borm, P., Hamers, H., Klijn, F., and 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., and Klijn, F. (2002). Sequencing games: a survey. In Borm, P. and Peters, H., editors, *Chapters in Game Theory (In Honor of Stef Tijs)*, pages 27–50. Kluwer Academic Publishers, Boston.
- Curiel, I., Pederzoli, G., and Tijs, S. (1989). Sequencing games. *European Journal of Operational Research*, 40:344–351.
- Curiel, I., Potters, J., Rajendra Prasad, V., Tijs, S., and Veltman, B. (1994). Sequencing and cooperation. *Operations Research*, 42:566–568.
- Hamers, H., Klijn, F., and Suijs, J. (1999). On the balancedness of m -sequencing games. *European Journal of Operational Research*, 119:678–691.
- Hamers, H. and Slikker, M. (1995). The PEGS-rule for probabilistic sequencing situations. *Cahiers du Centre d'Études de Recherche Opérationnelle*, 37:109–119.
- Pritsker, A., Miller, L., and Zinkl, R. (1971). Sequencing n products involving m independent jobs on m machines. *A.I.I.E. Transactions*, 3:49–60.
- Shapley, L. (1967). On balanced sets and cores. *Naval Research Logistics Quarterly*, 14:453–460.
- Slikker, M. (2002). An m -sequencing game with an empty core. Working paper, Technische Universiteit Eindhoven, Eindhoven, The Netherlands.
- van den Nouweland, A., Krabbenborg, M., and Potters, J. (1992). Flow-shops with a dominant machine. *European Journal of Operational Research*, 62:38–46.