Tilburg University

Proportionate Flow Shop Games

Estevez Fernandez, M.A.; Mosquera, M.A.; Borm, P.E.M.; Hamers, H.J.M.

Publication date:
2006

CentER

Discussion Paper

No. 2006–63

PROPORTIONATE FLOW SHOP GAMES

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

June 2006

ISSN 0924-7815

Tilburg University

Proportionate Flow Shop Games

Arantza Estévez-Fernández[1,2] Manuel A. Mosquera[3,4] Peter Borm[1]
Herbert Hamers[1]

June 29, 2006

[1]CentER and Department of Econometrics and Operations Research, Tilburg University, P.O.Box 90153, 5000 LE Tilburg, The Netherlands.

[3]Department of Statistics and Operations Research, Faculty of Mathematics, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain.

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 an additional problem: how to allocate the cost savings obtained by ordering the jobs optimally? In this paper, 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 expression for the Shapley value and a specific type of equal gain splitting rule which leads to core elements of the PFS game are proposed.

Keywords: Proportionate flow shop problems, core, convexity.

JEL Classification Numbers: C71

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

[2]Corresponding author.

[4]The author acknowledges the financial support of *Ministerio de Ciencia y Tecnología*, FEDER, *Xunta de Galicia* (projects BEC2002-04102-C02-02, SEJ2005-07637-C02-02 and PGIDT03PXIC20701PN).

is associated dependent on its completion time. In this paper we will 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, each with a different product, 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, Hoogeveen and Pinedo (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 and setup times 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 the clients jointly face is an optimization problem: the problem of finding an optimal reordering of all jobs, i.e., a schedule that maximizes joint cost savings. The subsequent problem is of a game theoretic nature: how to reallocate these cost savings in a fair way. By defining the value of a coalition of clients as the maximal attainable costs 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 non-empty core.

The above game-theoretic approach to sequencing situations has been initiated by Curiel, Pederzoli and Tijs (1989) for the class of one-machine sequencing situations. Generalizations to e.g. ready times, due dates, multiple ownership and more machines have been studied in Hamers, Borm and Tijs (1995); Borm, Fiestras-Janeiro, Hamers, Sánchez and Voorneveld (2002); Calleja, Estévez-Fernández, Borm and Hamers (2006); Estévez-Fernández, Calleja, Borm and Hamers (2004); Hamers, Klijn and Suijs (1999); Calleja, Borm, Hamers, Klijn and Slikker (2002). A recent review on sequencing games can be found in Curiel, Hamers and Klijn (2002). Finally, within the context of flow shop problems, van den Nouweland, Krabbenborg and Potters (1990) and van den Nouweland (1993) have studied the specific case of a dominant machine.

This article analyzes proportionate flow shop problems and related PFS games. It is shown that PFS games are balanced. Moreover, PFS games turn out to be convex if the initial order is the urgency order, in which case the Shapley value is in the core of the game. We provide an explicit expression for the Shapley value. Under this assumption, we also provide a context-specific allocation rule (the γ-rule) in the same spirit as the equal gain splitting (EGS) rule introduced in Curiel et al. (1989). This allocation rule follows the algorithm in Shakhlevich et al. (1998). In this way, the optimization problem of determining the optimal order of the grand coalition and the allocation problem of how to share joint savings can be solved in an integrated way.

The remainder of the paper is organized as follows. Section 2 provides the basic definitions and terminology of proportionate flow shop problems. Moreover, two useful results in Shakhlevich et al. (1998) are recalled. Section 3 deals with cooperation within proportionate flow shop problems. The γ-rule is introduced as a specific allocation rule of the maximal joint cost savings. In Section 4 PFS games are defined. It is shown that these games are convex provided that the initial order is an urgency order and an expression of the Shapley value is provided. Moreover, it is seen that in this case also the γ-rule will provide a core element.

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. 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 $M = \{M_1, \ldots, M_m\}$ the set of machines, $N = \{1, \ldots, n\}$ the set of jobs, and $p \in \mathbb{R}^N_+$ 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 will be 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 in the machines as $\sigma = (\sigma^1, \ldots, \sigma^m)$ with $\sigma^r : N \to \{1, \ldots, |N|\}$ a bijection describing the processing order in machine M_r. We will denote by $\Pi(N, M)$ the set of all schedules of the jobs in 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 in machine* M_r, i.e., $P(\sigma^r, i) = \{j \in N | \sigma^r(j) < \sigma^r(i)\}$. 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 in machine* M_r, i.e., $p(\sigma^r, i) \in N$ such that $\bar{P}(\sigma^r, p(\sigma^r, i)) = P(\sigma^r, i)$. Note that in principle the order in machines need not be the same. A schedule $\sigma = (\sigma^1, \ldots, \sigma^m)$ with $\sigma^1 = \ldots = \sigma^m$ is called a *permutation schedule* or *order*. With minor abuse of notation, σ will then denote the order in each machine. We will 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* in machine M_r *with respect to* an arbitrary *schedule* σ, $C_i^\sigma(r)$, can be recursively determined by

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

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

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

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 the schedule σ. Hence, there exist positive numbers α_i, $i \in N$, such that $c_i(\sigma) = \alpha_i C_i^\sigma(m)$. From now on we will denote the overall completion time $C_i^\sigma(m)$ by C_i^σ.

Given a PFS situation (M, N, p) and a linear cost associated to each job, which will be represented by $\alpha \in \mathbb{R}^N$, the associated *PFS problem*, (M, N, p, α) has as objective to find a schedule that minimizes the total cost originated in the system, i.e., find $\hat{\sigma}$ 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 will recall three lemmas from Shakhlevich et al. (1998) that will be 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

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

Since every optimal schedule is a permutation schedule, we will restrict our study to permutation schedules from now on.

Example 2.1. 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 Figure 1.

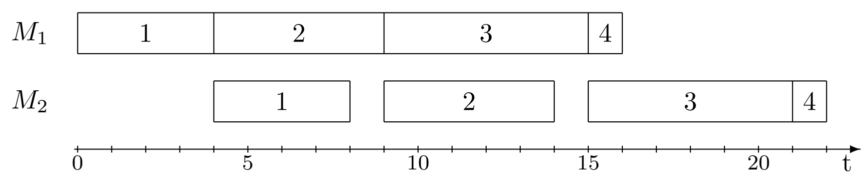

Figure 1: Gantt Chart of the PFS situation in Example 2.1

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 $c_N(\sigma) = 1490$. $\diamond$

Since the processing time of a job is the same in 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 with larger processing time, then the one with higher urgency will be processed first in an optimal order.

Lemma 2.2 (Shakhlevich et al. (1998))**.** Let (M, N, p, α) be a PFS problem and σ 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 $\sigma(i) < \sigma(j)$.

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 $\sigma(a_1^\sigma) < \ldots < \sigma(a_s^\sigma)$. 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, since $\sigma(a_1^\sigma) = 1$, $P(\sigma, a_1^\sigma) = \emptyset$. 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 decreasing urgency order.

Lemma 2.3 (Shakhlevich et al. (1998))**.** Let (M, N, p, α) be a PFS problem and σ 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 $u_i \geq u_j$.

3 Cooperation in proportionate flow shops

In this section we will recall the algorithm to find an optimal schedule for PFS problems given in Shakhlevich et al. (1998) and propose an allocation rule to share the costs savings obtained by reordering the jobs into an optimal order if the initial order is in decreasing urgency order.

We first describe the algorithm in Shakhlevich et al. (1998). Let (M, N, p, α) be a PFS problem. We define the *urgency order*, σ_u, as the order in which the jobs are ordered in decreasing urgency. 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 will generate orders $\hat{\sigma}_1, \ldots, \hat{\sigma}_n$ where $\hat{\sigma}_1 := \sigma_u$ and $\hat{\sigma}_n$ is optimal. Note that associated to the order $\hat{\sigma}_{i-1}$ 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 $r = 2,\ldots,s_i$.

Here, we have numbered the segments from right to left (instead from left to right) for convenience of the description of the rule that we will give later on. Subsequently, $\hat{\sigma}_i$ is obtained from $\hat{\sigma}_{i-1}$ by placing i in first position or in between two consecutive segments or remain in its initial position. The decision will be taken in such a way that $c_N(\hat{\sigma}_i)$ is minimal and $\max_{k\in \bar{P}(\hat{\sigma}_i,i)}\{p_k\}$ is maximal.

Now we turn to interactive proportionate flow shop situations and assume that each job belongs to a player. We define the γ-rule which allocates the gains $\sum_{i\in N}(c_N(\hat{\sigma}_{i-1}) - c_N(\hat{\sigma}_i))$. Here, we will decompose the gain $c_N(\hat{\sigma}_{i-1}) - c_N(\hat{\sigma}_i)$ into "positive jumps" and the associated "positive gains" will be shared among the jobs involved. For this, we will need some additional notation. We define

$$g_{A(i,r)i} := \sum_{j\in A(i,r)} (\alpha_i p_j - \alpha_j p_i) + \alpha_i(m-1)(\max_{j\in A(i,r)\cup\{i\}}\{p_j\} - \max_{j\in A(i,r+1)\cup\{i\}}\{p_j\}), \tag{3.1}$$

for $r = 1,\ldots,s_i$, with $A(i,s_i+1) := \emptyset$.

Hence, $g_{A(i,r)i}$ represents the cost savings obtained when job i goes from the tale of $A(i,r)$ to its front. Note that $g_{A(i,r)i}$ can be negative. Define $N(i,1) := A(i,1)$, $g_{N(i,1)} := g_{A(i,1)i}$, and $h_{N(i,1)} := (g_{N(i,1)})_+$. For $r = 1,\ldots,s_i$ we define recursively

$$N(i,r) := \begin{cases} N(i,r-1)\cup A(i,r) & \text{if } h_{N(i,r-1)} = 0,\\ A(i,r) & \text{otherwise,}\end{cases}$$

$$g_{N(i,r)} := \begin{cases} g_{N(i,r-1)} + g_{A(i,r)i} & \text{if } h_{N(i,r-1)} = 0,\\ g_{A(i,r)i} & \text{otherwise,}\end{cases}$$

and

$$h_{N(i,r)} := (g_{N(i,r)})_+.$$

Easily, $c(\hat{\sigma}_{i-1}) - c(\hat{\sigma}_i) = \sum_{r=1}^{s_i} h_{N(i,r)}$ and therefore $\sum_{i\in N}\sum_{r=1}^{s_i} h_{N(i,r)}$ gives the total cost savings gained by means of cooperation. The γ-rule simply gives half of $h_{N(i,r)}$ to i while the other half is shared equally among the jobs in $N(i,r)$ for each $i \in N$ and $1 \le r \le s_i$. Formally, we define

$$\gamma(M,N,p,\alpha) = \sum_{i\in N}\sum_{r=1}^{s_i}\left(\frac{h_{N(i,r)}}{2}e^{\{i\}} + \frac{h_{N(i,r)}}{2|N(i,r)|}e^{N(i,r)}\right)$$

with $e^R \in \mathbb{R}^N$ a vector of zeros and ones with $e_i^R = 1$ if $i \in R$ and $e_i^R = 0$ otherwise, for $R \subset N$.

The following example illustrates the computation of the γ-rule.

Example 3.1. Let (M,N,p,α) be a PFS problem with $M = \{M_1,M_2,M_3\}$, $N = \{1,2,3,4,5,6,7,8,9\}$, $p = (20,30,10,30,10,30,20,10,40)$ and $\alpha = (200,270,80,210,69,180,130,59,200)$. Hence, 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, $c_N(\sigma_u) = 224320$. The situation is represented in Figure 2.

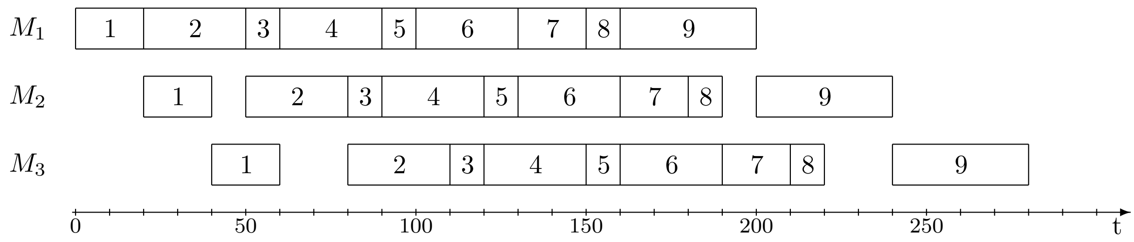

Figure 2: Gantt Chart of the PFS situation in Example 3.1.

The allocation of the total cost savings after reordering the jobs in the optimal order is summarized in Table 1.

	1	2	3	4	5	6	7	8	9
$i=1$	0	0	0	0	0	0	0	0	0
$i=2$	0	0	0	0	0	0	0	0	0
$i=3$	600	650	$650+600$	0	0	0	0	0	0
$i=4$	0	0	0	0	0	0	0	0	0
$i=5$	380	180	0	180	$360+380$	0	0	0	0
$i=6$	0	0	0	0	0	0	0	0	0
$i=7$	0	$\frac{550}{3}$	0	$\frac{550}{3}$	0	$\frac{550}{3}$	550	0	0
$i=8$	13	13	0	13	0	13	13	65	0
$i=9$	0	0	0	0	0	0	0	0	0
	993	$1026\frac{1}{3}$	1250	$376\frac{1}{3}$	740	$196\frac{1}{3}$	563	65	0

Table 1: Allocation of the cost savings in Example 3.1.

We explain below how the cost savings are shared when jobs 5 and 8 are reordered.

First, we will study the case in which job 5 is reordered. We leave it to the reader to verify that the order obtained after reordering jobs 1, 2, 3 and 4 is $\hat{\sigma}_4 = (3\ 1\ 2\ 4\ 5\ 6\ 7\ 8\ 9)$ and $Seg(\hat{\sigma}_4) = \{\{3\}, \{1\}, \{2,4,5,6,7,8\}, \{9\}\}$. Take $i = 5$ and as previous order $\hat{\sigma}_4$. Hence, $A(5,1) = \{2,4\}$, $A(5,2) = \{1\}$, and $A(5,3) = \{3\}$. Moreover,

$$g_{A(5,1)5} = c_N(\hat{\sigma}_4) - c_N(\tau_5^1) = (\alpha_2 C_2^{\hat{\sigma}_4} + \alpha_4 C_4^{\hat{\sigma}_4} + \alpha_5 C_5^{\hat{\sigma}_4}) - (\alpha_2 C_2^{\tau_5^1} + \alpha_4 C_4^{\tau_5^1} + \alpha_5 C_5^{\tau_5^1}) = 720,$$

with $\tau_5^1 = (3\ 1\ 5\ 2\ 4\ 6\ 7\ 8\ 9)$,

$$g_{A(5,2)5} = c_N(\tau_5^1) - c_N(\tau_5^2) = (\alpha_1 C_1^{\tau_5^1} + \alpha_5 C_5^{\tau_5^1}) - (\alpha_1 C_1^{\tau_5^2} + \alpha_5 C_5^{\tau_5^2}) = 760,$$

with $\tau_5^2 = (3\ 5\ 1\ 2\ 4\ 6\ 7\ 8\ 9)$, and

$$g_{A(5,3)5} = c_N(\tau_5^2) - c_N(\tau_5^3) = (\alpha_3 C_3^{\tau_5^2} + \alpha_5 C_5^{\tau_5^2}) - (\alpha_3 C_3^{\tau_5^3} + \alpha_5 C_5^{\tau_5^3}) = -110,$$

with $\tau_5^3 = (5\ 3\ 1\ 2\ 4\ 6\ 7\ 8\ 9)$.
Hence, $N(5,1) := A(5,1) = \{2,4\}$, $g_{N(5,1)} = 720$, $h_{N(5,1)} = 720$, $N(5,2) = \{1\}$, $g_{N(5,2)} = 760$, $h_{N(5,2)} = 760$, $N(5,3) = \{3\}$, $g_{N(5,3)} = -110$, $h_{N(5,3)} = 0$. In this step, the owner of job 5 gets 360 from $h_{N(5,1)}$ and the owners of jobs in $N(5,1)$ share equally 360, i.e., 2 and 4 get 180 each. Similarly, the owner of job 5 gets 380 from $h_{N(5,2)}$ and the owner of the job in $N(5,2)$ gets 380, i.e., 1 gets 380.

Hence, an optimal order after reallocating 5 is $\hat{\sigma}_5 = \tau_5^2 = (3\ 5\ 1\ 2\ 4\ 6\ 7\ 8\ 9)$ and the cost savings obtained after this reorder are $h_{N(5,1)} + h_{N(5,2)} + h_{N(5,3)} = 1480$.

Next, we will study the case in which job 8 is reordered. In this case, $\hat{\sigma}_7 = (3\ 5\ 1\ 7\ 2\ 4\ 6\ 8\ 9)$ and $Seg(\hat{\sigma}_7) = \{\{3,5\},\{1,7\},\{2,4,6,8\},\{9\}\}$. Take $i = 8$ and as previous order $\hat{\sigma}_7$. Here, $A(8,1) = \{2,4,6\}$, $A(8,2) = \{1,7\}$, and $A(8,3) = \{3,5\}$. Moreover,

$$g_{A(8,1)8} = c_N(\hat{\sigma}_7) - c_N(\tau_8^1) = (\alpha_2 C_2^{\hat{\sigma}_7} + \alpha_4 C_4^{\hat{\sigma}_7} + \alpha_6 C_6^{\hat{\sigma}_7} + \alpha_8 C_8^{\hat{\sigma}_7}) - (\alpha_2 C_2^{\tau_8^1} + \alpha_4 C_4^{\tau_8^1} + \alpha_6 C_6^{\tau_8^1} + \alpha_8 C_8^{\tau_8^1}) = -110,$$

with $\tau_8^1 = (3\ 5\ 1\ 7\ 8\ 2\ 4\ 6\ 9)$,

$$g_{A(8,2)8} = c_N(\tau_8^1) - c_N(\tau_8^2) = (\alpha_1 C_1^{\tau_8^1} + \alpha_7 C_7^{\tau_8^1} + \alpha_8 C_8^{\tau_8^1}) - (\alpha_1 C_1^{\tau_8^2} + \alpha_7 C_7^{\tau_8^2} + \alpha_8 C_8^{\tau_8^2}) = 240,$$

with $\tau_8^2 = (3\ 5\ 8\ 1\ 7\ 2\ 4\ 6\ 9)$.
Note that job 8 can not be reallocated in an earlier position since it would violate Lemma 2.2. Hence, $N(8,1) := A(8,1) = \{2,4,6\}$, $g_{N(8,1)} = -110$, $h_{N(8,1)} = 0$, $N(8,2) = \{1,2,4,6,7\}$, $g_{N(8,2)} = -110 + 240 = 130$, $h_{N(8,2)} = 130$, $N(8,3) = \{3,5\}$, $g_{N(8,3)} < 0$, $h_{N(8,3)} = 0$. In this step, the owner of job 8 gets 65 from $h_{N(8,2)}$ and the owners of jobs in $N(8,2)$ share equally 65, i.e., 1, 2, 4, 6 and 7 get 13 each.

Hence, an optimal order after reallocating 8 is $\hat{\sigma}_8 = \tau_8^2 = (3\ 5\ 8\ 1\ 7\ 2\ 4\ 6\ 9)$ and the cost savings obtained after this reorder are $h_{N(8,1)} + h_{N(8,2)} + h_{N(8,3)} = 130$. $\diamond$

4 Proportionate flow shop games

In this section we study proportionate flow shop games and show that they are balanced. Moreover, if the initial order is the urgency order, then they are convex and an explicit expression of the Shapley value is provided based on the decomposition of the proportionate flow shop games into unanimity games. Besides, it is shown that the γ-rule leads to a core element.

Before stating our main results we will 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 $v(\emptyset) = 0$. The *core* of a cooperative TU-game (N, v) is defined by

$$Core(v) = \{x \in \mathbb{R}^N | \sum_{i \in N} x_i = v(N), \sum_{i \in S} x_i \geq v(S) \text{ for all } S \in 2^N\},$$

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

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T) \tag{4.1}$$

for all $S, T \subset N$.

Let (N, v) be a game and let $\pi : \{1, \ldots, |N|\} \to N$ be a bijection. The *marginal vector* $m^\pi(v)$, is defined by

$$m^\pi_{\pi(k)}(v) := v(\{\pi(1), \ldots, \pi(k)\}) - v(\{\pi(1), \ldots, \pi(k-1)\})$$

for all $k \in \{1, \ldots, |N|\}$. It is known that convexity of a game is equivalent to every marginal vector being a core element (see Shapley (1953) and Ichiishi (1981)). The Shapley value of a game (N, v) is defined as the average of its marginal vectors.

Next, we start the game theoretical study of proportionate flow shops. Let (M, N, p, α) be a PFS problem and let $\sigma_0 \in \Pi(N)$ be an initial order on the jobs. We assume without loss of generality that $\sigma_0 = (1 \ \ldots \ n)$. By associating jobs with players (or clients) the associated PFS game (N, v) is defined by

$$v(S) := \max_{\sigma \in \mathcal{A}(S)} \{c_N(\sigma_0) - c_N(\sigma)\} \tag{4.2}$$

for every $S \subset N$, where $\mathcal{A}(S)$ is the set of admissible rearrangements for coalition S. An order $\sigma \in \Pi(N)$ is said to be *admissible for* coalition S if $P(\sigma_0, j) = P(\sigma, j)$ for all $j \in N \setminus S$. This implies that in an admissible rearrangement the initial schedule for jobs outside S does not change, i.e., the starting time in each machine of each player outside S does not change with respect to the initial order. Moreover, agents of S are only allowed to be reordered within maximally connected components of S with regard to σ_0. Here, a *coalition* R is called *connected* (with respect to σ_0) if for all $i, j \in R$ and $k \in N$ such that $\sigma_0(i) < \sigma_0(k) < \sigma_0(j)$ it holds that $k \in R$. Given a coalition $S \subset N$, we denote by S/σ_0 the set of all maximally connected components of S according to σ_0. Due to the definition of admissible rearrangements, we can write the value of coalition $S \subset N$ as

$$v(S) = \sum_{R \in S/\sigma_0} v(R). \tag{4.3}$$

It is readily seen that PFS games are σ_0-component additive and therefore balanced (see Curiel et al. (1994)).

Example 4.1. 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 $\sigma_0 = (1\ 2\ 3)$. The situation is represented in Figure 3.

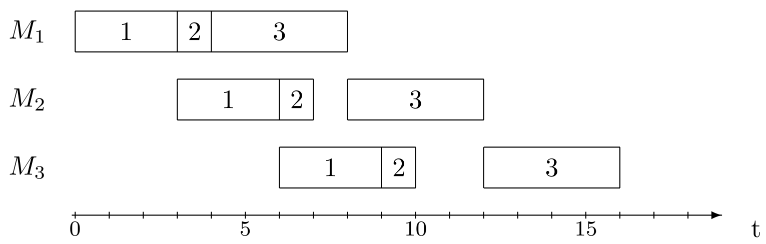

Figure 3: Gantt Chart of the PFS situation in Example 4.1.

The corresponding PFS game (N, v) is

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	N
$v(S)$	0	0	0	0	3	0	1	3

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 $\mathcal{A}(\{1, 2\}) = \{\sigma_0, \sigma_1\}$, with $\sigma_1 = (2\ 1\ 3)$, and the total cost for the order σ_1 is $c_N(\sigma_1) = 155$. Then,

$$v(\{1, 2\}) = \max_{\sigma \in \{\sigma_0, \sigma_1\}} \{c_N(\sigma_0) - c_N(\sigma)\} = \max\{0, 3\} = 3. \qquad \diamond$$

Note that the initial order in Example 4.1 is not an urgency order. Moreover, the game is balanced but not convex (take $S = \{1, 2\}$ and $T = \{2, 3\}$).

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

$$\sigma_0 = \sigma_u = (1\ 2\ \ldots\ n).$$

We will 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 Section 3 is followed. Due to equation (4.3) we will restrict our study to connected coalitions. Let $S \subset N$ be a connected coalition, $S = \{j, j + 1, \ldots, k - 1, k\}$. To find the optimal order for S we will generate orders $\hat{\sigma}^S_j, \ldots, \hat{\sigma}^S_k$ in the following way: $\hat{\sigma}^S_j := \sigma_u$ and $\hat{\sigma}^S_i$ is obtained from $\hat{\sigma}^S_{i-1}$ as follows. If for every r it follows that $A_r^{\hat{\sigma}^S_{i-1}} \cap \{j, \ldots, i - 1\}$ does not contain any new-max job according to $\hat{\sigma}^S_{i-1}$, then i and j belong to the same segment and j is not a new-max job.

In this case, $\hat{\sigma}_i^S = \sigma_u$ by Lemma 2.3. If $A_r^{\hat{\sigma}_{i-1}^S} \cap \{j, \dots, i-1\}$ contains a new-max job according to $\hat{\sigma}_{i-1}^S$ for some r, then we define $r_i = \min\{r \mid A_r^{\hat{\sigma}_{i-1}^S} \cap \{j, \dots, i-1\}$ contains a new-max job according to $\hat{\sigma}_{i-1}^S\}$, $t_i = \max\{r \mid A_r^{\hat{\sigma}_{i-1}^S} \cap \{j, \dots, i-1\}$ contains a new-max job according to $\hat{\sigma}_{i-1}^S\}$, and $s_i = t_i - r_i + 1$. Analogously than in the method described in Section 3, we define $A^S(i,1), \dots, A^S(i,s_i)$ as $A^S(i,1) = A_{t_i}^{\hat{\sigma}_{i-1}^S} \cap \{j, \dots, i-1\}$, $A^S(i,r) = A_{t_i-r+1}^{\hat{\sigma}_{i-1}^S}$ for $r = 2 \dots s_i$. Subsequently, $\hat{\sigma}_i^S$ is obtained by placing i in position j or in between two consecutive segments or remain in its initial position. The decision will be taken in such a way that

$$c_N(\hat{\sigma}_i^S) \text{ is minimal and } \max_{k \in \bar{P}(\hat{\sigma}_i^S, i)} \{p_k\} \text{ is maximal.} \tag{4.4}$$

Note that Lemma 2.2 and Lemma 2.3 are still applicable in S. Hence, job i may be placed in j-th position only if $A_{r_i}^{\hat{\sigma}_{i-1}^S} \cap \{j, \dots, i-1\} = A_{r_i}^{\hat{\sigma}_{i-1}^S}$, otherwise Lemma 2.2 would be violated. Hence, the value of coalition S can be written as

$$v(S) = \sum_{i=j}^{k} (c_N(\hat{\sigma}_{i-1}^S) - c_N(\hat{\sigma}_i^S))$$

with $\hat{\sigma}_{j-1}^S := \sigma_u$. We define $G_i^S := c_N(\hat{\sigma}_{i-1}^S) - c_N(\hat{\sigma}_i^S)$ for $i \in S$. Here, G_i^S denotes the cost savings obtained after reordering job $i \in S$ in S. Hence,

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

Note that as a consequence of Lemma 2.2 it follows that if $j \in P(\sigma_u, a)$, then $j \in P(\hat{\sigma}_i^S, a)$ with a a new-max job according to σ_u.

Next, we provide some lemmas that will be used in the proofs of our main results. Their proofs can be found in the appendix. 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 4.1. Let (M, N, p, α) be a PFS problem and let σ_u 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 $i \in S$.

Next, we provide a result on the "monotonicity" of new-max jobs and cost savings.

Lemma 4.2. 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 $i_S < \dots < j_S$, 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) $\hat{\sigma}_i^T(i) = \hat{\sigma}_i^S(i)$ for every $i \in S$ with $p_i \geq p_a$.

(ii) $\hat{\sigma}_i^T(i) \leq \hat{\sigma}_i^S(i)$ for every $i \in S$ with $p_i < p_a$. Moreover, if $\hat{\sigma}_i^T(i) < \hat{\sigma}_i^S(i)$, then $\hat{\sigma}_i^T(i) < \hat{\sigma}_i^T(a)$.

(iii) Every new-max job according to $\hat{\sigma}_i^S$ is also a new-max job according to $\hat{\sigma}_i^T$.

(iv) $G_i^S \leq G_i^T$. Moreover, if $p_i \geq p_a$, then $G_i^S = G_i^T$.

The following lemma states that the cost savings achievable for a coalition by the reallocation of job i are at most the total cost savings that job i can achieve for the grand coalition during its reallocation.

Lemma 4.3. Let (M, N, p, α) be a PFS problem and let σ_u be the initial order. Let $S \subset N$ be a connected coalition. Then,

$$G_i^S = \sum_{r:N(i,r)\subset S} h_{N(i,r)}$$

for all $i \in S$.

Next, we will show that the γ-rule leads to a core element of the associated PFS game.

Theorem 4.4. Let (M, N, p, α) be a PFS problem and let σ_u be the initial order. Then, the γ-rule provides a core element of the associated PFS game.

Proof. Efficiency holds by definition. Let $S \subset N$ be a connected coalition, then

$$\begin{aligned}
\sum_{i\in S} \gamma_i(M,N,p,\alpha) &= (e^S)^t \sum_{i\in N}\sum_{r=1}^{s_i} \left(\frac{h_{N(i,r)}}{2} e^{\{i\}} + \frac{h_{N(i,r)}}{2|N(i,r)|} e^{N(i,r)} \right) \\
&= \frac{1}{2}\sum_{i\in S}\sum_{r=1}^{s_i} h_{N(i,r)} + \frac{1}{2}(e^S)^t \sum_{i\in N}\sum_{r=1}^{s_i} \frac{h_{N(i,r)}}{|N(i,r)|} e^{N(i,r)} \\
&\geq \frac{1}{2}\sum_{i\in S}\sum_{r=1}^{s_i} h_{N(i,r)} + \frac{1}{2}(e^S)^t \sum_{i\in S}\sum_{r=1}^{s_i} \frac{h_{N(i,r)}}{|N(i,r)|} e^{N(i,r)} \\
&\geq \frac{1}{2}\sum_{i\in S}\sum_{r=1}^{s_i} h_{N(i,r)} + \frac{1}{2}(e^S)^t \sum_{i\in S}\sum_{r:N(i,r)\subset S} \frac{h_{N(i,r)}}{|N(i,r)|} e^{N(i,r)} \\
&= \frac{1}{2}\sum_{i\in S}\sum_{r=1}^{s_i} h_{N(i,r)} + \frac{1}{2}\sum_{i\in S}\sum_{r:N(i,r)\subset S} h_{N(i,r)} \\
&\geq \sum_{i\in S}\sum_{r:N(i,r)\subset S} h_{N(i,r)} \\
&= \sum_{i\in S} G_i^S = v(S),
\end{aligned}$$

where $(e^S)^t$ is the transposed matrix of e^S. The first, second, and third inequalities follow because $h_{N(i,r)} \geq 0$ and the last equality is a consequence of Lemma 4.3. □

The next result gives the decomposition into unanimity games of a PFS game. We denote by $\{a_1, \ldots, a_s\}$, with $a_1 < \ldots < a_s$, 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 4.5. 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,

$$v(T) = \sum_{k \in N} \sum_{r=1}^{r(k)} \left(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}} \right) u_{\{a_r,\dots,k\}}(T)$$

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

Proof. Let $T \subset N$ be a connected coalition and set $T = \{i, \dots, j\}$. We will distinguish between two cases.
<u>Case 1:</u> $T \cap \{a_1, \dots, a_s\} = \emptyset$. Then, $\hat{\sigma}_k^T(k) = k$ for all $k \in T$ by Lemma 2.3 and therefore $G_k^T = 0$ for all $k \in T$. Hence, $v(T) = 0$. Moreover, $\{a_r, \dots, k\} \not\subset T$ for every new-max job a_r and every $k \geq a_r$. Hence, $u_{\{a_r,\dots,k\}}(T) = 0$ and

$$\sum_{k \in N} \sum_{r=1}^{r(k)} \left(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}} \right) u_{\{a_r,\dots,k\}}(T) = 0 = v(T).$$

<u>Case 2:</u> $T \cap \{a_1, \dots, a_s\} = \{a_v, \dots, a_w\}$ with $a_v \leq \dots \leq a_w$. Then, $\hat{\sigma}_k^T(k) = k$ for all $k < a_v$ by Lemma 2.3 and $\hat{\sigma}_k^T(k) = \hat{\sigma}_k^{\{a_v,\dots,n\}}(k)$ for all $k \geq a_v$ by the mechanism of the algorithm. Hence, $G_k^T = 0$ for all $i \leq k < a_v$ and $G_k^T = G_k^{\{a_v,\dots,n\}}$ for all $k \geq a_v$. Therefore, $v(T) = \sum_{k=a_v}^{j} G_k^{\{a_v,\dots,n\}}$. Moreover,

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

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

(i) $a_r < a_v \leq k$, then $\{a_r, \dots, k\} \not\subset \{i, \dots, j\} = T$ and $u_{\{a_r,\dots,k\}}(T) = 0$,

(ii) $k > j$ and $a_r \leq k$, then $\{a_r, \dots, k\} \not\subset \{i, \dots, j\} = T$ and $u_{\{a_r,\dots,k\}}(T) = 0$.

The second equality is satisfied because if k and r are such that $a_v \leq a_r \leq a_{r(k)}$, $a_r \leq k \leq j$, then $\{a_r, \dots, k\} \subset \{i, \dots, j\} = T$ 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

$$\sum_{k\in N}\sum_{r=1}^{r(k)} \left(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}\right) u_{\{a_r,\dots,k\}}(T) = \sum_{U\in T/\sigma_0}\sum_{k\in N}\sum_{r=1}^{r(k)} \left(G_k^{\{a_r,\dots,n\}} - G_k^{\{a_{r+1},\dots,n\}}\right) u_{\{a_r,\dots,k\}}(U)$$

since the unanimity games are defined for connected coalitions. □

As a direct consequence of Lemma 4.2 (iv) and Theorem 4.5 we have that PFS games are convex.

Corollary 4.6. PFS games are convex if the initial order is the urgency order.

Proof. By Theorem 4.5 and by Lemma 4.2 (iv) we know that PFS games are decomposed in non-negative linear combination of unanimity games. Hence, PFS games are convex. □

If the initial order is an urgency order, PFS games are convex and the Shapley value belongs to the core. The next result provides a game independent expression of the Shapley value for PFS games.

Theorem 4.7. 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 the 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, $G_i^{\{a_{r(i)-1},\dots,i\}} - G_i^{\{a_{r(i)},\dots,i\}} = G_i^{\{a_{r(i)-1},\dots,n\}} - G_i^{\{a_{r(i)},\dots,n\}} \geq 0$ can be obtained by Lemma 4.2 (iv). 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 responsible.

Appendix

Proof of Lemma 4.1. 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 2.2. Hence,

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

□

For the proof of Lemma 4.2, we need the following additional lemmas. The first lemma is an immediate consequence of Lemma 2.2 and the definition of new-max job and therefore the proof will be omitted. It states that a new-max job a according to σ_u does not change its initial position in $\hat{\sigma}_a^S$, for all coalition $S \subset N$, $a \in S$.

Lemma A.1. Let (M, N, p, α) be a PFS problem and let σ_u be the initial order. Let $S \subset N$ and let $a \in S$ be a new-max job according to σ_u. Then, $\hat{\sigma}_a^S(a) = a$.

The following result is a direct consequence of the algorithm. 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 σ_u 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 will show that if a job becomes new-max job during its reordering, then it will remain new-max job during the successive application of the algorithm.

Lemma A.3. Let (M, N, p, α) be a PFS problem and let σ_u 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 4.1 and the result follows. Hence, we may assume that j is not new-max job according to σ_u.

We will 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 2.2. Hence, j is a new-max job according to $\hat{\sigma}_i^S$. □

For the proof of Lemma 4.2 we need some additional notation. Let $S \subset N$, $S = \{i_S, \ldots, j_S\}$, with $i_S < \ldots < j_S$, and $i \in S$. 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 $r \in \{1, \ldots, s_i\}$. We denote by $a(i, S)$ the new-max job according to $\hat{\sigma}_{i-1}^S$ such that i is placed at the tail of the segment defined by $a(i,S)$ after being reordered. Note that $p_{a(i,S)} = \max_{k\in P(\hat{\sigma}_i^S,i)}\{p_k\}$. Let $i \in S$ and let $a \in S$ be a new-max job according to $\hat{\sigma}_{i-1}^S$. We denote by $r(i,a)$ the index of the segment defined by a according to $\hat{\sigma}_{i-1}^S$, i.e., $r(i,a) \in \{1,\ldots,s_i\}$ such that $a \in B^S(i, r(i,a))$. Moreover, we denote by $g_{B^S(i,r)i}$ the gains obtained when, starting from $\hat{\sigma}_{i-1}^S$, we change i from the tail of segment $B^S(i,r)$ to the tail of segment $B^S(i,r-1)$. Formally, it can be written as

$$g_{B^S(i,r)i} := \sum_{j\in B^S(i,r)} (\alpha_i p_j - \alpha_j p_i) + \alpha_i(m-1)\left(\max_{j\in B^S(i,r)\cup\{i\}}\{p_j\} - \max_{j\in B^S(i,r-1)\cup\{i\}}\{p_j\}\right), \tag{A.1}$$

with $r \in \{1,\ldots,s_i\}$ and $B^S(i,0) := \begin{cases} \emptyset & \text{if } i_S = 1 \\ A_{r(i,a^*)}^{\hat{\sigma}_{i-1}^S} & \text{if } i_S > 1 \end{cases}$, where a^* is the new-max job according to σ_u such that $p_{a^*} = \max_{k\in P(\sigma_u,i_S)}\{p_k\}$.

Recall that $\{a_1,\ldots,a_s\}$, with $a_1 < \ldots < a_s$, is the set of new-max jobs according to σ_u. Let $S \subset N$ be a connected coalition, $S = \{i_S,\ldots,j_S\}$, satisfying:

(i) $S \cap \{a_1,\ldots,a_s\} = \{a_u,\ldots,a_v\}$, with $a_u \leq \ldots \leq a_v$ and $a_u \neq a_1$;

(ii) there exists $l_1 \in S$ verifying the following three conditions

$$p_{l_1} < p_{a_{u-1}} \tag{A.2}$$

$$a_{u-1} \neq a(l_1,S) \tag{A.3}$$

$$a_{u-1} = a(l,S) \text{ for every } l \in S \text{ with } l < l_1 \text{ and } p_l < p_{a_{u-1}}. \tag{A.4}$$

Consider the following partition of S

$$\big\{\{i_S,\ldots,l_1-1\},\{l_1\},\{l_1+1,\ldots,l_2-1\},\{l_2\},\ldots,\{l_m\},\{l_m+1,\ldots,j_S\}\big\} \tag{A.5}$$

where $p_j \geq p_{a(l_k,S)}$ for every $j \in \{l_k+1,\ldots,l_{k+1}-1\}$ and every $k \in \{1,\ldots,m\}$ (with $l_{m+1}-1 := j_S$), and l_k satisfying $p_{l_k} < p_{a(l_{k-1},S)}$, with $l_0 := a_{u-1}$.

Lemma A.4. The two following assertions hold

(i) for every $k,\tilde{k} \in \{1,\ldots,m\}$ with $\tilde{k} > k$ we have

$$g_{B^S(l_{\tilde{k}},r)l_{\tilde{k}}} \leq \frac{p_{l_{\tilde{k}}}}{p_{l_k}} g_{B^S(l_k,r)l_k}$$

for every $r \in \{r(l_{\tilde{k}}, a(l_{k-1},S))+1,\ldots,r(l_{\tilde{k}},a(l_k,S))\}$ with $r(l_{\tilde{k}},a(l_0,S)) := 0$;

(ii) $\hat{\sigma}_{l_{\tilde{k}}}^S(l_{\tilde{k}}) > \hat{\sigma}_{l_{\tilde{k}}}^S(l_{\tilde{k}-1})$ for every $\tilde{k} \in \{2,\ldots,m\}$.

Proof. First, we prove the result for $\tilde{k} = 2$. Since $p_j \geq p_{a(l_1,S)}$ for every $j \in \{l_1+1,\ldots,l_2-1\}$ we have $\hat{\sigma}_j^S(j) > \hat{\sigma}_j^S(l_1)$ by Lemma 2.2 and Lemma 2.3. Therefore,

$$\hat{\sigma}_{l_2-1}^S(j) > \hat{\sigma}_{l_2-1}^S(l_1) \text{ for every } j \in \{l_1+1,\ldots,l_2-1\}.$$

Hence, the set of new-max jobs preceding $a(l_1,S)$ according to $\hat{\sigma}_{l_1-1}^S$ and $\hat{\sigma}_{l_2-1}^S$ coincide. Consequently,

$$r(l_2, a(l_1,S)) = r(l_1, a(l_1,S))$$

and we will denote $r(a(l_1,S)) = r(l_1, a(l_1,S))$. Moreover,

$$B^S(l_2,r) = B^S(l_1,r) \text{ for every } r \in \{1,\ldots,r(a(l_1,S))-1\} \tag{A.6}$$

and

$$B^S(l_2, r(a(l_1,S))) = B^S(l_1, a(l_1,S)) \cup \{l_1\}. \tag{A.7}$$

We first show (i). For $r \in \{1,\ldots,r(a(l_1,S))-1\}$

$$\begin{aligned}
g_{B^S(l_2,r)l_2} &= \sum_{j\in B^S(l_2,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2}) + \alpha_{l_2}(m-1)(\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\} - \max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}) \\
&= \sum_{j\in B^S(l_1,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2}) + \alpha_{l_2}(m-1)(\max_{j\in B^S(l_2,r)\cup\{l_2\}}\{p_j\} - \max_{j\in B^S(l_2,r-1)\cup\{l_2\}}\{p_j\}) \\
&\leq \sum_{j\in B^S(l_1,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2}) + \alpha_{l_2}(m-1)(\max_{j\in B^S(l_2,r)}\{p_j\} - \max_{j\in B^S(l_2,r-1)}\{p_j\}) \\
&= \sum_{j\in B^S(l_1,r)} (\alpha_{l_2}p_j - \alpha_j p_{l_2}) + \alpha_{l_2}(m-1)(\max_{j\in B^S(l_1,r)\cup\{l_1\}}\{p_j\} - \max_{j\in B^S(l_1,r-1)\cup\{l_1\}}\{p_j\}) \qquad (A.8) \\
&\leq \sum_{j\in B^S(l_1,r)} (\frac{p_{l_2}}{p_{l_1}}\alpha_{l_1}p_j - \alpha_j p_{l_2}) + \frac{p_{l_2}}{p_{l_1}}\alpha_{l_1}(m-1)(\max_{j\in B^S(l_1,r)\cup\{l_1\}}\{p_j\} - \max_{j\in B^S(l_1,r-1)\cup\{l_1\}}\{p_j\}) \\
&= \frac{p_{l_2}}{p_{l_1}}\Big(\sum_{j\in B^S(l_1,r)} (\alpha_{l_1}p_j - \alpha_j p_{l_1}) + \alpha_{l_1}(m-1)(\max_{j\in B^S(l_1,r)\cup\{l_1\}}\{p_j\} - \max_{j\in B^S(l_1,r-1)\cup\{l_1\}}\{p_j\})\Big) \\
&= \frac{p_{l_2}}{p_{l_1}} g_{B^S(l_1,r)l_1}
\end{aligned}$$

where the second equality follows by equation (A.6). For the first 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 $p_{l_2} \geq \max_{j\in B^S(l_2,r)}\{p_j\} > \max_{j\in B^S(l_2,r-1)}\{p_j\}$, 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 $\max_{j\in B^S(l_2,r)}\{p_j\} > p_{l_2} \geq \max_{j\in B^S(l_2,r-1)}\{p_j\}$, then

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

finally, if $\max_{j\in B^S(l_2,r)}\{p_j\} > \max_{j\in B^S(l_2,r-1)}\{p_j\} \geq p_{l_2}$, 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 equation (A.6) together with the fact that l_1 does not become new-max job since $p_{l_1} < p_{a_{u-1}} < p_{a(l_1,S)}$ by definition of l_1. The second inequality follows since $l_1 < l_2$, then: $u_{l_1} = \frac{\alpha_{l_1}}{p_{l_1}} \geq \frac{\alpha_{l_2}}{p_{l_2}} = u_{l_2}$ and therefore $\frac{p_{l_2}}{p_{l_1}}\alpha_{l_1} \geq \alpha_{l_2}$.

Analogously, one can see that $g_{B^S(l_2,r(a(l_1,S)))l_2} \leq \frac{p_{l_2}}{p_{l_1}} g_{B^S(l_1,r(a(l_1,S)))l_1}$. The only difference is that the second equality becomes an inequality by equation (A.7) and the fact that $\alpha_{l_2}p_{l_1} - \alpha_{l_1}p_{l_2} \leq 0$ since $u_{l_1} \geq u_{l_2}$.

Next, we will show (ii). Note that by definition of $r(a(l_1,S))$ and assumption (4.4) we have

$$\sum_{r=\bar{r}}^{r(a(l_1,S))} g_{B^S(l_1,r)l_1} \leq 0 \text{ for every } \bar{r} \in \{1,\ldots,r(a(l_1,S))\}. \tag{A.9}$$

Then,

$$\sum_{r=\bar{r}}^{r(a(l_1,S))} g_{B^S(l_2,r)l_2} \leq \frac{p_{l_2}}{p_{l_1}} \sum_{r=\bar{r}}^{r(a(l_1,S))} g_{B^S(l_1,r)l_1} \leq 0 \text{ for every } \bar{r} \in \{1,\ldots,r(a(l_1,S))\},$$

where the first inequality holds by (i) and the second one by equation (A.9). Therefore, $\hat{\sigma}^S_{l_2}(l_2) > \hat{\sigma}^S_{l_2}(l_1)$ by assumption (4.4) and Lemma 2.3.

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

$$\hat{\sigma}^S_{l_{\tilde{k}}-1}(l_1) < \hat{\sigma}^S_{l_{\tilde{k}}-1}(l_2) < \ldots < \hat{\sigma}^S_{l_{\tilde{k}}-1}(l_{\tilde{k}-1}). \tag{A.10}$$

Since $p_j \geq p_{a(l_k,S)}$ for every $k \in \{1,\ldots,\tilde{k}-1\}$ and every $j \in \{l_k+1,\ldots,l_{k+1}-1\}$, we have that $\hat{\sigma}^S_j(j) > \hat{\sigma}^S_j(l_k)$ by Lemma 2.2 and Lemma 2.3. Therefore, for every $k \in \{1,\ldots,\tilde{k}-1\}$ it follows

$$\hat{\sigma}^S_{l_{\tilde{k}}-1}(j) > \hat{\sigma}^S_{l_{\tilde{k}}-1}(l_k) \text{ for every } j \in \{l_k+1,\ldots,l_{k+1}-1\}. \tag{A.11}$$

Hence, for every $k \in \{2,\ldots,\tilde{k}-1\}$ we have that the set of new-max jobs in between $a(l_{k-1},S)$ and $a(l_k,S)$ according to $\hat{\sigma}^S_{l_k}$ and $\hat{\sigma}^S_{l_{\tilde{k}}}$ coincide. Therefore, for $k \in \{1,\ldots,\tilde{k}-1\}$ we have

$$r(l_{\tilde{k}}, a(l_k,S)) = r(l_k, a(l_k,S)) \tag{A.12}$$

then, we can denote $r(a(l_k,S)) = r(l_k,a(l_k,S))$. Moreover, for every $k \in \{1,\ldots,\tilde{k}-1\}$ we have

$$B^S(l_{\tilde{k}}, r) = B^S(l_k, r) \tag{A.13}$$

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

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

with

$$B^S(l_{\tilde{k}}, r(a(l_k,S))) \setminus B^S(l_k, r(a(l_k,S))) \subset \{l_k, \ldots, l_{\tilde{k}-1}\}. \tag{A.14}$$

In order to show (i) take $k \in \{1, \ldots, \tilde{k}-1\}$ and $r \in \{r(a(l_{k-1},S))+1, \ldots, r(a(l_k,S))\}$, then one can see that

$$g_{B^S(l_{\tilde{k}},r)l_{\tilde{k}}} \leq \frac{p_{l_{\tilde{k}}}}{p_{l_k}} g_{B^S(l_k,r)l_k}$$

by using the same kind of arguments as in equation (A.8).

Next, we will show (ii). Note that by definition of $r(a(l_k,S))$, $k \in \{1, \ldots, \tilde{k}-1\}$, and assumption (4.4) we have

$$\sum_{r=\bar{r}}^{r(a(l_k,S))} g_{B^S(l_k,r)l_k} \leq 0 \tag{A.15}$$

for every $\bar{r} \in \{r(a(l_{k-1},S))+1, \ldots, r(l_k, a(l_k,S))\}$. Then, for every $k^* \in \{1, \ldots, \tilde{k}-1\}$ and every $\bar{r} \in \{r(a(l_{k^*-1},S))+1, \ldots, r(a(l_{k^*},S))\}$ we have

$$\begin{aligned}
\sum_{r=\bar{r}}^{r(a(l_{\tilde{k}-1},S))} g_{B^S(l_{\tilde{k}},r)l_{\tilde{k}}} &= \sum_{k=k^*+1}^{\tilde{k}-1} \sum_{r=r(a(l_{k-1},S))+1}^{r(a(l_k,S))} g_{B^S(l_{\tilde{k}},r)l_{\tilde{k}}} + \sum_{r=\bar{r}}^{r(a(l_{k^*},S))} g_{B^S(l_{\tilde{k}},r)l_{\tilde{k}}} \\
&\leq \sum_{k=k^*+1}^{\tilde{k}-1} \frac{p_{l_{\tilde{k}}}}{p_{l_k}} \sum_{r=r(a(l_{k-1},S))+1}^{r(a(l_k,S))} g_{B^S(l_k,r)l_k} + \frac{p_{l_{\tilde{k}}}}{p_{l_{k^*}}} \sum_{r=\bar{r}}^{r(a(l_{k^*},S))} g_{B^S(l_{k^*},r)l_{k^*}} \\
&\leq 0
\end{aligned}$$

where the first inequality holds by (i) and the second one by equation (A.15). Therefore, $\hat{\sigma}^S_{l_{\tilde{k}}}(l_{\tilde{k}}) > \hat{\sigma}^S_{l_{\tilde{k}}}(l_{\tilde{k}-1})$ by assumption (4.4) and Lemma 2.3. □

Proof of Lemma 4.2. Recall that $\sigma_0 = \sigma_u = (1 \ \ldots \ n)$ and $\{a_1, \ldots, a_s\}$ is the set of new-max jobs according to σ_u with $a_1 < \ldots < a_s$. We distinguish three cases.

Case 1: $S \cap \{a_1, \ldots, a_s\} = \emptyset$. 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 4.1, and assertion (iv) follows since $G^T_i \geq 0 = G^S_i$ by definition of G^T_i.

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

Case 3: $S \cap \{a_1, \ldots, a_s\} = \{a_u, \ldots, a_v\}$ and $T \cap \{a_1, \ldots, a_s\} = \{a_{\tilde{u}}, \ldots, a_{\tilde{v}}\}$ with $a_{\tilde{u}} < a_u \leq a_v \leq a_{\tilde{v}}$. Let $S = \{i_S, \ldots, j_S\}$ and partition S according to (A.5). Let $i \in S$ and let a be a new-max job according to $\hat{\sigma}^S_{i-1}$ $(\hat{\sigma}^T_{i-1})$. During the remaining of this proof we will denote by $r^S(i,a)$ $(r^T(i,a))$ the index of the segment defined by a according to $\hat{\sigma}^S_{i-1}$ $(\hat{\sigma}^T_{i-1})$. Moreover, by $s_i(S)$ we denote the number of segments before reordering player i in S.

Note that for every $i \in \{i_S, \ldots, a_u\}$ we have $\hat{\sigma}^S_i = \sigma_u$ and therefore assertions (i), (ii), (iii), and (iv) follow using the same kind of reasoning as in Case 1.

Subsequently, assume that the result holds for $\{i_S, \ldots, l_k - 1\}$ for some $k \in \{1, \ldots, m\}$. Then, we have

$$B^T(l_k, r^T(l_k, a_{u-1}) + r) = B^S(l_k, r) \tag{A.16}$$

for every $r \in \{1, \ldots, s_{l_k}(S)\} \setminus \{r^S(l_k, a(l_1, S)), \ldots, r^S(l_k, a(l_{k-1}, S))\}$.

Besides, $B^T(l_k, r^T(l_k, a_{u-1}) + r) \subset B^S(l_k, r)$ for every $r \in \{r^S(l_k, a(l_1, S)), \ldots, r^S(l_k, a(l_{k-1}))$ with

$$B^S(l_k, r) \setminus B^T(l_k, r^T(l_k, a_{u-1}) + r) \subset \{l_1, \ldots, l_{k-1}\}. \tag{A.17}$$

Note that it may be the case that $a(l_k, S) = a(l_{k+1}, S)$ for some $k \in \{1, \ldots, k-2\}$. We define recursively k^*_w, with $w \in \{1, \ldots, t\}$, as

$$k^*_w = \min\{\bar{k} \in \{k^*_{w-1} + 1, \ldots, k-1\} \,|\, a(l_{\bar{k}}, S) \neq a(l_{\bar{k}-1}, S)\} \tag{A.18}$$

where $k^*_0 = 0$. Note that $\hat{\sigma}^S_{l_k-1}(l_{k^*_{w-1}}) < \hat{\sigma}^S_{l_k-1}(l_{k^*_w})$ for every $w \in \{2, \ldots, t\}$ by Lemma A.4 (ii).

Then, by equation (A.16) it follows

$$g_{B^T(l_k, r^T(l_k, a_{u-1})+r)l_k} = g_{B^S(l_k, r)l_k} \tag{A.19}$$

for every $r \in \{1, \ldots, s_{l_k}(S)\} \setminus \{r^S(l_k, a(l_1, S)), \ldots, r^S(l_k, a(l_{k-1}, S))\}$.

Besides, one can see analogously as in equation (A.8) that

$$g_{B^T(l_k, r^T(l_k, a_{u-1})+r^S(l_k, a(l_{k^*_w}, S)))l_k} \leq \frac{p_{l_k}}{p_{l_{k^*_w}}} g_{B^S(l_{k^*_w}, r^S(l_{k^*_w}, a(l_{k^*_w}, S)))l_{k^*_w}} \tag{A.20}$$

for every $w \in \{1, \ldots, t\}$.

By definition of $r^S(l_{k^*_w}, S)$, $w \in \{1, \ldots, t\}$, and assumption (4.4) we know that

$$\sum_{r=\bar{r}}^{r^S(l_{k^*_w}, a(l_{k^*_w}, S))} g_{B^S(l_{k^*_w}, r)l_{k^*_w}} \leq 0 \tag{A.21}$$

for every $\bar{r} \in \{r^S(l_{k^*_w}, a(l_{k^*_{w-1}}, S)) + 1, \ldots, r^S(l_{k^*_w}, a(l_{k^*_w}, S))\}$. Then,

$$\begin{aligned} \sum_{r=\bar{r}}^{r^S(l_k, a(l_{k^*_t}, S))} g_{B^T(l_k, r^T(l_k, a_{u-1})+r)l_k} &= \sum_{w=\bar{w}+1}^{t} \sum_{r=r^S(l_k, a(l_{k^*_{w-1}}, S))}^{r^S(l_k, a(l_{k^*_w}, S))} g_{B^T(l_k, r^T(l_k, a_{u-1})+r)l_k} + \sum_{r=\bar{r}}^{r^S(l_k, a(l_{k^*_{\bar{w}}}, S))} g_{B^T(l_k, r^T(l_k, a_{u-1})+r)l_k} \\ &\leq \sum_{w=\bar{w}+1}^{t} \frac{p_{l_k}}{p_{l_{k^*_w}}} \sum_{r=r^S(l_k, a(l_{k^*_{w-1}}, S))}^{r^S(l_k, a(l_{k^*_w}, S))} g_{B^S(l_{k^*_w}, r)l_{k^*_w}} + \frac{p_{l_k}}{p_{l_{k^*_{\bar{w}}}}} \sum_{r=\bar{r}}^{r^S(l_k, a(l_{k^*_{\bar{w}}}, S))} g_{B^S(l_{k^*_{\bar{w}}}, r)l_{k^*_{\bar{w}}}} \\ &\leq 0 \end{aligned} \tag{A.22}$$

for every $\bar{w} \in \{1, \ldots, t\}$ and every $\bar{r} \in \{r^S(l_k, a(l_{k^*_{\bar{w}-1}}, S))+1, \ldots, r^S(l_k, a(l_{k^*_{\bar{w}}}, S))\}$. Here, the first inequality holds by equation (A.20) and the second one by equation (A.21).

First, suppose that $p_{l_k} \geq p_{a_u}$. Then, $\hat{\sigma}^T_{l_k}(l_k) > \hat{\sigma}^T_{l_k}(a_u)$ by Lemma 2.2. Hence, $\hat{\sigma}^T_{l_k}(l_k) = \hat{\sigma}^S_{l_k}(l_k)$ by equations (A.19) and (A.22), and by assumption (4.4).

Second, suppose that $p_{l_k} < p_{a_u}$. Then, it may be the case that for some $\hat{r} \in \{1, \ldots, r^T(l_k, a_{u-1})\}$ we have $\sum_{r=\hat{r}}^{r^T(l_k, a(l_k,S))} g_{B^T(l_k,r)l_k} > 0$. In such a case $\hat{\sigma}^T_{l_k}(l_k) < \hat{\sigma}^S_{l_k}(l_k)$ and $\hat{\sigma}^T_{l_k}(l_k) < \hat{\sigma}^T_{l_k}(a_u)$, otherwise $\hat{\sigma}^T_{l_k}(l_k) = \hat{\sigma}^S_{l_k}(l_k)$ by equation (A.19) and assumption (4.4).

Hence (i) and (ii) are satisfied. Assertion (iii) is an immediate consequence of Lemma A.3, and the fact that l_k is not a new-max job according $\hat{\sigma}^S_{l_k}$ since $p_{l_k} < p_{a(l_{k-1},S)} < p_{a(l_k,S)}$. Assertion (iv) is a direct consequence of (ii) together with equation (A.19).

Finally, suppose that the result is true for $\{i_S, \ldots, i-1\}$ with $l_k < i < l_{k+1}$. Then, we have

$$B^T(i, r^T(l_k, a_{u-1}) + r) = B^S(i, r) \tag{A.23}$$

and

$$g_{B^T(i,r^T(l_k,a_{u-1})+r)i} = g_{A^S(i,r)i} \tag{A.24}$$

for every $r \in \{1, \ldots, r(i, a(l_k, S)) - 1\}$.

Moreover, $\hat{\sigma}^T_i(i) > \hat{\sigma}^T_i(a_u)$ by Lemma 2.2. Hence, $\hat{\sigma}^T_i(i) = \hat{\sigma}^S_i(i)$ by equation (A.24) and assumption (4.4). Assertion (iii) follows by induction together with Lemma A.3, and (i). Assertion (iv) is a direct consequence of (i) together with equation (A.24). □

Proof of Lemma 4.3. Recall that $\sigma_0 = \sigma_u = (1 \ \ldots \ n)$ and $\{a_1, \ldots, a_s\}$ is the set of new-max jobs according to σ_u with $a_1 < \ldots < a_s$. We will distinguish three cases.

<u>Case 1:</u> $S \cap \{a_1, \ldots, a_s\} = \emptyset$. Then, $\hat{\sigma}^S_i = \sigma_u$ for every $i \in S$ and

$$G^S_i = 0 = \sum_{r:N(i,r)\subset S} h_{N(i,r)}$$

by definition of $h_{N(i,r)}$.

<u>Case 2:</u> $a_1 \in S$. Then, we have $\hat{\sigma}^S_i = \hat{\sigma}^N_i$ for every $i \in S$ and

$$G^S_i = G^N_i = \sum_{r:N(i,r)\subset S} h_{N(i,r)}.$$

<u>Case 3:</u> $a_1 \notin S$, $S \cap \{a_1, \ldots, a_s\} = \{a_u, \ldots, a_v\}$ with $a_u \leq \ldots \leq a_v$. Let $S = \{i_S, \ldots, j_S\}$ and consider the partition (A.5). Let $k \in \{1, \ldots, m\}$ and let $i \in \{l_k + 1, \ldots, l_{k+1} - 1\}$. By Lemma 4.2 (i) and (iv) we know

that $\hat{\sigma}_i^N(i) = \hat{\sigma}_i^S(i)$ and $G_i^N = G_i^S$. Hence,

$$G_i^S = G_i^N = \sum_{r:N(i,r)\subset S} h_{N(i,r)}$$

where the second equality follows by $\hat{\sigma}_i^N(i) = \hat{\sigma}_i^S(i)$ and the fact that $h_{N(i,r)} = 0$ for every $r \geq r(i, a(i,S))$.

Next, consider l_k with $k \in \{1, \ldots, m\}$. If $\hat{\sigma}_{l_k}^N(l_k) = \hat{\sigma}_{l_k}^S(l_k)$ we are in the previous situation. Assume that $\hat{\sigma}_{l_k}^N(l_k) < \hat{\sigma}_{l_k}^S(l_k)$. Then,

$$G_i^S = \sum_{r=1}^{r(i,a(S,i))-1} g_{A^S(i,r)i} = \sum_{r=1}^{r(i,a(S,i))-1} g_{A^N(i,r)i} = \sum_{r=1}^{r(i,a(S,i))-1} h_{N(i,r)} = \sum_{r:N(i,r)\subset S} h_{N(i,r)}$$

where the first equality follows by definition of G_i^S and $r(i, a(S,i))$, the second one by equation (A.19) with $T = N$ and the fact that $B^S(i,r) = A^S(i, s_i - r + 1)$, the third equality is a direct consequence of the definition of $h_{N(i,r)}$ and the last one follows by equation (A.22) with $B^S(i,r) = A^S(i, s_i - r + 1)$ and the definition of $h_{N(i,r)}$. □

References

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

Allahverdi, A. and 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. and 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. and 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. and Hamers, H. (2006), 'Job scheduling, cooperation, and control', *Operations Research Letters* **34**, 22–28.

Cheng, T. C. E. and Shakhlevich, N. (1999), 'Proportionate flow shop with controllable processing times', *Journal of Scheduling* **2**(6), 253–265.

Curiel, I., Hamers, H. and Klijn, F. (2002), Sequencing games: a survey, *in* P. Borm and H. Peters, eds, 'Chapters in Game Theory: in Honor of Stef Tijs', Kluwer Academic Publishers, Boston, pp. 27–50.

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

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

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

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

Hamers, H., Klijn, F. and Suijs, J. (1999), 'On the balancedness of m-sequencing games', *European Journal of Operational Research* **119**, 678–691.

Ichiishi, T. (1981), 'Super-modularity: Applications to convex games and to the greedy algorithm for LP', *Journal of Economic Theory* **25**, 283–286.

Shakhlevich, N., Hoogeveen, H. and 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 and A. W. Tucker, eds, 'Contribution to the Theory of Games II', Vol. 28 of Annals of Mathematics Studies, Princeton University Press, pp. 307–317.

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. and 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. (1993), Games and Graphs in Economic Situations, PhD thesis, Tilburg University.

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