On the Convexity of Games corresponding to Sequencing Situations with Due Dates

Peter Borm[*], Gloria Fiestras-Janeiro[†], Herbert Hamers[*], Estela Sánchez[†], and Mark Voorneveld[*].

Abstract

This paper considers sequencing situations with due date criteria. Three different types of criteria are considered: the weighted penalty criterion, the weighted tardiness criterion and the completion time criterion. The main focus is on convexity of the associated cooperative games.

Keywords: Sequencing Situations; Due Date Criteria; Cooperative Games; Convexity.

1 Introduction

In one-machine sequencing situations a number of jobs has to be processed on a single machine. We assume that associated to each job there is an agent (player) who has a specific cost function which among other things depends on the completion time of his job. Further, there is assumed to be an initial order on the jobs of the agents before the processing of the machine starts. The objective is to find a processing order of the jobs that minimizes the aggregate cost function of all players.

Once this order has been obtained, a new question arises: how to allocate the corresponding cost savings with respect to the initial order among the agents? *Curiel, Pederzoli and Tijs (1989)* analyzed this problem by considering corresponding cooperative sequencing games for the special class of sequencing situations in which all players use a weighted completion time criterion. It was shown that all sequencing games of this type are convex games, so that allocation rules which always result in outcomes that are stable against coalitional deviations (core elements) can be devised. The EGS-rule, which is based on an allocation procedure that follows the algorithm to go from the initial order to an optimal one, was proposed as a particular choice of such a rule.

In *Curiel, Potters, Rajendra Prasad, Tijs, and Veltman (1993)* a more general class of sequencing situations is considered. For each agent, the cost

[*]Department of Econometrics and CentER, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands. E-mail: pemborm@kub.nl

[†]Department of Statistics and Operations Research, University of Vigo, Spain.

function is just assumed to be weakly monotonic in the completion time of his job. The corresponding class of sequencing games were called σ_0-component additive games, where σ_0 represents the initial order of the jobs. These games are, in general, not convex but core elements do exist. The $\beta-$rule was proposed as an extension of the $EGS-$rule, always yielding stable outcomes within the core of the corresponding game.

Hamers (1995) and *Hamers, Borm, and Tijs (1995)* considered sequencing situations where all agents use the weighted completion time criterion but where also to each job a ready time is associated: the earliest time the processing of this job can begin. It was shown that if in the initial order the jobs are arranged such that the ready times are non-decreasing, the corresponding games are σ_0-component additive games. Moreover, if all the jobs have equal processing times, then these games are convex.

In this paper we deal with sequencing situations where a due date is associated to each job: a time moment before which the processing of the job should be finished. Moreover, for each agent the cost criterion not only depends on the completion time but also on the due date of the corresponding job. We will assume that all players will use the same type of criterion. Three types of criteria are considered: the weighted penalty criterion, the weighted tardiness criterion, and the weighted completion time criterion. In each of these cases, the associated sequencing game is σ_0-component additive.

Our aim is to analyze the convexity property for each of the three corresponding classes of cooperative games. The convexity condition expresses that the incentives of an arbitrary agent for joining a certain coalition increase as the coalition grows. In the context of cooperative games, the property of convexity has drawn the interest of several researchers. The class of convex TU games has several nice properties. *Shapley (1971)* and *Ichiischi (1981)* showed that the extreme points of the core are the marginal vectors of the game if and only if the game is convex. Hence, convex games have a non-empty core. Moreover, with respect to one-point game theoretical solution concepts, it holds that the Shapley value (*Shapley (1953)*), which is by definition the average of the marginal vectors, is the barycenter of the core. Besides, the convexity property has been also extended and applied to the class of NTU games (*Vilkov (1977)* and *Sharkey (1982)*) and to the class of stochastic cooperative games (*Suijs and Borm (1999)*).

It turns out that convexity is not satisfied in general for the classes of games we deal with. It depends on the different parameters of the model: the processing times, the due dates, and the exact penalties for being late. We will show which classes of parameters do and do not necessary lead to convexity.

The organization of the paper is as follows. In section 2, we describe the underlying sequencing model and provide a characterization of the property of convexity for the class of σ_0-component additive games. In section 3 and 4, we analyze this property for the class of sequencing games that arise from sequencing situations where the aggregate cost function is based on weighted penalty criteria and weighted tardiness criteria, respectively. In each of these sections, we first describe a procedure that leads to an optimal order. Several examples illustrate that not all games associated to these sequencing situations are convex. Nevertheless, convexity holds by fixing some parameters in the model. In section 5, we show that sequencing situations in which all jobs have equal processing times, the due date of each job is a multiple of its processing time and the cost of each job is given by the weighted completion time function, yield the same class of convex games as the sequencing situations in which all jobs have equal processing times, the ready time of each job is a multiple of its processing time and the cost of each job is determined by a weighted completion time criterion, i.e., the class considered by *Hamers, Borm, and Tijs (1995)*.

2 Sequencing Situations and Games with Due Dates

A sequencing situation with due dates or briefly a $d-$sequencing situation is given by a 5-tuple (N, σ_0, p, d, c) where N is the set of jobs to be processed on a machine, σ_0 is the initial order[1] of the jobs, $p = (p_i)_{i \in N}$ is a vector specifying the processing times, $d = (d_i)_{i \in N}$ is a vector specifying the due dates, such that $d_{\sigma_0^{-1}(1)} \leq \ldots \leq d_{\sigma_0^{-1}(n)}$, and $c = (c_i)_{i \in N}$ specifies the cost function $c_i : [0, \infty) \to I\!R$ where $c_i(t)$ is to be interpreted as the cost incurred by agent i if his job is completed at time t. In this paper we consider three types of cost functions:

(C1) "weighted penalty" $\mathrm{c}_i^1(t) = \begin{cases} 0 & if\, t \leq d_i \\ \alpha_i & if \quad t > d_i \end{cases}$ where $\alpha_i > 0$. So if job $i \in N$ is completed after its due date, it incurs a fixed cost α_i.

(C2) "weighted tardiness[2]" $\mathrm{c}_i^2(t) = \alpha_i \max\{t - d_i, 0\} = \alpha_i(t - d_i)_+$ with $\alpha_i > 0$. Hence, job i incurs no costs if it is completed in time and a cost proportional to its tardiness if it is completed after its due date.

(C3) "weighted completion time" $\mathrm{c}_i^3(t) = \alpha_i t$ where $\alpha_i > 0$. The cost for job i is proportional to its completion time. There is, however, one obvious restriction here. It is assumed that we only consider orders in which all jobs are on time.

If the jobs are arranged by an order $\sigma \in \Pi(N)$, let $t_{\sigma,i}$ be the starting time of job i, i.e.,

$\mathrm{t}_{\sigma,i} = \sum_{k \in N : \sigma(k) < \sigma(i)} p_k$ and let $C(\sigma, S)$ be the time moment that all jobs in S are completed if the order is given by σ, i.e., $\mathrm{C}(\sigma, S) = \sum_{k \in N : \sigma(k) \leq \sigma(m)} p_k$ where $m \in S$ is the last player in S according to the order given by σ, i.e.,

$\sigma(m) \geq \sigma(k)\ forall\ k \in S$. With minor abuse of notation we will write $C(\sigma, i)$ instead of $C(\sigma, \{i\})$. Let $c_\sigma(S)$ be the aggregate cost of S in the order

[1]Formally, σ_0 is a bijection from N to $\{1, \ldots, n\}$ where $\sigma_0(i) = s$ means that job i is in position s in the queue before the machine. We will denote the class of all these bijections by $\Pi(N)$.

[2]Given $t \in \boldsymbol{R}$, we will denote $[t]_+ = \max\{t, 0\}$.

given by σ, $c_{\sigma}(S)=\sum\limits_{i\in S} c_i(C(\sigma,i))$.

The (maximal) cost savings of a coalition S depend on the set of admissible rearrangements of this coalition. We call a bijection $\sigma : N \to \{1,...,n\}$ *admissible* for S if it satisfies $P(\sigma,i)=P(\sigma_0,i)$ for all $i\in N\backslash S$, where $P(\sigma,i)=\{j\in N \mid \sigma(j)<\sigma(i)\}$. Hence, we consider an order to be admissible for S if each agent outside S has the same starting time as in the initial order. Moreover, the agents of S are not allowed to jump over players outside S. The set of all admissible rearrangements for a coalition S is denoted by Σ_S .

Given a $d-$sequencing situation (N,σ_0,d,p,α) the corresponding sequencing game is defined in such a way that the worth of a coalition S is equal to the maximal cost savings the coalition can achieve by means of admissible rearrangements. Formally, we have $v(S)=\max_{\sigma\in\Sigma_S}\{c_{\sigma_0}(S)-c_{\sigma}(S)\}=c_{\sigma_0}(S)-c_{\hat{\sigma}}(S)$ where $\hat{\sigma}\in\Sigma_S$ is an optimal order for the coalition S.

From the definition of admissible rearrangements it follows that the essential coalitions for sequencing games are the connected ones. A coalition S is called *connected* with respect to σ_0 if for all $i,j\in S$ and $k\in N$ such that $\sigma_0(i)<\sigma_0(k)<\sigma_0(j)$ it holds that $k\in S$. For convenience, we use the following obvious notations for the different types of connected coalitions:

$(m,j]_{\sigma_0}:=\{k \mid \sigma_0(m)<\sigma_0(k)\leq\sigma_0(j)\}$

$[m,j)_{\sigma_0}:=\{k \mid \sigma_0(m)\leq\sigma_0(k)<\sigma_0(j)\}$

$(m,j)_{\sigma_0}:=\{k \mid \sigma_0(m)<\sigma_0(k)<\sigma_0(j)\}$

$[m,j]_{\sigma_0}:=\{k \mid \sigma_0(m)\leq\sigma_0(k)\leq\sigma_0(j)\}$ if m and j are jobs such that $\sigma_0(m)<\sigma_0(j)$.

A game (N,v) is called *convex* if $v(T\cup\{i\})-v(T)\geq v(S\cup\{i\})-v(S)$ for all $i\in N$ and all $S\subset T\subset N\setminus\{i\}$.

Curiel, Potters, Rajendra Prasad, Tijs and Veltman (1993) introduced the class of σ_0-*component additive games.* Given $\sigma_0\in\Pi(N)$, a cooperative game (N,v) is called a σ_0- *component additive game* if the following three conditions are satisfied:

- $v(\{i\})=0$ for each $i\in N$,
- v is superadditive: for each $S,T\in 2^N$if $S\cap T=\emptyset$, then $v(S\cup T)\geq v(S)+v(T)$, and
- $v(S)=\sum_{T\in S/\sigma_0} v(T)$, where S/σ_0 is the set of all maximally connected components of S.

Notice that from the conditions on admissible rearrangements it follows that $d-$sequencing games are σ_0-*component additive games.*

Given a set of players N and a coalition $T\subset N$, the $T-$unanimity game u_T is defined by $u_T(S)=1$ if $T\subset S$ and $u_T(S)=0$ for all other coalitions. Every game (N,v) can be expressed as a linear combination of the T-unanimity games as follows $v=\sum_{T\subset N}\Delta_v(T)\,u_T$, where $\Delta_v(T)=\sum\limits_{S\subset T}(-1)^{|T|-|S|}v(S)$.

These coefficients are called the dividends and its computation can be a hard task. Given a σ_0-component additive game $\Delta_v(T) = 0$ for every non-connected coalition T as a direct consequence of theorem 2 in *Owen (1986)*. Then, in this class of games, only the dividends associated to the connected coalitions appear. In the next result we obtain a simple expression for the value of these coefficients for an arbitrary σ_0-component additive game.

Proposition 1 *Let (N,v) be a σ_0-component additive game. Then, the characteristic function v can be written as $v=\sum_{[k,l]_{\sigma_0}\subset N:\sigma_0(k)<\sigma_0(l)} g_{[k,l]_{\sigma_0}} u_{[k,l]_{\sigma_0}}$, where $g_{[k,l]_{\sigma_0}}=v([k,l]_{\sigma_0}) - v([k,l)_{\sigma_0}) - v((k,l]_{\sigma_0}) + v((k,l)_{\sigma_0})$.*

PROOF.

W.l.o.g. we assume $N = \{1,\ldots,n\}$ and $\sigma_0 \in \Pi(N)$ such that $\sigma_0(i) = i$ for all $i \in N$. We also omit subscripts. Define $w = \sum_{[k,l]\subset N:k<l} g_{[k,l]} u_{[k,l]}$. Let $T = [i,j] \subset N$ be a connected coalition with $i < j$. Then

$$\begin{aligned}
w(T) &= \sum_{[k,l]\subset[i,j]:k<l} g_{[k,l]} \\
&= \sum_{k=i}^{j-1}\sum_{l=k+1}^{j} g_{[k,l]} \\
&= \sum_{k=i}^{j-1}\sum_{l=k+1}^{j} [v([k,l]) - v([k,l)) - v((k,l]) + v((k,l))] \\
&= \sum_{k=i}^{j-1}\sum_{l=k+1}^{j} [v([k,l]) - v([k,l))] - \sum_{k=i}^{j-1}\sum_{l=k+1}^{j} [v((k,l]) - v((k,l))] \\
&= \sum_{k=i}^{j-1} [v([k,j]) - v(\{k\})] - \sum_{k=i}^{j-1} v((k,j]) \\
&= v([i,j]) \\
&= v(T).
\end{aligned}$$

Now, let $T \subset N$ be a coalition. Then using the σ_0-component additivity and the proof above we find v(T)$=\sum_{S\in T\backslash\sigma_0} v(S) = \sum_{S\in T\backslash\sigma_0} w(S)$.

Moreover,

$$\begin{aligned}
\sum_{S\in T\backslash\sigma_0} w(S) &= \sum_{S\in T\backslash\sigma_0}\left(\sum_{[k,l]\subset N:k<l} g_{[k,l]} u_{[k,l]}(S)\right) \\
&= \sum_{[k,l]\subset N:k<l} g_{[k,l]}\left(\sum_{S\in T\backslash\sigma_0} u_{[k,l]}(S)\right) \\
&= \sum_{[k,l]\subset N:k<l} g_{[k,l]} u_{[k,l]}(T) \\
&= w(T).
\end{aligned}$$

Hence, we find $w(T) = v(T)$. ■

Remark. Notice we can write the coefficients in two ways:

$$\begin{aligned} g_{[k,l]} &= [v([k,l]) - v([k,l))] - [v((k,l]) - v((k,l))] \\ &= [v([k,l]) - v((k,l])] - [v([k,l)) - v((k,l))] . \end{aligned}$$

The first expression can be interpreted as follows: the first part, $v([k,l]) - v([k,l))$, measures the contribution of player l (the last player of the coalition $[k,l]$) if he joins to the end of the ordered coalition $[k,l)$, and the second part, $v((k,l]) - v((k,l))$ measures the contribution of player l if he joins to the end of the ordered coalition (k,l). So, the difference specifies the role of player k to the marginal contribution of player l. The second expression can be interpreted in a similar way.

The next theorem is a direct consequence of the results above.

Theorem 2 *Let (N,v) be a σ_0-component additive game. Then (N,v) is convex if and only if the coefficients $g_{[k,l]_{\sigma_0}}$ are non negative for all $k,l \in N$ such that $\sigma_0(k) < \sigma_0(l)$.*

The next sections are devoted to the study of the convexity of the sequencing games arising from $d-$sequencing situations. As a consequence of theorem 2, in order to check whether a σ_0-component additive game is convex, it suffices to check the non-negativity of all the coefficients $g_{[k,l]_{\sigma_0}}$. This fact implies a significant reduction in the number of conditions that need to be checked for the convexity of these games. We have to verify $\frac{1}{2}(n-1)(n-2)$ conditions[3]. This clearly improves the $\sum_{m=2}^{n} \binom{n}{m}\binom{m}{2}$ conditions that *Zumsteg (1995)* indicates for the general case.

3 Convexity of Sequencing Games arising from d-Sequencing Situations with cost criterion C1

In this section we study the convexity of the sequencing games that arise from $d-$sequencing situations (N, σ_0, p, d, c^1) when all players use a cost criterion fitting C1. These situations we call $C1$-sequencing situations. The next example shows that, in general, the associated game need not be convex.

Example 1. Let us consider the following $C1-$sequencing situation:

1. $N = \{1,2,3,4\}$,
2. $\sigma_0(i) = i$ for all $i \in N$,
3. $p = (300, 201, 201, 100)$,

[3] When $\sigma_0(l) = \sigma_0(k) + 1$, $g_{[k,l]_{\sigma_0}} = v\left([k,l]_{\sigma_0}\right)$ and then, clearly $g_{[k,l]_{\sigma_0}} \geq 0$.

4. $d_i = 500$ for all $i \in N$,

5. $\alpha_i = 1$ for all $i \in N$.

Easy calculations shows that $g_{[1,4]} = v[1,4] - v(1,4] - v[1,4) + v(1,4) = 1 - 1 - 1 + 0 = -1 < 0$. Hence, the corresponding game is not convex.

itbpFU3.1047in2.3341in0inOptimal orders w.r.t. $C1$posterin.wmf

The $C1-$sequencing situation of example 1 illustrates the fact that the associated cooperative games to $C1-$sequencing situations with equal unitary penalties or equal due dates, in general, need not be convex. If we consider $C1-$sequencing situations where all jobs having equal processing times, the associated sequencing game will be convex. In order to prove this result we only need to check the non-negativity of the coefficients $g_{[k,l]_{\sigma_0}}$ for every connected coalition $[k,l]_{\sigma_0}$. We will use the Lawler's algorithm (*Lawler (1976)*) to find an optimal order of every connected coalition w.r.t. the initial order in the associated sequencing game, and then we easily can compute the marginal contributions of each player who joins to the end of the connected coalition.

Let us consider $\sigma_0 \in \Pi(N)$ and let $m, j \in N$ such that $\sigma_0(m) < \sigma_0(j)$. We will denote $V_{[m,j]_{\sigma_0}}$ as a set of players which is in time in an optimal order of $[m,j]_{\sigma_0}$ where every job $i \in N \setminus [m,j]_{\sigma_0}$ is located in the position $\sigma_0(i)$. For each $V_{[m,j]_{\sigma_0}}$, $G_{[m,j]_{\sigma_0}}$is the set of jobs of $[m,j]_{\sigma_0}$that can not be completed in time in the optimal order associated to $V_{[m,j]_{\sigma_0}}$, *i.e.*, $G_{[m,j]_{\sigma_0}} = [m,j]_{\sigma_0} \setminus V_{[m,j]_{\sigma_0}}$. Moreover, $a_{[m,j]_{\sigma_0}}$will be the difference between the corresponding cost of an optimal order of $[m,j]_{\sigma_0}$ and the associated cost of an optimal order of $[m,j)_{\sigma_0}$.

For simplifying the notation, if σ_0 is the identity permutation, we will denote $V_{[1,j]_{\sigma_0}} = V_j$, $G_{[1,j]_{\sigma_0}} = G_j$, and $a_{[1,j]_{\sigma_0}} = a_j$. V_j will be called a $j - optimal$ set[4]. In example 1, $V_4 = \{4,2\}$, $G_4 = \{3,1\}$ and $a_4 = 0$.

Lawler (1976) gave an $O(n\,logn)$ algorithm to find an $n - optimal\ set$ to minimize the weighted number of tardy jobs under one additional assumption (w.l.o.g. we will assume that σ_0 is the identity permutation): Given i,j$\in N \quad such that p_i < p_j, then \alpha_i \geq \alpha_j$. This means that if a job has a shorter processing time than another, its penalty is at least equal or larger. This condition is trivially satisfied if all the processing times are equal.

Take $p_i = q$ for all $i \in N$. *Lawler (1976)* set $\widehat{V}_0 = \emptyset$ and, recursively,
$$\widehat{V}_j = \begin{cases} \widehat{V}_{j-1} \cup \{j\} & if \;\; C(\widehat{\sigma}^{j-1}, \widehat{V}_{j-1}) + q \leq d_j \\ \left(\widehat{V}_{j-1} \cup \{j\}\right) \setminus \{l\} & otherwise \end{cases}$$
where $\widehat{\sigma}^{j-1}$ represents an optimal order associated to the $(j-1) - optimal$ set $\widehat{V}_{j-1}$, and l is the minimum element w.r.t. the following relation between elements of $\widehat{V}_{j-1} \cup \{j\}$:

$$i \prec k if and only if \begin{cases} \alpha_i < \alpha_k, or \\ \alpha_i = \alpha_k \; and \widehat{\sigma}^{j-1}(i) < \widehat{\sigma}^{j-1}(k) \end{cases} \tag{1}$$

[4]An $n - optimal$ set is formed by the jobs that are in time in an optimal order of N.

Next, we will carefully compute the gains $\widehat{a}_j$ and locate the jobs in a specific position step by step.

Let us consider the initial situation given by $\widehat{V}_0 = \emptyset$, $\widehat{\sigma}^0 = \sigma_0$.

First step:

Consider the first job. If it can be processed in time, go to the second step. If not, label it as garbage and go to the second step. That means

- If $p_1 \leq d_1$, $\widehat{V}_1 = \{1\}$, and $\widehat{\sigma}^1 = \sigma_0$.
- If $p_1 > d_1$, $\widehat{V}_1 = \emptyset$, $\widehat{G}_1 = \{1\}$, and $\widehat{\sigma}^1 = \sigma_0$.

In both cases $\widehat{a}_1 = 0$.

j-th step:

- If no garbage jobs exist ($\widehat{G}_{j-1} = \emptyset$), put job j behind job $j-1$.
  - If j is on time, proceed to the next step. That means $\widehat{V}_j = \widehat{V}_{j-1} \cup \{j\}$, $\widehat{G}_j = \emptyset$, and $\widehat{a}_j = 0$.
  - If job j is not on time, take the job l determined by (1) and put it right behind j.

    Then, $\widehat{V}_j = (\widehat{V}_{j-1} \cup \{j\}) \setminus \{l\}$. One of the following two cases must happen:
    - $l = j$, then $\widehat{V}_j = \widehat{V}_{j-1}$, $\widehat{G}_j = \{j\}$ and $\widehat{a}_j = 0$.

    . itbpFU2.4249in1.8256in0ingarb1.wmf
    - If $l \neq j$, $\widehat{V}_j = (\widehat{V}_{j-1} \cup \{j\}) \setminus \{l\}$, $\widehat{G}_j = \{l\}$ and in this case there is a positive gain $\widehat{a}_j = \alpha_j - \alpha_l$. itbpFU2.4059in1.8109in0ingarb2.wmf
- If garbage jobs exist ($\widehat{G}_{j-1} \neq \emptyset$), put job j right in front of garbage jobs.
  - If now job j is processed in time, there are two possibilities:
    - Job j was already in time before it was moved. Then, it is certainly in time now. Hence, the movement causes no gain, $\widehat{a}_j = 0$, since all garbage jobs ahead of j were late and now still are. All other jobs are still on time.
    - Job j was not in time before it was moved, moving it ahead the garbage yields a positive gain $\widehat{a}_j = \alpha_j$.

    In both cases, $\widehat{V}_j = \widehat{V}_{j-1} \cup \{j\}$ and $\widehat{G}_j = \widehat{G}_{j-1}$. itbpFU2.4059in1.8109in0ingarb3.wmf
  - If now job j is not processed in time, then it was certainly not in time behind the garbage jobs, hence moving it ahead them does not yield a gain. Now choose the job l to be removed from $\widehat{V}_{j-1} \cup \{j\}$ to the garbage can as in (1). Put this new garbage job right behind the old garbage that remains in place. Hence, there are again two possibilities:

* $l = j$. Job j was the job that was added to the garbage. Then $\widehat{a}_j = 0$, $\widehat{V}_j = \widehat{V}_{j-1}$, and $\widehat{G}_j = \widehat{G}_{j-1} \cup \{j\}$. itbpFU2.4059in1.8109in0ingarb4.wmf

* $l \neq j$. Job j was not the job that was added to the garbage. We see that the job j must now be in time and a gain was made, $\widehat{a}_j = \alpha_j - \alpha_l$. In this case $\widehat{V}_j = (\widehat{V}_{j-1} \cup \{j\}) \setminus \{l\}$ and $\widehat{G}_j = \widehat{G}_{j-1} \cup \{l\}$.

  itbpFU2.4059in1.8109in0ingarb5.wmf

If there are still jobs behind the garbage, go to the next step. Otherwise stop.

Note that the final $n-$optimal set $\widehat{V}_n$ has been achieved by non-negative switches only, $\widehat{a}_j \geq 0$ for all $j \in \{1, 2, ..., n\}$.

We have just described how to obtain an $n-optimal$ set. Of course with obvious modifications this procedure can be applied to obtain an optimal order of the coalition $[m, j]$ with $m < l$.

The following lemma shows that given a $C1-$sequencing situation where all processing times equal q, it can be established a relation between the set $\widehat{V}_{[k,l]}$, and the set $\widehat{V}_{(k,l]}$.

Lemma 3 *Let (N, σ_0, p, d, c^1) be a $C1-$sequencing situation where $p_i = q$ for all $i \in N$. It is verified that for all $k, l \in N$ such that $\sigma_0(k) < \sigma_0(l)$,*

$$\widehat{V}_{(k,l]_{\sigma_0}} \subset \widehat{V}_{[k,l]_{\sigma_0}}, \tag{2}$$

$$0 \leq \left|\widehat{G}_{[k,l]_{\sigma_0}}\right| - \left|\widehat{G}_{(k,l]_{\sigma_0}}\right| \leq 1, and \tag{3}$$

$$\widehat{a}_{[k,l]_{\sigma_0}} - \widehat{a}_{(k,l]_{\sigma_0}} \geq 0. \tag{4}$$

Proof.
See the appendix.

Example 2. This example shows that to have (2), (3), and (4) we really have to restrict to equal processing times. In example 1, it is easy to check that

$$\begin{aligned} \left|\widehat{G}_{[1,3]}\right| - \left|\widehat{G}_{(1,3]}\right| &= 1 - 2 = -1, and \\ \widehat{a}_{[1,4]} - \widehat{a}_{(1,4]} &= 0 - 1 < 0. \end{aligned}$$

Moreover in the $C1-$sequencing situation with

1. $N = \{1, 2, 3, 4\}$,
2. $\sigma_0(i) = i$ for all $i \in N$,
3. $p = (2, 2, 3, 3)$,
4. $d = (2, 2, 5, 7)$, and
5. $\alpha = (5, 5, 5, 3)$

one readily sees that $\widehat{V}_{[2,4]} = \{3\}$ and $\widehat{V}_{(2,4]} = \{4\}$.

Theorem 4 *Let (N, σ_0, p, d, c^1) be a $C1-$sequencing situation where $p_i = q$ for all $i \in N$. Then, the corresponding sequencing game (N, v) is a convex game.*

PROOF.
Taking into account the result of theorem 2, it suffices to prove that $g_{[k,l]_{\sigma_0}} \geq 0$ for all $k, l \in N$ such that $\sigma_0(k) < \sigma_0(l)$. And, this is derived from lemma 3, since $g_{[k,l]_{\sigma_0}} = \widehat{a}_{[k,l]_{\sigma_0}} - \widehat{a}_{(k,l]_{\sigma_0}}$. ■

In the next table we summarize the convexity results for sequencing games arising from $C1$-sequencing situations.

$\alpha_i = b$, $d_i = e$, and $p_i = q$ for all $i \in N$	convex (Theorem 4)
$\alpha_i = b$ and $p_i = q$ for all $i \in N$	convex (Theorem 4)
$\alpha_i = b$ and $d_i = e$ for all $i \in N$	not convex (Example 1)
$d_i = e$ and $p_i = q$ for all $i \in N$	convex (Theorem 4)
$p_i = q$ for all $i \in N$	convex (Theorem 4)
$d_i = e$ for all $i \in N$	not convex (Example 1)
$\alpha_i = b$ for all $i \in N$	not convex (Example 1)

4 Convexity of Sequencing Games arising from d-Sequencing Situations with cost criterion C2

In this section we study d-sequencing situations (N, σ_0, p, d, c^2) where all players use a cost criterion fitting $C2$. This means that the associated cost to each job is proportional to its tardiness. We refer to these sequencing situations as $C2-$sequencing situations. The associated games need not be convex in general as the following examples illustrate.

Example 3. Let us consider the following $C2-$sequencing situation:

1. $N = \{1, 2, 3\}$,
2. $\sigma_0(i) = i$ for all $i \in N$,
3. $p = (2, 3, 1)$,
4. $d_i = 3$, for all $i \in N$, and
5. $\alpha = (4, 5, 8)$.

Easy calculations gives us the value of $g_{[1,3]}$: $g_{[1,3]} = v[1,3] - v(1,3] - v[1,3) + v(1,3) = -2 < 0$. Thus, the associated game to this $C2$-sequencing situation is not convex.

Example 4. Let us consider the following $C2-$sequencing situation:

1. $N = \{1, 2, 3, 4, 5\}$,
2. $\sigma_0(i) = i$, for all $i \in N$,
3. $p = (19, 17, 16, 9, 9)$,
4. $d = (20, 22, 28, 35, 40)$, and
5. $\alpha_i = 1$, for all $i \in N$.

In this case, the value of $g_{[1,5]}$ is given by

$$\begin{aligned} g_{[1,5]} &= v[1,5] - v(1,5] - v[1,5) + v(1,5) \\ &= 21 - 21 - 11 + 9 \\ &= -2 < 0 \end{aligned}$$

From examples 3 and 4, one can derive that in order to guarantee the convexity of the associated sequencing game to a $C2-$sequencing situation, it is not enough to consider jobs with either the same due dates or the same penalties for their tardiness. Nevertheless, convexity appears when all jobs have both equal penalties for their tardiness and equal processing times, or both equal due dates and equal processing times, or both equal due dates and equal penalties for their tardiness.

In the first case, the associated sequencing game is a zero game, since all jobs are arranged in a non-decreasing way of their due dates in both the initial and an optimal order *(Smith (1956))*. In order to study the two remaining cases, we proceed in the following way: first we state the gains attainable for player i and j in case player i is directly in front of player j ; then, we establish a result that gives us a way to find an optimal order; and, finally, we prove the convexity results.

Lemma 5 *Let (N, σ, p, d, c^2) be a C2-sequencing situation where $d_l = e$ for all $l \in N$. Let us take $i, j \in N$, such that $\sigma(j) = \sigma(i) + 1$.*

a) If $\alpha_l = b$ for all $l \in N$ the gains of switching i and j are given by

$$g_{[i,j]_\sigma} = \left[\min\left\{[b\,(C(\sigma,i) - e)]_+\,, b\,(p_i - p_j)\right\}\right]_+ . \tag{5}$$

b) If $p_l = q$ for all $l \in N$ the gains of switching i and j are given by

$$g_{[i,j]_\sigma} = \left[\min\left\{[(\alpha_j - \alpha_i)\,((|P(\sigma,i)| + 2)q - e)]_+\,, q(\alpha_j - \alpha_i)\right\}\right]_+ . \tag{6}$$

Proof.
See the appendix.

Next we describe an optimal order for a C2-sequencing situation when all players have both equal due dates and equal processing times, or both equal due dates and equal penalties for their tardiness. This result is directly derived from lemma 5.

Lemma 6 *Let (N, σ_0, p, d, c^2) a C2-sequencing situation where $d_i = e$ for all $i \in N$.*

a) If $\alpha_i = b$ for all $i \in N$. Then, $\hat{\sigma}$ is an optimal order if

$$p_{\hat{\sigma}^{-1}(1)} \leq \ldots \leq p_{\hat{\sigma}^{-1}(k)} \leq p_{\hat{\sigma}^{-1}(k+1)} \leq \ldots \leq p_{\hat{\sigma}^{-1}(n)}. \tag{7}$$

b) If $p_i = q$ for all $i \in N$. Then, $\hat{\sigma}$ is an optimal order if

$$\alpha_{\hat{\sigma}^{-1}(1)} \geq \ldots \geq \alpha_{\hat{\sigma}^{-1}(k)} \geq \alpha_{\hat{\sigma}^{-1}(k+1)} \geq \ldots \geq \alpha_{\hat{\sigma}^{-1}(n)}. \tag{8}$$

Remark. From the previous lemma a processing order in which all the jobs are processed in a non-decreasing way w.r.t. the urgency index, defined by $u_i = \frac{\alpha_i}{p_i}$ for all $i \in N$, maximizes the total gain. Nevertheless, there are several optimal orders. In case a), two optimal orders differ in the position where the jobs in time or the jobs with equal processing times are placed on; in case b), clearly, these differences are in the position of the jobs in time or the jobs with equal penalties for their tardiness. Moreover, for any proper connected coalition S, we can obtain an optimal order just applying to the jobs in S the constraints (7) (case a)) or (8) (case b)), respectively. On the other hand, given an optimal order, $\hat{\sigma}_{(i,j]}$, for a coalition $(i,j]_{\sigma_0}$, it is clear that we can find an optimal order for coalition $[i,j]_{\sigma_0}$ through switches of player i and any player $k \in (i,j]_{\sigma_0}$ such that $\sigma_0(i) < \hat{\sigma}_{(i,j]}(k) \leq \sigma_0(j)$ and $p_i > p_k$ (case a)) or $\alpha_i < \alpha_k$ (case b)). The aggregate gains, which are obtained from these switches, equal the difference between $v\left([i,j]_{\sigma_0}\right)$ and $v\left((i,j]_{\sigma_0}\right)$.

Theorem 7 *Let (N, σ_0, p, d, c^2) be a C2-sequencing situation such that $d_i = e$ for all $i \in N$. The associated sequencing game is convex when $\alpha_i = b$ for all $i \in N$, or $p_i = q$ for all $i \in N$.*

Proof.
We only proof the theorem for the case $\alpha_i = b$ for all $i \in N$.

Let (N, v) be the associated game to (N, σ_0, p, d, c^2). Taking into account theorem 2, we just have to check that $v([i,j]_{\sigma_0}) - v((i,j]_{\sigma_0}) \geq v([i,j)_{\sigma_0}) - v((i,j)_{\sigma_0})$ for all connected coalition $[i,j]_{\sigma_0} \subset N$ with $\sigma_0(i) < \sigma_0(j)$.

W.l.o.g. we will suppose that σ_0 is the identity permutation, $[i,j] = [1,n] = N$, and $\hat{\sigma} \in \Pi(N)$ is an optimal order of N such that $\hat{\sigma}(i_k) = k$ (player i_k is located in position k in an optimal order $\hat{\sigma}$ of N). So, due to the remark above, we can suppose that $p_{i_1} \leq p_{i_2} \leq \ldots \leq p_{i_n}$.

Let $\hat{\sigma}(1) = s$ and $\hat{\sigma}(n) = t$, where $s, t \in \{1, 2, ..., n\}$ and $s \neq t$. Two cases may be considered:

1. $t > s$. That means that player n is coming behind player 1 in the optimal order $\hat{\sigma}$. Then, as a consequence of lemma 6, player 1 switches positions with other players in $(1, n)$ until he attains position s. As a direct consequence of lemma 5, v[1,n]-v(1,n]=v[1,n)-v(1,n).

2. $t < s$. In this case, player n is coming ahead player 1 in the optimal order $\hat{\sigma}$. Then, applying lemma 6, player 1 reaches position s in $\hat{\sigma}$ by switches with players $i_1, \ldots, i_{s-1} \in (1, n]$. Using lemma 5, the marginal contribution of player 1 to $(1, n]$ is given by

v[1,n]-v(1,n]=

$$\sum_{k=2}^{s}\left[\min\left\{\left[b\left(C(\sigma_0,1)+\sum_{l=1}^{k-2}p_{i_l}-e\right)\right]_+, b\left(p_1-p_{i_{k-1}}\right)\right\}\right]_+ . \tag{9}$$

Meanwhile, player 1 reached position $s-1$ in an optimal order of $[1, n)$ by switches with players $i_1, \ldots, i_{t-1}, i_{t+1}, \ldots, i_{s-1} \in (1, n)$. Then, v[1,n)-v(1,n)=

$$\sum_{k=2}^{t}\left[\min\left\{\left[b\left(C(\sigma_0,1)+\sum_{l=1}^{k-2}p_{i_l}-e\right)\right]_+, b\left(p_1-p_{i_{k-1}}\right)\right\}\right]_+ + \tag{10}$$

$$\sum_{k=t+2}^{s}\left[\min\left\{\left[b\left(C(\sigma_0,1)+\sum_{l=1}^{k-2}p_{i_l}-p_{i_t}-e\right)\right]_+, b\left(p_1-p_{i_{k-1}}\right)\right\}\right]_+ \tag{11}$$

Easily, expressions (9) and (10) can be compared: all the terms in both expressions are non-negative; each one of the $t-1$ initial terms coincides in both expressions; and for each $k = t+2, ..., s$ $\sum_{l=1}^{k-2} p_{i_l} \geq \sum_{l=1}^{k-2} p_{i_l} - p_{i_t}$. Therefore, $v[1, n) - v(1, n) \leq v[1, n] - v(1, n]$.

Hence, we conclude that the associated game is convex. ■

In this section we discussed the convexity property of the games arising from $C2-$sequencing situations according to the different parameters of the model. Nevertheless, one case is still unsolved. When all jobs have equal processing times, *Slikker(1993)* proved that if the job number is less or equal than 4, the game is convex. But, in case of a larger number of jobs, the convexity problem is still open.

In the next table we summarize the convexity results for sequencing games arising from $C2$-sequencing situations.

$d_i = e$ for all $i \in N$	not convex (Example 3)
$\alpha_i = b$ for all $i \in N$	not convex (Example 4)
$\alpha_i = b$ and $d_i = e$ for all $i \in N$	convex (Theorem 7)
$p_i = q$ and $d_i = e$ for all $i \in N$	convex (Theorem 7)
$\alpha_i = b$ and $p_i = q$ for all $i \in N$	convex (zero game)
$p_i = q$ for all $i \in N$	if $\|N\| \leq 4$, convex if $\|N\| > 4$, open problem

5 Convexity of Sequencing Games arising from d-Sequencing Situations with cost criterium C3

In this section we concentrate on $d-$sequencing situations that satisfy (A1) $d_i \in \{1, ..., n\}$ and $p_i = 1$ for all $i \in N$ Further, it is assumed that there is an initial bijection $\sigma_0 : N \to \{1, ..., n\}$ on the jobs of the players before the processing of the machine starts with the properties (A2) $d_i \leq d_j$ for all $i, j \in N$ with $\sigma_0(i) < \sigma_0(j)$, and $C(\sigma_0, i) \leq d_i$ for all $i \in N$

and

(A3) $\sigma_0(i) = C(\sigma_0, i)$ for all $i \in N$. Note that the assumptions $(A1) - (A2)$ imply that in the initial bijection there is no time gap in the job processing and that in particular the last job that is processed according to σ_0 is completed at time n. In spite of the conclusion that assumption $(A3)$ is superfluous, we have added it here for the sake of convenience and symmetry with ready time sequencing situations discussed later on. Moreover, the cost function of each job is proportional to its completion time. (A4) $c_i^3(t) = \alpha_i t$, for all $i \in N$. These $d-$sequencing situations will be called $C3-$sequencing situations.

Since each job has to be completed before its due date, we will consider for each coalition S only those orders $\sigma \in \sum_S$ such that satisfy $C(\sigma, i) \leq d_i$. Note that by the assumptions on the initial and admissible bijections we have for any $\sigma \in \Sigma_S$ that $\sigma(i) = C(\sigma, i)$ for all $i \in N$.

Next, we describe the special class of one-machine sequencing situations, in which all jobs have equal processing times and the ready time of each job is a multiple of the processing time and the corresponding class of games. The description of these sequencing games is identical to the $d-$sequencing situations. The only difference is that there is no due date imposed on a agent but a ready time. The ready time r_i of the job of agent i is the earliest time that the job can be processed on the machine. We will concentrate on sequencing situations that satisfy (B1) $r_i \in \{0, ..., n-1\}$ and $p_i = 1$ for all $i \in N$. The initial order σ_0 has the properties (B2) $r_i \leq r_j$ for all $i, j \in N$ with $\sigma_0(i) < \sigma_0(j)$ and $C(\sigma_0, i) \geq r_i + 1$ for all $i \in N$ and (B3) $\sigma_0(i) = C(\sigma_0, i)$ for all $i \in N$. Note that the assumptions $(B1) - (B3)$ imply that in the initial bijection σ_0 there are no time gaps in the job processing and that the job that is processed last is completed at time n. The cost for agent i is given by $(A4)$. A sequencing situation as described above is denoted by $(N, \sigma_0, r, p, \alpha)$ and will be refered to as an r-sequencing situation.

In r-sequencing situations we will only consider those bijections $\sigma : N \to \{1, ..., n\}$ that satisfy $C(\sigma, i) \geq r_i + 1$ for all $i \in N$. The set of admissible rearrangements, denoted by $\mathcal{A}_S$, has the same restrictions with respect to interchanging positions between players of a coalition S as before. Hence, we may again conclude that for any $\sigma \in \mathcal{A}_S$ we have that $\sigma(i) = C(\sigma, i)$. The corresponding sequencing game is defined by v(S)=$\max_{\sigma \in \mathcal{A}_S} \{\sum_{i \in S} \alpha_i C(\sigma_0, i) -$

$\sum_{i\in S}\alpha_i C(\sigma,i)\}$

Hamers, Borm and Tijs (1995) show that sequencing games arising from r-sequencing situations are convex by establishing relations between optimal orders of subcoalitions. These relations are obtained by analyzing the procedure described in *Rinnooy Kan (1976)* that provides an optimal order. For the optimal order in d-sequencing situations we can use the procedure of *Smith (1956)*, which operates similar to the procedure of *Rinnooy Kan (1976)*. Both procedures aim for having the jobs with the largest cost coefficient α_i as far as possible at the front of the queue. The Smith-procedure has to take into account the due dates, whereas the Rinnooy Kan-procedure has to take into account the ready times. For this reason the Smith-procedure starts at the end of the queue, whereas the Rinnooy Kan-procedure starts at the front of the queue. In spite of this difference it is possible for d-sequencing situations to establish similar relations between optimal orders of various subcoalitions as for r-sequencing situations. However, where in the Rinnooy Kan-procedure these relations are established if a player is added at the end of a (sub)queue, in the Smith-procedure these relations can be established if a player is added at the front of a (sub)queue. Following exactly the same line of argument it can be infered that sequencing games arising from d-sequencing situations are convex games.

In fact, we will show even a stronger result: both classes of sequencing situations generate the same class of sequencing games.

Theorem 8 *Let $R(N)$ and $D(N)$ be the class of sequencing games that arise from r-sequencing situations and $C3$-sequencing situations, respectively. Then $R(N) = D(N)$.*

Proof.

We show that $R(N)\subseteq D(N)$. Let $(N,v)\in R(N)$. Let (N,σ_0,r,p,α) be an r-sequencing situation that generates the game (N,v). W.l.o.g. we can take $\sigma_0(i)=i$ for all $i\in N$. Let $S=\{i,i+1,...,j\}$, be a connected set w.r.t. σ_0. Then

$$v(S)=\max\{\sum_{k=i}^{j}\alpha_k k-\sum_{k=i}^{j}\alpha_k x_k \mid x_k\geq r_k+1\ \forall k\in S,\{x_i,...,x_j\}=\{i,...,j\}\}. \tag{12}$$

Consider the d-sequencing situation (N,τ_0,d,p,β) in which for all $i\in N$ we define $\tau_0(i)=n+1-i$, $d_i=n-r_i$ and $\beta_i=c+(\alpha_n-\alpha_i)$ with $c=\max_{i\in N}\alpha_i$. We first show that (N,τ_0,d,p,β) satisfies the assumptions $(A1)-(A3)$. Obviously, $(A3)$ is a consequence of $(B1)$, while $(A1)$ follows immediately from the definition of d and $(B1)$. If $\tau_0(l)<\tau_0(m)$ then $m<l$ which implies that $r_m\leq r_l$. The definition of d yields immediately that $d_l\leq d_m$. Further, we have for any $l\in N$ that $\sigma_0(l)=l\geq r_l+1=n+1-d_l$. This implies that $d_l\geq n+1-l=\tau_0(l)=C(\tau_0,l)$. Hence $(A2)$ is satisfied.

Note that from the definition of τ_0 it follows that S is also connected w.r.t. τ_0.

Then for the game (N, w) corresponding to (N, τ_0, d, p, β) it holds that

$$
\begin{aligned}
w(S) &= \max\{\sum_{k=i}^{j} \beta_k(n+1-k) - \sum_{k=i}^{j} \beta_k y_k \mid y_k \leq d_k \ \forall k \in S, && (13)\\
&\qquad \{y_i, ..., y_j\} = \{n+1-j, ..., n+1-i\}\} && (14)
\end{aligned}
$$

Let $\hat{y}$ be an optimal solution of (12). By defining $\hat{x}$ by $\hat{x}_k = n+1-\hat{y}_k$ for all $k \in \{i, ...j\}$ we have

$$
\begin{aligned}
w(S) &= \sum_{k=i}^{j} \beta_k(n+1-k) - \sum_{k=i}^{j} \beta_k \hat{y}_k \\
&= \sum_{k=i}^{j} (c + \alpha_n - \alpha_k)(n+1-k) - \sum_{k=i}^{j} (c + \alpha_n - \alpha_k)(n+1-\hat{x}_k) \\
&= (c + \alpha_n) \sum_{k=i}^{j} (\hat{x}_k - k) + \sum_{k=i}^{j} \alpha_k (k - \hat{x}_k) \\
&= \sum_{k=i}^{j} \alpha_k (k - \hat{x}_k) \\
&\leq v(S),
\end{aligned}
$$

where the first equality holds since $\hat{y}$ is optimal, the second equality by the definition of τ_0, β and $\hat{x}$, the third equality and fourth equality by straightforward calculations. The inequality holds by (11) since $\hat{x}_k = n+1-\hat{y}_k \geq n+1-d_k = n+1-(n-r_k) = r_k+1$ and $\{\hat{x}_i, ..., \hat{x}_j\} = \{i, ..., j\}$.
Let $\hat{x}$ be an optimal solution of (11). By defining $\hat{y}$ by $\hat{y}_k = n+1-\hat{x}_k$ for all $k \in S$ we can show in the same way as above that $v(S) \leq w(S)$, which completes the first part of this proof.
Obviously, the second part, $D(N) \subseteq R(N)$, can be dealt with in an analogous way. ∎

Acknowledgments

G. Fiestras-Janeiro and E. Sánchez would like to thank the Spanish Ministry of Education and Culture for financial support through the grants PB94 0648-C02-02 and PB97 0550-C02-02. M. Voorneveld's research is financially supported by Dutch Foundation for Mathematical Research (SWON) through project 613-04-051.

Appendix

In this section we present the proofs that were omitted in sections 3 and 4.

PROOF OF LEMMA 3.

We will prove the result by induction in the size of $[k,l]_{\sigma_0}$ with $\sigma_0(k) < \sigma_0(l)$. W.l.o.g. we will assume $q = 1$ and $\sigma_0(i) = i$ for all $i \in N$. Notice that with this assumption the position of each job in any order is its completion time.

Let us suppose that $l = k+1$. We distinguish two cases:

- $k > d_k$. Clearly,
  - if $\widehat{V}_{(k,k+1]} = \emptyset$, then (2) holds. Moreover $\left|\widehat{G}_{(k,k+1]}\right| = 1$ and $\hat{a}_{(k,k+1]} = 0$. Since $k > d_k$, $\left|\widehat{G}_{[k,k+1]}\right| \geq 1$ and $\hat{a}_{[k,k+1]} \geq 0$. So (3) and (4) hold.
  - if $\widehat{V}_{(k,k+1]} = \{k+1\}$, then $\widehat{G}_{(k,k+1]} = \emptyset$ and $\hat{a}_{(k,k+1]} = 0$. Since $k > d_k$, $\widehat{V}_{[k,k+1]} = \{k+1\}$, $\widehat{G}_{[k,k+1]} = \{k\}$, and $\hat{a}_{[k,k+1]} = 0$. Then (2), (3), and (4) follow.
- $k \leq d_k$. Clearly,
  - if $\widehat{V}_{(k,k+1]} = \emptyset$, then (2) holds. Moreover $\left|\widehat{G}_{(k,k+1]}\right| = 1$, $\hat{a}_{(k,k+1]} = 0$, and $\left|\widehat{V}_{[k,k+1]}\right| = 1$ since $k+1 > d_{k+1} \geq d_k \geq k$. So $\left|\widehat{G}_{[k,k+1]}\right| = 1$ and (3) follows. Furthermore, $\hat{a}_{[k,k+1]} = 0$ if $\alpha_{k+1} \leq \alpha_k$ or $\hat{a}_{[k,k+1]} = \alpha_{k+1} - \alpha_k$ if $\alpha_{k+1} > \alpha_k$. In both cases $\hat{a}_{(k,k+1]} \leq \hat{a}_{[k,k+1]}$ and (4) holds.
  - if $\widehat{V}_{(k,k+1]} = \{k+1\}$, then $\widehat{V}_{[k,k+1]} = \{k, k+1\}$, $\widehat{G}_{(k,k+1]} = \emptyset$, $\widehat{G}_{[k,k+1]} = \emptyset$, $\hat{a}_{(k,k+1]} = 0 = \hat{a}_{[k,k+1]}$. Thus (2), (3), and (4) follow.

Let $[k,l]$ be such that $l \geq k+2$. We may assume that $\widehat{V}_{(k,r]} \subset \widehat{V}_{[k,r]}$, $0 \leq \left|\widehat{G}_{[k,r]}\right| - \left|\widehat{G}_{(k,r]}\right| \leq 1$, and $\widehat{a}_{[k,r]} - \widehat{a}_{(k,r]} \geq 0$ for all r such that $k \leq r < l$. So, taking[5] $r = l-1$

$$\widehat{V}_{(k,l)} \subset \widehat{V}_{[k,l)}, \tag{15}$$

$$0 \leq \left|\widehat{G}_{[k,l)}\right| - \left|\widehat{G}_{(k,l)}\right| \leq 1, and \tag{16}$$

$$\widehat{a}_{[k,l)} - \widehat{a}_{(k,l)} \geq 0. \tag{17}$$

Three cases can happen in the algorithm in the step in which job l is added to $[k,l)$ for getting a $[k,l] - optimal\ set$.

- If $\widehat{V}_{[k,l]} = \widehat{V}_{[k,l)}$, then $\widehat{G}_{[k,l]} = \widehat{G}_{[k,l)} \cup \{l\}$. Taking into account (13), (14), and the Lawler´s algorithm, player l will not be in time in a $(k,l]-optimal\ set$. Then, $\widehat{V}_{(k,l]} = \widehat{V}_{(k,l)}, \widehat{G}_{(k,l]} = \widehat{G}_{(k,l)} \cup \{l\}, and \widehat{a}_{[k,l]} = \widehat{a}_{(k,l]} = 0$.

  Then, (2), (3), and (4) hold.

[5]Let us notice that $\widehat{V}_{(k,l)} = \widehat{V}_{(k,l-1]}$.

- If $\widehat{V}_{[k,l]} = \widehat{V}_{[k,l)} \cup \{l\}$, then $\widehat{G}_{[k,l)} = \widehat{G}_{[k,l]}$ and $\widehat{a}_{[k,l]} = 0$ if job l is in time initially or $\widehat{a}_{[k,l]} = \alpha_l$ if job l is not in time initially. We distinguish two cases:
  - If $\left|\widehat{G}_{[k,l)}\right| = \left|\widehat{G}_{(k,l)}\right|$, then $\widehat{V}_{(k,l]} = \widehat{V}_{(k,l)} \cup \{l\}$, $\widehat{G}_{(k,l)} = \widehat{G}_{(k,l]}$, *and* $\widehat{a}_{(k,l]} = \widehat{a}_{[k,l]}$.
  - If $\left|\widehat{G}_{[k,l)}\right| = \left|\widehat{G}_{(k,l)}\right| + 1$, then either
    - $\widehat{V}_{(k,l]} = \widehat{V}_{(k,l)} \cup \{l\}$, $\widehat{G}_{(k,l)} = \widehat{G}_{(k,l]}$, and $\widehat{a}_{(k,l]} = \widehat{a}_{[k,l]}$.

    or
    - $\widehat{V}_{(k,l]} = \left(\widehat{V}_{(k,l)} \cup \{l\}\right) \setminus \{m\}$ and $\widehat{G}_{(k,l]} = \widehat{G}_{(k,l)} \cup \{m\}$. Since l is not in time in the initial order and taking into account the selection of the job m, then $\widehat{a}_{[k,l]} - \widehat{a}_{(k,l]} = \alpha_l - (\alpha_l - \alpha_m) = \alpha_m \geq 0$.

  So, it is easy to check that (2), (3), and (4) hold.

- If $\widehat{V}_{[k,l]} = (\widehat{V}_{[k,l)} \cup \{l\}) \setminus \{m\}$, where $m \in \widehat{V}_{[k,l)}$ with $\alpha_m = \min\left\{\alpha_i \mid i \in \widehat{V}_{[k,l)}\right\}$. Then $\widehat{G}_{[k,l]} = \widehat{G}_{[k,l)} \cup \{m\}$ and $\widehat{a}_{[k,l]} = \alpha_l - \alpha_m$. Clearly l is not in time in front of garbage jobs of $\widehat{G}_{[k,l)}$ nor of $\widehat{G}_{(k,l)}$. Now, two cases must be taken into account,
  - If $m \in \widehat{V}_{(k,l)}$, then $\widehat{V}_{(k,l]} = (\widehat{V}_{(k,l)} \cup \{l\}) \setminus \{m\}$, $\widehat{G}_{(k,l]} = \widehat{G}_{(k,l)} \cup \{m\}$, and $\widehat{a}_{(k,l]} = \alpha_l - \alpha_m = \widehat{a}_{[k,l]}$.
  - If $m \notin \widehat{V}_{(k,l)}$, then $\widehat{V}_{(k,l]} = (\widehat{V}_{(k,l)} \cup \{l\}) \setminus \{s\}$ where s$\in \widehat{V}_{(k,l)} \cup \{l\} \subset \widehat{V}_{[k,l)} \cup \{l\}$ such that $\alpha_m \leq \alpha_s = \min\left\{\alpha_i \mid i \in \widehat{V}_{(k,l)} \cup \{l\}\right\}$. Then $\widehat{G}_{(k,l]} = \widehat{G}_{(k,l)} \cup \{s\}$ and $\widehat{a}_{(k,l]} = \alpha_l - \alpha_s \leq \widehat{a}_{[k,l]}$.

  Then (2), (3), and (4) hold. ■

Proof of lemma 5.

a) Let (N, σ, p, d, c^2) be a C2-sequencing situation where $d_l = e$ for all $l \in N$. Let us take $i, j \in N$, such that $\sigma(j) = \sigma(i) + 1$. Let us consider the ordering τ defined by $\tau(l) = \sigma(l)$ for all $l \in N \setminus \{i, j\}$, $\tau(i) = \sigma(j)$ and $\tau(j) = \sigma(i)$. The definition of $g_{[i,j]_\sigma}$ and τ clearly implies that $g_{[i,j]_\sigma} = v\left([i,j]_\sigma\right) = \left[c^2_\sigma\left([i,j]_\sigma\right) - c^2_\tau\left([i,j]_\sigma\right)\right]_+$.

Then, in order to consider the cost saving between both orders, we must take into account the following cases:

A) $e \geq C(\sigma, i)$ and $e \geq C(\sigma, j)$. In this case players i and j are in time in the order σ, and then they are also in time when they switch their positions. So, $g_{[i,j]_\sigma} = 0$ and $\left[\min\left\{\left[b\left(C(\sigma,i) - e\right)\right]_+, b\left(p_i - p_j\right)\right\}\right]_+ = \left[\min\left\{0, b\left(p_i - p_j\right)\right\}\right]_+ = 0$.

B) $e \geq C(\sigma, i)$ and $e < C(\sigma, j)$. In this case player i is in time in the order σ, meanwhile player j is not in time. It is trivial to check that $g_{[i,j]_\sigma} = 0$ considering that $\alpha_i = \alpha_j = b$. Moreover, $\left[\min\left\{\left[b\left(C(\sigma, i) - e\right)\right]_+, b\left(p_i - p_j\right)\right\}\right]_+ = \left[\min\left\{0, b\left(p_i - p_j\right)\right\}\right]_+ = 0$.

C) $e < C(\sigma, i)$. In this case, player i is not in time in the order σ and, hence, player j is not in time too. We will distinguish two cases:

C1) $C(\sigma, i) - p_i + p_j \leq e$. If the players switch their positions, then player j will be in time and player i will still be late. Thus, the gains are $g_{[i,j]_\sigma} = b\left(\sum_{k \in P(\sigma,i)} p_k + p_i - e\right) = b\left(C(\sigma, i) - e\right)$.
Clearly,

$$
\begin{aligned}
\left[\min\left\{\left[b\left(C(\sigma, i) - e\right)\right]_+, b\left(p_i - p_j\right)\right\}\right]_+ &= \\
\min\left\{b\left(C(\sigma, i) - e\right), b\left(p_i - p_j\right)\right\} &= b\left(C(\sigma, i) - e\right).
\end{aligned}
$$

C2) $C(\sigma, i) - p_i + p_j > e$. If the players switch their positions, both will still not be in time, and its trivial to check that $g_{[i,j]_\sigma} = \begin{cases} b\left(p_i - p_j\right) & if \quad p_i > p_j \\ 0 & otherwise \end{cases}$.

In this case,

- if $p_i > p_j$,

$$
\begin{aligned}
\left[\min\left\{\left[b\left(C(\sigma, i) - e\right)\right]_+, b\left(p_i - p_j\right)\right\}\right]_+ &= \\
\min\left\{b\left(C(\sigma, i) - e\right), b\left(p_i - p_j\right)\right\} &= b\left(p_i - p_j\right).
\end{aligned}
$$

- if $p_i \leq p_j$, $\left[\min\left\{\left[b\left(C(\sigma, i) - e\right)\right]_+, b\left(p_i - p_j\right)\right\}\right]_+ = \left[b\left(p_i - p_j\right)\right]_+ = 0$.

b) Let (N, σ, p, d, c^2) be a C2-sequencing situation where $d_l = e$ for all $l \in N$. Let us take $i, j \in N$, such that $\sigma(j) = \sigma(i) + 1$. Let us consider the ordering τ defined by $\tau(l) = \sigma(l)$ for all $l \in N \setminus \{i, j\}$, $\tau(i) = \sigma(j)$ and $\tau(j) = \sigma(i)$. The definition of $g_{[i,j]_\sigma}$ and τ clearly implies that $g_{[i,j]_\sigma} = v\left([i,j]_\sigma\right) = \left[c_\sigma^2\left([i,j]_\sigma\right) - c_\tau^2\left([i,j]_\sigma\right)\right]_+$.

Then, in order to consider the saving costs between both orders, we must take into account the following cases:

A) $e \geq C(\sigma, i)$ and $e \geq C(\sigma, j)$. In this case players i and j are in time in the order σ, and then they are also in time when they switch their positions. So, $g_{[i,j]_\sigma} = 0$ and

$$
\begin{aligned}
\left[\min\left\{\left[(\alpha_j - \alpha_i)\left((|P(\sigma, i)| + 2)q - e\right)\right]_+, q(\alpha_j - \alpha_i)\right\}\right]_+ &= \\
\left[\min\left\{0, q(\alpha_j - \alpha_i)\right\}\right]_+ &= 0.
\end{aligned}
$$

B) $e \geq C(\sigma,i)$ and $e < C(\sigma,j)$. In this case player i is in time in the order σ, meanwhile player j is not in time. It is trivial to check that $c^2_\sigma([i,j]) - c^2_\tau([i,j]) = (\alpha_j - \alpha_i)((|P(\sigma,i)|+2)q - e)$. Thus, $g_{[i,j]_\sigma} = [(\alpha_j - \alpha_i)((|P(\sigma,i)|+2)q - e)]_+$. Moreover, since $C(\sigma,i) = (|P(\sigma,i)| + 1)q \leq e < (|P(\sigma,i)|+2)q = C(\sigma,j)$ it follows that $\left[\min\left\{[(\alpha_j - \alpha_i)((|P(\sigma,i)|+2)q - e)]_+, q(\alpha_j - \alpha_i)\right\}\right]$ $[(\alpha_j - \alpha_i)((|P(\sigma,i)|+2)q - e)]_+$.

C) $C(\sigma,i) > e$. In this situation, both players are not in time in the order σ, then both will still be late when they switch their positions. Then, $g_{[i,j]_\sigma} = [q(\alpha_j - \alpha_i)]_+$. And, since $(|P(\sigma,i)|+1)q > e$, we have

- if $\alpha_j \geq \alpha_i$, then $\left[\min\left\{[(\alpha_j - \alpha_i)((|P(\sigma,i)|+2)q - e)]_+, q(\alpha_j - \alpha_i)\right\}\right]_+ = q(\alpha_j - \alpha_i)$.
- if $\alpha_j < \alpha_i$, then $\left[\min\left\{[(\alpha_j - \alpha_i)((|P(\sigma,i)|+2)q - e)]_+, q(\alpha_j - \alpha_i)\right\}\right]_+ = 0$. ■

References

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

Curiel I., Potters J., Rajendra Prasad V., Tijs S. and Veltman B. (1993) Cooperation in one machine scheduling. *Zeitschrift für Operations Research*, 38, 113-129.

Hamers H. (1995) Sequencing and delivery situations: a game theoretic approach. PhD dissertation, University Tilburg, The Netherlands.

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

Ichiischi T. (1981) Super-modularity: applications to convex games and the greedy algorithm for LP. *Journal of Economic Theory*, 25, 283-286.

Lawler E. (1976) Sequencing to minimize the weighted number of tardy jobs. *Revue Francaise d'Automatique, Informatique, Recherche Operationelle, 10.5 Suppl., 27-33.*

Owen G. (1986). Values of graph-restricted games. *SIAM J. Alg. Disc. Meth.*, 7, 210-220.

Rinnooy Kan A. (1976), Machine scheduling problems. Martinus Nijhof, The Hague.

Shapley L. (1953) A value for n-person games. In: Kuhn H.W., Tucker A.W. (eds) Contributions to the Theory of Games II, Princeton University Press, Princeton NJ: 307-317.

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

SLIKKER M. (1993) On games arising from sequencing situations with due dates. Term Paper. Tilburg University (In Dutch).

SMITH W. (1956) Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3, 59-66.

SUIJS J. AND BORM P. (1999) Stochastic cooperative games: superadditivity, convexity and certainty equivalents. *To appear in Games and Economic Behavior.*

ZUMSTEG S.M. (1995) Non-cooperative aspects of cooperative Game Theory and related computational problems. PhD dissertation, Zürich.