On the Balancedness of m-Sequencing Games

Herbert Hamers Flip Klijn Jeroen Suijs

Department of Econometrics and CentER, Tilburg University,

P.O.Box 90153, 5000 LE Tilburg, The Netherlands

March 4, 1998

Abstract: This paper studies m-sequencing games that arise from sequencing situations with m parallel and identical machines. These m-sequencing games, which involve n players, give rise to m-machine games, which involve m players. Here, n corresponds to the number of jobs in an m-sequencing situation, and m corresponds to the number of machines in the same m-sequencing situation. We prove that an m-sequening game is balanced if and only if the corresponding m-machine game is balanced. Furthermore, it is shown that m-sequencing games are balanced if $m \in \{1, 2\}$. Finally, if $m \geq 3$, balancedness is established for two special classes of m-sequencing games.

Journal of Economic Literature Classification Number: C71

Keywords: cooperative games, sequencing situations

1 Introduction

Sequencing or scheduling situations consist of a number of jobs (tasks, operations) that have to be processed on a number of machines. The processing time of a job on a specific machine is the time this machine takes to handle this job. Sequencing situations can be classified by the number of machines, the specific properties of the machines (e.g. parallel, identical), the chosen restrictions on the jobs (e.g. ready times, due dates, flow constraints), and the chosen cost criterion (e.g. weighted completion time, make-span).

By assuming that there exists an initial schedule before the machines start processing, we can establish a relation between cooperative games and sequencing situations in the following way. Letting each agent own exactly one of the scheduled jobs a group of agents (a coalition) can save costs by rearranging their jobs in a way that is admissible with respect to this initial schedule. By defining the value of a coalition as the maximal cost savings a coalition can make in this way, we obtain a cooperative sequencing game related to a sequencing situation.

The above game-theoretic approach was initiated by *Curiel, Pederzoli, and Tijs (1989)*. They considered the class of one-machine sequencing situations in which no restrictions on the jobs are imposed and the weighted completion time criterion was chosen as the cost criterion. Convexity was shown for the corresponding sequencing games. *Hamers, Borm, and Tijs (1995)* extended the class of one-machine sequencing situations considered by *Curiel et al. (1989)* by imposing ready times on the jobs. In this case the corresponding sequencing games are balanced, but not necessarily convex. However, for a special subclass of sequencing games convexity could be established. Similar results are also obtained in *Borm and Hamers (1998)* in which due dates are imposed on the jobs. Instead of imposing restrictions on the jobs, *van den Nouweland, Krabbenborg, and Potters (1992)* extended the number of machines. They considered m-machine sequencing situations with respect to flow shops and a dominant machine. Convexity was established for the special class in which the first machine is dominant. In general the corresponding sequencing games need not be balanced.

This paper considers sequencing situations with m parallel and identical machines in which no restrictions on the jobs are imposed. Again, the weighted completion time criterion is used. Furthermore, each agent has one job that has to be processed on precisely one machine. These sequencing situations, which will be refered to as m-machine sequencing situations, give rise to the class of m-sequencing games. A formal description of the model and the corresponding games can be found in Section 2.

For 1-machine sequencing situations the corresponding class of sequencing games coincides with the class of sequencing games introduced by *Curiel et al. (1989)*. This paper shows that sequencing games arising from 2-machine sequencing situations are balanced, but not necessarily convex. In fact, we show that these sequencing games are σ-component additive games. Generally, m-sequencing games, whose cardinality depends on the number of jobs, give rise to m-machine games, whose cardinality depends on the number of machines. We prove that an m-sequening game is balanced if and only if the corresponding m-machine game is balanced. Finally, for two special subclasses of m-machine sequencing situations, with $m \geq 3$, we show that the related games are balanced. For the proof of this result we turn to the class of permutation games, which are totally balanced (cf. *Tijs et al. (1984)*).

2 The model

This section describes the sequencing situations with m parallel and identical machines, which will be referred to as m-machine sequencing situations, and the corresponding class of m-sequencing games.

In an m-machine sequencing situation each agent has one job that has to be processed on precisely one machine. Each job can be processed on any machine. The finite set of machines is denoted by $M = \{1, ..., m\}$ and the finite set of agents is denoted by $N = \{1, ..., n\}$. We assume that each machine starts processing at time 0 and that the processing time of each job is independent of the machine the job is processed on. The processing time of the job of agent i is denoted by $p_i \geq 0$. We assume that every agent has a linear monetary cost function $c_i : [0, \infty) \rightarrow \mathbb{R}$ defined by $c_i(t) = \alpha_i t$ where $\alpha_i > 0$ is a (positive) cost coefficient.

By a one to one map $b : N \rightarrow \{1, ..., m\} \times \{1, ..., n\}$ we can describe on which machine and in which position on that machine the job of an agent will be processed. Specifically,

$b(i) = (r, j)$ means that agent i is assigned to machine r and that (the job of) agent i is in position j on machine r. Such a map b will be called a (processing) schedule.

In the following an m-machine sequencing situation will be described by (M, N, b^0, p, α), where $M = \{1, ..., m\}$ is the set of machines, $N = \{1, ..., n\}$ the set of agents, b^0 the initial schedule, $p \in \mathbb{R}^N_+$ the processing times, and $\alpha = (\alpha_i)_{i \in N} \in \mathbb{R}^N_{++}$ the cost coefficients.

The starting time $t(b, i)$ of the job of agent i if processed in a semi-active way according to a schedule b equals

$$t(b, i) = \sum_{j \in N : b(j) \prec b(i)} p_j,$$

where $b(j) \prec b(i)$ if and only if the job of the agents j and i are on the same machine (i.e. $b(j)_1 = b(i)_1$) and j precedes i (i.e. $b(j)_2 < b(i)_2$). Consequently, the completion time $C(b, i)$ of the job of agent i with respect to b is equal to $t(b, i) + p_i$. The total costs $c_b(S)$ of a coalitions $S \subseteq N$ with respect to the schedule b is given by

$$c_b(S) = \sum_{i \in S} \alpha_i (C(b, i)). \tag{1}$$

We will restrict attention to m-machine sequencing situations (M, N, b^0, p, α) that satisfy the following condition: the starting time of a job that is in the last position on a machine with respect to b^0 is smaller than or equal to the completion time of each job that is in the last position with respect to b^0 on the other machines. Formally, let i_k be the last agent on machine k with respect to b^0, then for any $k \in M$ we demand that

$$t(b^0, i_k) \leq C(b^0, i_s) \text{ for all } s \in M. \tag{2}$$

This condition states that each job that is in the last position of a machine cannot make any profit by joining the end of a queue of any other machine. These schedules can arise in the following way. Let the agents enter one by one the machines before the processing starts. If an agent enters he will choose the queue of a machine that gives him the shortest waiting time.

The (maximal) cost savings of a coalition S depend on the set of admissible rearrangements of this coalition. We call a schedule $b : N \to \{1, ..., m\} \times \{1, ..., n\}$ *admissible for* S with respect to b^0 if it satisfies the following two conditions:
(i) Two agents $i, j \in S$ which are on the same machine can only switch if all agents in between i and j on that machine are also members of S;
(ii) Two agents $i, j \in S$ which are on different machines can only switch places if the tail of i and the tail of j are contained in S. The tail of an agent i is the set of agents that follow agent i on his machine, i.e. the set of agents $k \in N$ with $b(i) \prec b(k)$.
The set of admissible schedules for a coalition S is denoted by B_S. An admissible schedule for coalition N will be called a schedule.

Before formally introducing sequencing games, we recall some facts concerning cooperative games.

A *cooperative* game is a pair (N, v) where N is a finite set of players (agents) and v is a map $v : 2^N \to \mathbb{R}$ with $v(\emptyset) = 0$, and 2^N the collection of all subsets of N.

Cooperative game theory focuses on 'fair' and/or 'stable' division rules for the value of $v(N)$ of the grand coalition. A core element $x = (x_i)_{i \in N} \in \mathbb{R}^N$ divides the value $v(N)$ among the players in such a way that no coalition has an incentive to split off, i.e.,

$$x(N) = v(N) \quad \text{and} \quad x(S) \geq v(S) \quad \text{for all } S \in 2^N,$$

where $x(S) = \sum_{i \in S} x_i$ for all $S \in 2^N$. The core $C(N, v)$ consists of all core elements. A game is called *balanced* if its core is non-empty. A game (N, v) is called *totally balanced* if each subgame $(S, v_{|S})$ is balanced, were $v_{|S}$ is defined by $v_{|S}(T) = v(T)$ for all $T \subseteq S$.
Convex games, for instance, are (totally) balanced games. A game (N, v) is called *convex* if for all coalitions $S, T \in 2^N$ and all $i \in N$ with $S \subseteq T \subseteq N \backslash \{i\}$ it holds that

$$v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T).$$

A nice property of convex games is that all marginal vectors belong to the core (cf. *Shapley (1971)*). A marginal vector $m^\sigma(v)$ is defined by

$$m_i^\sigma(v) := v(\{j : \sigma(j) \leq \sigma(i)\}) - v(\{j : \sigma(j) < \sigma(i)\}),$$

for all $i \in N$, and all permutations σ of N.

Permutation games, introduced in *Tijs et al. (1984),* are also totally balanced games. Let $A = [a_{ij}]_{i=1}^{n}{}_{j=1}^{n}$ be a square matrix. Then a *permutation game* (N, r) is defined by

$$r(S) = \max_{\pi \in \Pi_S} \sum_{i \in S} [a_{ii} - a_{i\pi(i)}] \text{ for all } S \subseteq N,$$

where Π_S is the set of permutations of coalition S.

Another class of balanced games is the class of σ-component additive games, introduced in *Curiel et al. (1994 A,B)*. For the definition of this class of games we need some preliminaries. A game (N, v) is called *superadditive* if $v(S) + v(T) \leq v(S \cup T)$ for all $S, T \in 2^N$ with $S \cap T = \emptyset$. Let $\sigma : N \to \{1, \ldots, n\}$ be an order of the player set N. A coalition T is called *connected* with respect to σ if for all $i, j \in T$ and $k \in N$ such that $\sigma(i) < \sigma(k) < \sigma(j)$ it holds that $k \in T$. A connected coalition $T \subseteq S$ is a *component* of S if $T \cup \{i\}$ is not connected for every $i \in S \backslash T$. The components of S form a partition of S, denoted by S/σ. A game (N, v) is called a *σ-component additive game* if it satisfies the following two conditions:

$$\begin{array}{ll} (i) & v(S) = \sum_{T \in S/\sigma} v(T) \quad \text{for all } S \subseteq N; \\ (ii) & (N, v) \text{ is superadditive.} \end{array}$$

By defining the worth of a coalition as the maximum cost savings a coalition can achieve by means of admissible schedules we obtain a cooperative game called an m-sequencing game. Formally, for an m-machine sequencing situation (M, N, b^0, p, α) the corresponding *m-sequencing game* (N, v) is defined by

$$v(S) = \max_{b \in B_S} \{ \sum_{i \in S} \alpha_i [C(b^0, i) - C(b, i)] \} \tag{3}$$

for all $S \in 2^N \backslash \{\emptyset\}$ and $v(\emptyset) = 0$.

3 On the balancedness of m-sequencing games

In this section we present our results with respect to the balancedness of m-sequencing games.

The definition of m-sequencing games implies that 1-sequencing games coincide with the class of sequencing games introduced in *Curiel et al. (1989)*. Since Curiel et al. showed that the sequencing games they considered are convex, we also have that 1-sequencing games are convex, and consequently, are balanced.

The next example shows that 2-sequencing games need not be convex.

Example 3.1 Let $M = \{1, 2\}$, $N = \{1, \ldots, 5\}$, $p = (2, 1, 1, 2, 1)$, and $\alpha = (1, 1, 1, 1, 1)$. The initial schedule b^0 is given in Figure 1.

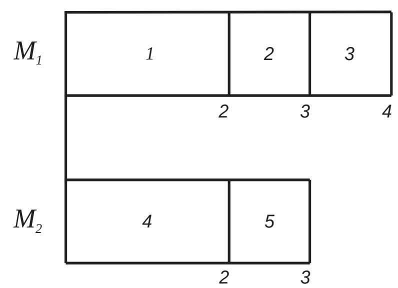

Figure 1: The schedule b^0

Take $T = \{1, 3, 4, 5\}$, $S = \{1, 3\}$, and $i = 2$. Let (N, v) be the corresponding 2-sequencing game. Then $v(T \cup \{i\}) = 3$, $v(T) = 2$, $v(S \cup \{i\}) = 2$, and $v(S) = 0$. From this we conclude that (N, v) is not a convex game:

$$v(T \cup \{i\}) - v(T) = 1 < 2 = v(S \cup \{i\}) - v(S).$$

$\diamond$

The following Theorem shows that 2-sequencing games are balanced.

Theorem 3.1 *Let (M, N, b^0, p, α) be such that $|M| = 2$. Then the corresponding 2-sequencing game (N, v) is balanced.*

Proof. Let $i_1, i_2, ..., i_{m_1}$ be the jobs on machine 1 such that $b^0(i_x) \prec b^0(i_y)$ if $x < y$ and let $i_n, ..., i_{m_1+1}$ be the jobs on machine 2 such that $b^0(i_x) \prec b^0(i_y)$ if $x > y$. Take $\sigma \in \Pi(N)$ such that $\sigma(j) = i_j$ for all $j \in N$. From superadditivity of (N, v) together with the conditions of admissible schedules it follows that (N, v) is a σ-component additive game. Since σ-component additive games are balanced, we have that 2-sequencing games are balanced. $\square$

Next, let us turn to m-sequencing games. Based on an m-machine sequencing situation we define a new cooperative game with m players. Let (M, N, b^0, p, α) be an m-machine sequencing situation and let (N, v) be the corresponding m-sequencing game. The set of players whose jobs are on machine $k \in M$ according to the initial schedule b^0 will be denoted by $N_k(b^0)$. Then an m-machine game (M, w) is defined by

$$w(K) := v(\bigcup_{k \in K} N_k(b^0)) - \sum_{k \in K} v(N_k(b^0)),$$

for every coalition $K \subseteq M$ of machines. The worth $w(K)$ of a coalition of machines $K \subseteq M$ is the extra cost savings the machines in K can make when they decide to cooperate with each other.

The next theorem says that the core of an m-sequencing game is non-empty whenever the core of the m-machine game is non-empty, and vice versa.

Theorem 3.2 *Let (M, N, b^0, p, α) be an m-sequencing situation. Let (N, v) be the corresponding m-sequencing game and let (M, w) be the corresponding m-machine game. Then (N, v) is balanced if and only if (M, w) is balanced.*

Proof. First, we prove the 'only if' part. Let $x \in C(N, v)$. For $k \in M$ we define

$$y_k := x(N_k(b^0)) - v(N_k(b^0)).$$

We show that $y \in C(N, w)$. Let $K \subseteq M$. Then,

$$\begin{aligned} y(K) &= \sum_{k \in K} x(N_k(b^0)) - \sum_{k \in K} v(N_k(b^0)) \\ &\geq v(\bigcup_{k \in K} N_k(b^0)) - \sum_{k \in K} v(N_k(b^0)) \\ &= w(K). \end{aligned}$$

For $K = M$ we have an equality, since $x(\bigcup_{k \in M} N_k(b^0)) = x(N) = v(N) = v(\bigcup_{k \in M} N_k(b^0))$.

Second, we prove the 'if' part. Let $y \in C(M, w)$. For convenience we introduce some notation. Let $n_k(b^0)$ be the number of jobs on machine k with respect to b^0, i.e. $n_k(b^0) = \mid N_k(b^0) \mid$, and for any machine $k \in M$ let $\sigma_k : N_k(b^0) \to \{1, \ldots, n_k(b^0)\}$ be the initial order on machine k, i.e. $\sigma_k(i) < \sigma_k(j)$ if and only if $b^0(i) \prec b^0(j)$ for all $i, j \in N_k(b^0)$. For a job $i \in N_k(b^0)$ and any $k \in M$ we define

$$x_i := v(\{j : \sigma_k(j) \leq \sigma_k(i)\}) - v(\{j : \sigma_k(j) < \sigma_k(i)\}),$$

and

$$\hat{x}_i := \begin{cases} x_i & \text{if } \sigma_k(i) \neq n_k(b^0); \\ x_i + y_k & \text{if } \sigma_k(i) = n_k(b^0). \end{cases}$$

Note that $(x_i)_{i \in N_k(b^0)}$ is a marginal vector of the subgame $(N_k(b^0), v_{|N_k(b^0)})$. We prove in four steps that $\hat{x} \in C(N, v)$.

The first step shows that $\hat{x}$ is efficient. This follows from

$$\begin{aligned}
\sum_{i\in N}\hat{x}_i &= \sum_{k\in M}\sum_{i\in N_k(b^0)}\hat{x}_i \\
&= \sum_{k\in M}[v(N_k(b^0))+y_k] \\
&= \sum_{k\in M}y_k+\sum_{k\in M}v(N_k(b^0)) \\
&= w(M)+\sum_{k\in M}v(N_k(b^0)) \\
&= v(N)-\sum_{k\in M}v(N_k(b^0))+\sum_{k\in M}v(N_k(b^0)) \\
&= v(N).
\end{aligned}$$

For the second step of the proof, take $T\subseteq N$. Define $T_k=T\cap N_k(b^0)$ for $k\in M$, and let $\tilde{T}_k$ be the component with respect to σ_k of T_k that contains the last player on machine k. Formally,

$$\tilde{T}_k := \{S\in T_k/\sigma_k \,:\, \sigma_k^{-1}(n_k(b^0))\in S\}.$$

Note that $\tilde{T}_k$ is the empty set if $\sigma_k^{-1}(n_k(b^0))\notin T_k$. Next, let $\tilde{T}_k$ be non-empty and let $i_1,i_2,\ldots,i_{\tilde{t}_k}\in\tilde{T}_k$ be the elements of $\tilde{T}_k$ such that $\sigma_k(i_1)<\sigma_k(i_2)<\cdots<\sigma_k(i_{\tilde{t}_k})$. Then

$$\begin{aligned}
\sum_{i\in\tilde{T}_k}x_i &= \sum_{i\in\tilde{T}_k}v(\{j:\sigma_k(j)\le\sigma_k(i)\})-v(\{j:\sigma_k(j)<\sigma_k(i)\}) \\
&= \sum_{l=1}^{\tilde{t}_k}v(\{j:\sigma_k(j)\le\sigma_k(i_l)\})-v(\{j:\sigma_k(j)<\sigma_k(i_l)\}) \\
&= v(\{j:\sigma_k(j)\le\sigma_k(i_{\tilde{t}_k})\})-v(\{j:\sigma_k(j)<\sigma_k(i_1)\}) \\
&= v(N_k(b^0))-v(N_k(b^0)\backslash\tilde{T}_k), \qquad (4)
\end{aligned}$$

where the third equality follows from

$$v(\{j:\sigma_k(j)\le\sigma_k(i_l)\})=v(\{j:\sigma_k(j)<\sigma_k(i_{l+1})\}) \quad \text{for } 1\le l<\tilde{t}_k.$$

In the third step, let $S\in T_k/\sigma_k$ be such that $S\neq\tilde{T}_k$. Since the subgame $(N_k(b^0),v_{|N_k(b^0)})$ is a 1-machine sequencing game it follows that this game is convex. Hence, the marginal vector $(x_i)_{i\in N_k(b^0)}\in C(N_k(b^0),v_{|N_k(b^0)})$ (cf. *Shapley (1971)*). This implies that

$$\sum_{i\in S}\hat{x}_i=\sum_{i\in S}x_i\ge v_{|N_k(b^0)}(S)=v(S). \qquad (5)$$

Finally, in the fourth part we show that $\sum_{i\in T}\hat{x}_i\ge v(T)$.

$$\sum_{i\in T}\hat{x}_i = \sum_{k\in M}\sum_{\substack{S\,\in\,T_k/\sigma_k\\ S\,\neq\,\tilde{T}_k}}\sum_{i\in S}x_i+\sum_{k\in M:\tilde{T}_k\neq\emptyset}\left((\sum_{i\in\tilde{T}_k}x_i)+y_k\right)$$

$$
\begin{aligned}
&\geq \sum_{k\in M} \sum_{\substack{S \in T_k/\sigma_k \\ S \neq \tilde{T}_k}} v(S) + \sum_{k\in M:\tilde{T}_k\neq\emptyset} y_k \\
&\quad + \sum_{k\in M:\tilde{T}_k\neq\emptyset} \sum_{i\in\tilde{T}_k} x_i \\
&= \sum_{k\in M} \sum_{\substack{S \in T_k/\sigma_k \\ S \neq \tilde{T}_k}} v(S) + \sum_{k\in M:\tilde{T}_k\neq\emptyset} y_k \\
&\quad + \sum_{k\in M:\tilde{T}_k\neq\emptyset} [v(N_k(b^0)) - v(N_k(b^0)\backslash\tilde{T}_k)] \\
&\geq \sum_{k\in M} \sum_{\substack{S \in T_k/\sigma_k \\ S \neq \tilde{T}_k}} v(S) + v(\bigcup_{k\in M:\tilde{T}_k\neq\emptyset} N_k(b^0)) - \sum_{k\in M:\tilde{T}_k\neq\emptyset} v(N_k(b^0)) \\
&\quad + \sum_{k\in M:\tilde{T}_k\neq\emptyset} [v(N_k(b^0)) - v(N_k(b^0)\backslash\tilde{T}_k)] \\
&= \sum_{k\in M} \sum_{\substack{S \in T_k/\sigma_k \\ S \neq \tilde{T}_k}} v(S) + v(\bigcup_{k\in M:\tilde{T}_k\neq\emptyset} N_k(b^0)) - \sum_{k\in M:\tilde{T}_k\neq\emptyset} v(N_k(b^0)\backslash\tilde{T}_k) \\
&\geq \sum_{k\in M} \sum_{\substack{S \in T_k/\sigma_k \\ S \neq \tilde{T}_k}} v(S) + v(\bigcup_{k\in M:\tilde{T}_k\neq\emptyset} \tilde{T}_k) \\
&= v(T),
\end{aligned}
$$

where the first inequality follows from (5), the second inequality follows from $y \in C(M, w)$, and the third inequality follows from the superadditivity of v. The second equality follows from (4). □

Theorem 3.2 implies that in order to check whether an 3-sequencing game with n players is balanced or not, it is sufficient to compute $w(\{1, 2\}), w(\{1, 3\}), w(\{2, 3\})$, and $w(\{1, 2, 3\})$ ($w(\{k\}) = 0$ for all $k \in M$), and then check whether this 3-machine game is balanced or not. The following example illustrates this.

Example 3.2 Let $M = \{1, 2, 3\}$, $N = \{1, ..., 20\}$, $\alpha = (1, \ldots, 1)$, and processing times and the initial schedule b^0 as in Figure 2. Let (N, v) be the corresponding 3-sequencing game and (M, w) be the corresponding 3-machine game. Some calculations give $w(\{1, 2\}) = 3, w(\{1, 3\}) = 7, w(\{2, 3\}) = 0$, and $w(\{1, 2, 3\}) = 7$. Clearly, $(7, 0, 0) \in C(M, w)$. Hence, the game (M, w) is balanced. By Theorem 3.2 (N, v) is balanced. $\diamond$

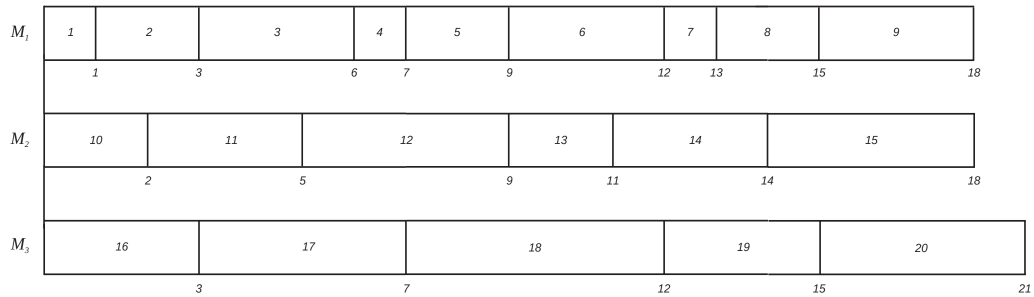

Figure 2: The schedule b^0

Moreover, Theorem 3.2 provides an alternative proof for Theorem 3.1. It readily follows that 2-machine games are convex games. Hence, 2-sequencing games are balanced.

Consider m-sequencing situations in which all cost coefficients are equal to one. The next theorem says that the corresponding m-sequencing games are balanced.

Theorem 3.3 *Let (N, v) be the m-sequencing game that arises from an m-machine sequencing situation (M, N, b^0, p, α) in which $\alpha_i = 1$ for all $i \in N$. Then (N, v) is balanced.*

In the remaining part of this section we will provide the proof of Theorem 3.3. First it is shown that we can restrict attention to m-sequencing games that arise from m-machine sequencing situations in which each machine initially has to proces an equal number of jobs. Second, we prove that the corresponding m-machine games corrrespond to permutation games. Third, we show that m-machine games are balanced. From Theorem 3.2 we can then conclude that m-sequencing games are balanced.

Let (M, N, b^0, p, α) be an m-machine sequencing situation in which $\alpha_i = 1$. An optimal schedule $\hat{b}(N)$ of coalition N is established (see e.g. *Conway, Maxwell, and Miller (1967)*) by first ordering the jobs of the players in N in a non-decreasing order, i.e., $p_{i_1} \leq p_{i_2} \leq ... \leq p_{i_n}$ where $\{i_1, i_2, ..., i_n\} = N$. Second, assign the jobs, after numbering the machines, in rotation to the machines:

$$\begin{array}{lccccccccccccccc} \text{Job of player} & i_1 & i_2 & ... & i_m & | & i_{m+1} & i_{m+2} & ... & i_{2m} & | & ... & | & i_{n-r} & ... & i_n & \\ \text{Machine} & 1 & 2 & ... & m & | & 1 & 2 & ... & m & | & ... & | & 1 & ... & r & ... & m \end{array}$$

Hence, for an optimal schedule that is obtained by the above described procedure, we can conclude that each machine in $\{1, ..., r\}$ has an equal number of jobs and each machine in $\{r+1, ..., m\}$ has an equal number of jobs. Moreover, the number of jobs on the first r machines is one higher than the jobs on the last $n - r$ machines. We can, however, construct a m-machine sequencing situation such that there exists an optimal schedule of its grand coalition, induced by $\hat{b}(N)$, in which each machine serves the same number of jobs. This m-machine sequencing situation is obtained by adding dummy jobs with processing time zero and cost coefficient one to the original m-machine sequencing situation. To see this, let $l = \max_{k \in M} n_k(b^0)$ be the length of the longest queue waiting for a machine w.r.t. b^0 in (M, N, b^0, p, α). Then for each machine k we put $l - n_k(b^0)$ jobs in front of the existing queue, so that a total of l jobs is waiting for service by machine k. Now we have a new m-machine sequencing situation $(M, \overline{N}, \overline{b^0}, \overline{p}, \overline{\alpha})$ with $\overline{N}$ the set of jobs, that is, N together with $ml - n$ dummy jobs, $\overline{b^0}$ the new initial serving order, $\overline{p}$ the new vector of cost coefficients. Note that for $i \in N$ it holds that

$$\begin{aligned} \overline{b^0}(i) &= b^0(i) + l - n_k(b^0) \\ \overline{p}_i &= p_i \\ \overline{\alpha}_i &= \alpha_i \quad (= 1) \end{aligned}$$

and for $i \in \overline{N} \backslash N$ it holds that

$$\begin{aligned} \overline{p}_i &= 0 \\ \overline{\alpha}_i &= 1 \end{aligned}$$

and

$$\{\overline{b^0}(i) \mid i \in \overline{N} \backslash N\} = \{(k,1), (k,2), ..., (k, l - n_k(b^0)) \mid k \in M\}.$$

The next lemma gives a relation between the m-sequencing games of the above described m-machine sequencing situations. The proof is omitted since it follows straightforwardly from the described procedure to find an optimal order and the fact that all new (dummy) jobs in the constructed m-machine sequencing situation have processing time zero.

Lemma 3.1 *Let (N, v) be the m-sequencing game corresponding to (M, N, b^0, p, α) in which $\alpha_i = 1$ for all $i \in N$. Let $(\overline{N}, \overline{v})$ be the m-sequencing game corresponding to $(M, \overline{N}, \overline{b^0}, \overline{p}, \overline{\alpha})$. Then*

$$v(S) = \overline{v}(S) = \overline{v}(S \cup T) \text{ for all } S \subseteq N, T \subseteq \overline{N} \backslash N.$$

From Lemma 3.1 immediately follows

Corollary 3.1 *Let (N, v) be the m-sequencing game corresponding to (M, N, b^0, p, α) in which $\alpha_i = 1$ for all $i \in N$. Let $(\overline{N}, \overline{v})$ be the m-sequencing game corresponding to $(M, \overline{N}, \overline{b^0}, \overline{p}, \overline{\alpha})$. Then $C(N, v) \neq \emptyset$ if and only if $C(\overline{N}, \overline{v}) \neq \emptyset$.*

So, for the proof of the balancedness of m-sequencing games we may restrict attention to m-machine sequencing situation (M, N, b^0, p, α) where exactly l jobs are scheduled on each machine in the initial order b^0. In the sequal of this section we therefore only consider m-machine sequencing situations where initially each machine has to process an equal number of jobs and in which all cost coefficient are equal to one.

To introduce a square matrix that defines the permutation game that arises from an m-machine sequencing situation (M, N, b^0, p, α), we need to take into account the following observations.

Since exactly l jobs are scheduled on each machine in the initial order b^0 as well as in an optimal order $b^{S(K)}$ for the jobs $S(K) = \bigcup_{k \in K} N_k(b^0)$, we can reduce the set $B_{S(K)}$ of admissible orders to

$$\overline{B}_{S(K)} = \{b \in B_{S(K)} : n_k(b) = l \text{ for all } k \in M\}.$$

Thus $\overline{B}_{S(K)}$ is the set of all admissible orders that schedule exactly l jobs on each machine. Then given an order $b \in \overline{B}_{S(K)}$ the total (waiting) costs for jobs $S(K)$ equals

$$c_b(S(K)) = \sum_{k \in K} \sum_{i=1}^{l} \sum_{j=1}^{i} p_{b^{-1}(k,j)} = \sum_{k \in K} \sum_{j=1}^{l} (l+1-j) p_{b^{-1}(k,j)}. \tag{6}$$

This implies that player $i = b^{-1}(k,j)$, which is in position j on machine k, contributes $(l+1-j)p_{b^{-1}(k,j)}$ to the total costs of coalition $S(K)$. Note that this amount is independent of the other jobs that are scheduled on this machine.

Now, we will define a permutation game (N, r) that arises from a m-machine sequencing situation (M, N, b^0, p, α). Let us start with introducing this permutation game for a specific case: 2 machines and 3 players on each machine. So, $M = \{1, 2\}$ and $N = \{1, 2, 3, 4, 5, 6\}$ and $p = (p_1, ..., p_6)$. If player $i \in N$ is scheduled in the first (second, third) position of machine $k \in M$, this player contributes $3p_i$ ($2p_i$, p_i) to the total costs of the players on machine k. We can describe these costs by the following 6×6 matrix A.

$$A = \begin{bmatrix} 3p_1 & 2p_1 & p_1 & 3p_1 & 2p_1 & p_1 \\ 3p_2 & 2p_2 & p_2 & 3p_2 & 2p_2 & p_2 \\ 3p_3 & 2p_3 & p_3 & 3p_3 & 2p_3 & p_3 \\ 3p_4 & 2p_4 & p_4 & 3p_4 & 2p_4 & p_4 \\ 3p_5 & 2p_5 & p_5 & 3p_5 & 2p_5 & p_5 \\ 3p_6 & 2p_6 & p_6 & 3p_6 & 2p_6 & p_6 \end{bmatrix}$$

Here, the rows correspond to the players $i \in N$ and the columns correspond to the positions in the processing order. For example, the entry a_{24} denotes the costs $3p_2$ of player 2 if it is processed on the first position of machine 2. A permutation $\pi : N \to \{1, \ldots, 6\}$ with $\pi(i) = j$ and $(k-1)l + 1 \leq j \leq kl$ we give the interpretation that player i is scheduled in position $j - l(k-1)$ on machine k.

Next, let us define the permutation game for the general case with m machines and l jobs on each machine. For all $i \in N$ and for all j with $(k-1)l + 1 \leq j \leq kl$, $k \in \{1, ..., m\}$ we define the square $ml \times ml$ matrix A by

$$a_{ij} = [lk - j + 1]p_i. \tag{7}$$

The permutation game (N, r) that arises from an m-sequencing situation (M, N, b^0, p, α) is defined by

$$r(S) = \max_{\pi \in \Pi_S} \sum_{i \in S} [a_{ii} - a_{i\pi(i)}]$$

for all $S \subseteq N$, where a_{ij} is given by (10).

Now we will show the relation between the m-sequencing game (M, w) and the permutation game (N, r).

Lemma 3.2 *Let (M, N, b^0, p, α) be a m-sequencing situation in which $\alpha_i = 1$ for all $i \in N$. Let (M, w) be the corresponding m-machine game and let (N, r) be the corresponding permutation game. Then*

$$w(K) = r(\bigcup_{k \in K} N_k(b^0)) for all K \subseteq M.$$

Proof. Consider $K \subseteq M$, then for each schedule $b \in \overline{B}_{S(K)}$ there exists a permutation $\pi^b \in \Pi_{S(K)}$ that puts each job on the same machine and in the same position as b does. This permutation is defined as

$$\pi^b(i) = (b_1(i) - 1)l + b_2(i) \tag{8}$$

for all $i \in N$. Furthermore, each permutation $\pi \in \Pi_{S(K)}$ can be written as an admissible order $b^\pi \in \overline{B}_{S(K)}$. For all $i \in N$ we define

$$\begin{aligned} b_1^\pi(i) &= \pi(i) - (k-1)l \\ b_2^\pi(i) &= k \end{aligned}$$

where k is such that $(k-1)l + 1 \leq \pi(i) \leq kl$.
Hence, for each $K \subseteq M$ we have that

$$\begin{aligned} w(K) &= \max_{b \in \overline{B}_{S(K)}} \left[\sum_{k \in K} \sum_{j=1}^{l} (l+1-j) p_{b^{0^{-1}}(k,j)} - \sum_{k \in K} \sum_{j=1}^{l} (l+1-j) p_{b^{-1}(k,j)} \right] \\ &= \max_{b \in \overline{B}_{S(K)}} \left[\sum_{k \in K} \sum_{j=1}^{l} a_{b^{0^{-1}}(k,j),(k-1)l+j} - \sum_{k \in K} \sum_{j=1}^{l} a_{b^{-1}(k,j),(k-1)l+j} \right] \\ &= \max_{b \in \overline{B}_{S(K)}} \left[\sum_{i \in S(K)} a_{i,\pi^{b^0}(i)} - \sum_{i \in S(K)} a_{i,\pi^b(i)} \right] \\ &= \max_{\pi \in \Pi_{S(K)}} \sum_{i \in S(K)} \left[a_{ii} - a_{i\pi(i)} \right] \\ &= r(S(K)) \\ &= r(\bigcup_{k \in K} N_k(b^0)), \end{aligned}$$

where the first equality holds by (6), the second equality by (7) and the third equality by (8). The fourth equality holds since we may assume, without loss of generality, that $\pi^{b^0}(i)$ is the identical permutation. The fifth equality holds by the definition of a permutation game, and the last equality by the definition of $S(K)$. □

In the next lemma we show that m-machine games are balanced.

Lemma 3.3 *Let (M, N, b^0, p, α) be a m-sequencing situation in which $\alpha_i = 1$ for all $i \in N$ and let (M, w) be the corresponding m-machine game. Then $C(M, w) \neq \emptyset$.*

Proof. Let (N, r) be the permutation game that arises from the m-machine sequencing situation (M, N, p, α). Since (N, r) is balanced, there exists an $x \in C(N, r)$. Define $y \in \mathbb{R}^M$ by

$$y_k = x(N_k(b^0))$$

for all $k \in M$. Then for $K \subset M$ we have

$$\begin{aligned} \sum_{k \in K} y_k &= \sum_{k \in S} x(N_k(b_0)) \\ &\geq r(\bigcup_{k \in S} N_k(b^0)) \\ &= w(K), \end{aligned} \tag{9}$$

where the first equality follows from the definiton of y, the inequality follows from $x \in C(N, r)$ and the second equality follows from Lemma 3.3. If $K = M$ the inequality becomes an equality, which implies that $y \in C(M, w)$. $\square$

The proof of Theorem 3.3 is now a consequence of Lemma 3.3 and Theorem 3.2.

4 Remarks

In Theorem 3.3 we assumed that all cost coefficients are equal to one. This implies that the class of m-sequencing games generated by the unweighted completion time criterion is a subclass of the class of balanced games. Clearly, the balancedness result also holds true in the case that all cost coefficients are equal to some positive constant $c > 0$. Furthermore, a slight adaptation of the proof of Theorem 3.3 gives a similar result for m-sequencing situations with identical processing times instead of identical cost coefficients.

Theorem 4.1 *Let (N, v) be the m-sequencing game that arises from an m-machine sequencing situation (M, N, b^0, p, α) in which $p_i = 1$ for all $i \in N$. Then (N, v) is balanced.*

Proof. Note that an optimal schedule $\hat{b}(N)$ of coalition N is established by first ordering the jobs of the players in N in a non-increasing order, i.e., $\alpha_{i_1} \geq \alpha_{i_2} \geq ... \geq \alpha_{i_n}$ where $\{i_1, i_2, ..., i_n\} = N$. Second, assign the jobs, after numbering the machines, in rotation to the machines:

$$\begin{array}{lllll|llll|l|lllll} \text{Job of player} & i_1 & i_2 & ... & i_m & i_{m+1} & i_{m+2} & ... & i_{2m} & ... & i_{n-r} & ... & i_n & & \\ \text{Machine} & 1 & 2 & ... & m & 1 & 2 & ... & m & ... & 1 & ... & r & ... & m \end{array}$$

Then the proof is similar to the proof of Theorem 3.3. The only difference is the matrix that defines the matrix of the permutation game. Here, we define for all $i \in N$ and for all j with $(k-1)l + 1 \leq j \leq kl$, $k \in \{1, ..., m\}$ the square $ml \times ml$ matrix A by

$$a_{ij} = [j - (k-1)l]\alpha_i. \tag{10}$$

$\square$

The following example shows that if condition (2) is not satisfied, then the corresponding m-sequencing game need not be balanced.

Example 4.1 Let $M = \{1, 2, 3\}$, $N = \{1, \ldots, 5\}$, $p = (2, 2, 1, 2, 2)$, and $\alpha = (1, 1, 1, 1, 1)$. The initial schedule b^0 is given in Figure 3. Let (N, v) be the corresponding 3-sequencing game. Suppose $x \in C(N, v)$ is a core element. Then $1 = v(N) = x(N) \geq \sum_{i \in N} v(i) = 0 + 1 + 0 + 0 + 1 = 2$. This contradiction shows that the core is empty. Hence the game (N, v) is not balanced. $\diamond$

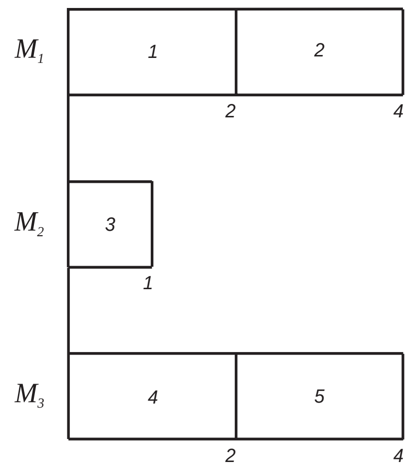

Figure 3: The schedule b^0

Finally, for m-machine sequencing situations ($m \geq 3$) with the weighted completion time criterion, the balancedness of the corresponding m-sequencing games is an open problem. If we follow the approach in this paper we need an optimal order for a coalition $S(K)$. The problem of finding such an optimal order, however, is difficult in the sense that it is NP-hard.

References

[1] BORM P. AND HAMERS H. (1998): "On the Convexity of Sequencing Games with Due Dates," *Working Paper, Tilburg University, The Netherlands*.

[2] CONWAY R., MAXWELL W., AND MILLER L. (1967): "Theory of Scheduling," *Addison-Wesley Publishing Company, London.*

[3] CURIEL I., PEDERZOLI G., AND TIJS S. (1989): "Sequencing Games," *European Journal of Operational Research*, 40, 344-351.

[4] CURIEL I., POTTERS J., RAJENDRA PRASAD V., TIJS S., AND VELTMAN B. (1994, A): "Cooperation in One Machine Scheduling," *Zeitschrift für Operations Research*, 38, 113-129.

[5] CURIEL I., POTTERS J., RAJENDRA PRASAD V., TIJS S., AND VELTMAN B. (1994, B): "Sequencing and Cooperation," *Operations research*, 42, 566-568.

[6] HAMERS H., BORM P., AND TIJS S. (1995): "On Games corresponding to Sequencing Situations with Ready Times," *Mathematical Programmming*, 70, 1-13.

[7] NOUWELAND A. VAN DEN, KRABBENBORG M., AND POTTERS J. (1992): "Flowshops with a Dominant Machine," *European Journal of Operational Research*, 62, 38-46.

[8] SHAPLEY L. (1971) , "Cores of Convex Games," *International Journal of Game Theory*, 1, 11-26.

[9] TIJS S., PARTHASARATHY T., POTTERS J., AND RAJENDRA PRASSAD V. (1984): "Permutation Games: another Class of Totally Balanced Games," *OR Spektrum*, 6, 119-123.