Tilburg University

Batch Sequencing and Cooperation

Ciftci, B.B.; Borm, P.E.M.; Hamers, H.J.M.; Slikker, M.

Publication date:
2008

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Ciftci, B. B., Borm, P. E. M., Hamers, H. J. M., & Slikker, M. (2008). *Batch Sequencing and Cooperation*. (CentER Discussion Paper; Vol. 2008-100). Operations research.

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

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

No. 2008–100

BATCH SEQUENCING AND COOPERATION

By Barış Çiftçi, Peter Borm, Herbert Hamers, Marco Slikker

December 2008

ISSN 0924-7815

Batch Sequencing and Cooperation

Barış Çiftçi^{*†}, Peter Borm[†], Herbert Hamers[‡], Marco Slikker[‡]

Abstract

Game theoretic analysis of sequencing situations has been restricted to manufacturing systems which consist of machines that can process only one job at a time. However, in many manufacturing systems, operations are carried out by batch machines which can simultaneously process multiple jobs. This paper aims to extend the game theoretical approach to the cost allocation problems arising from sequencing situations on systems that consist of batch machines. We first consider sequencing situations with a single batch machine and analyze cooperative games arising from these situations. It is shown that these games are convex and an expression for the Shapley value of these games is provided. We also introduce an equal gain splitting rule for these sequencing situations and provide an axiomatic characterization. Second, we analyze various aspects of flow-shop sequencing situations which consist of batch machines only. In particular, we provide two cases in which the cooperative game arising from the flow-shop sequencing situation is equal to the game arising from a sequencing situation that corresponds to one specific machine in the flow-shop.

Keywords: Sequencing situations, sequencing games, batch machines.

JEL code: C71.

^{*}Corresponding author. Phone: +31 13 466 2051. Fax: +31 13 466 3280.

[†]CentER and Department of Econometrics and OR, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands.

[‡]Department of Technology Management, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

E-mail addresses: P.E.M.Borm@uvt.nl (P. Borm), B.B.Ciftci@uvt.nl (B. Ciftci), H.J.M.Hamers@uvt.nl (H. Hamers), M.Slikker@tue.nl (M. Slikker).

1 Introduction

Game theoretic analysis of the cost allocation problems arising from sequencing situations is initiated by Curiel, Pederzoli and Tijs (1989). This study considered one machine sequencing situations in which a finite number of agents, each having one job, are queued in front of a machine waiting for their jobs to be processed. Agents have linear cost functions and each group of agents is allowed to obtain cost savings by reordering their jobs. The problem of the distribution of the maximal cost savings is tackled by analyzing corresponding cooperative sequencing games. It was shown that these games are convex and hence balanced. Curiel et al. (1989) also introduced the equal gain splitting ($\mathcal{EGS}$) rule for these one machine sequencing situations and provided an axiomatic characterization.

The following studies in this strand of literature have extended the basic model by considering ready times (Hamers et al., 1995), due dates (Borm et al., 2002), precedence relations (Hamers et al., 2005) and controllable processing times (van Velzen, 2006). In each of these papers, the convexity of the corresponding class of games or of some special subclasses is established. Curiel et al. (1993) considered a larger class of sequencing situations by allowing more general cost functions for the agents. It was shown that these games are not convex in general but core elements do exist. The β -rule was proposed as an extension of the $\mathcal{EGS}$ rule. This rule was shown to yield outcomes in the core of the corresponding games. Other papers have investigated multiple-machine sequencing situations. Van den Nouweland et al. (1992) considered sequencing situations in flow-shops while Hamers et al. (1999) and Slikker (2005, 2006) studied sequencing situations with multiple parallel machines.

The manufacturing systems considered in all studies above consist only of machines which can process no more than one job at a time. Although these models are realistic for many existing manufacturing systems, there are also various systems which include batch machines: machines that can simultaneously process multiple jobs (a batch) subject to the capacity of the machine. Transportation of the semi-finished jobs from one machine to another or the delivery of the finished jobs to the customers/warehouses (cf. Lee and Chen, 2001) constitute very common examples of batch machines in manufacturing systems since transporters, i.e., the machines in these operations usually carry a batch of jobs at the same time. Other well-known examples include heat-treat ovens which can process multiple jobs with the same processing requirement (temperature, processing time etc.) simultaneously in a batch (cf. Lee et al., 1992) and also numerically controlled (NC) routers which cut a stack of metal sheets simultaneously during the cutting operation (cf. Ahmadi et al., 1992). We refer to Webster and Baker (1995) and Potts and Kovalyov (2000) for a review of the scheduling literature on batch sequencing.

This paper aims to present a first game theoretical analysis of the cost allocation problems in batch sequencing situations. We first consider sequencing situations with a single batch machine. These situations give rise to the class of so-called batch sequencing games.

It is shown that these games are convex: an agent's marginal contribution does not decrease if he joins a larger coalition. In particular, we show that these games can be written as a non-negative linear combination of unanimity games. This observation also leads to an expression for the Shapley value of these games. We also consider an equal gain splitting rule and provide an axiomatic characterization of this rule by using efficiency, symmetry and consistency axioms along the lines of Suijs et al. (1997) and Gerichhausen and Hamers (2008).

Finally, we consider sequencing situations in flow-shops which consist of a sequence of finitely many batch machines. We show that when each batch machine has the same batch size or when each batch machine has the same batch processing time, the associated cooperative batch sequencing game is equal to the batch sequencing game corresponding one particular batch machine in the flow-shop. Hence, the games corresponding to these two special classes are convex. However, it is also shown that the games corresponding to batch sequencing situations in flow-shops are not convex in general.

The outline of the paper is as follows. Section 2 recalls some basic notions and results from cooperative game theory. Section 3 describes batch sequencing situations with a single batch machine. Section 4 introduces the corresponding batch sequencing games and presents our main results. Section 5 introduces and analyzes flow-shop batch sequencing situations and corresponding games.

2 Preliminaries

A cooperative game is a pair (N, v) , where N is a nonempty, finite set of players and v is a mapping, $v : 2^N \rightarrow \mathbb{R}$ with $v(\emptyset) = 0$. The mapping v assigns to every coalition $S \subset N$ its worth $v(S)$. A game (N, v) is monotonic if $v(T) \geq v(S)$ for every $S, T \in 2^N$ with $S \subset T$ and it is called superadditive if $v(S) + v(T) \leq v(S \cup T)$ for every $S, T \in 2^N$ with $T \cap S = \emptyset$. A game (N, v) is convex if a player's marginal contribution does not decrease if he joins a larger coalition, i.e., $v(T \cup \{i\}) - v(T) \geq v(S \cup \{i\}) - v(S)$ for every $i \in N$ and $S, T \subset N \setminus \{i\}$ with $S \subset T$.

The core $C(v)$ of a game (N, v) is defined as the set of efficient allocations for which no coalition has an incentive to split off from the grand coalition, i.e.,

$$C(v) = \{x \in \mathbb{R}^N \mid \sum_{i \in N} x_i = v(N) \text{ and } \sum_{i \in S} x_i \geq v(S) \text{ for all } S \in 2^N\}.$$

A game with a nonempty core is called balanced. In particular, convex games are balanced.

For any coalition $T \in 2^N \setminus \{\emptyset\}$, the unanimity game u_T is defined by $u_T(S) = 1$ if $T \subset S$ and $u_T(S) = 0$ for all other coalitions S . It is well known that every cooperative game (N, v) can be written as a unique linear combination of unanimity games by

$$v = \sum_{S \subset N} \lambda_S u_S, \tag{1}$$

where $\lambda_S = \sum_{T \subset S} (-1)^{|S|-|T|} v(T)$ for every $S \subset N$ (cf. Shapley, 1953).

The Shapley value (Shapley, 1953) is one of the most important solution concepts in cooperative game theory and has been studied extensively. The Shapley value of a game can be calculated by making use of the decomposition of a cooperative game into unanimity games. More precisely, given a cooperative game (N, v) , the Shapley value Φ assigns to agent $i \in N$

$$\Phi_i(v) = \sum_{S \subset N: i \in S} \frac{\lambda_S}{|S|},$$

for every $i \in N$.

An order on the players is a bijection $\sigma : N \rightarrow \{1, \dots, n\}$. We denote the set of all orders on N by Π_N . A coalition S is called connected with respect to σ if for all $i, j \in S$ and $k \in N$ such that $\sigma(i) < \sigma(k) < \sigma(j)$ it holds that $k \in S$. We denote with $con(\sigma)$ the set of coalitions that are connected with respect to σ . For a coalition S , $S \setminus \sigma$ is the set of σ -components of S , a σ -component of S being a maximally connected subset of S with respect to σ . We call a cooperative game (N, v) σ -component additive if it satisfies the following three conditions:

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

Le Breton et al. (1992) showed that σ -component additive games are balanced.

3 Batch Sequencing Situations

In a batch sequencing situation a finite number of agents, each having one job, are queued in front of a single batch machine, waiting for their jobs to be processed. The set of agents is denoted by $N = \{1, 2, \dots, n\}$. The machine can process one batch of jobs at one time. At most $z \in \mathbb{Z}_{++}$ jobs can be placed in one batch. Each batch is processed in t time units which is independent of the number of jobs placed in the batch. We assume that there is an initial order σ_0 on the agents before the processing of the machine starts. Specifically, $\sigma_0(i) = j$ means that agent i is in position j . For each agent $i \in N$, the costs of spending time in the system is assumed to be linear and the corresponding cost function $c_i : \mathbb{R}_+ \rightarrow \mathbb{R}$ is defined by $c_i(k) = \alpha_i k$ with $\alpha_i > 0$.

A *batch sequencing situation* as described above is denoted by $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ where $\sigma_0 \in \Pi_N$, $\alpha = (\alpha_i)_{i \in N} \in \mathbb{R}_{++}^N$, $z \in \mathbb{Z}_{++}$ and $t \in \mathbb{R}_{++}$.

In a batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$, it can easily be observed that as long as there is a sufficient number of jobs to fill up a batch, it is profitable to run full batches of size z on the machine. Hence, the first z jobs are placed in the first batch to be

processed by the machine, the following z jobs are placed in the second batch and so on. So, if the jobs are processed according to the order σ , then $\left\lceil \frac{\sigma(i)}{z} \right\rceil$ gives the number of the batch that the job of agent i is placed in¹. Hence the completion time $C(\sigma, i)$ of the job of agent i with respect to σ is given by $C(\sigma, i) = \left\lceil \frac{\sigma(i)}{z} \right\rceil t$.

The total costs of all agents if the jobs are processed according to the order σ equal $\sum_{i \in N} \alpha_i C(\sigma, i)$. By reordering the jobs the total costs can be reduced. Since the number of possible orderings of jobs is finite, there exists an order for which total costs are minimized. We call such an order *optimal*. The following proposition establishes the optimality of an HWCF (highest waiting cost first) order: a processing order in which jobs are processed in nonincreasing order of the waiting cost parameters α_i .

Proposition 3.1 *An HWCF order is optimal for every batch sequencing situation.*

Proposition 3.1 is a direct consequence of the independence of the batch processing time from the composition of each batch and can be proved by using a straightforward argument based on adjacent pairwise interchanges. Notice that the optimal order is unique up to reorderings of the jobs in the same batch and up to reorderings of the jobs of the agents with the same cost parameter.

4 Batch Sequencing Games

For a batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$, the costs of a coalition S with respect to a processing order σ equal $\sum_{i \in S} \alpha_i C(\sigma, i)$. We want to determine the maximal cost savings of a coalition S when its members decide to cooperate. For this aim, we have to define which reorderings of the jobs of coalition S are admissible with respect to the initial order. In this paper, we follow the approach of Curiel et al. (1989), which has become an established line of research, and assume that an order σ is *admissible* for S with respect to σ_0 if $P(\sigma, j) = P(\sigma_0, j)$ for all $j \in N \setminus S$, where for every $\tau \in \Pi_N$ the set of predecessors of an agent $j \in N$ with respect to $\tau \in \Pi_N$ is defined as $P(\tau, j) = \{i \in N \mid \tau(i) \leq \tau(j)\}$. The set of admissible reorderings of a coalition S is denoted by $\mathcal{A}(S)$.

The value of a coalition S is defined as the maximum cost savings coalition S can achieve by means of an admissible reordering. Formally, the batch sequencing game (N, v) corresponding to $\Gamma(N)$ is defined by

$$v(S) = \max_{\sigma \in \mathcal{A}(S)} \left\{ \sum_{i \in S} \alpha_i t \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma(i)}{z} \right\rceil \right) \right\}, \quad (2)$$

for every $S \subset N$.

Clearly, batch sequencing games are monotonic and superadditive. Notice that, by definition of an admissible ordering, a coalition S can produce cost savings only by changing positions within σ_0 -components. Hence, the value of a coalition S is equal to the sum of

¹ $\lceil q \rceil$ denotes the smallest integer which is greater than or equal to q for any real number $q \in \mathbb{R}$.

the values of its σ_0 -components, i.e., $v(S) = \sum_{T \in S \setminus \sigma_0} v(T)$. Notice further that one-person coalitions can not generate any cost savings, i.e., $v(\{i\}) = 0$ for every $i \in N$. So, batch sequencing games are σ_0 -component additive and hence they are balanced.

In the following, we will denote by $\sigma_S \in \Pi_N$ an ordering which is attained from σ_0 by reordering the members in each σ_0 -component of a coalition S with respect to the HWCF rule, i.e., $\sigma_S(i) = \sigma_0(i)$ for every $i \in N \setminus S$ and $\sigma_S(i) < \sigma_S(j)$ for every $T \in S \setminus \sigma_0$ and every $i, j \in T$ such that $\alpha_i > \alpha_j$. Clearly, $\sigma_S \in \mathcal{A}(S)$. Moreover, it follows by Proposition 3.1 and the σ_0 -component additivity of batch sequencing games that σ_S is optimal for S , i.e.,

$$v(S) = \sum_{i \in S} \alpha_i t \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_S(i)}{z} \right\rceil \right),$$

for every $S \subset N$.

Example 4.1 Consider the batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ with $N = \{1, 2, \dots, 7\}$. Assume that σ_0 is given by $\sigma_0(i) = i$ for every $i \in N$, $\alpha = (1, 3, 4, 6, 8, 9, 12)$, $z = 2$, and $t = 1$. Consider the coalition $S = \{2, 3, 5, 6, 7\}$. It can easily be observed that $\sigma_S = (1, 3, 2, 4, 7, 6, 5)$. The orders σ_0 and σ_S are depicted in Figure 1.

Figure 1: The orders σ_0 and σ_S in Example 4.1

Hence

$$\begin{aligned} v(S) &= \sum_{i \in S} \alpha_i t \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_S(i)}{z} \right\rceil \right) \\ &= \alpha_2(1 - 2) + \alpha_3(2 - 1) + \alpha_5(3 - 4) + \alpha_6(3 - 3) + \alpha_7(4 - 3) = 5. \end{aligned}$$

◇

In the following, we will show that batch sequencing games can be written as a non-negative linear combination of unanimity games. For this aim, we first need the following notation and two lemmas.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let (N, v) be the corresponding batch sequencing game. For any coalition $T \subset N$, we will denote the member of T which stands in front of the other members of T with respect to σ_0 by $f(T)$ and the member which stands behind the other members of T by $l(T)$, i.e.,

$$f(T) = \operatorname{argmin}_{i \in T} \sigma_0(i) \quad \text{and} \quad l(T) = \operatorname{argmax}_{i \in T} \sigma_0(i).$$

For every agent $i \in N$ and an order $\sigma \in \Pi_N$, the number of the batch that the job of agent i is placed in with respect to σ is denoted by $b_\sigma(i)$, i.e., $b_\sigma(i) = k$ if and only if $\left\lceil \frac{\sigma(i)}{z} \right\rceil = k$. Also we denote by (N, v_{σ_S}) the batch sequencing game corresponding to the batch sequencing situation $(N, \sigma_S, \alpha, z, t)$ for any $S \subset N$.

Lemma 4.1 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and $S, T \subset N$ such that $S \subset T$. Then,*

$$v(T) = v(S) + v_{\sigma_S}(T).$$

Proof.

$$\begin{aligned} v(S) + v_{\sigma_S}(T) &= \sum_{i \in S} \alpha_i \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_S(i)}{z} \right\rceil \right) + \\ &\quad + \left(\sum_{i \in S} \alpha_i \left(\left\lceil \frac{\sigma_S(i)}{z} \right\rceil - \left\lceil \frac{\sigma_T(i)}{z} \right\rceil \right) + \sum_{i \in T \setminus S} \alpha_i \left(\left\lceil \frac{\sigma_S(i)}{z} \right\rceil - \left\lceil \frac{\sigma_T(i)}{z} \right\rceil \right) \right) \\ &= \sum_{i \in T} \alpha_i \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_T(i)}{z} \right\rceil \right) \\ &= v(T), \end{aligned}$$

where the second equality follows from the fact that $\sigma_0(i) = \sigma_S(i)$ for every $i \in T \setminus S$. $\square$

We will also make use of the following lemma, proved by Borm et al. (2002), which characterizes the coefficients in the unique linear decomposition of a σ_0 -component additive game into unanimity games.

Lemma 4.2 (Borm et al., 2002, Proposition 1) *Let (N, v) be a σ_0 -component additive game and let $\sum_{S \subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games. Then, for every coalition $S \subset N$*

$$\lambda_S = \begin{cases} v(S) - v(S \setminus \{f(S)\}) - v(S \setminus \{l(S)\}) + v(S \setminus \{f(S), l(S)\}), & \text{if } S \in \text{con}(\sigma_0), \\ 0, & \text{otherwise.} \end{cases}$$

In the following proposition, we show that in case of batch sequencing games the formula provided by Lemma 4.2 boils down to an expression in terms of the differences between certain players' waiting cost parameters. More specifically, Proposition 4.1 reveals that the coefficient of a connected coalition S is equal to the sum of differences between the weight of the last player of a batch with respect to σ_S and the weight of the first player of the subsequent batch with respect to σ_S , where the sum is taken over all batches that are crossed by both the first and the last players of S with respect to σ_0 when the order changes from σ_0 to σ_S .

Proposition 4.1 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation. Let (N, v) be the corresponding batch sequencing game and let $\sum_{S \subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games. Then, for every $S \in \text{con}(\sigma_0)$*

$$\lambda_S = \begin{cases} \sum_{k: b_{\sigma_S}(l(S)) \leq k < b_{\sigma_S}(f(S))} t(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)}), & \text{if } b_{\sigma_S}(l(S)) < b_{\sigma_S}(f(S)), \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Let $S \subset N$ be a connected coalition with respect to σ_0 . Let us denote $f(S)$ by i , $l(S)$ by j and $S \setminus \{f(S), l(S)\}$ by S' . By using Lemmas 4.1 and 4.2, we obtain

$$\begin{aligned}\lambda_S &= v(S) - v(S \setminus \{i\}) - v(S \setminus \{j\}) + v(S'), \\ &= (v_{\sigma_{S \setminus \{i\}}}(S) + v(S \setminus \{i\})) - v(S \setminus \{i\}) - (v_{\sigma_{S'}}(S \setminus \{j\}) + v(S')) + v(S'), \\ &= v_{\sigma_{S \setminus \{i\}}}(S) - v_{\sigma_{S'}}(S \setminus \{j\}).\end{aligned}$$

Moreover, since $S \setminus \{i\}$ is already ordered optimally in $\sigma_{S \setminus \{i\}}$,

$$v_{\sigma_{S \setminus \{i\}}}(S) = \sum_{k: b_{\sigma_0}(i) \leq k < b_{\sigma_S}(i)} t(\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} - \alpha_i).$$

Similarly,

$$v_{\sigma_{S'}}(S \setminus \{j\}) = \sum_{k: b_{\sigma_0}(i) \leq k < b_{\sigma_{S \setminus \{j\}}}(i)} t(\alpha_{\sigma_{S'}^{-1}(kz+1)} - \alpha_i).$$

Hence,

$$\lambda_S = \sum_{k: b_{\sigma_0}(i) \leq k < b_{\sigma_S}(i)} t(\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} - \alpha_i) - \sum_{k: b_{\sigma_0}(i) \leq k < b_{\sigma_{S \setminus \{j\}}}(i)} t(\alpha_{\sigma_{S'}^{-1}(kz+1)} - \alpha_i). \quad (3)$$

Observe also that $\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} = \alpha_{\sigma_{S'}^{-1}(kz+1)}$ for every k such that $b_{\sigma_0}(i) \leq k < b_{\sigma_S}(j)$. So, equation (3) can be rewritten as

$$\lambda_S = \sum_{k: b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)} t(\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} - \alpha_i) - \sum_{k: b_{\sigma_S}(j) \leq k < b_{\sigma_{S \setminus \{j\}}}(i)} t(\alpha_{\sigma_{S'}^{-1}(kz+1)} - \alpha_i). \quad (4)$$

It immediately follows from equation (4) that $\lambda_S = 0$ when $b_{\sigma_S}(j) \geq b_{\sigma_S}(i)$. So, assume that $b_{\sigma_S}(j) < b_{\sigma_S}(i)$. Observe that $b_{\sigma_S}(i)$ is either equal to $b_{\sigma_{S \setminus \{j\}}}(i)$ or equal to $b_{\sigma_{S \setminus \{j\}}}(i) + 1$.

First, let $b_{\sigma_S}(i) = b_{\sigma_{S \setminus \{j\}}}(i)$. Then, equation (4) can be rewritten as

$$\begin{aligned}\lambda_S &= \sum_{k: b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)} t(\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} - \alpha_{\sigma_{S'}^{-1}(kz+1)}) \\ &= \sum_{k: b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)} t(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)}),\end{aligned}$$

where the second equality follows from the fact that $\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} = \alpha_{\sigma_S^{-1}(kz)}$ and $\alpha_{\sigma_{S'}^{-1}(kz+1)} = \alpha_{\sigma_S^{-1}(kz+1)}$ for every k such that $b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)$.

Next let $b_{\sigma_S}(i) = b_{\sigma_{S \setminus \{j\}}}(i) + 1$. Denote $b_{\sigma_{S \setminus \{j\}}}(i)$ by $\bar{k}$. Then, obviously, $\alpha_i = \alpha_{\sigma_S^{-1}(\bar{k}z+1)}$. By equation (4)

$$\begin{aligned}\lambda_S &= \sum_{k: b_{\sigma_S}(j) \leq k < \bar{k}} t(\alpha_{\sigma_{S \setminus \{i\}}^{-1}(kz+1)} - \alpha_{\sigma_{S'}^{-1}(kz+1)}) + t(\alpha_{\sigma_{S \setminus \{i\}}^{-1}(\bar{k}z+1)} - \alpha_i), \\ &= \sum_{k: b_{\sigma_S}(j) \leq k < \bar{k}} t(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)}) + t(\alpha_{\sigma_S^{-1}(\bar{k}z)} - \alpha_{\sigma_S^{-1}(\bar{k}z+1)}), \\ &= \sum_{k: b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)} t(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)}).\end{aligned}$$

Hence, we can conclude that $\lambda_S = \sum_{k: b_{\sigma_S}(j) \leq k < b_{\sigma_S}(i)} t (\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)})$ when $b_{\sigma_S}(j) < b_{\sigma_S}(i)$. $\square$

We illustrate the expression provided by Proposition 4.1 in the following example.

Example 4.2 Take the batch sequencing situation $\Gamma(N)$ considered in Example 4.1. Let (N, v) be the batch sequencing game corresponding to $\Gamma(N)$ and consider the coalition $S = \{1, 2, \dots, 6\}$. We will calculate λ_S , the coefficient corresponding to S in the decomposition of (N, v) into unanimity games. Obviously, $f(S) = 1$, $l(S) = 6$ and $\sigma_S = (6, 5, 4, 3, 2, 1, 7)$ (See Figure 2).

Figure 2: The orders σ_0 and σ_S in Example 4.2

Observe that the job of agent 1 is processed in the third batch and the job of agent 6 is processed in the first batch with respect to σ_S . That is $b_{\sigma_S}(l(S)) = 1$ and $b_{\sigma_S}(f(S)) = 3$. Since $b_{\sigma_S}(l(S)) < b_{\sigma_S}(f(S))$, by Proposition 4.1

$$\begin{aligned} \lambda_S &= (\alpha_{\sigma_S^{-1}(2)} - \alpha_{\sigma_S^{-1}(3)}) + (\alpha_{\sigma_S^{-1}(4)} - \alpha_{\sigma_S^{-1}(5)}) \\ &= (\alpha_5 - \alpha_4) + (\alpha_3 - \alpha_2) = (8 - 6) + (4 - 3) = 3. \end{aligned}$$

The whole decomposition of (N, v) into unanimity games is provided below:

$$\begin{aligned} v &= u_{\{2,3\}} + 2u_{\{4,5\}} + 3u_{\{6,7\}} + 2u_{\{1,2,3\}} + 2u_{\{2,3,4\}} + 2u_{\{3,4,5\}} + u_{\{4,5,6\}} + u_{\{5,6,7\}} \\ &\quad + u_{\{1,2,3,4\}} + 3u_{\{2,3,4,5\}} + 2u_{\{3,4,5,6\}} + 5u_{\{4,5,6,7\}} + 4u_{N \setminus \{6,7\}} + 3u_{N \setminus \{1,7\}} \\ &\quad + 3u_{N \setminus \{1,2\}} + 3u_{N \setminus \{7\}} + 6u_{N \setminus \{1\}} + 5u_N. \end{aligned}$$

$\diamond$

Proposition 4.1 reveals that batch sequencing games are nonnegative combinations of unanimity games. Since unanimity games are convex, this establishes the convexity of the batch sequencing games. As a side result, we also obtain an expression of the Shapley value of batch sequencing games.

Theorem 4.1 Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let (N, v) be the corresponding batch sequencing game. Then,

- (i) (N, v) is convex.

(ii) For all $i \in N$

$$\Phi_i(v) = \sum_{S \in \text{con}(\sigma_0), i \in S} \sum_{k: b_{\sigma_S}(l(S)) \leq k < b_{\sigma_S}(f(S))} \frac{t(\alpha_{\sigma_S^{-1}(kz)} - \alpha_{\sigma_S^{-1}(kz+1)})}{|S|}.$$

In the remainder of this section, we introduce and characterize a non-aggregated equal gain splitting ($\mathcal{EGS}$) solution for batch sequencing situations. Non-aggregated solutions, which are first introduced by Suijs et al. (1997) for classical sequencing situations, can be considered as a specification of all components of the total reward an agent obtains. In our setting, a non-aggregated solution Ψ is a map assigning to each batch sequencing situation $\Gamma(N)$ a matrix $W \in \mathbb{R}_+^{N \times N}$, where an element w_{ij} of W represents the nonnegative gain assigned to agent i for cooperating with agent j . The aggregated solution corresponding to $W \in \Psi(\Gamma(N))$ can be found by multiplying W with the vector e^N of all ones in $\mathbb{R}^N$.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and (N, v) be the corresponding batch sequencing game. Also let $\sum_{S \subset N} \lambda_S u_S$ be the linear decomposition of (N, v) into unanimity games. For any $i, j \in N$, let $[i, j]$ be the set of all players in between i and j with respect to σ_0 , i.e., $[i, j] = \{k \in N \mid \min\{\sigma_0(i), \sigma_0(j)\} \leq \sigma_0(k) \leq \max\{\sigma_0(i), \sigma_0(j)\}\}$. The non-aggregated equal gain splitting solution $\mathcal{EGS}$ assigns to each batch sequencing situation $\Gamma(N)$ a solution $\mathcal{EGS}(\Gamma(N)) \in \mathbb{R}_+^{N \times N}$ such that

$$\mathcal{EGS}(\Gamma(N))_{ij} = \frac{\lambda_{[i,j]}}{2}$$

for every $i, j \in N$.

Example 4.3 Consider the batch sequencing situation of Example 4.1. Recall that the decomposition of the corresponding batch sequencing game into unanimity games is provided in Example 4.2. By making use of this decomposition, the equal gain splitting solution $\mathcal{EGS}(\Gamma(N))$ equals to

$$\mathcal{EGS}(\Gamma(N)) = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} & 2 & \frac{3}{2} & \frac{5}{2} \\ 0 & 0 & \frac{1}{2} & 1 & \frac{3}{2} & \frac{3}{2} & 3 \\ 1 & \frac{1}{2} & 0 & 0 & 1 & 1 & \frac{3}{2} \\ \frac{1}{2} & 1 & 0 & 0 & 1 & \frac{1}{2} & \frac{5}{2} \\ 2 & \frac{3}{2} & 1 & 1 & 0 & 0 & \frac{1}{2} \\ \frac{3}{2} & \frac{3}{2} & 1 & \frac{1}{2} & 0 & 0 & \frac{3}{2} \\ \frac{5}{2} & 3 & \frac{3}{2} & \frac{5}{2} & \frac{1}{2} & \frac{3}{2} & 0 \end{bmatrix}$$

◇

Now we introduce the notions of dummy agents and reduced batch sequencing situations that will be used for the axiomatization of the $\mathcal{EGS}$ solution for batch sequencing situations.

Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation. An agent $i \in N$ is called a *dummy agent* in $\Gamma(N)$ if $b(i) = b_{\sigma_S}(i)$ for every $S \subset N$ with $i \in S$, i.e., the job of agent

i stays in its initial batch no matter which coalition of agents i cooperates with. Roughly speaking, a batch sequencing situation reduced to a connected coalition $S \in \text{con}(\sigma_0)$ is a batch sequencing situation obtained when the agents outside S are replaced with dummy agents. Formally, a *reduced batch sequencing situation with respect to S* is described by $\Gamma|_S(N) = (N, \sigma_0, \beta, z, t)$ where $\beta = (\beta_i)_{i \in N} \in \mathbb{R}_+^N$ is such that $\beta_i = \alpha_i$ for every $i \in S$ and

$$\beta_j = \begin{cases} \max_{i \in N} 2\alpha_i, & \text{if } j < f(S), \\ \min_{i \in N} \alpha_i/2, & \text{if } j > l(S), \end{cases}$$

for every agent $j \in N \setminus S$.

Let Ψ be a non-aggregated solution for batch sequencing situations. We consider the following three properties of Ψ .

- **Efficiency:** Ψ is efficient if

$$\sum_{i,j \in N} \Psi(\Gamma(N))_{ij} = \sum_{i \in N} \alpha_i (C(\sigma_0, i) - C(\sigma_N, i)) \text{ for all batch sequencing situations } \Gamma(N).$$

- **Symmetry:** Ψ is symmetric if $\Psi(\Gamma(N))$ is a symmetric matrix for all batch sequencing situations $\Gamma(N)$.
- **Consistency:** Ψ is consistent if for all batch sequencing situations $\Gamma(N)$ and all $S \in \text{con}(\sigma_0)$ the following is satisfied:

$$\Psi(\Gamma(N))_{ij} = \Psi(\Gamma|_S(N))_{ij} \text{ for every } i, j \in S.$$

The efficiency axiom states that the total amount allocated to the agents is equal to the maximal total cost savings that the agents can jointly obtain. Symmetry states that the (extra) gain two agents can obtain is equally divided among the two agents. Consistency states that connected coalitions obtain the same division if they renegotiate on the basis of the same solution concept to the reduced situation with outside dummies.

In Theorem 4.2 we will characterize the $\mathcal{EGS}$ solution with the three properties mentioned above. For the proof, we need the following lemma which states that, for non-aggregate solutions that satisfy both efficiency and consistency, the total amount allocated to connected coalitions must be equal to the maximal total cost savings that these coalitions can achieve.

Lemma 4.3 *Let $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$ be a batch sequencing situation and let Ψ be a non-aggregated solution for batch sequencing situations. If Ψ is both efficient and consistent, then*

$$\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} = \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)),$$

for every $S \in \text{con}(\sigma_0)$.

Proof. Suppose first that there exists $S \in \text{con}(\sigma_0)$ such that

$$\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} > \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)).$$

Now on the one hand efficiency of Ψ implies that

$$\sum_{i,j \in N} \Psi(\Gamma_{|S}(N))_{ij} = \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)),$$

since all agents outside S are dummy agents in $\Gamma_{|S}(N)$. On the other hand, consistency of Ψ implies that

$$\begin{aligned} \sum_{i,j \in S} \Psi(\Gamma_{|S}(N))_{ij} &= \sum_{i,j \in S} \Psi(\Gamma(N))_{ij} \\ &> \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)) = \sum_{i,j \in N} \Psi(\Gamma_{|S}(N))_{ij}. \end{aligned}$$

A contradiction, because $\sum_{i,j \in N} \Psi(\Gamma_{|S}(N))_{ij} \geq \sum_{i,j \in S} \Psi(\Gamma_{|S}(N))_{ij}$ since $\Psi(\Gamma_{|S}(N))_{ij} \geq 0$ for every $i, j \in N$.

Now suppose that there exists $S \in \text{con}(\sigma_0)$ such that

$$\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} < \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)).$$

Clearly, $S \neq N$ and there exists $S_1, S_2 \in \text{con}(\sigma_0) \cup \{\emptyset\}$ such that $S_1 \cap S_2 = \emptyset$ and $N \setminus S = S_1 \cup S_2$. Observe that

$$\sum_{i,j \in N} \Psi(\Gamma(N))_{ij} = \sum_{i \in N} \alpha_i (C(\sigma_0, i) - C(\sigma_N, i)) \geq \sum_{T \in \{S, S_1, S_2\}} \sum_{i \in T} \alpha_i (C(\sigma_0, i) - C(\sigma_T, i)),$$

where the equality follows from Ψ being efficient and the inequality follows from the fact that $N = S \cup S_1 \cup S_2$ and $S \cap S_1 = S \cap S_2 = S_1 \cap S_2 = \emptyset$.

Then, since $\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} < \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i))$, we have either

$$\sum_{i,j \in S_1} \Psi(\Gamma(N))_{ij} > \sum_{i \in S_1} \alpha_i (C(\sigma_0, i) - C(\sigma_{S_1}, i))$$

or

$$\sum_{i,j \in S_2} \Psi(\Gamma(N))_{ij} > \sum_{i \in S_2} \alpha_i (C(\sigma_0, i) - C(\sigma_{S_2}, i)).$$

However, as we have already shown, the existence of a coalition $S \in \text{con}(\sigma_0)$ with $\sum_{i,j \in S} \Psi(\Gamma(N))_{ij} > \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i))$ leads to a contradiction. $\square$

Theorem 4.2 *The $\mathcal{EGS}$ solution is the unique non-aggregated solution satisfying efficiency, symmetry and consistency.*

Proof. Obviously, $\mathcal{EGS}$ satisfies efficiency, symmetry and consistency.

Now let Ψ be a nonempty solution which satisfies efficiency, symmetry and consistency. Pick a batch sequencing situation $\Gamma(N) = (N, \sigma_0, \alpha, z, t)$. We will show that $\Psi(\Gamma(N))_{ij}$ is uniquely determined for every $i, j \in N$ with induction to the number $|\sigma_0(j) - \sigma_0(i)|$.

Pick $i, j \in N$. Assume first that $|\sigma_0(j) - \sigma_0(i)| = 0$. Then, $i = j$ and $\Psi(\Gamma(N))_{ij} = 0 = \mathcal{EGS}(\Gamma(N))_{ij}$ by Lemma 4.3. Now, assume that $\Psi(\Gamma(N))_{ij} = \mathcal{EGS}(\Gamma(N))_{ij}$ for every $i, j \in N$ with $|\sigma_0(j) - \sigma_0(i)| \leq k$ for some $k \in \{0, \dots, n-2\}$. Pick $i, j \in N$ such that $|\sigma_0(j) - \sigma_0(i)| = k+1$. Let $S = [i, j]$. Then,

$$\begin{aligned} \Psi(\Gamma(N))_{ij} + \Psi(\Gamma(N))_{ji} &= \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)) - \sum_{(p,r) \in S \times S \setminus \{(i,j), (j,i)\}} \Psi(\Gamma(N))_{pr}, \\ &= \sum_{i \in S} \alpha_i (C(\sigma_0, i) - C(\sigma_S, i)) - \sum_{(p,r) \in S \times S \setminus \{(i,j), (j,i)\}} \mathcal{EGS}(\Gamma(N))_{pr}, \end{aligned}$$

where the first equality follows from Lemma 4.3 and the second equality follows from the fact that $|\sigma_0(p) - \sigma_0(r)| \leq k$ for every $(p, r) \in S \times S \setminus \{(i, j), (j, i)\}$ and the induction assumption.

Then $\Psi(\Gamma(N))_{ij} + \Psi(\Gamma(N))_{ji}$ is determined uniquely. Consequently, by symmetry, Ψ is uniquely determined and hence $\Psi(\Gamma(N)) = \mathcal{EGS}(\Gamma(N))$. $\square$

5 Flow-Shops with Batch Machines

Flow-shop batch sequencing (FSBS) situations consist of a sequence of finitely many batch machines $B_1, B_2, \dots, B_m$ and a finite number of agents $N = \{1, 2, \dots, n\}$ each having one job to be processed in the order $B_1, B_2, \dots, B_m$. Each batch machine B_k has a batch size of z_k and processes a batch in t_k time units independent of the number of jobs placed in the batch. As it is the case in batching situations on a single batch machine, we assume that there is an initial order σ_0 on the jobs before the processing of the jobs on the flow-shop begins. That is if agent i is in front of agent j in the queue, then at all machines in the flow-shop, the job of agent i has to be processed before the job of agent j is processed or together with the job of agent j in the same batch. An FSBS situation as described above is denoted by $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ where $M = \{1, 2, \dots, m\}$ and $m \in \mathbb{Z}_{++}$ is the number of batch machines in the flow-shop, $\sigma_0 \in \Pi_N$, $\alpha = (\alpha_i)_{i \in N} \in \mathbb{R}_{++}^N$, $z \in \mathbb{Z}_{++}^M$ and $t \in \mathbb{R}_{++}^M$.

For every FSBS situation, a *production schedule* τ fixes for every agent i and for every machine k a starting time $T_{i,k}(\tau)$ of the job of agent i at machine k . A production schedule is feasible if it conforms to the batch capacity constraints of the machines, to the order of the flow-shop and to the order on the jobs. Formally, we call a production schedule *feasible* with respect to the order $\sigma \in \Pi_N$ if it satisfies the following:

- (i) $T_{i,k}(\tau) \geq 0$ for all $i \in N$ and all $k \in M$.

- (ii) $T_{i,k}(\tau) + t_k \leq T_{i,k+1}(\tau)$ for all $i \in N$ and all $k \in M$.
- (iii) If $\sigma(i) \leq \sigma(j)$, then $T_{i,k}(\tau) \leq T_{j,k}(\tau)$ for all $i, j \in N$ and all $k \in M$.
- (iv) $|\{i \in N | T_{i,k}(\tau) = s\}| \leq z_k$ for all $k \in M$ and $s \geq 0$.
- (v) If $T_{i,k}(\tau) \neq T_{j,k}(\tau)$, then $|T_{i,k}(\tau) - T_{j,k}(\tau)| \geq t_k$ for all $i, j \in N$ and all $k \in M$.

We denote by $F_{i,k}(\tau)$ the time at which machine k finishes the processing of the batch in which job i is placed, i.e., $F_{i,k}(\tau) = T_{i,k}(\tau) + t_k$. Then, the completion time $C_i(\tau)$ of job i under production schedule τ is $F_{i,m}(\tau)$. The total costs of all agents if the jobs are processed according to the production schedule τ equal $\sum_{i \in N} \alpha_i F_{i,m}(\tau)$.

Example 5.1 Consider an FSBS $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ where $N = \{1, \dots, 5\}$, $M = \{1, 2\}$, $\sigma_0(i) = i$ for every $i \in N$, $\alpha = (1, 7, 1, 1, 1)$, $z = (1, 2)$ and $t = (1, 5)$. In Figure 3, we depict two feasible production schedules with respect to the initial order σ_0 : τ_0 and τ_1 .

Figure 3: Two feasible schedules for the flow-shop situation in Example 5.1

It can easily be observed that with respect to both schedules, every job is immediately processed by B_1 as soon as B_1 is available. If the jobs are processed with respect to production schedule τ_0 , then $F_{1,1}(\tau_0) = T_{1,2}(\tau_0) = 1$. That is job 1's processing at B_2 starts as soon as its processing at B_1 ends. If the jobs are processed with respect to production schedule τ_1 , then $F_{1,1}(\tau_1) = 1$ and $T_{1,2}(\tau_1) = T_{2,2}(\tau_1) = 2$. That is, with respect to production schedule τ_1 , although job 1 is available for processing at B_2 at time one, it waits for one time unit for job 2 to become available for B_2 and at time two their processing by B_2 starts together in the same batch. $\diamond$

Finding an optimal schedule for a general FSBS situation is still an open problem². However, as we establish in the following proposition, there always exists an optimal schedule

²For an FSBS situation with two machines and equal waiting time parameters, Ahmadi et al. (1992) showed that the problem of finding the optimal schedule can be solved within $O(N^3)$ operations.

which processes the jobs according to the HWCF-sequence σ_N and runs full batches in the first batch machine.

Proposition 5.1 *For every FSBS situation $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$, there exists an optimal schedule τ such that*

- (i) τ is a feasible schedule with respect to σ_N .
- (ii) τ runs full batches of size z_1 in the first batch machine.

We note that Proposition 5.1 can be proved by using a simple interchange argument.

Observe that there can be more than one production schedule which is feasible with respect to the initial order in an FSBS situation. We assume in this study that, in the initial case, i.e., when the agents do not cooperate, the corresponding *initial production schedule* is the one in which no agent waits for another because waiting for other jobs, without any further compensation, to get processed together in the same batch will only increase the costs of the agent. That is the initial production schedule is the one which is feasible with respect to σ_0 and satisfies the condition that every job which is ready to be processed at a machine is processed as soon as the machine is also available. We denote the initial production schedule in flow-shop batch sequencing situations by τ_0 (Note that the production schedule τ_0 in Example 5.1 indeed fits this description.).

Next, for a coalition of cooperating agents, we must decide on which production schedule rearrangements are admissible. We assume that a coalition S can choose any production schedule which is feasible with respect to an order $\sigma \in \mathcal{A}(S)$ as long as they do not harm the players outside S . Formally, a production schedule τ is admissible for coalition S if it satisfies the following two conditions:

- (i) τ is a feasible production schedule with respect to an order in $\mathcal{A}(S)$.
- (ii) $F_{i,m}(\tau) \leq F_{i,m}(\tau_0)$ for all $i \in N \setminus S$.³

The set of admissible production schedules for S is denoted by $\mathcal{APS}(S)$.

We define the value of a coalition S as the maximum cost savings coalition S can achieve by means of an admissible production schedule. Formally, a flow-shop batch sequencing game (N, w) corresponding to a flow-shop batch sequencing situation $\Gamma(N, M)$ is defined by

$$w(S) = \max_{\tau \in \mathcal{APS}(S)} \sum_{i \in S} \alpha_i (F_{i,m}(\tau_0) - F_{i,m}(\tau)), \quad (5)$$

³Condition (ii) enables agents to wait for other agents' jobs in order to get processed together in the same batch. That is agents can create savings both by reordering their jobs and also by waiting for some other jobs. Another option for admissibility of a rearrangement could be to require that $F_{i,m}(\tau) = F_{i,m}(\tau_0)$ for all $i \in N \setminus S$. Under this more restrictive condition, waiting would no longer be possible in a flow-shop batching situation, i.e., the agents could create savings only by reordering their jobs. We want to remark that, under this more restrictive condition, the results obtained for single machine batch sequencing games can easily be extended to corresponding flow-shop batch sequencing games.

for every $S \subset N$.

Using a simple interchange argument, it can be shown that there exists an optimal admissible production schedule for S which is feasible with respect to σ_S . Recall that σ_S is an ordering which is attained from σ_0 by reordering the members in each σ_0 -component of a coalition S with respect to the HWCF rule.

Clearly, FSBS games are monotonic and superadditive. However, as illustrated by the following example, FSBS games are neither σ_0 -component additive nor convex in general.

Example 5.2 Consider the flow-shop batch sequencing situation $\Gamma(N, M)$ given in Example 5.1 and the coalition $S = \{1, 3, 4, 5\}$. The initial production schedule, τ_0 is also given in Example 5.1. Notice that S can not create savings just by reordering the jobs since $\alpha_i = \alpha_j$ for every $i, j \in S$. However, observe that τ_1 in Example 5.1 is an admissible production schedule for S (since $F_{2,2}(\tau_1) < F_{2,2}(\tau_0)$) and the cost savings obtained if S uses τ_1 is:

$$\begin{aligned} \sum_{i \in S} \alpha_i (F_{i,2}(\tau_0) - F_{i,2}(\tau_1)) &= \sum_{i \in S} (F_{i,2}(\tau_0) - F_{i,2}(\tau_1)) \\ &= (6 - 7) + (11 - 12) + (16 - 12) + (16 - 17) = 1. \end{aligned}$$

So, when agent 1 waits for agent 2, the only agent outside S , agent 2 profits from an earlier completion time, agents 3 and 5 are harmed indirectly but agent 4 profits. As a result, S could obtain cost savings of 1. Actually, τ_1 is the optimal production schedule for S , i.e., $w(S) = 1$.

For $T \in 2^N \setminus \{S\}$ one finds that: $w(T) = 30$ for every $T \supset \{1, 2\}$ and $w(T) = 0$ otherwise.

Observe that this FSBS game is not convex:

$$0 = w(N) - w(\{1, 2, 3, 4\}) < w(\{1, 3, 4, 5\}) - w(\{1, 3, 4\}) = 1.$$

(N, w) is not σ_0 -component additive either: $\{1\}$ and $\{3, 4, 5\}$ are the σ_0 -components of $\{1, 3, 4, 5\}$, but

$$1 = w(\{1, 3, 4, 5\}) \neq w(\{1\}) + w(\{3, 4, 5\}) = 0.$$

◇

In the following, we will examine two particular FSBS situations: situations where all batch machines have the same batch size and situations where all batch machines have the same batch processing time. First, it is shown that, in both of these situations, an optimal order for a coalition of agents can be obtained by reordering the jobs of the agents. That is although waiting for other jobs to produce savings is allowed in our model, in these FSBS situations it need not be employed by the coalitions to obtain maximal cost savings. Second, it is shown that the FSBS games arising from these situations are equal to the game arising from the “bottleneck” machine in the flow-shop: the machine with the highest batch processing time when all machines have the same batch capacity and the machine with the minimum batch capacity when all machines have the same batch processing time.

We need the following notation in order to present our results. Consider an *FSBS* situation $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$. We denote by P_k the sum of the first k machines' batch processing times, by $\bar{t}$ the maximum batch processing time and by $\bar{z}$ the minimum batch size, i.e., $P_k = \sum_{p=1}^k t_p$, $\bar{t} = \max\{t_k | k \in M\}$ and $\bar{z} = \min\{z_k | k \in M\}$. Lastly, we denote by τ_σ the production schedule which is obtained from τ_0 only by reordering the agents with respect to σ . That is

$$T_{\sigma^{-1}(p),k}(\tau_\sigma) = T_{\sigma_0^{-1}(p),k}(\tau_0)$$

for every $p \in \{1, 2, \dots, n\}$ and $k \in M$.

Proposition 5.2 *Let $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ be an *FSBS* situation. If $z_k = z$ for every $k \in M$ or if $t_k = t$ for every $k \in M$, then τ_{σ_S} is an optimal production schedule for every $S \subset N$.*

Proof. First assume that $z_k = z$ for every $k \in M$ and consider the initial production schedule τ_0 . Observe that with respect to τ_0 first z jobs in the initial order σ_0 are processed together in the first batch in each machine; the second z jobs in σ_0 are processed together in the second batch in each machine and so on. Since the initial production schedule τ_0 runs full batches in each machine in the flow-shop, waiting for other jobs is not an option for the players. We know that there exists an optimal production schedule for every coalition S which is feasible with respect to σ_S . Then, clearly, τ_{σ_S} is an optimal production schedule for any coalition $S \subset N$.

Now assume that $t_k = t$ for every $k \in M$ and consider the initial production schedule τ_0 . Observe that τ_0 may run batches with less jobs than full capacity and the jobs in these batches have the option to wait for other jobs to get processed together in the same batch. However, observe that it is not possible to decrease the completion times of the other jobs by waiting for them, because the batch processing time of each machine is the same. That is the coalitions can not produce savings through waiting for other jobs. We know that there exists an optimal production schedule which is feasible with respect to σ_S for every coalition S . Then, since waiting is not profitable, τ_{σ_S} is an optimal production schedule for any coalition $S \subset N$. $\square$

In Theorem 5.1 we show that in both of these *FSBS* situations the corresponding *FSBS* games are equal to the batch sequencing game corresponding to the bottleneck machine in the flow-shop. For the proof, we need the following lemma which states that in both of these *FSBS* situations the completion time of a job is determined up to a constant by the bottleneck machine.

Lemma 5.1 *Let $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ be an *FSBS* situation.*

- (i) *If $z_k = z$ for every $k \in M$, then $F_{i,m}(\tau_\sigma) = P_m + (\lceil \frac{\sigma(i)}{z} \rceil - 1)\bar{t}$ for every $i \in N$ and every $\sigma \in \Pi_N$.*

(ii) If $t_k = t$ for every $k \in M$, then $F_{i,m}(\tau_\sigma) = mt + (\left\lceil \frac{\sigma(i)}{z} \right\rceil - 1)t$ for every $i \in N$ and every $\sigma \in \Pi_N$.

Proof. (i). Let $\sigma \in \Pi_N$ and let $i \in N$ be such that $\left\lceil \frac{\sigma(i)}{z} \right\rceil = 1$. Clearly, i is processed in the first batch by each machine and $F_{i,k}(\tau_\sigma) = P_k$ for every $k \in M$. Now, pick $j \in N$ with $\left\lceil \frac{\sigma(j)}{z} \right\rceil = 2$, i.e., j is processed in the second batch by each machine. Let $\bar{t}_{[1,k]}$ be the maximum batch processing time among the first k machines, i.e., $\bar{t}_{[1,k]} = \max\{t_l | l \in \{1, 2, \dots, k\}\}$. We will show, by induction on k , that

$$F_{j,k}(\tau_\sigma) = F_{i,k}(\tau_\sigma) + \left(\left\lceil \frac{\sigma(j)}{z} \right\rceil - 1 \right) \bar{t}_{[1,k]} = P_k + \bar{t}_{[1,k]}.$$

When $k = 1$, the assertion holds trivially. So, assume that the assertion holds for every $k < l$. Then, $F_{j,l-1}(\tau_\sigma) = P_{l-1} + \bar{t}_{[1,l-1]}$.

Now, if $t_l < \bar{t}_{[1,l-1]}$, then $\bar{t}_{[1,l]} = \bar{t}_{[1,l-1]}$ and

$$F_{j,l-1}(\tau_\sigma) = P_{l-1} + \bar{t}_{[1,l-1]} > P_{l-1} + t_l = P_l = F_{i,l}(\tau_\sigma),$$

i.e., the second batch's processing at machine $l-1$ finishes after the first batch's processing finishes at machine l . Hence, the second batch immediately starts to be processed by machine l at time $F_{j,l-1}(\tau_\sigma)$. So,

$$F_{j,l}(\tau_\sigma) = F_{j,l-1}(\tau_\sigma) + t_l = P_{l-1} + \bar{t}_{[1,l-1]} + t_l = P_l + \bar{t}_{[1,l]}.$$

If $t_l \geq \bar{t}_{[1,l-1]}$, then $\bar{t}_{[1,l]} = t_l$. Also we have that

$$F_{j,l-1}(\tau_\sigma) = P_{l-1} + \bar{t}_{[1,l-1]} \leq P_{l-1} + t_l = F_{i,l}(\tau_\sigma).$$

That is the second batch's processing at machine $l-1$ finishes before the first batch's processing finishes at machine l . Hence, the second batch starts to be processed by machine l at time $F_{i,l}(\tau_\sigma)$. So, $F_{j,l}(\tau_\sigma) = F_{i,l}(\tau_\sigma) + t_l = P_l + t_l = P_l + \bar{t}_{[1,l]}$.

Now, one can repeat the whole argument given above for a job which is processed in the third batch by each machine, then for the fourth batch and so on to prove that $F_{i,m}(\tau_\sigma) = P_m + (\left\lceil \frac{\sigma(i)}{z} \right\rceil - 1)\bar{t}$ for every $i \in N$.

(ii) can be proven similarly. □

Theorem 5.1 Let $\Gamma(N, M) = (N, M, \sigma_0, \alpha, z, t)$ be an FSBS situation and (N, w) be the corresponding flow-shop batch sequencing game.

- (i) Let $z_k = z$ for every $k \in M$. Define $\Gamma(N) = (N, \sigma_0, \alpha, z, \bar{t})$ to be the batch sequencing situation corresponding to the bottleneck batch machine and let (N, v) be the corresponding batch sequencing game. Then, $v = w$.
- (ii) Let $t_k = t$ for every $k \in M$. Define $\Gamma(N) = (N, \sigma_0, \alpha, \bar{z}, t)$ to be the batch sequencing situation corresponding to the bottleneck batch machine and let (N, v) be the corresponding batch sequencing game. Then, $v = w$.

Proof. (i). We know by Lemma 5.1 that

$$F_{i,m}(\tau_0) = P_m + \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - 1\right)\bar{t} \text{ and } F_{i,m}(\tau_{\sigma_S}) = P_m + \left(\left\lceil \frac{\sigma_S(i)}{z} \right\rceil - 1\right)\bar{t},$$

for every $S \subset N$ and every $i \in N$. Moreover, we know by Proposition 5.2 that in $\Gamma(N, M)$ τ_{σ_S} is an optimal schedule for every $S \subset N$. Then,

$$\begin{aligned} w(S) &= \sum_{i \in S} \alpha_i (F_{i,m}(\tau_0) - F_{i,m}(\tau_{\sigma_S})) \\ &= \sum_{i \in S} \alpha_i \left((P_m + \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - 1\right)\bar{t}) - (P_m + \left(\left\lceil \frac{\sigma_S(i)}{z} \right\rceil - 1\right)\bar{t}) \right) \\ &= \sum_{i \in S} \alpha_i \bar{t} \left(\left\lceil \frac{\sigma_0(i)}{z} \right\rceil - \left\lceil \frac{\sigma_S(i)}{z} \right\rceil \right) = v(S), \end{aligned}$$

for every $S \subset N$.

(ii) can be proven similarly. □

References

- [1] Ahmadi, J. H., Ahmadi, R. H., Dasu, S., Tang, C. S., (1992). Batching and scheduling jobs on batch and discrete processors. *Operations Research*, 40: 750-763.
- [2] 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.
- [3] Curiel, I., Pederzoli, G., Tijs, S., (1989). Sequencing games. *European Journal of Operational Research*, 40: 344-351.
- [4] Curiel, I., Potters, J., Prasad, R., Tijs, S., Veltman, B., (1993). Cooperation in one machine scheduling. *Mathematical Methods of Operations Research*, 38: 113-129.
- [5] Gerichhausen, M., Hamers, H., (2008). Partitioning sequencing situations and games. *European Journal of Operational Research*. (in press).
- [6] Hamers, H., Borm, P., Tijs, S., (1995). On games corresponding to sequencing situations with ready times. *Mathematical Programming*, 70: 1-13.
- [7] Hamers, H., Klijn, F., Suijs, J., (1999). On the balancedness of multimachine sequencing games. *European Journal of Operational Research*, 119: 678-691.
- [8] Hamers, H., Klijn, F., van Velzen, B., (2005). On the convexity of precedence sequencing games. *Annals of Operations Research*, 137: 161-175.
- [9] Le Breton, M., Owen, G., Weber, S., (1992). Strongly balanced cooperative games. *International Journal of Game Theory*, 20: 419-427.
- [10] Lee, C. Y., Uzsoy, R., Martin-Vega, L. A., (1992). Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research*, 40: 764-775.

- [11] Lee, C. Y., Chen, Z. L., (2001). Machine scheduling with transportation considerations. *Journal of Scheduling*, 4: 3-24.
- [12] van den Nouweland, A., Krabbenborg, M., Potters, J., (1992). Flow-shops with a dominant machine. *European Journal of Operational Research*, 62: 38-46.
- [13] Potts, N. C., Kovalyov, Y. M., (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120: 228-249.
- [14] Shapley, L. S., (1953). A value for n-person games. In: Kuhn, H.,W., Tucker, A.,W., (Eds.), Contributions to the Theory of Games II. Princeton Univ. Press. In: *Annals of Mathematics Studies*, 28: 307-317.
- [15] Slikker, M., (2005). Balancedness of sequencing games with multiple parallel machines. *Annals of Operations Research*, 137: 177-189.
- [16] Slikker, M., (2006). Balancedness of multiple machine sequencing games revisited. *European Journal of Operational Research*, 174: 1944-1949.
- [17] Suijs, J., Hamers, H., Tijs, S., (1997). On consistency of reward allocation rules in sequencing situations. In: Hanveld, K., Vrieze, O., and Kallenberg, L., (Eds.), *Ten Years LNMB*, 223-232. Amsterdam: CWI Tract.
- [18] van Velzen, B., (2006). Sequencing games with controllable processing times. *European Journal of Operational Research*, 172: 64-85.
- [19] Webster, S., Baker, R. K., (1995). Scheduling groups of jobs on a single machine. *Operations Research*, 43: 692-703.