DOI 10.1007/s10951-008-0062-z

Proportionate flow shop games

Arantza Estévez-Fernández · Manuel A. Mosquera · Peter Borm · Herbert Hamers

Received: 25 July 2006 / Accepted: 3 March 2008 / Published online: 29 March 2008

Abstract In a proportionate flow shop problem several jobs have to be processed through a fixed sequence of machines and the processing time of each job is equal on all machines. By identifying jobs with agents whose costs linearly depend on the completion time of their jobs and assuming an initial processing order on the jobs, we face two problems: the first is how to obtain an optimal order that minimizes the total processing cost, the second is how to allocate the cost savings obtained by ordering the jobs optimally. In this paper we focus on the allocation problem. PFS games are defined as cooperative games associated to proportionate flow shop problems. It is seen that PFS games have a nonempty core. Moreover, it is shown that PFS games are convex if the jobs are initially ordered in decreasing urgency. For this case an explicit game independent expression for the Shapley value is provided.

The authors thank two referees for their valuable suggestions for improvement. M.A. Mosquera acknowledges the financial support of *Ministerio de Ciencia y Tecnología*, FEDER, *Xunta de Galicia* (projects SEJ2005-07637-C02-02 and PGIDIT06PXIC207038PN).

A. Estévez-Fernández (✉)
Department of Econometrics and Operations Research,
VU University Amsterdam, De Boelelaan 1105, 1081 HV
Amsterdam, The Netherlands
e-mail: mestevez@feweb.vu.nl

M.A. Mosquera
Department of Statistics and Operations Research, Faculty of
Business Administration and Tourism, University of Vigo,
32004 Ourense, Spain

P. Borm · H. Hamers
CentER and Department of Econometrics and Operations
Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg,
The Netherlands

Keywords Proportionate flow shop problems · Core · Convexity · Shapley value

1 Introduction

In a flow shop problem a group of jobs has to be processed through a fixed number of machines and the order of the machines in which the jobs have to be processed is the same for all jobs. To each job a cost is associated dependent on its completion time. In this paper we consider proportionate flow shop problems. A proportionate flow shop problem is a flow shop problem in which additionally every job has the same processing time on each machine (e.g., a wooden door needs several layers of paint and each different layer of paint has a different machine associated to it, but the painting time is always the same). Proportionate flow shop problems have gained considerable attention lately and various papers have been devoted to this topic. In Shakhlevich et al. (1998) an algorithm is provided to obtain an optimal schedule for this kind of problems. Shiau and Huang (2004) generalize this type of problems by considering multiple identical machines at any stage. In Allahverdi (1996) and Allahverdi and Savsar (2001) proportionate flow shop problems with breakdowns (machines can have failures and some extra time is needed for repairs) and setup times (situations where each job needs some specific preparations on the machine before being processed, thus incurring some additional extra time) are studied, respectively. Cheng and Shakhlevich (1999) propose algorithms for proportionate flow shop problems where the processing times can be controlled by incurring extra costs.

By associating jobs to clients, a proportionate flow shop problem gives rise to an interactive decision making problem. Each client incurs costs, which we assume to depend linearly on the completion time of its job. By assuming an initial order on the jobs, the first problem clients jointly face is an optimization problem which is to find an optimal reordering of all jobs, i.e., a schedule that maximizes joint cost savings. The subsequent problem is of a game theoretic nature and asks how to reallocate these cost savings in a fair way. By defining the value of a coalition of clients as the maximal attainable cost savings by means of an optimal admissible reordering, we obtain a cooperative proportionate flow shop game (a PFS game) related to the proportionate flow shop problem. The core of this game provides insight in the allocation problem at hand, since core elements lead to a stable reallocation of the joint cost savings. A game is said to be balanced if it has a nonempty core.

The above game-theoretic approach to sequencing situations has been initiated by Curiel et al. (1989) for the class of one-machine sequencing situations. Generalizations to, e.g., ready times, due dates, multiple ownership, more machines, and controllable processing times have been studied in Hamers et al. (1995), Borm et al. (2002), Calleja et al. (2006), Estévez-Fernández et al. (2004), Hamers et al. (1999), Calleja et al. (2002), and van Velzen (2006). A review on sequencing games can be found in Curiel et al. (2002). Finally, within the context of flow shop problems, van den Nouweland et al. (1990) have studied the specific case of a dominant machine. Other approaches to cooperation in sequencing situations can be found in Maniquet (2003) and Moulin and Stong (2002). For queuing situations especially, strategic behavior and competition have been analyzed; for a clear survey we refer to Hassin and Haviv (2003). Two papers on cooperative aspects of queuing situations are González and Herrero (2004) and García-Sanz et al. (2008).

The current article analyzes proportionate flow shop problems and related PFS games following the lines set out by Curiel et al. (1989). It is shown that PFS games are balanced. Moreover, PFS games turn out to be convex if the initial order is an urgency order. An urgency order is an order in which jobs are ordered nonincreasingly with respect to their urgency index. Here, as usual, the urgency index of a job is the quotient of its cost coefficient (i.e., the coefficient of the linear cost function) and its processing time. Besides, in this setting we provide an explicit game independent expression for the Shapley value.

The remainder of the paper is organized as follows. Section 2 provides the basic definitions and terminology of proportionate flow shop problems. Here, three useful results in Shakhlevich et al. (1998) and the algorithm presented there are recalled. In Sect. 3, PFS games are defined and it is shown that they are balanced. Besides, these games are convex provided that the initial order is an urgency order. In such case, an expression of the Shapley value, which does not depend on coalitional values, is provided.

2 Proportionate flow shop problems

A flow shop situation consists of a fixed sequence of m machines, and a finite set of jobs N that have to be processed on all machines. Without loss of generality, we assume in this paper that each job must be processed first on machine M_1, then on M_2, etc. A proportionate flow shop (PFS) situation is a flow shop situation where the processing time of every job is the same on each machine. Hence, a PFS situation can be described by a 3-tuple (M, N, p) with $M = \{M_1, \dots, M_m\}$ being the set of machines, $N = \{1, \dots, n\}$ denoting the set of jobs, and $p \in \mathbb{R}^N_+$ as the vector of processing times of the jobs.

A *schedule* fixes for every job i and every machine r a time interval of length p_i in which job i is processed in such a way that neither a job is processed on two different machines at the same time, nor a machine processes two different jobs at the same time. Given a PFS situation (M, N, p) we denote a schedule of the jobs on the machines as $\sigma = (\sigma^1, \dots, \sigma^m)$ with $\sigma^r : N \to \{1, \dots, |N|\}$ a bijection describing the processing order on machine M_r. We denote by $\Pi(N, M)$ the set of all schedules of the jobs on the machines. Given $\sigma \in \Pi(N, M)$, $i \in N$, and $M_r \in M$, we denote by $P(\sigma^r, i)$ the *set of predecessors of job i on machine* M_r, i.e., $P(\sigma^r, i) = \{j \in N \mid \sigma^r(j) < \sigma^r(i)\}$. Further, we define $\bar{P}(\sigma^r, i) := P(\sigma^r, i) \cup \{i\}$. We denote by $p(\sigma^r, i)$ the *immediate predecessor of job i on machine M_r*, i.e., $p(\sigma^r, i) \in N$ such that $\bar{P}(\sigma^r, p(\sigma^r, i)) = P(\sigma^r, i)$. Notice that $p(\sigma^r, i)$ does not exist when $P(\sigma^r, i) = \emptyset$.

In principle, the processing order on machines need not be the same. A schedule $\sigma = (\sigma^1, \dots, \sigma^m)$ with $\sigma^1 = \dots = \sigma^m$ is called a *permutation schedule*, or *order*. We denote by $\Pi(N)$ the set of all permutations schedules of the jobs.

Assuming that processing starts at time 0 and that there are no unnecessary delays, the *completion time of job i* on machine M_r *with respect to* an arbitrary *schedule* σ, $C^\sigma_i(r)$, can be recursively determined by

$$C^\sigma_i(1) = \sum_{j \in \bar{P}(\sigma^1, i)} p_j,$$

and for $r = 2, \dots, m$,

$$C^\sigma_i(r) = \begin{cases} C^\sigma_i(r-1) + p_i & \text{if } P(\sigma^r, i) = \emptyset, \\ \max\{C^\sigma_{p(\sigma^r,i)}(r), C^\sigma_i(r-1)\} + p_i & \text{otherwise.} \end{cases}$$

It is assumed that each job $i \in N$ incurs costs, c_i, which are linear with respect to the time in which the job leaves the system according to schedule σ. Hence, there exist positive numbers α_i, $i \in N$, such that $c_i(\sigma) = \alpha_i C^\sigma_i(m)$. From now on we denote the overall completion time $C^\sigma_i(m)$ by C^σ_i.

Given a PFS situation (M, N, p) and a linear cost associated to each job, which is represented by $\alpha \in \mathbb{R}^N$, the associated *PFS problem*, (M, N, p, α) has the objective of finding a schedule that minimizes the total cost originated in the

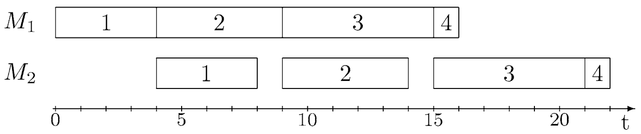

Fig. 1 Gantt chart of the PFS situation in Example 2.2

system, i.e., finding $\hat{\sigma}$ such that

$$c_N(\hat{\sigma}) = \min_{\sigma \in \Pi(N,M)} c_N(\sigma)$$

with $c_N(\sigma) = \sum_{i \in N} c_i(\sigma) = \sum_{i \in N} \alpha_i C_i^{\sigma}$. Note that $\Pi(N, M)$ is finite, and therefore there exists at least one optimal solution.

Next, we recall three lemmas from Shakhlevich et al. (1998) that are used throughout the article.

Lemma 2.1 (Shakhlevich et al. 1998) *Let (M, N, p, α) be a PFS problem. Then,*

(i) *every optimal schedule is a permutation schedule*;
(ii) *for a permutation schedule σ and $i \in N$, the completion time C_i^{σ} is given by*

$$C_i^{\sigma} = \sum_{j \in \bar{P}(\sigma,i)} p_j + (m-1) \max_{j \in \bar{P}(\sigma,i)} \{p_j\}.$$

Example 2.2 Let (M, N, p, α) be a PFS problem with machines $M = \{M_1, M_2\}$, jobs $N = \{1, 2, 3, 4\}$, vector of processing times $p = (4, 5, 6, 1)$, and vector of cost coefficients $\alpha = (32.5, 32, 32, 5)$. Let $\sigma = (1\ 2\ 3\ 4)$ be a permutation schedule. This situation is represented in Fig. 1.

Here, $C_1^{\sigma} = 8$, $C_2^{\sigma} = 14$, $C_3^{\sigma} = 21$, and $C_4^{\sigma} = 22$. We illustrate how to calculate C_3^{σ} below.

$$\begin{aligned} C_3^{\sigma} &= p_1 + p_2 + p_3 + (m-1)\max\{p_1, p_2, p_3\} \\ &= 4 + 5 + 6 + (2-1)\max\{4, 5, 6\} = 21. \end{aligned}$$

Hence, the total weighted completion time according to σ is $c_N(\sigma) = 1490$.

Since the processing time of a job is the same on all machines, we can define an *urgency* (index) of job $i \in N$ as $u_i = \frac{\alpha_i}{p_i}$. The next lemma states that if a job has higher urgency than another job with larger processing time, then the one with higher urgency is processed first in an optimal order.

Lemma 2.3 (Shakhlevich et al. 1998) *Let (M, N, p, α) be a PFS problem and let σ be an optimal order. If $i, j \in N$ are such that $u_i \geq u_j$ and $p_i < p_j$, or $u_i > u_j$ and $p_i \leq p_j$, then $\sigma(i) < \sigma(j)$.*

We now recall some terminology from Shakhlevich et al. (1998). Let (M, N, p, α) be a PFS problem and let $\sigma \in \Pi(N)$. We say that job $i \in N$ is a *new-max job according to σ* if $p_i > \max_{j \in P(\sigma,i)}\{p_j\}$. Let $a_1^{\sigma}, \ldots, a_s^{\sigma}$ be the new-max jobs according to σ, with $\sigma(a_1^{\sigma}) < \cdots < \sigma(a_s^{\sigma})$. Note that $p_{a_1^{\sigma}} < \cdots < p_{a_s^{\sigma}}$. Then, N can be partitioned into s so-called *segments* $A_1^{\sigma}, \ldots, A_s^{\sigma}$ in the following way

$$A_r^{\sigma} := \begin{cases} P(\sigma, a_{r+1}^{\sigma}) \setminus P(\sigma, a_r^{\sigma}) & \text{if } 1 \leq r < s, \\ N \setminus P(\sigma, a_r^{\sigma}) & \text{if } r = s. \end{cases}$$

Note that $P(\sigma, a_1^{\sigma}) = \emptyset$, since $\sigma(a_1^{\sigma}) = 1$. The above partition into segments is denoted by $Seg(\sigma)$.

The lemma below states that in any optimal order the jobs in a segment are processed in nonincreasing urgency order.

Lemma 2.4 (Shakhlevich et al. 1998) *Let (M, N, p, α) be a PFS problem and let σ be an optimal order. Let A_r^{σ} be a segment corresponding to σ and $i, j \in A_r^{\sigma}$. If $\sigma(i) < \sigma(j)$, then $u_i \geq u_j$.*

Subsequently, we recall the algorithm to find an optimal schedule for PFS problems given in Shakhlevich et al. (1998). In our description, we already incorporate the results in Lemmas 2.3 and 2.4. Let (M, N, p, α) be a PFS problem. We define an *urgency order*, σ_u, as an order in which the jobs are ordered in nonincreasing urgency and jobs with the same urgency are ordered according to nondecreasing processing time. Since the starting point of the algorithm is σ_u, we can assume without loss of generality that $\sigma_u = (1\ \ldots\ n)$. To find the optimal order we generate orders $\hat{\sigma}_1, \ldots, \hat{\sigma}_n$, where $\hat{\sigma}_1 := \sigma_u$ and $\hat{\sigma}_n$ is optimal. Note that associated with $\hat{\sigma}_{i-1}$ we have the segments $A_1^{\hat{\sigma}_{i-1}}, \ldots, A_s^{\hat{\sigma}_{i-1}}$ which give a partition of N. Now, we explain how to obtain $\hat{\sigma}_i$ from $\hat{\sigma}_{i-1}$. Let $s_i \in \{1, \ldots, s\}$ be such that $A_{s_i}^{\hat{\sigma}_{i-1}} \cap \{1, \ldots, i-1\} \neq \emptyset$ and $A_{s_i+1}^{\hat{\sigma}_{i-1}} \cap \{1, \ldots, i-1\} = \emptyset$. We define $A(i, 1), \ldots, A(i, s_i)$ as $A(i, 1) = A_{s_i}^{\hat{\sigma}_{i-1}} \cap \{1, \ldots, i-1\}$ and $A(i, r) = A_{s_i-r+1}^{\hat{\sigma}_{i-1}}$ for $r = 2, \ldots, s_i$, i.e., we have numbered the segments from right to left.

Associated to each $A(i, r)$, we denote by $a(i, r)$ the new-max job of this (sub)segment and by σ_{ir} the order we obtain from $\hat{\sigma}_{i-1}$ by placing i at the tail of $A(i, r)$. Note that $\sigma_{i1} = \hat{\sigma}_{i-1}$. We denote by σ_{is_i+1} the order we obtain from $\hat{\sigma}_{i-1}$ by placing i in front of $a(i, s_i)$. Figure 2 gives an illustration of these concepts where $s_i = 2$. In this figure, each rectangle represents a job and the height of the rectangle represents its processing time. Shaded rectangles are new-max jobs according to the urgency order.

Let $r^N(i) \in \{1, \ldots, s_i + 1\}$ be such that

(1) $c_N(\sigma_{ir}) - c_N(\sigma_{ir^N(i)}) > 0$ for every $r < r^N(i)$;
(2) $c_N(\sigma_{ir^N(i)}) - c_N(\sigma_{ir}) \leq 0$ for every $r > r^N(i)$.

We define $\hat{\sigma}_i = \sigma_{ir^N(i)}$. Obviously, (1) and (2) imply that $c_N(\hat{\sigma}_i)$ is minimal with respect to the $s_i + 1$ considered orders. In Shakhlevich et al. (1998), it is shown that $\hat{\sigma}_i =$

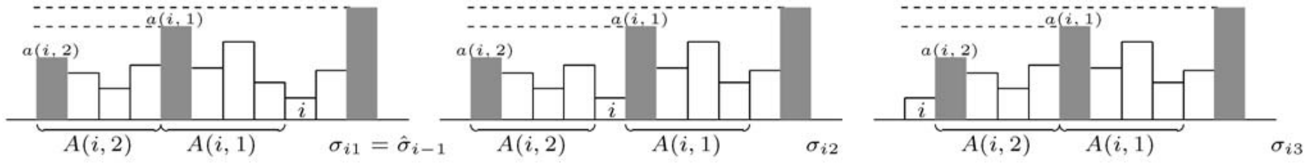

Fig. 2 Reordering of job *i*

$\sigma_{ir^N(i)}$ is optimal for the case in which jobs $\{i+1,\dots,n\}$ are not allowed to be reordered. Note that (2) guarantees that i is processed latest in case of ties. Equations (1) and (2) are equivalent to $c_N(\hat{\sigma}_i)$ is "minimal" and $p_{a(i,r^N(i))} > p_{a(i,r)}$ for every $r \neq r^N(i)$ with $c_N(\sigma_{ir}) = c_N(\sigma_{ir^N(i)})$.

3 Proportionate flow shop games

In this section we introduce proportionate flow shop games and show balancedness. Moreover, if the initial order is an urgency order, then the corresponding proportionate flow shop game is convex and an explicit game independent expression of the Shapley value is provided based on the decomposition of proportionate flow shop games into unanimity games.

Before stating our main results we recall some basic notions from cooperative game theory.

A cooperative TU-game in characteristic function form is an ordered pair (N, v) where N is a finite set (the set of players) and $v: 2^N \to \mathbb{R}$ satisfies $v(\emptyset) = 0$. Here, 2^N denotes the set of subsets (coalitions) of N. A game (N, v) is called *super-additive* if $v(S) + v(T) \leq v(S \cup T)$ for all $S, T \subset N$ such that $S \cap T = \emptyset$. A special class of TU-games is the class of unanimity games. Let $T \subset N$, we define the *unanimity game* (N, u_T) by

$$u_T(S) = \begin{cases} 1 & \text{if } T \subset S, \\ 0 & \text{otherwise.} \end{cases}$$

Recall that every TU-game (N, v) can be written in a unique way as linear combination of unanimity games.

The *core* of a cooperative TU-game (N, v) is defined by

$$\mathcal{C}\mathrm{ore}(v) = \left\{ x \in \mathbb{R}^N \,\middle|\, \sum_{i \in N} x_i = v(N),\ \sum_{i \in S} x_i \geq v(S) \right.$$
$$\left. \text{for all } S \in 2^N \right\},$$

i.e., the core is the set of efficient allocations of $v(N)$ such that there is no coalition with an incentive to split off. A game is said to be *balanced* (see Bondareva 1963 and Shapley 1967) if the core is nonempty.

An important subclass of balanced games is the class of convex games (cf. Shapley 1971). A game (N, v) is said to be *convex* if

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T)$$

for all $S, T \subset N$.

An important and well studied one point solution in cooperative game theory is the Shapley value (Shapley 1953). Let (N, v) be a TU-game and let $v = \sum_{T \subset N,\, T \neq \emptyset} c_T u_T$ be the unique decomposition into unanimity games. Then, the Shapley value of (N, v) is defined by

$$\Phi_i(v) = \sum_{T: i \in T} \frac{c_T}{|T|} \tag{3.1}$$

for every $i \in N$. It turns out that the Shapley value of a convex game is always in the core (Shapley 1953).

Next, we start the game theoretical study of proportionate flow shops. Let (M, N, p, α) be a PFS problem and let $\sigma_0 \in \Pi(N, M)$ be an initial schedule on the jobs. By associating jobs with players (or clients) we consider an associated PFS game. In order to define the value of a coalition $S \subset N$, one has to decide which schedules are admissible for S. For this, we first need the notion of connected components in a machine. A *coalition* U is called *connected* (with respect to σ_0^r) if for all $i, j \in U$ and $k \in N$ such that $\sigma_0^r(i) < \sigma_0^r(k) < \sigma_0^r(j)$ it holds that $k \in U$. A connected coalition $U \subset S$ is called *maximal on machine r* if $U \cup \{i\}$ is not connected with respect to σ_0^r for every $i \in S \setminus U$. Given a coalition $S \subset N$, we denote by S/σ_0^r the set of all maximally connected components of S according to σ_0^r.

Given a coalition $S \subset N$, the set of *admissible schedules* of S with respect to σ_0 consists of all schedules σ such that

(i) $S/\sigma_0^r = S/\sigma^r$ for all $r \in M$;
(ii) $C_i^{\sigma}(r) = C_i^{\sigma_0}(r)$ for every $i \in N \setminus S$ and for all $r \in M$.

In words, a schedule is admissible for a coalition S with respect to σ_0 if changes on any machine r with respect to σ_0^r are only allowed among elements within connected components of S on r in such a way that no job outside the coalition has its completion time affected. These two conditions and Lemma 2.1 imply that

(iii) all optimal schedules for the grand coalition are admissible;
(iv) if the initial schedule is an order, then any order satisfying (i) is admissible for any coalition $S \subset N$.

We now proceed with the definition of PFS games. Let (M,N,p,α) be a PFS problem and let $\sigma_0 \in \Pi(N,M)$ be an initial schedule on the jobs. The associated PFS game (N,v) is defined by

$$v(S) := \max_{\sigma\in\mathcal{A}(S,\sigma_0)} \bigl\{c_S(\sigma_0) - c_S(\sigma)\bigr\}$$

for every $S \subset N$, where $c_S(\sigma) = \sum_{i\in S} c_i(\sigma) = \sum_{i\in S} \alpha_i C_i^\sigma$ and $\mathcal{A}(S,\sigma_0)$ is the set of admissible rearrangements for coalition S with respect to σ_0. Note that by Condition (ii) of admissibility, the characteristic function can also be expressed as

$$v(S) := \max_{\sigma\in\mathcal{A}(S,\sigma_0)} \bigl\{c_N(\sigma_0) - c_N(\sigma)\bigr\}.$$

Theorem 3.1 *PFS games are balanced.*

Proof[1] We distinguish between two cases.

Case 1: σ_0 is an order. In this case, $\sigma_0^1 = \cdots = \sigma_0^m$ and $S/\sigma_0 := S/\sigma_0^1 = \cdots = S/\sigma_0^m$. It is readily seen that the game is σ_0-component additive (i.e., $v(\{i\}) = 0$ for all $i \in N$, (N,v) is super-additive, and $v(S) = \sum_{T\in S/\sigma_0} v(T)$) and therefore balanced (see LeBreton et al. 1992 and Curiel et al. 1994).

Case 2: σ_0 is an arbitrary schedule. Let (N,v) be the PFS game associated to the PFS problem (M,N,p,α) and the initial schedule σ_0. Define the order $\tilde{\sigma}_0$ by $\tilde{\sigma}_0^r = \sigma_0^m$ for every $r \in M$. Let (N,w) be the PFS game associated to the PFS problem (M,N,p,α) and initial order $\tilde{\sigma}_0$. Defining $a_i = c_i(\sigma_0) - c_i(\tilde{\sigma}_0)$ for all $i \in N$, it follows that

$$v(N) = w(N) + \sum_{i\in N} a_i \tag{3.2}$$

by definition of (N,v), (N,w), and a_i, and the fact that the optimal schedule for N, $\hat{\sigma}$, belongs to both admissible sets as pointed out in (iii), i.e., $\hat{\sigma} \in \mathcal{A}(N,\sigma_0)$, $\hat{\sigma} \in \mathcal{A}(N,\tilde{\sigma}_0)$. Besides,

$$v(S) \le w(S) + \sum_{i\in S} a_i. \tag{3.3}$$

Equation (3.3) follows because for every $\sigma \in \mathcal{A}(S,\sigma_0)$ we can define $\tilde{\sigma}$ by $\tilde{\sigma}^r = \sigma^m$ for every $r \in M$. Since $\tilde{\sigma}_0$ is an order, we have that the order $\tilde{\sigma}$ is admissible as pointed out in (iv), i.e., $\tilde{\sigma} \in \mathcal{A}(S,\tilde{\sigma}_0)$. Hence,

$$\begin{aligned}
v(S) &= \max_{\sigma\in\mathcal{A}(S,\sigma_0)} \bigl\{c_S(\sigma_0) - c_S(\sigma)\bigr\} \\
&= c_S(\sigma_0) - \min_{\sigma\in\mathcal{A}(S,\sigma_0)} \bigl\{c_S(\sigma)\bigr\} \\
&= c_S(\sigma_0) - c_S(\tilde{\sigma}_0) + c_S(\tilde{\sigma}_0) - \min_{\sigma\in\mathcal{A}(S,\sigma_0)} \bigl\{c_S(\sigma)\bigr\} \\
&\le c_S(\sigma_0) - c_S(\tilde{\sigma}_0) + c_S(\tilde{\sigma}_0) - \min_{\sigma\in\mathcal{A}(S,\sigma_0)} \bigl\{c_S(\tilde{\sigma})\bigr\} \\
&\le c_S(\sigma_0) - c_S(\tilde{\sigma}_0) + c_S(\tilde{\sigma}_0) - \min_{\sigma\in\mathcal{A}(S,\tilde{\sigma}_0)} \bigl\{c_S(\sigma)\bigr\} \\
&= c_S(\sigma_0) - c_S(\tilde{\sigma}_0) + \max_{\sigma\in\mathcal{A}(S,\tilde{\sigma}_0)} \bigl\{c_S(\tilde{\sigma}_0) - c_S(\sigma)\bigr\} \\
&= \sum_{i\in S} a_i + w(S),
\end{aligned}$$

where the first inequality follows from the fact that $c_S(\sigma) \ge c_S(\tilde{\sigma})$ for every $\sigma \in \mathcal{A}(S,\sigma_0)$ by Lemma 1 in Shakhlevich et al. (1998), the second inequality holds because $\mathcal{A}(S,\tilde{\sigma}_0)$ may be larger than the set of orders $\tilde{\sigma}$ defined from admissible schedules $\sigma \in \mathcal{A}(S,\sigma_0)$. The last equality follows by definition of a_i, since $c_S(\sigma_0) - c_S(\tilde{\sigma}_0) = \sum_{i\in S} c_i(\sigma_0) - \sum_{i\in S} c_i(\tilde{\sigma}_0) = \sum_{i\in S} a_i$.

By case 1, (N,w) is balanced, and therefore $\mathcal{C}\mathrm{ore}(w) \neq \emptyset$. Let $y \in \mathcal{C}\mathrm{ore}(w)$ and define $x = y + a$. Then, from (3.2) and (3.3) it follows that $x \in \mathcal{C}\mathrm{ore}(v)$, and therefore (N,v) is balanced. □

Example 3.2 Let (M,N,p,α) be a PFS situation where $N = \{1,2,3\}$, $M = \{M_1,M_2,M_3\}$, $p = (3,1,4)$ and $\alpha = (4,1,7)$. Let σ_0 be an order, with $\sigma_0 = (1\ 2\ 3)$. The situation is represented in Fig. 3.

The corresponding PFS game (N,v) is given in Table 1.

We explain in detail how to calculate the value of coalition $\{1,2\}$ below. The total cost with the initial order is $c_N(\sigma_0) = 158$. The set of admissible rearrangements for coalition $\{1,2\}$ is $\mathcal{A}(\{1,2\},\sigma_0) = \{\sigma_0,\sigma_1,\sigma_2,\sigma_3,\sigma_4,\sigma_5,\sigma_6,\sigma_7\}$, where the different schedules and their associated costs and cost savings are given in Table 2.

It turns out that σ_7 has the minimum associated cost and then

$$\begin{aligned}
v\bigl(\{1,2\}\bigr) &= \max_{\sigma\in\mathcal{A}(\{1,2\},\sigma_0)} \bigl\{c_S(\sigma_0) - c_S(\sigma)\bigr\} \\
&= c_S(\sigma_0) - c_S(\sigma_7) = 3.
\end{aligned}$$

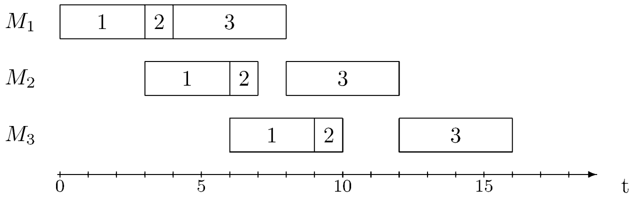

Fig. 3 Gantt chart of the PFS situation in Example 3.2

Table 1 Coalitional values of the game in Example 3.2

S	{1}	{2}	{3}	{12}	{13}	{23}	N
$v(S)$	0	0	0	3	0	1	3

[1] We are indebted to a referee for the idea of this proof.

Table 2 Admissible schedules for coalition $\{1, 2\}$ with respect to σ_0 in Example 3.2

σ_l	$c_S(\sigma_l)$	$c_S(\sigma_0) - c_S(\sigma_l)$
$\sigma_0 = ((1\ 2\ 3), (1\ 2\ 3), (1\ 2\ 3))$	46	0
$\sigma_1 = ((1\ 2\ 3), (1\ 2\ 3), (2\ 1\ 3))$	52	−6
$\sigma_2 = ((1\ 2\ 3), (2\ 1\ 3), (1\ 2\ 3))$	56	−10
$\sigma_3 = ((2\ 1\ 3), (1\ 2\ 3), (1\ 2\ 3))$	51	−5
$\sigma_4 = ((1\ 2\ 3), (2\ 1\ 3), (2\ 1\ 3))$	50	−4
$\sigma_5 = ((2\ 1\ 3), (1\ 2\ 3), (2\ 1\ 3))$	57	−11
$\sigma_6 = ((2\ 1\ 3), (2\ 1\ 3), (1\ 2\ 3))$	51	−5
$\sigma_7 = ((2\ 1\ 3), (2\ 1\ 3), (2\ 1\ 3))$	43	3

Note that the initial order in Example 3.2 is not an urgency order. Moreover, the game is balanced but not convex.

From now on we study PFS games *with an urgency order as the initial schedule*, i.e.,

$$\sigma_0 = \sigma_u.$$

Moreover, we assume without loss of generality that

$$\sigma_0 = (1\ 2\ \dots\ n).$$

Since the initial schedule is an order, we have $S/\sigma_0^r = S/\sigma_0^m$ for every $r \in M$. With minor abuse of notation, we denote by S/σ_0 the set of maximally connected components on each machine. Due to the conditions of admissible rearrangements, we can write the value of coalition $S \subset N$ as

$$v(S) = \sum_{R \in S/\sigma_0} v(R), \tag{3.4}$$

as it has been pointed out in the proof of Theorem 3.1.

Next, we give an expression for the value of a coalition based on the cost savings that each player can obtain if a similar procedure as the method in Sect. 2 is followed. Due to (3.4) we can restrict our study to connected coalitions. Let $S \subset N$ be a connected coalition, $S = \{i_S, i_S + 1, \dots, j_S - 1, j_S\}$. To find the optimal order for S we generate orders $\hat{\sigma}_{i_S}^S, \dots, \hat{\sigma}_{j_S}^S$ where $\hat{\sigma}_{i_S}^S := \sigma_u$ and $\hat{\sigma}_{j_S}^S$ is optimal in S. Now, we explain how $\hat{\sigma}_i^S$ is obtained from $\hat{\sigma}_{i-1}^S$. If for every r it holds that $A_r^{\hat{\sigma}_{i-1}^S} \cap \{i_S, \dots, i-1\}$ does not contain any new-max job according to $\hat{\sigma}_{i-1}^S$, then i and i_S belong to the same segment in σ_u and i_S is not a new-max job. In this case, $\hat{\sigma}_i^S = \sigma_u$ by Lemma 2.4. This situation is illustrated in Fig. 4.

If $A_r^{\hat{\sigma}_{i-1}^S} \cap \{i_S, \dots, i-1\}$ contains a new-max job according to $\hat{\sigma}_{i-1}^S$ for some r, then we define

$$r_i = \min\bigl\{r \mid A_r^{\hat{\sigma}_{i-1}^S} \cap \{i_S, \dots, i-1\} \text{ contains a new-max job according to } \hat{\sigma}_{i-1}^S\bigr\},$$

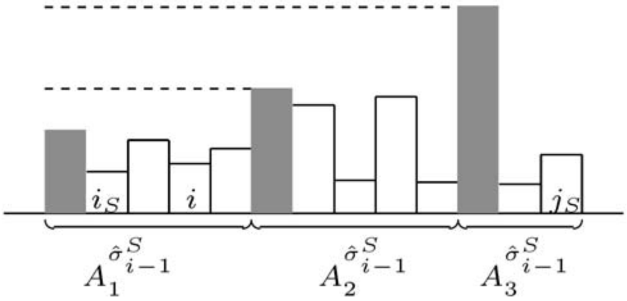
i_S
i
j_S
$A_1^{\hat{\sigma}_{i-1}^S}$
$A_2^{\hat{\sigma}_{i-1}^S}$
$A_3^{\hat{\sigma}_{i-1}^S}$

Fig. 4 Reordering of job i in coalition S

$$t_i = \max\bigl\{r \mid A_r^{\hat{\sigma}_{i-1}^S} \cap \{i_S, \dots, i-1\} \text{ contains a new-max job according to } \hat{\sigma}_{i-1}^S\bigr\},$$

$$s_i = t_i - r_i + 1.$$

Analogously to the method described in Sect. 2, we define $A^S(i, 1), \dots, A^S(i, s_i)$ as

$$A^S(i, 1) = A_{t_i}^{\hat{\sigma}_{i-1}^S} \cap \{i_S, \dots, i-1\},$$

$$A^S(i, r) = A_{t_i - r + 1}^{\hat{\sigma}_{i-1}^S} \quad \text{for } r = 2 \dots s_i.$$

Associated to each $A^S(i, r)$, we denote by $a^S(i, r)$ the new-max job of this (sub)segment and by σ_{ir}^S the order we obtain from $\hat{\sigma}_{i-1}^S$ by placing i at the tail of $A^S(i, r)$. Note that $\sigma_{i1}^S = \hat{\sigma}_{i-1}^S$. We denote by $\sigma_{is_i+1}^S$ the order obtained from $\hat{\sigma}_{i-1}^S$ by placing i in front of $a^S(i, s_i)$. This procedure is clarified in Fig. 5 where $s_i = 2$.

Let $r^S(i) \in \{1, \dots, s_i + 1\}$ be such that

(3) $c_N(\sigma_{ir}) - c_N(\sigma_{ir^S(i)}) > 0$ for every $r < r^S(i)$;
(4) $c_N(\sigma_{ir^S(i)}) - c_N(\sigma_{ir}) \leq 0$ for every $r > r^S(i)$.

We define $\hat{\sigma}_i^S = \sigma_{ir^S(i)}^S$. As in Sect. 2, the idea behind (3) and (4) is to make $c_N(\hat{\sigma}_i^S)$ minimal. Besides, (4) guarantees that i is processed latest in case of ties. Note that this is equivalent to $c_N(\hat{\sigma}_i^S)$ being minimal and $p_{a(i, r^S(i))} > p_{a(i,r)}$ for every $r \neq r^S(i)$ with $c_N(\sigma_{ir}^S) = c_N(\sigma_{ir^S(i)}^S)$.

In order to find the optimal order for a coalition S one has to realize that there might be new-max jobs preceding the elements of S that do not belong to the coalition. This, however, does not constitute a problem and Lemmas 2.1, 2.3, and 2.4 are still applicable in S. The proof that $\hat{\sigma}_{j_S}^S$ is optimal among the admissible orders of S can be provided along the same lines as in Shakhlevich et al. (1998) and is therefore omitted.

Hence, the value of coalition S can be written as

$$v(S) = \sum_{i=i_S}^{j_S} \bigl(c_N(\hat{\sigma}_{i-1}^S) - c_N(\hat{\sigma}_i^S)\bigr)$$

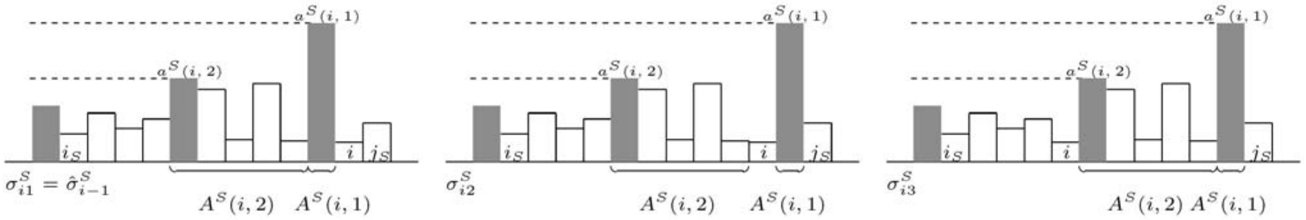

Fig. 5 Reordering of job i in coalition S

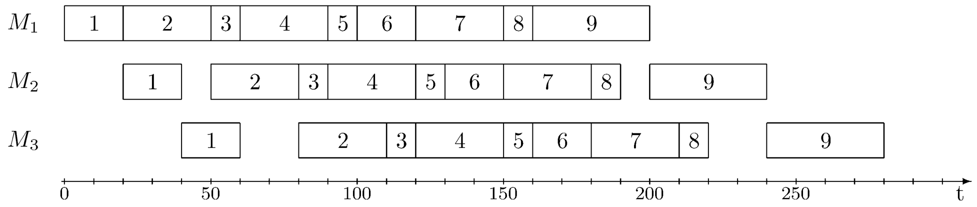

Fig. 6 Gantt chart of the PFS situation in Example 3.3

with $\hat{\sigma}^S_{i_S-1} := \sigma_u$. We define

$$G^S_i := c_N\big(\hat{\sigma}^S_{i-1}\big) - c_N\big(\hat{\sigma}^S_i\big)$$

for $i \in S$. Here, $G^S_i \geq 0$ denotes the cost savings obtained after reordering job $i \in S$ in S. Then,

$$v(S) = \sum_{i \in S} G^S_i.$$

Note that, as a consequence of Lemma 2.3, it follows that if $j \in P(\sigma_u, a)$, then $j \in P(\hat{\sigma}^S_i, a)$ for every $i \in S$, with a a new-max job according to σ_u.

The following example illustrates how to compute coalitional values by means of individual cost savings.

Example 3.3 Let (M, N, p, α) be a PFS problem with $M = \{M_1, M_2, M_3\}$, $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $p = (20, 30, 10, 30, 10, 20, 30, 10, 40)$, and $\alpha = (200, 270, 80, 210, 69, 130, 180, 59, 200)$. The urgency order is $\sigma_u = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$. Suppose that initially the jobs are processed according to the urgency order. Then, $c_N(\sigma_u) = 224020$. The situation is represented in Fig. 6.

It can be checked that the optimal order for this problem is $\hat{\sigma} = (3\ 5\ 8\ 1\ 6\ 2\ 4\ 7\ 9)$ with an associated cost of $c_N(\hat{\sigma}) = 219110$. Hence, the total value of the grand coalition is $v(N) = 224020 - 219110 = 4910$. The total value can be decomposed in the sum of the cost savings that are obtained each time a job is reordered, i.e., $v(N) = \sum_{i=1}^{9} G^N_i$ with $G^N_i = c_N(\hat{\sigma}^N_{i-1}) - c_N(\hat{\sigma}^N_i)$. Easy but tedious calculations show that $G^N_1 = G^N_2 = 0$, $G^N_3 = 2500$, $G^N_4 = 0$, $G^N_5 = 1480$, $G^N_6 = 800$, $G^N_7 = 0$, $G^N_8 = 130$, and $G^N_9 = 0$.

The next result gives the decomposition into unanimity games of a PFS game. We denote by $\{a_1, \ldots, a_s\}$, with $a_1 < \cdots < a_s$, the set of new-max jobs according to σ_u. For $i \in N$ we denote by $r(i)$ either the index of the new-max job which precedes i if i is not a new-max job according to σ_u, or the index of i if i is a new-max job according to σ_u (i.e., $i = a_{r(i)}$). Consequently, $p_{a_{r(i)}} = \max_{k \in \bar{P}(\sigma_u, i)}\{p_k\}$.

Theorem 3.4 *Let (M, N, p, α) be a PFS problem and let σ_u be the initial order. Let (N, v) be the associated PFS game. Then,*

$$v(T) = \sum_{k \in N} \sum_{r=1}^{r(k)} \big(G^{\{a_r, \ldots, n\}}_k - G^{\{a_{r+1}, \ldots, n\}}_k\big) u_{\{a_r, \ldots, k\}}(T)$$

for every $T \subset N$, where $G^{\{a_{r(k)+1}, \ldots, n\}}_k$ is defined as 0.

Proof Let $T \subset N$ be a connected coalition and set $T = \{i_T, \ldots, j_T\}$. We distinguish between two cases.
Case 1: $T \cap \{a_1, \ldots, a_s\} = \emptyset$. Then, $\hat{\sigma}^T_k(k) = k$ for all $k \in T$ by Lemma 2.4, and therefore $G^T_k = 0$ for all $k \in T$. Hence, $v(T) = 0$. Moreover, $\{a_r, \ldots, k\} \not\subset T$ for every new-max job a_r and every $k \geq a_r$. Hence, $u_{\{a_r, \ldots, k\}}(T) = 0$ and

$$\sum_{k \in N} \sum_{r=1}^{r(k)} \big(G^{\{a_r, \ldots, n\}}_k - G^{\{a_{r+1}, \ldots, n\}}_k\big) u_{\{a_r, \ldots, k\}}(T) = 0 = v(T).$$

Case 2: $T \cap \{a_1, \ldots, a_s\} = \{a_v, \ldots, a_w\}$ with $a_v \leq \cdots \leq a_w$. Then, $\hat{\sigma}^T_k(k) = k$ for all $i_T \leq k < a_v$ by Lemma 2.4 and $\hat{\sigma}^T_k(k) = \hat{\sigma}^{\{a_v, \ldots, n\}}_k(k)$ for all $k \geq a_v$ by the mechanism of the algorithm. Hence, $G^T_k = 0$ for all $i_T \leq k < a_v$ and $G^T_k = G^{\{a_v, \ldots, n\}}_k$ for all $j_T \geq k \geq a_v$. Therefore, $v(T) = \sum_{k=a_v}^{j_T} G^{\{a_v, \ldots, n\}}_k$. Moreover,

$$\begin{aligned}
&\sum_{k\in N}\sum_{r=1}^{r(k)}\big(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}\big)u_{\{a_r,\dots,k\}}(T)\\
&\quad=\sum_{k=a_v}^{j_T}\sum_{r=v}^{r(k)}\big(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}\big)u_{\{a_r,\dots,k\}}(T)\\
&\quad=\sum_{k=a_v}^{j_T}\Big[\big(G_k^{\{a_v,\dots,n\}} - G_k^{\{a_{v+1},\dots,n\}}\big)\\
&\qquad+\big(G_k^{\{a_{v+1},\dots,n\}} - G_k^{\{a_{v+2},\dots,n\}}\big)+\cdots\\
&\qquad+\big(G_k^{\{a_{r(k)-1},\dots,n\}} - G_k^{\{a_{r(k)},\dots,n\}}\big)+G_k^{\{a_{r(k)},\dots,n\}}\Big]\\
&\quad=\sum_{k=a_v}^{j_T}G_k^{\{a_v,\dots,n\}} = v(T),
\end{aligned}$$

where the first equality follows because if k and r are such that

(i) $a_r < a_v \le k$, then $\{a_r,\dots,k\} \not\subset \{i_T,\dots,j_T\} = T$ and $u_{\{a_r,\dots,k\}}(T) = 0$;
(ii) $k > j_T$ and $a_r \le k$, then $\{a_r,\dots,k\} \not\subset \{i_T,\dots,j_T\} = T$ and $u_{\{a_r,\dots,k\}}(T) = 0$.

The second equality is satisfied because if k and r are such that $a_v \le a_r \le a_{r(k)}$, $a_r \le k \le j_T$, then $\{a_r,\dots,k\} \subset \{i_T,\dots,j_T\} = T$ and $u_{\{a_r,\dots,k\}}(T) = 1$.

Let $T \subset N$. If T is unconnected, then $v(T) = \sum_{U\in T/\sigma_0} v(U)$ and

$$\begin{aligned}
&\sum_{k\in N}\sum_{r=1}^{r(k)}\big(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}\big)u_{\{a_r,\dots,k\}}(T)\\
&\quad=\sum_{U\in T/\sigma_0}\sum_{k\in N}\sum_{r=1}^{r(k)}\big(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}\big)u_{\{a_r,\dots,k\}}(U),
\end{aligned}$$

since the unanimity games are defined for connected coalitions. □

Next, we provide a result on the "monotonicity" of new-max jobs and cost savings. The proof can be found in the appendix (cf. Lemma A.5(iv)).

Lemma 3.5 *Let (M,N,p,α) be a PFS problem and let σ_u be the initial order. Let $S,T \subset N$, with $S \subset T \subset N$, be connected coalitions. Let $S = \{i_S,\dots,j_S\}$ with $i_S < j_S$, and let a be the new-max job according to σ_u such that $p_a = \max_{k\in P(\sigma_u,i_S)}\{p_k\}$. Then,*

$$G_i^S \le G_i^T$$

for every $i \in S$. Moreover, if $p_i \ge p_a$, then $G_i^S = G_i^T$ for every $i \in S$.

Corollary 3.6 *PFS games where the initial order is an urgency order are convex.*

Proof By Theorem 3.4 and Lemma 3.5 we know that PFS games have a nonnegative decomposition into unanimity games. Since unanimity games are convex, it follows that PFS games are also convex. □

If the initial order is an urgency order, PFS games are convex and the Shapley value belongs to the core. As a direct consequence of Theorem 3.4 and (3.1) we have the following result that provides a game independent expression of the Shapley value for PFS games where the initial order is an urgency order.

Theorem 3.7 *Let (M,N,p,α) be a PFS problem and let σ_u be the initial order. Then, the Shapley value of the associated PFS game (N,v) is given by*

$$\Phi_i(v) = \sum_{k=i}^{n}\sum_{r=1}^{r(i)}\frac{G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}}{|\{a_r,\dots,k\}|}$$

for every $i \in N$.

The Shapley value of PFS games can be interpreted as follows: player $i \in N$ needs players $a_{r(i)},\dots,i-1$ in order to obtain some cost savings, and the Shapley value shares the gain $G_i^{\{a_{r(i)},\dots,i\}} = G_i^{\{a_{r(i)},\dots,n\}}$ equally among all the players involved, i.e., $a_{r(i)},\dots,i$. If a new segment is added to the left of this group of jobs, i.e., if $a_{r(i)-1},\dots,a_{r(i)},\dots,i-1$ cooperate with i, extra gains $G_i^{\{a_{r(i)-1},\dots,i\}} - G_i^{\{a_{r(i)},\dots,i\}} = G_i^{\{a_{r(i)-1},\dots,n\}} - G_i^{\{a_{r(i)},\dots,n\}} \ge 0$ can be obtained by Lemma 3.5. The Shapley value shares equally these extra gains among all the players involved, i.e., $a_{r(i)-1},\dots,a_{r(i)},\dots,i$. Step by step, additive-gains are shared equally among all who are involved.

Example 3.8 Let (M,N,p,α) be the PFS problem of Example 3.3. Let $\sigma_0 = \sigma_u = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$ and let (N,v) be the associated PFS game. Initially, there are only three new-max jobs, which are jobs 1, 2, and 9. In order to calculate the Shapley value of this game we need to compute G_i^N, with $i \in N$, $G_i^{\{2,\dots,9\}}$, with $i \in \{2,\dots,9\}$, and $G_9^{\{9\}}$. These values are represented in Table 3.

Table 4 illustrates the calculation of the Shapley value according to Theorem 3.7. Here, we have only given the values of the (extra) gains.

Table 3 Cost savings achieved by job i in coalitions N, $\{2,\dots,9\}$, and $\{9\}$ in Example 3.8

	1	2	3	4	5	6	7	8	9
G_i^N	0	0	2500	0	1480	800	0	130	0
$G_i^{\{2,\dots,9\}}$		0	1300	0	720	800	0	0	0
$G_i^{\{9\}}$									0

Table 4 Computation of the Shapley value for the PFS game in Example 3.8

Players	1	2	3	4	5	6	7	8	9
$G_3^N - G_3^{\{2,\ldots,9\}} = 1200$	400	400	400						
$G_5^N - G_5^{\{2,\ldots,9\}} = 760$	152	152	152	152	152				
$G_8^N - G_8^{\{2,\ldots,9\}} = 130$	16.25	16.25	16.25	16.25	16.25	16.25	16.25	16.25	
$G_3^{\{2,\ldots,9\}} - G_3^{\{9\}} = 1300$		650	650						
$G_5^{\{2,\ldots,9\}} - G_5^{\{9\}} = 720$		180	180	180	180				
$G_6^{\{2,\ldots,9\}} - G_6^{\{9\}} = 800$		160	160	160	160	160			
Shapley value	568.25	1558.25	1558.25	508.25	508.25	176.25	16.25	16.25	0

Appendix

For the proof of Lemma 3.5, we need the following additional lemmas. The first lemma states that, in a PFS problem, the new-max jobs according to the urgency order remain new-max jobs during the proposed process of finding an optimal order for an arbitrary coalition S.

Lemma A.1 *Let (M, N, p, α) be a PFS problem and let $\sigma_u = (1, 2, \ldots, n)$ be the initial order. Let $S \subset N$. Then, every new-max job according to σ_u is also new-max job according to $\hat{\sigma}_i^S$ for every $i \in S$.*

Proof Let a be a new-max job according to σ_u and let $i \in S$. Then, $p_a > \max_{j \in P(\sigma_u, a)}\{p_j\}$. We have to show that a is a new-max job according to $\hat{\sigma}_i^S$, i.e., $p_a > \max_{j \in P(\hat{\sigma}_i^S, a)}\{p_j\}$.

Note that $P(\sigma_u, a) \subset P(\hat{\sigma}_i^S, a)$. Moreover, $p_j < p_a$ for all $j \in P(\hat{\sigma}_i^S, a) \setminus P(\sigma_u, a)$ by Lemma 2.3. Hence,

$$p_a > \max\Big\{\max_{j \in P(\sigma_u, a)}\{p_j\}, \max_{j \in P(\hat{\sigma}_i^S, a) \setminus P(\sigma_u, a)}\{p_j\}\Big\}$$
$$= \max_{j \in P(\hat{\sigma}_i^S, a)}\{p_j\}.$$

□

The following result is a direct consequence of the algorithm, and therefore the proof is omitted. It says that the set of predecessors of a certain job once reordered can only increase with the consecutive application of the algorithm.

Lemma A.2 *Let (M, N, p, α) be a PFS problem and let $\sigma_u = (1, 2, \ldots, n)$ be the initial order. Let $S \subset N$ and $i, j \in S$ with $j < i$. Then, $P(\hat{\sigma}_j^S, j) \subset P(\hat{\sigma}_i^S, j)$.*

Next, we show that if a job becomes new-max job during its reordering, then it remains new-max job during the successive application of the algorithm.

Lemma A.3 *Let (M, N, p, α) be a PFS problem and let $\sigma_u = (1, 2, \ldots, n)$ be the initial order. Let $S \subset N$ and $i, j \in S$ with $j < i$. Then, j is new-max job according to $\hat{\sigma}_i^S$ if and only if j is new-max job according to $\hat{\sigma}_j^S$.*

Proof If j is new-max job according to σ_u, then j is new-max job according to $\hat{\sigma}_k^S$ for all $k \in S$ by Lemma A.1, and the result follows. Hence, we may assume that j is not new-max job according to σ_u.

We first show the "only if" part. Let j be a new-max job according to $\hat{\sigma}_i^S$. Then,

$$p_j > \max_{k \in P(\hat{\sigma}_i^S, j)}\{p_k\} \geq \max_{k \in P(\hat{\sigma}_j^S, j)}\{p_k\},$$

where the first inequality follows since j is new-max job according to $\hat{\sigma}_i^S$ and the second one by Lemma A.2. Hence, j is a new-max job according to $\hat{\sigma}_j^S$.

Next, we show the "if" part. Let j be a new-max job according to $\hat{\sigma}_j^S$. By Lemma A.2 we have that $P(\hat{\sigma}_j^S, j) \subset P(\hat{\sigma}_i^S, j)$. Hence, we can write $P(\hat{\sigma}_i^S, j) = P(\hat{\sigma}_j^S, j) \cup (P(\hat{\sigma}_i^S, j) \setminus P(\hat{\sigma}_j^S, j))$. Observe that $p_j > p_k$ for all $k \in P(\hat{\sigma}_j^S, j)$, since j is a new-max job according to $\hat{\sigma}_j^S$ by assumption. Besides, $p_j > p_k$ for all $k \in P(\hat{\sigma}_i^S, j) \setminus P(\hat{\sigma}_j^S, j)$ by Lemma 2.3. Hence, j is a new-max job according to $\hat{\sigma}_i^S$. □

For the proof of Lemma 3.5 we need some additional notation. Recall that $\{a_1, \ldots, a_s\}$, with $a_1 < \cdots < a_s$, is the set of new-max jobs according to σ_u. Let $S \subset N$ be a connected coalition, $S = \{i_S, \ldots, j_S\}$, in the conditions of Lemma 3.5. Note that this is equivalent to

$$S \cap \{a_1, \ldots, a_s\} = \{a_u, \ldots, a_v\}$$

with $a_u \leq \cdots \leq a_v$ and $a_u \neq a_1$.

Partition S as follows:

$$\big\{\{i_S, \ldots, l_1 - 1\}, \{l_1\}, \{l_1 + 1, \ldots, l_2 - 1\}, \{l_2\}, \ldots, \{l_m\},$$
$$\{l_m + 1, \ldots, j_S\}\big\}, \tag{A.1}$$

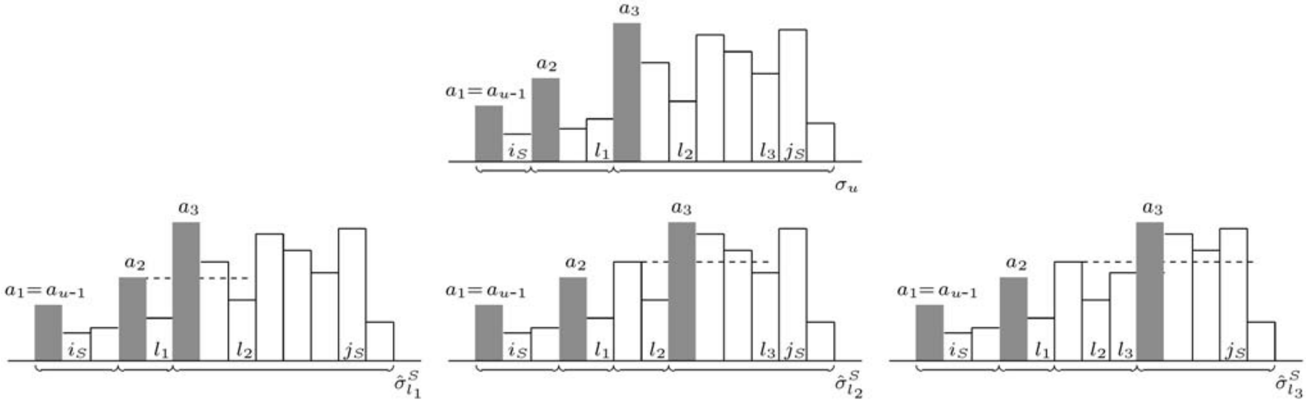

Fig. 7 Partition of S

where $l_1 \in S$ is the first job with a processing time smaller than a_{u-1} that is not assigned to the segment defined by a_{u-1} after reordering. Formally:

$$p_{l_1} < p_{a_{u-1}},$$

$$a_{u-1} \neq a^S\big(l_1, r^S(l_1)\big),$$

$$a_{u-1} = a^S\big(l, r^S(l)\big)$$

$$\text{for every } l \in S \text{ with } l < l_1 \text{ and } p_l < p_{a_{u-1}},$$

and l_k, $k \geq 2$, is determined by making use of the processing time of the new-max job which is in front of l_{k-1} after its reordering in the following way:

$$p_{l_k} < p_{a^S(l_{k-1}, r^S(l_{k-1}))},$$

$$p_j \geq p_{a^S(l_{k-1}, r^S(l_{k-1}))} \quad \text{for every } j \in \{l_{k-1}+1, \ldots, l_k - 1\}.$$

Figure 7 illustrates how this partition is obtained. In each order, all segments are indicated for clarity.

The idea behind this partition is the following: l_1 is the first job with a processing time smaller than a_{u-1} that is not placed in front of $A^S(l_1, s_{l_1})$ (i.e., $r^S(l_1) \neq s_{l_1} + 1$). Still, in a bigger coalition T containing a_{u-1}, l_1 could overtake a_{u-1} during its reordering. The remaining jobs $l_2, \ldots, l_m$ are those jobs that can overtake $a^S(l_{k-1}, r^S(l_{k-1}))$ during their reordering. It turns out that $l_2, \ldots, l_m$ are the only jobs that cannot overtake l_1 in S but could be moved to earlier positions in T (cf. Lemma A.4 and Lemma A.5(i), (ii)).

We show that $\hat{\sigma}^S_i(l_1) < \hat{\sigma}^S_i(l_2) < \cdots < \hat{\sigma}^S_i(l_m)$ for every $i \in S$. For the proof of this result we need to reorder segments $A^S(i,r)$ from left to right. This will be done by introducing segments $B^S(i,r)$. We define $B^S(i,1), \ldots, B^S(i,s_i)$ as $B^S(i,r) = A^S(i, s_i - r + 1)$ for every $r \in \{1, \ldots, s_i\}$. Now $r^S(i)$, $a^S(i,r)$, and σ^S_{ir} have the same interpretation as before, but it refers to $B^S(i,r)$. Hence, σ^S_{ir} here is the order we obtain from $\hat{\sigma}^S_{i-1}$ by placing i at the tail of $B^S(i,r)$. Note that $\sigma^S_{is_i} = \hat{\sigma}^S_{i-1}$. We denote by σ^S_{i0} the order obtained from $\hat{\sigma}^S_{i-1}$ by placing i in front of $a^S(i,1)$.

With our new notation, (3) and (4) become

(5) $c_N(\sigma_{ir}) - c_N(\sigma_{ir^S(i)}) > 0$ for every $r > r^S(i)$;
(6) $c_N(\sigma_{ir^S(i)}) - c_N(\sigma_{ir}) \leq 0$ for every $r < r^S(i)$.

We denote by $a(i,S)$ the new-max job according to $\hat{\sigma}^S_{i-1}$ such that i is placed at the tail of the segment defined by $a(i,S)$ after being reordered i.e., $a(i,S) = a^S(i, r^S(i))$. Note that $p_{a(i,S)} = \max_{k \in P(\hat{\sigma}^S_i, i)}\{p_k\}$. Let $i \in S$ and let $a \in S$ be a new-max job according to $\hat{\sigma}^S_{i-1}$. We denote by $r^S(i,a)$ the index of the segment to which a belongs according to $\hat{\sigma}^S_{i-1}$ i.e., $r^S(i,a) \in \{1, \ldots, s_i\}$ such that $a \in B^S(i, r^S(i,a))$. In the proof of Lemma A.4 we consider $c_N(\sigma^S_{ir}) - c_N(\sigma^S_{ir-1})$, $r = 1, \ldots, s_i$, (i.e., the gains obtained when, starting from $\hat{\sigma}^S_{i-1}$, we change i from the tail of segment $B^S(i,r)$ to the tail of segment $B^S(i,r-1)$). From Lemma 2.1(ii) and the definition of $B^S(i,r)$ we obtain

$$\begin{aligned} c_N\big(\sigma^S_{ir}\big) - c_N\big(\sigma^S_{ir-1}\big) &= \sum_{j \in B^S(i,r)} (\alpha_i p_j - \alpha_j p_i) \\ &\quad + \alpha_i (m-1)\Big(\max_{j \in B^S(i,r) \cup \{i\}} \{p_j\} - \max_{j \in B^S(i,r-1) \cup \{i\}} \{p_j\}\Big) \\ &\quad + \sum_{j \in B^S(i,r)} \alpha_j (m-1)\big(p_{a^S(i,r)} - \max\{p_i, p_{a^S(i,r)}\}\big), \end{aligned} \tag{A.2}$$

where $B^S(i,0) = A^{\hat{\sigma}^S_{i-1}}_{\tilde{r}}$ with $a_{u-1} \in A^{\hat{\sigma}^S_{i-1}}_{\tilde{r}}$. Note that if $p_i \leq p_{a^S(i,r)}$, then

$$\begin{aligned} c_N\big(\sigma^S_{ir}\big) - c_N\big(\sigma^S_{ir-1}\big) &= \sum_{j \in B^S(i,r)} (\alpha_i p_j - \alpha_j p_i) \end{aligned}$$

$$+\alpha_i(m-1)\Big(\max_{j\in B^S(i,r)\cup\{i\}}\{p_j\}-\max_{j\in B^S(i,r-1)\cup\{i\}}\{p_j\}\Big). \tag{A.3}$$

Lemma A.4 $\hat{\sigma}^S_{l_k}(l_k) > \hat{\sigma}^S_{l_k}(l_{k-1})$ *for every* $k \in \{2,\dots,m\}$.

Proof The outline of the proof is the following: we start by noting that jobs in between $l_{k-1}+1,\dots,l_k-1$ cannot overtake l_{k-1} when they are reordered. This means that the segments in front of l_{k-1} are the same in $\hat{\sigma}^S_{l_{k-1}}$ and $\hat{\sigma}^S_{l_k-1}$. This allows us to see that $c_N(\sigma^S_{l_k r^S(l_{k-1})}) - c_N(\sigma^S_{l_k r}) \le 0$ for every $r < r^S(l_{k-1})$, which implies that l_k does not overtake l_{k-1} after reordering.

First, we prove the result for $k=2$. Since $p_j \ge p_{a(l_1,S)}$ for every $j \in \{l_1+1,\dots,l_2-1\}$ we have $\hat{\sigma}^S_j(j) > \hat{\sigma}^S_j(l_1)$ by Lemma 2.3 and Lemma 2.4. Therefore,

$$\hat{\sigma}^S_{l_2-1}(j) > \hat{\sigma}^S_{l_2-1}(l_1) \quad \text{for every } j \in \{l_1+1,\dots,l_2-1\}. \tag{A.4}$$

By (A.4) and Lemma A.3 it follows that the set of new-max jobs preceding $a(l_1,S)$ according to $\hat{\sigma}^S_{l_1-1}$ and $\hat{\sigma}^S_{l_2-1}$ coincide. Consequently,

$$r^S\big(l_2,a(l_1,S)\big) = r^S\big(l_1,a(l_1,S)\big).$$

To shorten notation, let $r^S(a(l_1,S)) = r^S(l_1,a(l_1,S))$. Moreover,

$$B^S(l_2,r) = B^S(l_1,r)$$
$$\text{for every } r \in \big\{1,\dots,r^S\big(a(l_1,S)\big)-1\big\} \tag{A.5}$$

and

$$B^S\big(l_1,r^S\big(a(l_1,S)\big)\big) \subset B^S\big(l_2,r^S\big(a(l_1,S)\big)\big). \tag{A.6}$$

Before showing our result for $k=2$, we prove that

$$c_N\big(\sigma^S_{l_2 r}\big) - c_N\big(\sigma^S_{l_2 r-1}\big) \le \frac{p_{l_2}}{p_{l_1}}\big(c_N\big(\sigma^S_{l_1 r}\big) - c_N\big(\sigma^S_{l_1 r-1}\big)\big) \tag{A.7}$$

for $r \in \{1,\dots,r^S(a(l_1,S))-1\}$.

$$\begin{aligned}
&c_N\big(\sigma^S_{l_2 r}\big) - c_N\big(\sigma^S_{l_2 r-1}\big)\\
&\quad= \sum_{j\in B^S(l_2,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2})\\
&\qquad+\alpha_{l_2}(m-1)\Big(\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}\Big)\\
&\qquad+\sum_{j\in B^S(l_2,r)} \alpha_j(m-1)\big(p_{a^S(l_2,r)}\\
&\qquad-\max\{p_{l_2},p_{a^S(l_2,r)}\}\big)\\
&\quad\le \sum_{j\in B^S(l_2,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2})\\
&\qquad+\alpha_{l_2}(m-1)\Big(\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}\Big)\\
&\quad= \sum_{j\in B^S(l_1,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2})\\
&\qquad+\alpha_{l_2}(m-1)\Big(\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}\Big)\\
&\quad\le \sum_{j\in B^S(l_1,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2})\\
&\qquad+\alpha_{l_2}(m-1)\Big(\max_{j\in B^S(l_2,r)}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_2,r-1)}\{p_j\}\Big)\\
&\quad= \sum_{j\in B^S(l_1,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2})\\
&\qquad+\alpha_{l_2}(m-1)\Big(\max_{j\in B^S(l_1,r)\cup\{l_1\}}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_1,r-1)\cup\{l_1\}}\{p_j\}\Big)\\
&\quad\le \sum_{j\in B^S(l_1,r)} \Big(\frac{p_{l_2}}{p_{l_1}}\alpha_{l_1}p_j - \alpha_j p_{l_2}\Big)\\
&\qquad+\frac{p_{l_2}}{p_{l_1}}\alpha_{l_1}(m-1)\Big(\max_{j\in B^S(l_1,r)\cup\{l_1\}}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_1,r-1)\cup\{l_1\}}\{p_j\}\Big)\\
&\quad=\frac{p_{l_2}}{p_{l_1}}\Big(\sum_{j\in B^S(l_1,r)} (\alpha_{l_1}p_j - \alpha_j p_{l_1})\\
&\qquad+\alpha_{l_1}(m-1)\Big(\max_{j\in B^S(l_1,r)\cup\{l_1\}}\{p_j\}\\
&\qquad-\max_{j\in B^S(l_1,r-1)\cup\{l_1\}}\{p_j\}\Big)\Big)\\
&\quad=\frac{p_{l_2}}{p_{l_1}}\big(c_N\big(\sigma^S_{l_1 r}\big) - c_N\big(\sigma^S_{l_1 r-1}\big)\big).
\end{aligned} \tag{A.8}$$

Here, the first equality follows by (A.2), the first inequality is a consequence of

$$\sum_{j\in B^S(l_2,r)} \alpha_j(m-1)\bigl(p_{a^S(l_2,r)} - \max\{p_{l_2}, p_{a^S(l_2,r)}\}\bigr) \le 0,$$

and the second equality follows by (A.5). For the second inequality note that

$$\max_{j\in B^S(l_2,r-1)}\{p_j\} < \max_{j\in B^S(l_2,r)}\{p_j\}.$$

Hence, if $p_{l_2} \ge \max_{j\in B^S(l_2,r)}\{p_j\} > \max_{j\in B^S(l_2,r-1)}\{p_j\}$, then

$$\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\} - \max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}$$
$$= 0 < \max_{j\in B^S(l_2,r)}\{p_j\} - \max_{j\in B^S(l_2,r-1)}\{p_j\};$$

if $\max_{j\in B^S(l_2,r)}\{p_j\} > p_{l_2} \ge \max_{j\in B^S(l_2,r-1)}\{p_j\}$, then

$$\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\} - \max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}$$
$$\le \max_{j\in B^S(l_2,r)}\{p_j\} - \max_{j\in B^S(l_2,r-1)}\{p_j\};$$

and finally, if $\max_{j\in B^S(l_2,r)}\{p_j\} > \max_{j\in B^S(l_2,r-1)}\{p_j\} \ge p_{l_2}$, then

$$\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\} - \max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}$$
$$= \max_{j\in B^S(l_2,r)}\{p_j\} - \max_{j\in B^S(l_2,r-1)}\{p_j\}.$$

The third equality is a consequence of (A.5) together with the fact that l_1 does not become a new-max job, since $p_{l_1} < p_{a_{u-1}} < p_{a(l_1,S)}$ by definition of l_1. The third inequality follows, since $l_2 > l_1$ implies $u_{l_2} = \frac{\alpha_{l_2}}{p_{l_2}} \le \frac{\alpha_{l_1}}{p_{l_1}} = u_{l_1}$, therefore $\alpha_{l_2} \le \frac{p_{l_2}}{p_{l_1}}\alpha_{l_1}$. The last equality follows from (A.3) which can be applied because, by definition of l_1, $p_{l_1} < p_{a_{u-1}} < p_{a^S(l_1,r)}$ holds.

Analogously to (A.8), one can see that

$$c_N\bigl(\sigma^S_{l_2r^S(a(l_1,S))}\bigr) - c_N\bigl(\sigma^S_{l_2r^S(a(l_1,S))-1}\bigr)$$
$$\le \frac{p_{l_2}}{p_{l_1}}\bigl(c_N\bigl(\sigma^S_{l_1r^S(a(l_1,S))}\bigr) - c_N\bigl(\sigma^S_{l_1r^S(a(l_1,S))-1}\bigr)\bigr).$$

The only difference is that the second equality becomes an inequality by (A.6) and the fact that $\alpha_{l_2}p_j - \alpha_j p_{l_2} \le 0$, since $u_{l_2} \le u_j$ for every $j \in B^S(l_2, r^S(a(l_1,S))) \setminus B^S(l_1, r^S(a(l_1,S)))$.

Hence, for every $r \in \{0,\dots,r^S(a(l_1,S))-1\}$ we have

$$c_N\bigl(\sigma^S_{l_2r^S(l_2)}\bigr) - c_N\bigl(\sigma^S_{l_2r}\bigr)$$
$$\le c_N\bigl(\sigma^S_{l_2r^S(a(l_1,S))}\bigr) - c_N\bigl(\sigma^S_{l_2r}\bigr)$$
$$= \sum_{\tilde r=r}^{r^S(a(l_1,S))-1} \bigl(c_N\bigl(\sigma^S_{l_2\tilde r+1}\bigr) - c_N\bigl(\sigma^S_{l_2\tilde r}\bigr)\bigr)$$
$$\le \sum_{\tilde r=r}^{r^S(a(l_1,S))-1} \frac{p_{l_2}}{p_{l_1}}\bigl(c_N\bigl(\sigma^S_{l_1\tilde r+1}\bigr) - c_N\bigl(\sigma^S_{l_1\tilde r}\bigr)\bigr)$$
$$= \frac{p_{l_2}}{p_{l_1}}\bigl(c_N\bigl(\sigma^S_{l_1r^S(a(l_1,S))}\bigr) - c_N\bigl(\sigma^S_{l_1r}\bigr)\bigr)$$
$$\le 0,$$

where the first inequality holds by definition of $r^S(l_2)$, the second inequality is a direct consequence of (A.7), and the third one follows by (6). Therefore, $r^S(l_2) \ge r^S(l_1)$ by (6). If $r^S(l_2) > r^S(l_1)$, then it is immediate that $\hat\sigma^S_{l_2}(l_2) > \hat\sigma^S_{l_2}(l_1)$. If $r^S(l_2) = r^S(l_1)$ then $\hat\sigma^S_{l_2}(l_2) > \hat\sigma^S_{l_2}(l_1)$ by Lemma 2.4 and the fact that $u_{l_1} \ge u_{l_2}$. Therefore, $\hat\sigma^S_{l_2}(l_2) > \hat\sigma^S_{l_2}(l_1)$.

Second, let $k > 2$ and suppose that the result is true for $l_1,\dots,l_{k-1}$. Then,

$$\hat\sigma^S_{l_k-1}(l_1) < \hat\sigma^S_{l_k-1}(l_2) < \cdots < \hat\sigma^S_{l_k-1}(l_{k-1}).$$

For every $\tilde k \in \{1,\dots,k-1\}$ and every $j \in \{l_{\tilde k}+1,\dots,l_{\tilde k+1}-1\}$ we have that $p_j \ge p_{a(l_{\tilde k},S)}$. Hence, $\hat\sigma^S_j(j) > \hat\sigma^S_j(l_{\tilde k})$ by Lemmas 2.3 and 2.4 (i.e., a job j that is initially in between $l_{\tilde k}$ and $l_{\tilde k+1}$ can never overtake $l_{\tilde k}$ during its reordering). Therefore, for every $\tilde k \in \{1,\dots,k-1\}$ it follows that

$$\hat\sigma^S_{l_k-1}(j) > \hat\sigma^S_{l_k-1}(l_{\tilde k})$$
$$\text{for every } j \in \{l_{\tilde k}+1,\dots,l_{\tilde k+1}-1\}. \tag{A.9}$$

By (A.9) and Lemma A.3 it follows that for every $\tilde k \in \{2,\dots,k-1\}$ the set of new-max jobs in between $a(l_{\tilde k-1},S)$ and $a(l_{\tilde k},S)$ according to $\hat\sigma^S_{l_{\tilde k}-1}$ and $\hat\sigma^S_{l_k-1}$ coincide (i.e., the number of segments in between $a(l_{\tilde k-1},S)$ and $a(l_{\tilde k},S)$ remains constant after the reordering of $l_{\tilde k+1}$). Therefore, for $\tilde k \in \{1,\dots,k-1\}$ we have

$$r^S\bigl(l_k, a(l_{\tilde k},S)\bigr) = r^S\bigl(l_{\tilde k}, a(l_{\tilde k},S)\bigr).$$

To shorten notation, let $r^S(a(l_{\tilde k},S)) = r^S(l_{\tilde k}, a(l_{\tilde k},S))$. In addition, for every $\tilde k \in \{1,\dots,k-1\}$ we have

$$B^S(l_k,r) = B^S(l_{\tilde k},r)$$

for every $r \in \{r^S(a(l_{\tilde k-1},S))+1,\dots,r^S(a(l_{\tilde k},S))-1\}$, and

$$B^S\bigl(l_{\tilde k}, r^S\bigl(a(l_{\tilde k},S)\bigr)\bigr) \subset B^S\bigl(l_k, r^S\bigl(a(l_{\tilde k},S)\bigr)\bigr).$$

Using the same kind of arguments as in (A.8), one can see that

$$c_N\bigl(\sigma^S_{l_kr}\bigr) - c_N\bigl(\sigma^S_{l_kr-1}\bigr) \le \frac{p_{l_k}}{p_{l_{\tilde k}}}\bigl(c_N\bigl(\sigma^S_{l_{\tilde k}r}\bigr) - c_N\bigl(\sigma^S_{l_{\tilde k}r-1}\bigr)\bigr) \tag{A.10}$$

for every $\tilde k \in \{1,\dots,k-1\}$ and every $r \in \{r^S(l_k, a(l_{\tilde k-1}, S))+1,\dots,r^S(l_k, a(l_{\tilde k},S))\}$.

Next, we want to prove that $\sigma^S_{l_k}(l_k) > \sigma^S_{l_k}(l_{k-1})$. For this, we first show that $c_N(\sigma^S_{l_k r^S(l_k)}) - c_N(\sigma^S_{l_k\bar{r}}) \le 0$ for every $\bar{r} < r^S(a(l_{k-1}, S))$. In order to use (A.7) and (A.10), we need to know where $B^S(l_k, \bar{r})$ is located with respect to $l_1, \dots, l_{k-1}$. Let $\bar{r} \in \{0, \dots, r^S(a(l_{k-1}, S)) - 1\}$ and $k^* \in \{1, \dots, k-1\}$ be such that $r^S(a(l_{k^*-1}, S)) \le \bar{r} \le r^S(a(l_{k^*}, S))$ with $r^S(a(l_0, S)) = 1$. We have that

$$
\begin{aligned}
&c_N\big(\sigma^S_{l_k r^S(l_k)}\big) - c_N\big(\sigma^S_{l_k\bar{r}}\big)\\
&\quad\le c_N\big(\sigma^S_{l_k r^S(a(l_{k-1},S))}\big) - c_N\big(\sigma^S_{l_k\bar{r}}\big)\\
&\quad= \sum_{r=\bar{r}}^{r^S(a(l_{k^*},S))-1} \big(c_N\big(\sigma^S_{l_k r+1}\big) - c_N\big(\sigma^S_{l_k r}\big)\big)\\
&\qquad+ \sum_{\tilde{k}=k^*+1}^{k-1} \sum_{r=r^S(a(l_{\tilde{k}-1},S))}^{r^S(a(l_{\tilde{k}},S))-1} \big(c_N\big(\sigma^S_{l_k r+1}\big) - c_N\big(\sigma^S_{l_k r}\big)\big)\\
&\quad\le \sum_{r=\bar{r}}^{r^S(a(l_{k^*},S))-1} \frac{p_{l_k}}{p_{l_{k^*}}}\big(c_N\big(\sigma^S_{l_{k^*} r+1}\big) - c_N\big(\sigma^S_{l_{k^*} r}\big)\big)\\
&\qquad+ \sum_{\tilde{k}=k^*+1}^{k-1} \sum_{r=r^S(a(l_{\tilde{k}-1},S))}^{r^S(a(l_{\tilde{k}},S))-1} \frac{p_{l_k}}{p_{l_{\tilde{k}}}}\big(c_N\big(\sigma^S_{l_{\tilde{k}} r+1}\big) - c_N\big(\sigma^S_{l_{\tilde{k}} r}\big)\big)\\
&\quad= \frac{p_{l_k}}{p_{l_{k^*}}}\big(c_N\big(\sigma^S_{l_{k^*} r^S(a(l_{k^*},S))}\big) - c_N\big(\sigma^S_{l_{k^*}\bar{r}}\big)\big)\\
&\qquad+ \sum_{\tilde{k}=k^*+1}^{k-1} \frac{p_{l_k}}{p_{l_{\tilde{k}}}}\big(c_N\big(\sigma^S_{l_{\tilde{k}} r^S(a(l_{\tilde{k}},S))}\big) - c_N\big(\sigma^S_{l_{\tilde{k}} r^S(a(l_{\tilde{k}-1},S))}\big)\big)\\
&\quad\le 0,
\end{aligned}
$$

where the first inequality holds by definition of $r^S(l_k)$, the second one is a direct consequence of (A.10), the second equality is obtained by adding up terms, and the third inequality follows by (6). Therefore, $r^S(l_k) \ge r^S(l_{k-1})$ by (6). If $r^S(l_k) > r^S(l_{k-1})$, then it is immediate that $\hat{\sigma}^S_{l_k}(l_k) > \hat{\sigma}^S_{l_k}(l_{k-1})$. If $r^S(l_k) = r^S(l_{k-1})$ then $\hat{\sigma}^S_{l_k}(l_k) > \hat{\sigma}^S_{l_k}(l_{k-1})$ by Lemma 2.4 and the fact that $u_{l_{k-1}} \ge u_{l_k}$. Therefore, $\hat{\sigma}^S_{l_k}(l_k) > \hat{\sigma}^S_{l_k}(l_{k-1})$. □

We now turn to the proof of Lemma 3.5. Note that Lemma 3.5 corresponds to Lemma A.5(iv) below.

Lemma A.5 *Let (M, N, p, α) be a PFS problem and let $\sigma_u = (1, 2, \dots, n)$ be the initial order. Let $S, T \subset N$, with $S \subset T \subset N$, be connected coalitions. Let $S = \{i_S, \dots, j_S\}$ with $i_S < j_S$, and let a be the new-max job according to σ_u such that $p_a = \max_{k\in P(\sigma_u, i_S)}\{p_k\}$. Then, the following assertions hold*:

(i) *$\hat{\sigma}^T_i(i) = \hat{\sigma}^S_i(i)$ for every $i \in S$ with $p_i \ge p_a$.*

(ii) *$\hat{\sigma}^T_i(i) \le \hat{\sigma}^S_i(i)$ for every $i \in S$ with $p_i < p_a$. Moreover, if $\hat{\sigma}^T_i(i) < \hat{\sigma}^S_i(i)$, then $\hat{\sigma}^T_i(i) < \hat{\sigma}^T_i(a)$ for every $i \in S$.*

(iii) *Every new-max job in S according to $\hat{\sigma}^S_i$ is also a new-max job in S according to $\hat{\sigma}^T_i$ for every $i \in S$.*

(iv) *$G^S_i \le G^T_i$ for every $i \in S$. Moreover, for every $i \in S$ if $p_i \ge p_a$, then $G^S_i = G^T_i$.*

Proof Recall that $\{a_1, \dots, a_s\}$ is the set of new-max jobs according to σ_u with $a_1 < \dots < a_s$. We distinguish among three cases.

Case 1: Either $S \cap \{a_1, \dots, a_s\} = \emptyset$, or $S \cap \{a_1, \dots, a_s\} = \{j_S\}$. Then, $\hat{\sigma}^S_i = \sigma_u$ for every $i \in S$ and Assertions (i) and (ii) are direct consequence of the definition of $\hat{\sigma}^T_i$, Assertion (iii) follows by Lemma A.1, and Assertion (iv) follows, since $G^T_i \ge 0 = G^S_i$ by definition of G^T_i.

Case 2: $S \cap \{a_1, \dots, a_s\} = \{a_u, \dots, a_v\}$ and $T \cap \{a_1, \dots, a_s\} = \{a_u, \dots, a_w\}$ with $a_u \le \dots \le a_v \le \dots \le a_w$. Then, we have $\hat{\sigma}^S_i = \hat{\sigma}^T_i$ for every $i \in S$. Hence, Assertions (i), (ii), (iii), and (iv) are immediate.

Case 3: $S \cap \{a_1, \dots, a_s\} = \{a_u, \dots, a_v\}$ and $T \cap \{a_1, \dots, a_s\} = \{a_{\tilde{u}}, \dots, a_{\tilde{v}}\}$ with $a_{\tilde{u}} < a_u \le a_v \le a_{\tilde{v}}$.

Since the proof of this case is quite technical, we first provide an informal idea of the proof. We consider $S = \{i_S, \dots, j_S\}$ and the partition of S as defined in (A.1). Then, we compare the reordering of a player $i \in S$ in S and T as follows: if $i \in \{i_S, \dots, l_1 - 1\}$, then it easily follows that $\hat{\sigma}^T_i(i) = \hat{\sigma}^S_i(i)$ if $p_i \ge p_{a_{u-1}}$ by Lemma 2.3; moreover, $\hat{\sigma}^T_i(i) \le \hat{\sigma}^S_i(i)$ if $p_i < p_{a_{u-1}}$, since i can be placed in T in front of the same new-max job as in S (obtaining the same savings as in S) or can overtake a_{u-1} if this gives more savings. By definition of l_1, every $i \in \{i_S, \dots, l_1 - 1\}$ with $p_i < p_{a_{u-1}}$ is placed at the tail of the segment defined by a_{u-1} in S, i.e., $i \in A^S(l_1, s_{l_1} + 1)$, and therefore it is placed either at the tail of the segment defined by a_{u-1}, or in an earlier position in T, i.e., $i \in A^T(l_1, r)$ for some $r \ge s_{l_1} + 1$. Besides, every $i \in \{i_S, \dots, l_1 - 1\}$ with $p_i \ge p_{a_{u-1}}$ cannot do better in T by Lemma 2.3. Thus, $A^S(l_1, r) = A^T(l_1, r)$ for $r \in \{1, \dots, s_{l_1}\}$. Hence, when reordering l_1 in T, it can be placed either in front of the same new-max job as when reordered in S, or it can overtake a_{u-1}. Therefore, $\hat{\sigma}^T_{l_1}(l_1) \le \hat{\sigma}^S_{l_1}(l_1)$. Again, $A^S(j, r) = A^T(j, r)$ for every $r \in \{1, \dots, s_j\}$ with $l_1 \notin A^S(j, r)$ and every $j \in \{l_1 + 1, \dots, l_2 - 1\}$. Hence, we can use the same kind of reasoning as before, and so on.

To provide the formal argument we need one additional notation. Let $U \subset N$ and $i, j \in U$ with $j < i$, we denote by $r^U(i, j)$ the index of the segment to which j belongs according to $\hat{\sigma}^U_i$, i.e., $j \in A^U(i, r^U(i, j))$. (Note that this definition generalizes the definition of $r^S(i, a)$ above Lemma A.4 job j in the coalition, taking into account our current notation.)

Let $S = \{i_S, \dots, j_S\}$ and partition S according to (A.1). Let $i \in S$ and let $a \in S$ be a new-max job according to $\hat{\sigma}^S_{i-1}$ ($\hat{\sigma}^T_{i-1}$). During the remainder of this proof we denote by

$s_i(S)$ ($s_i(T)$) the number of segments in S (T) before reordering player i in S (T).

Note that $\hat{\sigma}_i^S = \sigma_0$ for every $i \in \{i_S, \dots, a_u\}$ by Lemma 2.3, and therefore Lemma A.5 follows.

First, let $i \in \{a_u + 1, \dots, l_1 - 1\}$ and assume that Lemma A.5 is satisfied for $i_S, \dots, i-1$. As a direct consequence of (i), (ii), and (iii) we have

$$A^T(j,r) = A^S(j,r),$$

for every $r \in \{1, \dots, s_j(S)\}$ and for every $j \in \{i_S + 1, \dots, i\}$. As a consequence, it follows that

$$\begin{aligned}
&c_N\big(\sigma^T_{ir}\big) - c_N\big(\sigma^T_{ir^S(i)}\big) = c_N\big(\sigma^S_{ir}\big) - c_N\big(\sigma^S_{ir^S(i)}\big)\\
&\quad \text{for every } r \in \big\{1, \dots, r^S(i) - 1\big\},\\
&c_N\big(\sigma^T_{ir^S(i)}\big) - c_N\big(\sigma^T_{ir}\big) = c_N\big(\sigma^S_{ir^S(i)}\big) - c_N\big(\sigma^S_{ir}\big)\\
&\quad \text{for every } r \in \big\{r^S(i) + 1, \dots, s_i(S) + 1\big\}.
\end{aligned} \tag{A.11}$$

We distinguish between $p_i \geq p_{a_{u-1}}$ and $p_i < p_{a_{u-1}}$.

Consider $p_i \geq p_{a_{u-1}}$. Here, we only have to show (i), (iii), and (iv). In this case,

$$\begin{aligned}
&c_N\big(\sigma^T_{ir^S(i)}\big) - c_N\big(\sigma^T_{ir}\big) \leq 0\\
&\quad \text{for every } r \in \big\{s_i(S) + 2, \dots, s_i(T) + 1\big\},
\end{aligned}$$

by Lemma 2.3. This together with (A.11) implies that $r^T(i) = r^S(i)$, and therefore (i) is satisfied. Note that $P(\hat{\sigma}_i^T, i) = P(\hat{\sigma}_i^S, i)$, and therefore i is a new-max job according to $\hat{\sigma}_i^S$ if and only if $\hat{\sigma}_i^T$ is a new-max job according to T, which implies (iii). Finally, (iv) is a direct consequence of $r^T(i) = r^S(i)$ and (A.11).

Consider $p_i < p_{a_{u-1}}$. Here, we have to show (ii), (iii), and (iv). By definition of l_1, we know that $r^S(i) = s_i(S) + 1$ and that i is not a new-max job according to $\hat{\sigma}_i^S$. Therefore, (iii) is satisfied. Note that it may happen that

$$\begin{aligned}
&c_N(\sigma^T_{ir^S(i)}) - c_N(\sigma^T_{ir}) > 0\\
&\quad \text{for some } r \in \big\{s_i(S) + 2, \dots, s_i(T) + 1\big\}.
\end{aligned} \tag{A.12}$$

Then, by (A.11) and (A.12), (ii) follows. Besides, $r^T(i) \geq r^S(i)$. Hence, $G_i^T = G_i^S$ if $r^T(i) = r^S(i)$ and $G_i^T > G_i^S$ if $r^T(i) > r^S(i)$. Therefore, (iv) is also satisfied.

Second, let $i = l_1$. Since $p_{l_1} < p_{a_{u-1}}$ by definition, we only have to show (ii), (iii), and (iv). We have shown that Lemma A.5 is satisfied for $i_S, \dots, i-1$. As a direct consequence of (i), (ii), and (iii) we have

$$A^T(l_1, r) = A^S(l_1, r), \quad r \in \big\{1, \dots s_{l_1}(S)\big\}.$$

As a consequence, it follows that

$$\begin{aligned}
&c_N\big(\sigma^T_{l_1 r}\big) - c_N\big(\sigma^T_{l_1 r^S(l_1)}\big) = c_N\big(\sigma^S_{l_1 r}\big) - c_N\big(\sigma^S_{l_1 r^S(l_1)}\big)\\
&\quad \text{for every } r \in \big\{1, \dots, r^S(l_1) - 1\big\},\\
&c_N\big(\sigma^T_{l_1 r^S(l_1)}\big) - c_N\big(\sigma^T_{l_1 r}\big) = c_N\big(\sigma^S_{l_1 r^S(l_1)}\big) - c_N\big(\sigma^S_{l_1 r}\big)\\
&\quad \text{for every } r \in \big\{r^S(l_1) + 1, \dots, s_{l_1}(S) + 1\big\}.
\end{aligned} \tag{A.13}$$

Since $p_{l_1} < p_{a_{u-1}}$ and l_1 is not new-max job according to $\hat{\sigma}_{l_1}^S$, (iii) is satisfied. Note that it may happen that

$$\begin{aligned}
&c_N\big(\sigma^T_{l_1 r^S(l_1)}\big) - c_N\big(\sigma^T_{l_1 r}\big) > 0\\
&\quad \text{for some } r \in \big\{s_{l_1}(S) + 2, \dots, s_{l_1}(T) + 1\big\}.
\end{aligned} \tag{A.14}$$

Then, by (A.13) and (A.14), (ii) follows. Besides, $r^T(l_1) \geq r^S(l_1)$. Hence, $G_{l_1}^T = G_{l_1}^S$ if $r^T(l_1) = r^S(l_1)$ and $G_{l_1}^T > G_{l_1}^S$ if $r^T(l_1) > r^S(l_1)$. Therefore, (iv) is also satisfied.

Third, let $k \in \{2, \dots, m+1\}$ and $i \in \{l_{k-1} + 1, \dots, l_k - 1\}$, where l_{m+1} is defined by $l_{m+1} = j_S + 1$. By Lemma A.4 and definition of $i \in \{l_{k-1} + 1, \dots, l_k - 1\}$, we have $p_i \geq p_{a(l_{k-1}, S)} \geq \cdots \geq p_{a(l_1, S)} > p_{a_{u-1}}$ and we only have to show (i), (iii), and (iv). We have shown that Lemma A.5 is satisfied for $i_S, \dots, l_1$. Assume that Lemma A.5 is satisfied for $l_1 + 1, \dots, i-1$ with $i \in \{l_{k-1} + 1, \dots, l_k - 1\}$. As a direct consequence of (i), (ii), and (iii) we have

$$\begin{aligned}
&A^T(i,r) = A^S(i,r)\\
&\quad \text{for all } r \in \big\{1, \dots, s_i(S)\big\} \setminus \big\{r^S(i, l_1), \dots, r^S(i, l_{k-1})\big\},\\
&A^T\big(i, r^S(i, l_{k'})\big) \subset A^S\big(i, r^S(i, l_{k'})\big)\\
&\quad \text{for all } k' \in \{1, \dots, k-1\},\\
&A^S\big(i, r^S(i, l_{k'})\big) \setminus A^T\big(i, r^S(i, l_{k'})\big) \subset \{l_1, \dots, l_{k-1}\}\\
&\quad \text{for all } k' \in \{1, \dots, k-1\}.
\end{aligned} \tag{A.15}$$

We have $\hat{\sigma}_i^S(i) > \hat{\sigma}_i^S(l_{k-1})$ by Lemma 2.3 and the fact that $p_i \geq p_{a(l_{k-1}, S)} \geq \cdots \geq p_{a(l_1, S)} > p_{a_{u-1}}$. By (A.15) and Lemma 2.3 it follows that

$$\begin{aligned}
&c_N\big(\sigma^T_{ir}\big) - c_N\big(\sigma^T_{ir^S(i)}\big) = c_N\big(\sigma^S_{ir}\big) - c_N\big(\sigma^S_{ir^S(i)}\big)\\
&\quad \text{for every } r \in \big\{1, \dots, r^S(i) - 1\big\},\\
&c_N\big(\sigma^T_{ir^S(i)}\big) - c_N\big(\sigma^T_{ir}\big) \leq 0\\
&\quad \text{for every } r \in \big\{r^S(i) + 1, \dots, s_i(T) + 1\big\},
\end{aligned} \tag{A.16}$$

which implies $r^T(i) = r^S(i)$. Therefore, (i) is satisfied. Moreover, note that by (A.15) and $r^T(i) = r^S(i)$ it follows that $P(\hat{\sigma}_i^T, i) = P(\hat{\sigma}_i^S, i)$, and therefore i is a new-max job according to $\hat{\sigma}_i^S$ if and only if i is a new-max job according to $\hat{\sigma}_i^T$, which implies (iii). Finally, (iv) is a direct consequence of $r^T(i) = r^S(i)$ and (A.16).

To finish, let $i = l_k$ with $k \in \{2, \ldots, m\}$. Assume that Lemma A.5 is satisfied for $i_S, \ldots, i-1$. As a direct consequence of (i), (ii), and (iii) we have that

$$\begin{aligned} &A^T(l_k, r) = A^S(l_k, r) \\ &\quad \text{for all } r \in \big\{1, \ldots, s_{l_k}(S)\big\} \setminus \big\{r^S(l_k, l_1), \ldots, r^S(l_k, l_k)\big\}, \\ &A^T\big(l_k, r^S(l_k, l_{k'})\big) \subset A^S\big(l_k, r^S(l_k, l_{k'})\big) \\ &\quad \text{for all } k' \in \{1, \ldots k\}, \\ &A^S\big(l_k, r^S(l_k, l_{k'})\big) \setminus A^T\big(l_k, r^S(l_k, l_{k'})\big) \subset \{l_1, \ldots, l_k\} \\ &\quad \text{for all } k' \in \{1, \ldots k\}. \end{aligned} \tag{A.17}$$

By definition of l_k it follows that $p_{l_k} < p_{a(l_{k-1},S)}$. Moreover, $\hat{\sigma}^S_{l_k}(l_k) > \hat{\sigma}^S_{l_k}(l_{k-1}) > \hat{\sigma}^S_{l_k}(a(l_{k-1}, S))$ (the first inequality follows by Lemma A.4 and the second one by definition of $a(l_{k-1}, S)$). Hence, l_k is not a new-max job according to $\hat{\sigma}^S_{l_k}$, and (iii) is satisfied. As a consequence of this and of (A.17), it follows that

$$\begin{aligned} &c_N\big(\sigma^T_{l_k r}\big) - c_N\big(\sigma^T_{l_k r^S(l_k)}\big) = c_N\big(\sigma^S_{l_k r}\big) - c_N\big(\sigma^S_{l_k r^S(l_k)}\big) \\ &\quad \text{for every } r \in \big\{1, \ldots, r^S(l_k) - 1\big\}. \end{aligned} \tag{A.18}$$

Note that it may happen that

$$\begin{aligned} &c_N\big(\sigma^T_{i r^S(i)}\big) - c_N\big(\sigma^T_{ir}\big) > 0 \\ &\quad \text{for some } r \in \big\{r^S(l_k) + 1, \ldots, s_i(T) + 1\big\}, \end{aligned} \tag{A.19}$$

since some of the $\{l_1, \ldots, l_{k-1}\}$ may be reordered in front of a_{u-1} in T. Then, by (A.18) and (A.19), (ii) follows. Besides, $r^T(l_k) \geq r^S(l_k)$. Hence, $G^T_{l_k} = G^S_{l_k}$ if $r^T(l_k) = r^S(l_k)$ and $G^T_{l_k} > G^S_{l_k}$ if $r^T(l_k) > r^S(l_k)$. Therefore, (iv) is also satisfied. □

References

Allahverdi, A. (1996). Two-machine proportionate flowshop scheduling with breakdowns to minimize maximum lateness. *Computers & Operations Research*, *23*(10), 909–916.

Allahverdi, A., & Savsar, M. (2001). Stochastic proportionate flowshop scheduling with setups. *Computers & Industrial Engineering*, *39*(3), 357–369.

Bondareva, O. N. (1963). Some applications of linear programming methods to the theory of cooperative games. *Problemy Kibernitiki*, *10*, 119–139 (in Russian).

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 class of parallel sequencing situations and related games. *Annals of Operations Research*, *109*, 263–276.

Calleja, P., Estévez-Fernández, A., Borm, P., & Hamers, H. (2006). Job scheduling, cooperation, and control. *Operations Research Letters*, *34*, 22–28.

Cheng, T. C. E., & Shakhlevich, N. (1999). Proportionate flow shop with controllable processing times. *Journal of Scheduling*, *2*(6), 253–265.

Curiel, I., Pederzoli, G., & Tijs, S. H. (1989). Sequencing games. *European Journal of Operational Research*, *40*(3), 344–351.

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

Curiel, I., Hamers, H., & Klijn, F. (2002). Sequencing games: a survey. In P. Borm & H. Peters (Eds.), *Chapters in game theory: in honor of Stef Tijs* (pp. 27–50). Boston: Kluwer Academic.

Estévez-Fernández, A., Calleja, P., Borm, P., & Hamers, H. (2004). *Sequencing games with repeated players* (CentER Discussion Paper 2004-128). Tilburg University, Tilburg, The Netherlands.

García-Sanz, M. D., Fernández, F. R., Fiestras-Janeiro, M. G., García-Jurado, I., & Puerto, J. (2008). Cooperation in Markovian queuing models. *European Journal of Operational Research*, *188*, 485–495.

González, P., & Herrero, C. (2004). Optimal sharing of surgical costs in the presence of queues. *Mathematical Methods of Operations Research*, *59*(3), 435–436.

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 m-sequencing games. *European Journal of Operational Research*, *119*, 678–691.

Hassin, R., & Haviv, M. (2003). *To queue or not to queue: equilibrium behavior in queuing systems*. Boston: Kluwer Academic.

LeBreton, M., Owen, G., & Weber, S. (1992). Strongly balanced cooperative games. *International Journal of Game Theory*, *20*, 419–427.

Maniquet, F. (2003). A characterization of the Shapley value in queuing problems. *Journal of Economic Theory*, *109*(1), 90–103.

Moulin, H., & Stong, R. (2002). Fair queuing and other probabilistic allocation methods. *Mathematics of Operations Research*, *27*(1), 1–30.

Shakhlevich, N., Hoogeveen, H., & Pinedo, M. (1998). Minimizing total weighted completion time in a proportionate flow shop. *Journal of Scheduling*, *1*(3), 157–168.

Shapley, L. S. (1953). A value for n-person games. In H. Kuhn & A. W. Tucker (Eds.), *Annals of mathematics studies: Vol. 28. Contribution to the theory of games II* (pp. 307–317). Princeton: Princeton University Press.

Shapley, L. S. (1967). On balanced sets and cores. *Naval Research Logistics Quarterly*, *14*, 453–460.

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

Shiau, D. F., & Huang, Y. M. (2004). Solving proportionate flexible flow shop scheduling problems: a column generation approach. In M. Hamza (Ed.), *Modelling, simulation, and optimization*.

van den Nouweland, A., Krabbenborg, M., & Potters, J. (1990). Flowshops with a dominant machine. *European Journal of Operational Research*, *62*(1), 38–46.

van Velzen, B. (2006). Sequencing games with controllable processing times. *European Journal of Operational Research*, *172*(1), 64–85.