European Journal of Operational Research 246 (2015) 894–906

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Step out–Step in sequencing games

M. Musegaas*, P. E. M. Borm, M. Quant

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

ARTICLE INFO

Article history:
Received 20 November 2014
Accepted 18 May 2015
Available online 27 May 2015

Keywords:
(Cooperative) game theory
Sequencing games
Core

ABSTRACT

In this paper a new class of relaxed sequencing games is introduced: the class of Step out–Step in sequencing games. In this relaxation any player within a coalition is allowed to step out from his position in the processing order and to step in at any position later in the processing order. First, we show that if the value of a coalition in a relaxed sequencing game is bounded from above by the gains made by all possible neighbor switches, then the game has a non-empty core. After that, we show that this is the case for Step out –Step in sequencing games. Moreover, this paper provides a polynomial time algorithm to determine the values of the coalitions in Step out–Step in sequencing games.

1. Introduction

In this paper one-machine sequencing situations are considered with a queue of players in front of a single machine, each with one job to be processed. Such a situation specifies for each player the processing time, time the machine takes to process the corresponding job of this player. In addition, it is assumed that each player has a linear cost function specified by an individual cost parameter. Moreover, there are no restrictive assumptions as due dates, ready times or precedence constraints imposed on the jobs. To minimize total joint costs, Smith (1956) showed that the players must be ordered with respect to weakly decreasing urgency, defined as the ratio of the individual cost parameter and the processing time. Assuming the presence of an initial order, this reordering will lead to cost savings. To analyze how to divide the maximal cost savings among the players, Curiel, Pederzoli, and Tijs (1989) introduced cooperative sequencing games. They show that sequencing games are convex and therefore have a non-empty core. This means that it is always possible to find a coalitionally stable cost savings division.

Several classes of sequencing games have been discussed before. These classes are all based on different features of the underlying sequencing situations, for example grouping of jobs, stochastic data, dynamic/multistage situations, multiple jobs and/or machines. Both Gerichhausen and Hamers (2009) and Grundel, Çiftçi, Borm, and Hamers (2013) applied grouping of jobs, but in a different way. Gerichhausen and Hamers (2009) considered partitioning sequencing games where jobs arrive in batches and those jobs that arrive in earlier batches have some privilege over jobs in later arrived batches. Grundel et al. (2013) introduced family sequencing situations and included the concept of set-up times in their model. Alparslan-Gök, Branzei, Fragnelli, and Tijs (2013) and Klijn and Sánchez (2006) considered stochastic data in sequencing games. Alparslan-Gök et al. (2013) considered sequencing games where uncertainty in the parameters (costs per time unit and/or processing time) is involved by means of interval data. Klijn and Sánchez (2006) considered the arrival pattern of the jobs to be stochastic, so they assumed that no initial order is specified. In multi-stage sequencing situations, the order arrived at after each stage becomes the starting order for the next stage. Multi-stage sequencing situatons are for example considered by Curiel (2015). In Lohmann, Borm, and Slikker (2014) another type of dynamic sequencing situations is considered, in which a player enters the system at the moment the player starts to prepare the machine for his job (there is a predecessor dependent set-up time) and leaves the system as soon as his job is finished. Çiftçi, Borm, Hamers, and Slikker (2013) considered machines which can simultaneously process multiple jobs. Multiple machine sequencing games are for example discussed in Estévez-Fernández, Mosquera, Borm, and Hamers (2008), Slikker (2006a) and Slikker (2005). Key questions in all of the above literature are finding optimal orders, allocation rules and properties of the corresponding cooperative games.

A common assumption underlying the definition of the values of the coalitions in many sequencing games is that two players of a certain coalition can only swap their positions if all players between them are also members of the coalition. Curiel, Potters, Prasad, Tijs, and Veltman (1993) argued that the resulting set of admissible reorderings for a coalition is too restrictive because there may be more reorderings possible which do not hurt the interests of the players outside the coalition. Relaxed sequencing games arise

* Corresponding author. Tel.: +31134663244.
E-mail address: m.musegaas@tilburguniversity.edu (M. Musegaas).

http://dx.doi.org/10.1016/j.ejor.2015.05.045

by relaxing the classical assumption about the set of admissible rearrangements for coalitions in a consistent way. In Curiel et al. (1993) four different relaxed sequencing games are introduced. These relaxations are based on requirements for the players outside the coalition regarding either their position in the processing order (position unchanged/may change) or their starting time (starting time unchanged/not increased). This means that a player in a certain coalition is allowed to jump over players outside the coalition as long as the exogenously imposed requirements are satisfied. As a consequence, a player may be moved to a position earlier in the processing order when another player moves backwards. Slikker (2006b) proved non-emptiness of the core for all four types of relaxed sequencing games considered in Curiel et al. (1993). In Van Velzen and Hamers (2003) two further classes of relaxed sequencing games are considered. In the first class there is a specific player allowed to switch with a player in front of him in the processing order if this player has a larger processing time, and with a player behind him in the processing order if this player has a smaller processing time. In the second class there are fixed time slots and thus only jobs with equal processing times can be switched. Van Velzen and Hamers (2003) proved that both classes of relaxed sequencing games have a non-empty core. In fact, a lot of attention has been paid to non-emptiness of the core of relaxed sequencing games. However, surprisingly enough, up to now for none of the relaxed sequencing games described above attention has been paid to finding polynomial time algorithms determining optimal processing orders for all possible coalitions.

In this paper another class of relaxed sequencing games is introduced: Step out–Step in (SoSi) sequencing games. This relaxation is intuitive from a practical point of view, because in this relaxation a member of a coalition is also allowed to step out from his position in the processing order and to step in at any position somewhere later in the processing order. In particular, each player outside the coalition will not obtain any new predecessors, possibly only fewer. For the time being we apply this relaxation on the classical sequencing situation as introduced by Curiel et al. (1989). However, this relaxation can also applied to other types of sequencing situations. We start with proving non-emptiness of the core for the class of relaxed sequencing games where the values of the coalitions are bounded from above by the value of this coalition in a classical sequencing game if this coalition would have been connected. After that, we show that the class of SoSi sequencing games belongs to this class, and thus every SoSi sequencing game has a non-empty core. Moreover, we provide a polynomial time algorithm determining an optimal processing order for a coalition and the corresponding value. The algorithm considers the players of the coalition in an order that is the reverse of the initial order, and for every player the algorithm checks whether moving the player to a position later in the processing order is beneficial. This algorithm works in a greedy way in the sense that every player is moved to the position giving the highest cost savings at that moment. Moreover, every player is considered in the algorithm exactly once and every player is moved to another position in the processing order at most once.

The organization of this paper is as follows. Section 2 recalls basic definitions on one-machine sequencing situations and formally introduces SoSi sequencing games. In Section 3 it is shown that every SoSi sequencing game has a non-empty core. Section 4 provides a polynomial time algorithm to determine the values of the coalitions in a SoSi sequencing game. Finally, in Section 5 we summarize our results and provide some directions for future research. In particular we consider another type of relaxed sequencing games, so-called Step out sequencing games.

2. SoSi sequencing games

A one-machine sequencing situation can be summarized by a tuple (N, σ_0, p, α), where $N = \{1, \ldots, n\}$ is the set of players, each with one job to be processed on the single machine. A processing order of the players can be described by a bijection $\sigma : N \to \{1, \ldots, n\}$. More specifically, $\sigma(i) = k$ means that player i is in position k. Let $\Pi(N)$ denote the set of all such processing orders. The processing order $\sigma_0 \in \Pi(N)$ specifies the initial order. The processing time $p_i > 0$ of the job of player i is the time the machine takes to process this job. The vector $p \in \mathbb{R}^N_{++}$ summarizes the processing times. Furthermore, the costs of player i of spending t time units in the system is assumed to be determined by a linear cost function $c_i : [0, \infty) \to \mathbb{R}$ given by $c_i(t) = \alpha_i t$ with $\alpha_i > 0$. The vector $\alpha \in \mathbb{R}^N_{++}$ summarizes the coefficients of the linear cost functions. It is assumed that the machine starts processing at time $t = 0$, and also that all jobs enter the system at $t = 0$.

Let $C_i(\sigma)$ be the completion time of the job of player i with respect to processing order σ via the associated semi-active schedule, i.e., a schedule in which there is no idle time between the jobs. Hence, the completion time of player i equals

$$C_i(\sigma) = \sum_{j \in N : \sigma(j) \leq \sigma(i)} p_j.$$

A processing order is called *optimal* if the total joint costs $\Sigma_{i \in N} \alpha_i C_i(\sigma)$ are minimized. In Smith (1956) it is shown that in each optimal order the players are processed in non-increasing order with respect to their urgency u_i defined by $u_i = \frac{\alpha_i}{p_i}$. Moreover, with g_{ij} representing the gain made by a possible neighbor switch of i and j if player i is directly in front of player j, i.e., with

$$g_{ij} = \max\{\alpha_j p_i - \alpha_i p_j, 0\},$$

the maximal total cost savings are equal to

$$\sum_{i \in N} \alpha_i C_i(\sigma_0) - \sum_{i \in N} \alpha_i C_i(\sigma^*) = \sum_{i,j \in N : \sigma_0(i) < \sigma_0(j)} g_{ij}, \tag{1}$$

where σ^* denotes an optimal order.

A *coalitional game* is a pair (N, v) where $N = \{1, \ldots, n\}$ denotes a non-empty, finite set of players and $v : 2^N \to \mathbb{R}$ assigns a monetary payoff to each coalition $S \in 2^N$, where 2^N denotes the collection of all subsets of N. The value $v(S)$ denotes the highest payoff the coalition S can jointly generate by means of optimal cooperation without help of players in $N \backslash S$. By convention, $v(\emptyset) = 0$.

To tackle the allocation problem of the maximal cost savings in a sequencing situation (N, σ_0, p, α) one can analyze an associated coalitional game (N, v). Here N naturally corresponds to the set of players in the game and, for a coalition $S \subset N$, $v(S)$ reflects the maximal cost savings this coalition can make with respect to the initial order σ_0. In order to determine these maximal cost savings, assumptions must be made on the possible reorderings of coalition S with respect to the initial order σ_0.

The classical (strong) assumption is that a member of a certain coalition $S \subset N$ can only swap with another member of the coalition if all players between these two players, according to the initial order, are also members of S. Given an initial order σ_0 the set of admissible orders for coalition S in a classical sequencing game is denoted by $\mathcal{A}^c(\sigma_0, S)$. This leads to the definition of a classical sequencing game. However, note that the resulting set of admissible reorderings for a coalition is quite restrictive, because there may be more reorderings possible which do not hurt the interests of the players outside the coalition.

In a SoSi sequencing game the classical assumption is relaxed by additionally allowing that a member of the coalition S steps out from his position in the processing order and steps in at any position later in the processing order. Hence, a processing order σ is called *admissible* for S in a SoSi sequencing game if

(i) $P(\sigma, i) \subset P(\sigma_0, i)$ for all $i \in N \backslash S$,
(ii) $\sigma^{-1}(\sigma(i) + 1) \in F(\sigma_0, i)$ for all $i \in N \backslash S$,

where $P(\sigma, i) = \{j \in N | \sigma(j) < \sigma(i)\}$ denotes the set of predecessors of player i with respect to processing order σ and $F(\sigma, i) = \{j \in$