

QUEUEING GAMES WITH AN
ENDOGENOUS NUMBER OF
MACHINES

Ata Atay

Christian Trudeau

UB Economics *Working Paper No. 429*

Title: Queueing games with an endogenous number of machines

Abstract: This paper studies queueing problems with an endogenous number of machines with and without an initial queue, the novelty being that coalitions not only choose how to queue, but also on how many machines. For a given problem, agents can (de)activate as many machines as they want, at a cost. After minimizing the total cost (processing costs and machine costs), we use a game theoretical approach to share the proceeds of this cooperation, and study the existence of stable allocations. First, we study queueing problems with an endogenous number of machines, and examine how to share the total cost. We provide an upper bound and a lower bound on the cost of a machine to guarantee the non-emptiness of the core (the set of stable allocations). Next, we study requeueing problems with an endogenous number of machines, where there is an existing queue. We examine how to share the cost savings compared to the initial situation, when optimally requeueing/changing the number of machines. Although, in general, stable allocation may not exist, we guarantee the existence of stable allocations when all machines are considered public goods, and we start with an initial schedule that might not have the optimal number of machines, but in which agents with large waiting costs are processed first.

JEL Codes: C44, C71, D61, D63

Keywords: Queueing problems, convexity, cost sharing, allocation problems

Authors:

Ata Atay

University of Barcelona and BEAT

Email: aatay@ub.edu

Christian Trudeau

University of Windsor

Email: trudeauc@uwindsor.ca

Date: July 2022

Acknowledgements: A. Atay is a Serra Hünter Fellow (Professor Lector Serra Hünter) under the Serra Hünter Plan (Pla Serra Hünter). A. Atay gratefully acknowledges financial support by the University of Barcelona through grant AS017672. C. Trudeau gratefully acknowledges financial support by the Social Sciences and Humanities Research Council of Canada [grant number 435-2019-0141].

QUEUEING GAMES WITH AN ENDOGENOUS NUMBER OF MACHINES

ATA ATAY AND CHRISTIAN TRUDEAU

ABSTRACT. This paper studies queueing problems with an endogenous number of machines with and without an initial queue, the novelty being that coalitions not only choose how to queue, but also on how many machines. For a given problem, agents can (de)activate as many machines as they want, at a cost. After minimizing the total cost (processing costs and machine costs), we use a game theoretical approach to share the proceeds of this cooperation, and study the existence of stable allocations. First, we study queueing problems with an endogenous number of machines, and examine how to share the total cost. We provide an upper bound and a lower bound on the cost of a machine to guarantee the non-emptiness of the core (the set of stable allocations). Next, we study requeueing problems with an endogenous number of machines, where there is an existing queue. We examine how to share the cost savings compared to the initial situation, when optimally requeueing/changing the number of machines. Although, in general, stable allocation may not exist, we guarantee the existence of stable allocations when all machines are considered public goods, and we start with an initial schedule that might not have the optimal number of machines, but in which agents with large waiting costs are processed first.

Keywords: queueing problems · convexity · cost sharing · allocation problems

JEL Classification: C44 · C71 · D61 · D63

Mathematics Subject Classification (2010): 60K25 · 90B22 · 91A12

Date: July 27, 2022.

A. Atay is a Serra Húnter Fellow (Professor Lector Serra Húnter) under the Serra Húnter Plan (Pla Serra Húnter). A. Atay gratefully acknowledges financial support by the University of Barcelona through grant AS017672. C. Trudeau gratefully acknowledges financial support by the Social Sciences and Humanities Research Council of Canada [grant number 435-2019-0141].

Atay: Department of Mathematical Economics, Finance and Actuarial Sciences, and Barcelona Economic Analysis Team (BEAT), University of Barcelona, Spain. E-mail: aatay@ub.edu.

Trudeau: Department of Economics, University of Windsor, Windsor, ON, Canada. E-mail: trudeauc@uwindsor.ca.

1. Introduction

Consider a set of agents with jobs that have to be executed by a number of machines in such a way that the aim is to minimize the total cost based on some criterion. We observe such problems in many real-life applications such as manufacturing, health care, logistics, etc. In this paper, we consider queueing problems from two different perspectives; (i) queueing problems that consider the problem of optimally queueing the agents before they arrive, (ii) queueing problems that consider the problem of reorganizing (rescheduling) an existing queue optimally. In both problems, a set of agents wait for their jobs to be processed on machines. Each agent has a job that needs the same amount of processing time with a different unit waiting cost. We refer to [Chun \(2016\)](#) for a comprehensive survey on queueing theory.

This paper is the first one that allows for an endogenous number of machines. [Maniquet \(2003\)](#) studied one machine queueing problem from a cooperative game theoretical perspective and showed that the rule assigning positions in the queue and compensations is the [Shapley \(1953\)](#) value of the associated TU-game. [Chun \(2006\)](#) introduced a pessimistic definition of the worth that can be generated by a subset of agents. It is proved that different definitions lead to very different rules. Towards a generalization to multiple machines, [Chun and Heo \(2008\)](#) consider queueing problems with two parallel machines. [Curiel et al. \(1989\)](#) are the first to study one-machine sequencing problems from a cooperative game theoretical point of view. That is, queueing problems with an initial queue where rescheduling is allowed to improve upon the initial situation. The rescheduling of jobs is allowed to reduce weighted completion time and the total savings by rescheduling can be shared by agents who own the jobs. [Hamers et al. \(1999\)](#) (see also [Slikker, 2006a](#)) consider multiple parallel sequencing situations where the number of machines is fixed. They guarantee the non-emptiness of the core for one and two machine situations, and moreover for two subclasses when there are at least three machines.

In this paper, we consider that the number of machines is endogenous. Each machine has a cost to activate. Hence, in both types of problems, a subset of agents can “buy” as many machines as they want to in exchange of its cost, and under some constraints in problems with an existing initial order, might be able to “sell” some of the existing machines. Moreover, each agent incurs some waiting cost until her job is processed and she can leave the system. Then, we take a game theoretical approach to address the question on how to distribute among the players the proceeds of their cooperation, whenever they (re)schedule their jobs to be processed in an optimal way, minimizing total costs. Following the vast literature on different problems on rescheduling an initial queue (see for instance [Calleja et al., 2002](#); [Musegaas et al., 2015](#); [Bahel and Trudeau, 2019](#); [Atay et al., 2021](#)), we examine conditions guaranteeing the existence of stable allocations.

First, we examine queueing problems that consider the problem of optimally queueing the agents before they arrive. Traditionally, queueing problems have a fixed number of machines, and their cost is sunk and thus ignored. The resulting cost game is then super-additive, as congestion implies that the total waiting cost for two agents is larger than the sum of their waiting costs if they are alone. With an endogenous number of machines, the cost function is always subadditive, as two agents can always each buy a machine, generating costs equal to the sum of their individual stand-alone costs. This allows a traditional definition of the core. We provide a lower bound and an upper bound on the cost of a machine for the non-emptiness of the core of a queueing problem with an endogenous number of machines (Theorems 1, 2). In the second case, we provide a full description of the core (Theorem 2).

Next, we consider the problem of rescheduling an existing queue optimally with an endogenous number of machines which we call the requeueing problem. While the sequencing literature has crucially depended on the assumptions regarding the admissible rearrangements of the initial queue by a coalition on a given machine, our results depend mostly on the assumptions made on the adjustment of the queue when new machines are added.

If we suppose that a coalition will be the sole users of a new machine it adds, we see that the core of a requeueing game with an endogenous number of machines may be empty (Example 2), and that under any assumptions on how a coalition can reorganize its members on the existing machines. Nevertheless, we provide sufficient conditions to guarantee the non-emptiness of the core (Theorems 3, 4). We also obtain positive results if we suppose that machines are public goods. That is, when we add machines, the whole queue moves up, and not only the members of the coalition that paid for the extra machines. The distinction is akin to establishing VIP machines and general-use machine. We show that under the assumption of public machines, whenever the initial schedule efficiently orders agents from high to low waiting costs but might not have the optimal number of machines, the public requeueing games always have a non-empty core (Theorem 5).

The paper is organized as follows. In Section 2 we present queueing problems with an endogenous number of machines. In Section 3 we introduce the associated TU-game for queueing problems with an endogenous number of machines. We derive upper and lower bounds on the cost of a machine to guarantee the existence of stable allocations as well as a full characterization of the set of stable allocations. In Section 4 we introduce two types of problems and their associated TU-games, the so-called private and public requeueing problems (and games). For private requeueing games, although stable allocations need not exist, we provide an upper-bound and a lower-bound to guarantee their existence. For

public requeuing games, we show that stable allocations always exist if the initial schedule serves agents with larger waiting costs first. Finally, we draw conclusions in Section 5.

2. Queueing problems with an endogenous number of machines

We examine first the queueing problem. We have a set of agents $N = \{1, 2, \dots, n\}$. When no confusion arises we denote by $|N| = n$ the cardinality of the set of agents. Each agent has one job to be processed on a machine. The agents have access to an unlimited number of machines, but they must pay $b \in \mathbb{R}_+$ for each machine that they use. All jobs and all machines are identical, and each machine can process one job per period. We assume that each machine starts processing at time 0.

Every agent $i \in N$ has a waiting cost that is linear with respect to the time it spends in the system. The waiting cost function of an agent $i \in N$ is $w_i t$ where $w_i > 0$ is the waiting (weight) cost per unit time of player i and t is the period at which the job has been processed. We refer to the vector of weights by $w := (w_i)_{i \in N}$. Let w_k^S be the waiting cost of the k^{th} agent (according to the order N) in S and $w_k^{-S} \equiv w_k^{N \setminus S}$ be the waiting cost of the k^{th} agent outside the coalition S .

A *queueing problem with an endogenous number of machines* can be described as (N, w, b) where N is the set of agents, w is the vector of unit waiting costs and $b \in \mathbb{R}_+$ is the cost of a machine. We suppose that $w_1 \geq w_2 \geq \dots \geq w_n$.

In a queueing problem, we examine the problem before agents arrive to queue: we are looking for the optimal number of machines and the optimal queueing of agents on those machines, the objective being the minimization of the total cost, consisting of the agents' waiting costs and the machine costs.

The solution consists in choosing a number of machines $m \in \{1, \dots, n\}$ and a schedule $\sigma = (\varphi, s)$, where $\varphi : N \rightarrow \{1, \dots, m\}$ assigns agents to machines and $s : N \rightarrow \mathbb{N} \cup \{0\}$ assigns to each agent a starting time. A schedule $\sigma = (\varphi, s)$ is admissible if for all $i, j \in N$, $\varphi(i) = \varphi(j) \Rightarrow s(i) \neq s(j)$. In words, if two agents are assigned to the same machine, they must have different starting times. The set of all possible schedules with m machines is denoted by $\Sigma(m)$. A scheduling plan is (m, σ) , with $\sigma \in \Sigma(m)$.

Let $N^k = \{i \in N : \varphi(i) = k\}$ be the set of agents assigned to machine k . A schedule is a *semi-active schedule* if there is no job which could be started earlier without altering the processing schedule. This has two implications for σ . First, we must have that if $\varphi(i) = k$ and $s(i) = l > 0$, there must be $j \in N^k$ such that $s(j) = l'$ for all $l' \in \{0, \dots, l-1\}$. Second, we must have $|N^k| - |N^{k'}| < 2$ for all $k, k' \in \{1, \dots, m\}$. In words, the first condition imposes that a schedule on a given machine has no downtime, and processes a job at all periods until all agents assigned to that machine have their job processed. The second condition imposes a difference in the number of agents assigned to pairs of machines to be at

most one; otherwise, we could move the last agent on the first machine to the last position on the second machine, reducing the processing time of that agent without affecting the processing time of other agents.

Since no preemption is allowed, the completion time of the job of agent i according to $\sigma = (\varphi, s)$ is $s(i) + 1$. Hence, the waiting cost of an agent $i \in N$ can be written as $c_\sigma(i) = w_i(s(i) + 1)$.

We thus need to find (m, σ) that optimizes the following objective function:

$$\min_{m \in \{1, \dots, n\}} \left(bm + \min_{\sigma \in \Sigma^m} \sum_{i \in N} c_\sigma(i) \right).$$

It is well-established in the literature that, for the one-machine case (with equal processing times), the total cost is minimal if the players are arranged according to their waiting costs in a decreasing order (see [Smith, 1956](#); [Curriel et al., 1989](#)). With multiple machines, it remains optimal to not process jobs of agents with larger waiting costs after those of agents with smaller waiting costs, i.e. $w_i < w_j \Rightarrow s(i) \leq s(j)$.

Given this result, if we install m machines, it is optimal to schedule the m agents with the highest waiting costs (agents $\{1, \dots, m\}$) at time 0, and it is irrelevant to which machine each agent is assigned to. The next m agents are then scheduled in the next period, and so on. Thus, the queueing problem reduces to finding the number of machines that solves

$$\min_{m \in \{1, \dots, n\}} \left(bm + \sum_{i \in N} \left(\left\lceil \frac{i}{m} \right\rceil \right) w_i \right).^1$$

We provide some initial results on the structure of the game. Let $m(S)$ be the optimal number of machines for coalition $S \subseteq N$.²

Lemma 1. *Fix the set of agents N . For any weight vector w there exists a non-increasing function $r^w : \{2, \dots, n\} \rightarrow \mathbb{R}_+$ such that:*

- (i) if $b \geq r^w(2)$, then $m(N) = 1$;
- (ii) if $r^w(k) > b \geq r^w(k+1)$ for some $1 < k < n$, then $m(N) = k$;
- (iii) if $r^w(n) > b$ then $m(N) = n$.

Proof. Fix N and w . The total cost when k machines are used would be cheaper than when $k - 1$ machines are used if

$$bk + \sum_{i \in N} \left(\left\lceil \frac{i}{k} \right\rceil \right) w_i \leq b(k-1) + \sum_{i \in N} \left(\left\lceil \frac{i}{k-1} \right\rceil \right) w_i$$

¹For all $x \in \mathbb{R}$, $\lceil x \rceil := \min\{k \in \mathbb{Z} | x \leq k\}$.

²There might be a tie, in which case pick the lowest number of machines among optimal ones.

which simplifies to

$$\begin{aligned}
 b &\leq \sum_{i \in N} \left(\left\lceil \frac{i}{k-1} \right\rceil - \left\lceil \frac{i}{k} \right\rceil \right) w_i \\
 \text{(a)} \quad &= w_k + \sum_{i=k+1}^n \left(\left\lceil \frac{i}{k-1} \right\rceil - \left\lceil \frac{i}{k} \right\rceil \right) w_i.
 \end{aligned}$$

The inequality (a) provides an upper-bound on the cost of a machine such that we prefer to use k machines to $k-1$ machines. Let us denote this number obtained in (a) by $r^w(k)$. This defines a function $r^w : \{2, \dots, n\} \rightarrow \mathbb{R}_+$.

We next show that this function is non-increasing. We show that $r^w(k) \leq r^w(k-1)$, that is,

$$\text{(b)} \quad w_k + \sum_{i=k+1}^n \left(\left\lceil \frac{i}{k-1} \right\rceil - \left\lceil \frac{i}{k} \right\rceil \right) w_i \leq w_{k-1} + \sum_{i=k}^n \left(\left\lceil \frac{i}{k-2} \right\rceil - \left\lceil \frac{i}{k-1} \right\rceil \right) w_i.$$

By assumption, $w_k \leq w_{k-1}$. We will show that

$$\sum_{i=k+1}^n \left(\left\lceil \frac{i}{k-1} \right\rceil - \left\lceil \frac{i}{k} \right\rceil \right) w_i \leq \sum_{i=k}^n \left(\left\lceil \frac{i}{k-2} \right\rceil - \left\lceil \frac{i}{k-1} \right\rceil \right) w_i,$$

which together with $w_k \leq w_{k-1}$ show that the inequality holds. To do so, we compare the right-hand side and the left-hand side summands of the same order in the inequality (b). We see that

$$\begin{aligned}
 \left(\left\lceil \frac{k+1}{k-1} \right\rceil - \left\lceil \frac{k+1}{k} \right\rceil \right) w_{k+1} &\leq \left(\left\lceil \frac{k}{k-2} \right\rceil - \left\lceil \frac{k}{k-1} \right\rceil \right) w_k \\
 \left(\left\lceil \frac{k+2}{k-1} \right\rceil - \left\lceil \frac{k+2}{k} \right\rceil \right) w_{k+2} &\leq \left(\left\lceil \frac{k+1}{k-2} \right\rceil - \left\lceil \frac{k+1}{k-1} \right\rceil \right) w_{k+1} \\
 &\vdots \\
 \left(\left\lceil \frac{n-1}{k-1} \right\rceil - \left\lceil \frac{n-1}{k} \right\rceil \right) w_{n-1} &\leq \left(\left\lceil \frac{n-2}{k-2} \right\rceil - \left\lceil \frac{n-2}{k-1} \right\rceil \right) w_{n-2} \\
 \left(\left\lceil \frac{n}{k-1} \right\rceil - \left\lceil \frac{n}{k} \right\rceil \right) w_n &\leq \left(\left\lceil \frac{n-1}{k-2} \right\rceil - \left\lceil \frac{n-1}{k-1} \right\rceil \right) w_{n-1} \\
 0 &\leq \left(\left\lceil \frac{k}{k-2} \right\rceil - \left\lceil \frac{k}{k-1} \right\rceil \right) w_n,
 \end{aligned}$$

and we see that removing a machine is costlier in terms of waiting costs if there are less machines in the initial problem. Applying the result recursively, starting with $r^w(n)$, we obtain that r^w is non-increasing.

It remains to show that we can define m using r^w . Let $C(N, k)$ be the cost for coalition N if it uses k machines. Suppose that $b \geq r^w(2)$. Then, since r^w is non-increasing $b \geq r^w(k)$

for all $k \in \{2, \dots, n\}$. This implies that $C(N, k) \leq C(N, k+1)$ for all $k = 1, \dots, n-1$. By transitivity, $C(N, 1) \leq C(N, k)$ for all $k \in \{2, \dots, n\}$ and thus $m(N) = 1$.

Suppose next that $r^w(k) > b \geq r^w(k+1)$ for some $1 < k < n$. By the same argument as above, $b \geq r^w(k+1)$ implies that $C(N, k) \leq C(N, l)$ for all $l \in \{k+1, \dots, n\}$. Since r^w is non-increasing, $r^w(k) > b$ implies that $r^w(l) > b$ for all $l = 2, \dots, k$. This implies that $C(N, l) < C(N, l-1)$ for all $l = 2, \dots, k$. By transitivity, $C(N, k) < C(N, l)$ for all $l \in \{1, \dots, k-1\}$. Combining with the previous result, we obtain $m(N) = k$.

Finally, suppose that $r^w(n) > b$. By the same argument as above, we have that $C(N, n) < C(N, l)$ for all $l \in \{1, \dots, n-1\}$ and we obtain $m(N) = n$. \square

We can similarly define a non-increasing function $r_S^w : \{2, \dots, |S|\} \rightarrow \mathbb{R}_+$ for all $S \subset N$ such that $|S| > 1$ to determine $m(S)$. For singletons, it is always optimal to use a single machine, and thus $m(\{i\}) = 1$ for all $i \in N$. Observing the structure of these functions r_S^w , the following result follows:

Lemma 2. *For all values of w and b , we have:*

- (i) $m(S) \leq m(T)$ for all $S \subset T \subseteq N$;
- (ii) $m(S \cup \{i\}) \leq m(S \cup \{j\})$ for all $S \subseteq N \setminus \{i, j\}$, and $i > j$.

Proof. Let $r_S^w(k)$ be the equivalent of $r^w(k)$ for coalition S .

(i) If $m(T) \geq |S|$, the result is immediate. Thus, suppose that $m(T) < |S|$.

We show that for any $S \subset T \subseteq N$ and $k = 2, \dots, |S|$, we have that $r_S^w(k) \leq r_T^w(k)$. That is,

$$\begin{aligned} r_S^w(k) &= w_k^S + \sum_{l=k+1}^{|S|} \left(\left\lceil \frac{l}{k-1} \right\rceil - \left\lfloor \frac{l}{k} \right\rfloor \right) w_l^S \\ &\leq w_k^T + \sum_{l=k+1}^{|S|} \left(\left\lceil \frac{l}{k-1} \right\rceil - \left\lfloor \frac{l}{k} \right\rfloor \right) w_l^T \\ &\leq w_k^T + \sum_{l=k+1}^{|T|} \left(\left\lceil \frac{l}{k-1} \right\rceil - \left\lfloor \frac{l}{k} \right\rfloor \right) w_l^T \\ &= r_T^w(k), \end{aligned}$$

where the first inequality comes from the fact that $w_k^S \leq w_k^T$ for all k .

Then, if $b \geq r_T^w(2)$, $b \geq r_S^w(2)$ and $m(S) = m(T) = 1$. Otherwise, $m(S)$ is the highest integer such that $b < r_S^w(m(S))$. But since $r_S^w(m(S)) \leq r_T^w(m(S))$, we have $b < r_S^w(m(T))$, and thus $m(S) \leq m(T)$, as desired.

ii) The proof is identical to part i), replacing S by $S \cup \{i\}$ and T by $S \cup \{j\}$. \square

In words, if we add agents to a coalition, it cannot be optimal to use less machines. The strategy to add an additional machine can only become more profitable (or less unprofitable) as the new agents might have higher waiting costs, and the additional agents might

lead to more saved waiting costs. The same is true if we replace an agent by one with a larger waiting cost.

3. Queueing games with an endogenous number of machines

A *cooperative transferable utility (TU) game* is defined by the pair (N, C) where N is the set of the players and the coalitional function C assigns to each coalition $T \subseteq N$ its cost $C(T) \in \mathbb{R}$, with $C(\emptyset) = 0$.

Cooperative game theory aims to allocate the value of the grand coalition in such a way that the cooperation is preserved among the agents. Given a cooperative game (N, C) , a *cost allocation* is $y \in \mathbb{R}^N$, where y_i stands for the cost paid by player $i \in N$. The total payment by a coalition $S \subseteq N$ is denoted by $y(S) = \sum_{i \in S} y_i$ with $y(\emptyset) = 0$.

In this section, we study the set of stable allocations of the total cost, where no coalition of agents pays more than its stand-alone cost. To do so, for any queueing problem with an endogenous number of machines, we will introduce a TU-game and study the *core* of the associated TU-game (Gillies, 1959).

Formally, let (N, w, b) be a queueing problem with an endogenous number of machines. Then, the corresponding *queueing game with an endogenous number of machines* is the pair (N, C) where N is the set of players, and C is the characteristic function that assigns the minimal cost $C(T)$ to each coalition $T \subseteq N$ to queue its members, with $C(\emptyset) = 0$. $C(T)$ includes both the waiting costs and the cost of machines. The core of a cooperative cost game (N, C) is:

$$\text{Core}(C) = \{y \in \mathbb{R}^N \mid y(N) = C(N), \quad y(S) \leq C(S) \quad \text{for all } S \subset N\}.$$

A game is called *balanced* if its core is non-empty.

Concave cost functions always have a non-empty core (Shapley, 1971). Formally, a game (N, C) is said to be *concave* if for all $i \in N$ and all $S \subseteq T \subseteq N \setminus \{i\}$, it holds $C(T \cup \{i\}) - C(T) \leq C(S \cup \{i\}) - C(S)$.

3.1. On the non-emptiness of queueing games with an endogenous number of machines. First, for queueing games with an endogenous number of machines, we provide an upper bound on the cost of a machine to guarantee the non-emptiness of the core.

For the sake of comprehensiveness, let us introduce some notation: Let $\mu \equiv \lceil \frac{n}{2} \rceil$. If n is even, then $\{1, \dots, \mu\}$ and $\{\mu + 1, \dots, n\}$ both contain μ agents, while if n is odd, then $\{1, \dots, \mu\}$ contains μ agents and $\{\mu + 1, \dots, n\}$ contains $\mu - 1$ agents.

Theorem 1. *Let (N, w, b) be a queueing problem with an endogenous number of machines, and (N, C) be the associated TU-game.*

- (i) *If $b \leq w_\mu$ then $y = (\min(b + w_i, 2w_i))_{i \in N} \in \text{Core}(C)$;*

(ii) If $b \leq w_{\lceil \frac{2n+1}{4} \rceil}$ then $\text{Core}(C) = (\min(b + w_i, 2w_i))_{i \in N}$.

Proof. (i) We first show that the allocation $y = (\min(b + w_i, 2w_i))_{i \in N}$ is budget balanced.

First, notice that if we use $k \geq \mu$ machines, than agents wait at most 2 periods. Adding an additional machine allows to reduce the waiting cost of agent $k + 1$ from 2 to 1 period, with all other waiting costs remaining the same. Thus, $r^w(k) = w_k$ for all $k > \mu$.

Notice also that when moving from $\mu - 1$ to μ machines the cost savings are larger: in addition to agent μ waiting for 1 period instead of 2, some other agents will wait 2 periods instead of 3. Thus, $r^w(\mu) \geq w_\mu$. Thus, given that $b \leq w_\mu \leq r^w(\mu)$ and by Lemma 1, $m(N) \geq \mu$.

Let $C(\cdot, k)$ be the cost function that assigns to each coalition the total cost if it uses k machines to process their jobs. For $k \geq \mu$,

$$C(N, k) = kb + \sum_{i=1}^k w_i + \sum_{i=k+1}^n 2w_i.$$

Thus,

$$\begin{aligned} C(N) &= \min_{k \in \{\mu, \dots, n\}} \left\{ kb + \sum_{i=1}^k w_i + \sum_{i=k+1}^n 2w_i \right\} \\ &= b(\mu - 1) + \sum_{i=1}^{\mu-1} w_i + \min_{k \in \{\mu, \dots, n\}} \left\{ b(k - \mu + 1) + \sum_{i=\mu}^k w_i + \sum_{i=k+1}^n 2w_i \right\} \\ &= \sum_{i=1}^{\mu-1} (b + w_i) + \sum_{i=\mu}^n \min(b + w_i, 2w_i) \\ &= \sum_{i \in N} \min(b + w_i, 2w_i) \\ &= \sum_{i \in N} y_i \end{aligned}$$

The third equality comes from the fact that for all $k > \mu$, $r^w(k) = w_k$, implying that we use at least k machines if and only $b + w_k \leq 2w_k$. While $r^w(\mu) \geq w_\mu$, by assumption $b \leq w_\mu$. The fourth equality also comes from the fact that by assumption, $b \leq w_\mu$.

It remains to prove that core constraints are satisfied, i.e., $y(T) \leq C(T)$ for all $T \subset N$. Fix $T \subset N$ and suppose that κ is the optimal number of machines for T .

We have that

$$\begin{aligned} \sum_{i \in T} y_i &\leq \kappa b + \sum_{i=1}^{\kappa} w_i^T + \sum_{i=\kappa+1}^{|T|} 2w_i^T \\ &\leq C(T), \end{aligned}$$

where the first inequality is obtained by assigning $b + w_i$ to the first κ agents in T and $2w_i$ to others, regardless of which of these two values is minimal, and the second inequality comes from the fact that the expression is exactly the cost of coalition T if $\kappa \geq \lfloor \frac{|T|}{2} \rfloor$, with the cost no smaller otherwise. Thus the core constraint is satisfied. Since T is arbitrarily chosen, the proof is complete.

(ii) Notice first that if n is odd, $\lceil \frac{2n+1}{4} \rceil = \mu$, while if n is even, $\lceil \frac{2n+1}{4} \rceil = \mu + 1$. In particular, for any n , we have that $\lceil \frac{2n+1}{4} \rceil - 1 \geq \frac{n-1}{2}$.

If $b < w_n$, then $C(S) = |S|b + \sum_{i \in S} w_i$ for all $S \subseteq N$ and the result is immediate. Thus, suppose that $b \geq w_n$.

Suppose that $w_{k+1} \leq b < w_k$ for $k \in \left\{ \lceil \frac{2n+1}{4} \rceil, \dots, n-1 \right\}$. Then, by Lemma 1, $C(N) = kb + \sum_{i=1}^k w_i + \sum_{i=k+1}^n 2w_i$.

Consider coalition $N \setminus \{i\}$ for $i \in \{1, \dots, k\}$. If they use k machines, the cost is $kb + \sum_{j=1}^k w_j + \sum_{j=k+1}^n 2w_j - w_i - w_{k+1}$. If they use $k-1$ machines, the cost is $(k-1)b + \sum_{j=1}^k w_j + \sum_{j=k+1}^n 2w_j - w_i$, as $k-1 \geq \lceil \frac{2n+1}{4} \rceil - 1 \geq \frac{n-1}{2}$. Thus, it prefers to use $k-1$ machines if $b \geq w_{k+1}$, which is satisfied. If they use $k-2$ machines, the cost is at least $(k-2)b + \sum_{j=1}^k w_j + \sum_{j=k+1}^n 2w_j - w_i + w_k$ (as some agents might have to wait more than 2 periods now), and as $b \leq w_k$ it prefers to use $k-1$ machines. Thus, $C(N \setminus \{i\}) = (k-1)b + \sum_{j=1}^k w_j + \sum_{j=k+1}^n 2w_j - w_i$.

Notice that $C(N \setminus \{i\}) + C(\{i\}) = C(N)$, and thus in any core allocation, we must have $y_i = C(\{i\}) = b + w_i$ for all $i \in \{1, \dots, k\}$.

Next, consider coalition $\{i, j\}$, with $i \in \{1, \dots, k\}$ and $j \in \{k+1, \dots, n\}$. If it uses a single machine, the cost is $b + w_i + 2w_j$. If it uses 2 machines, the cost is $2b + w_i + w_j$. It prefers to use a single machine as $b \geq w_{k+1} \geq w_j$. Thus, $C(\{i, j\}) = b + w_i + 2w_j$. Since $y_i = b + w_i$, we obtain a core constraint of $y_j \leq 2w_j$ for all $j \in \{k+1, \dots, n\}$. Given the value of $C(N)$, our only core candidate is $y_i = b + w_i$ for all $i \in \{1, \dots, k\}$ and $y_j = 2w_j$ for all $j \in \{k+1, \dots, n\}$. Given that we have shown in part i) that it is a core allocation, our proof is complete. \square

Following an upper-bound on the cost of a machine for the non-emptiness of the core, we provide a full characterization of the core making use of a lower-bound for the non-emptiness of the core.

Theorem 2. *Let (N, w, b) be a queueing problem with an endogenous number of machines, and (N, C) be the associated TU-game. Then,*

(i) *if $b \geq \sum_{i=1}^n (i-1)w_i$, $\text{Core}(C) = \text{Core}(\hat{C}) \neq \emptyset$ with $\hat{C}(T) := C(T) - \sum_{i=1}^{n-|T|-1} iw_{i+1}^{-T}$ for all*

$\emptyset \neq T \subseteq N$. Moreover, \hat{C} is concave.

(ii) *if $b \in \left[w_2 + \sum_{i=3}^n \left(i - \left\lceil \frac{i}{2} \right\rceil \right) w_i, \sum_{i=1}^n (i-1)w_i \right)$, then $\text{Core}(C) = \emptyset$.*

Proof. Notice that $w_2 + \sum_{i=3}^n \left(i - \left\lfloor \frac{i}{2} \right\rfloor\right) w_i = r^w(2)$, and thus by Lemma 1, $m(N) = 1$. By Lemma 2, $m(S) = 1$ for all $S \subseteq N$, and thus all coalitions use a single machine.

i) Notice first that $\hat{C}(T) = C(T)$ if $|T| \geq n - 1$. Let $j, k \in N$ and consider $N \setminus \{j, k\}$. By the grand coalition efficiency and individual rationality, if y is a core allocation, then $y(N \setminus \{j, k\}) \leq C(N \setminus \{j\}) + C(N \setminus \{k\}) - C(N)$.

Suppose that we have shown that $y(T) \leq \hat{C}(T)$ in any core allocation if $|T| > m$. We need to show that it implies that $y(T) \leq \hat{C}(T)$ in any core allocation if $|T| = m$.

Fix T such that $|T| = m$. Following the grand coalition efficiency and individual rationality, if y is a core allocation, then $y(T) \leq \hat{C}(T \cup \{k\}) + \hat{C}(N \setminus \{k\}) - \hat{C}(N)$.

Now, consider the cost function $\hat{C}(T) = C(T) - \sum_{i=1}^{n-|T|-1} iw_{i+1}^{-T}$, for all $\emptyset \neq T \subseteq N$. By definition, $\hat{C} \leq C$ and $\hat{C}(N) = C(N)$. We will show that \hat{C} is concave whenever the lower-bound on the cost of a machine in (i) is satisfied.

First, notice that for all $k \in N$, $\hat{C}(\{k\}) = b + w_k - \sum_{i=1}^{n-2} iw_{i+1}^{-\{k\}} = b + w_k - \sum_{i < k} (i - 1)w_i - \sum_{i > k} (i - 2)w_i$.

Recall that w_k^T denotes the waiting cost of the k^{th} agent in T , according to the order in N and $w_k^{-T} \equiv w_k^{N \setminus T}$. Next, fix $\emptyset \neq T \subseteq N \setminus \{k\}$. Then, we have that

$$\begin{aligned} \hat{C}(T \cup \{k\}) - \hat{C}(T) &= \sum_{i=1}^{|T|+1} iw_i^{T \cup \{k\}} - \sum_{i=1}^{|T|} iw_i^T - \sum_{i=1}^{n-|T|-2} iw_{i+1}^{-(T \cup \{k\})} + \sum_{i=1}^{n-|T|-1} iw_{i+1}^{-T} \\ &= \sum_{i > k} w_i + kw_k \end{aligned}$$

The equality is based on the following observations: if $i < k$ and $i \in T$, then its rank in $T \cup \{k\}$ is the same as in T , and the terms cancel out. The same is true if $i \in N \setminus T$. If $i > k$ and $i \in T$, the rank of i is one higher in $T \cup \{k\}$ than in T . If $i > k$ and $i \in N \setminus T$, the rank of i is one smaller in $N \setminus (T \cup \{k\})$ than in $N \setminus T$. In all cases, the difference is w_i . As for k , it appears in the first and fourth terms. The weight on its waiting cost is its rank in $T \cup \{k\}$ plus its rank in $N \setminus T$ minus 1. For all agents, that equals k .

This result is independent of T , as long as $T \neq \emptyset$. Making use of this result, we proceed to show $y(T) \leq \hat{C}(T)$. We have seen that $\hat{C}(T \cup \{k\}) = b + \sum_{i=1}^{|T|+1} iw_i^{T \cup \{k\}} - \sum_{i=1}^{n-|T|-2} iw_{i+1}^{-(T \cup \{k\})}$ and $\hat{C}(N \setminus \{k\}) = b + \sum_{i=1}^{n-1} iw_i^{N \setminus \{k\}}$ and $\hat{C}(N) = b + \sum_{i=1}^n iw_i$. Thus,

$$\hat{C}(T \cup \{k\}) + \hat{C}(N \setminus \{k\}) - \hat{C}(N) = b + \sum_{i=1}^{|T|+1} iw_i^{T \cup \{k\}} - \sum_{i=1}^{n-|T|-2} iw_{i+1}^{-(T \cup \{k\})} + \sum_{i=1}^{n-1} iw_i^{N \setminus \{k\}} - \sum_{i=1}^n iw_i.$$

Next, let us distinguish several cases.

Case 1: $i \in T$ such that $i < k$. Then, i is the i^{th} agent in $N \setminus \{k\}$ and in N . The agent i has the same rank in T and in $T \cup \{k\}$.

Case 2: $i \in T$ such that $i > k$. Then, i is the $(i - 1)^{th}$ agent in $N \setminus \{k\}$ and i^{th} in N . The agent i has the same rank in T is one lower than in $T \cup \{k\}$.

Case 3: $j \notin T$ such that $j < k$. Then, j is the j^{th} agent in $N \setminus \{k\}$ and in N . The agent j has the same rank in $N \setminus T$ and in $N \setminus (T \cup \{k\})$.

Case 4: $j \notin T$ such that $j > k$. Then, j is the $(j - 1)^{th}$ agent in $N \setminus \{k\}$ and j^{th} in N . The rank of j in $N \setminus T$ is one higher than in $N \setminus (T \cup \{k\})$.

Note that in the second term agent k his waiting cost gets assigned a weight equal to his rank in $T \cup \{k\}$. In the fifth term, his waiting cost gets assigned a weight equal to k , his rank in N . These cases imply that

$$\begin{aligned} \hat{C}(T \cup \{k\}) + \hat{C}(N \setminus \{k\}) - \hat{C}(N) &= b + \sum_{i=1}^{|T|} iw_i^T - \sum_{i=1}^{n-|T|-1} iw_{i+1}^{-T} \\ &= \hat{C}(T). \end{aligned}$$

Thus, we have the core constraint $y(T) \leq \hat{C}(T)$. In order to finish the proof of (i), we will show that \hat{C} is concave.

To verify concavity, it remains to check that $\hat{C}(T \cup \{k\}) - \hat{C}(T) \leq \hat{C}(\{k\}) - \hat{C}(\emptyset) = \hat{C}(\{k\})$ or equivalently,

$$\sum_{i>k} w_i + kw_k \leq b + w_k - \sum_{i<k} (i-1)w_i - \sum_{i>k} (i-2)w_i$$

which simplifies to

$$b \geq \sum_{i=1}^n (i-1)w_i$$

which is satisfied by assumption. Thus, \hat{C} is concave and hence $Core(\hat{C})$ is non-empty which finishes the proof of (i).

To prove the statement (ii), recall that we have shown that $C(N \setminus \{l\}) = b + \sum_{i=1}^{n-1} iw_i^{N \setminus \{l\}}$ for $l \in N$ and $C(N) = b + \sum_{i=1}^n iw_i$. Thus,

$$\begin{aligned} \sum_{l \in N} C(N \setminus \{l\}) &= \sum_{l \in N} \left(b + \sum_{i=1}^{n-1} iw_i^{N \setminus \{l\}} \right) \\ &= nb + \sum_{l \in N} \sum_{i=1}^{n-1} iw_i^{N \setminus \{l\}} \\ &= nb + \sum_{i \in N} (i(n-2) + 1) w_i. \end{aligned}$$

The last equality is obtained as follows: agent i does not appear in $N \setminus \{i\}$, appears at rank i in $N \setminus \{l\}$ if $l > i$, and appears at rank $i - 1$ if $l < i$. We have $n - i$ coalitions where $l > i$ and $i - 1$ coalitions where $l < i$. Thus, the coefficient associated to w_i is $(n - i)i + (i - 1)(i - 1) = i(n - 2) + 1$.

Then, we have that

$$\begin{aligned}
 \sum_{l \in N} C(N \setminus \{l\}) - (n-1)C(N) &= nb + \sum_{i \in N} (i(n-2) + 1)w_i - \left((n-1)b + \sum_{i \in N} (n-1)iw_i \right) \\
 &= b - \sum_{i \in N} (i-1)w_i \\
 &< 0,
 \end{aligned}$$

and thus $\sum_{l \in N} C(N \setminus \{l\}) < (n-1)C(N)$, by our assumption on b . Thus, $\text{Core}(C)$ is empty. \square

We provide an example that shows that the conditions in Theorems 1 and 2i) are not necessary for the core to be non-empty.

Example 1. Suppose that $N = \{1, 2, 3, 4\}$ and that $w_i = 25 - 5i$ for all $i \in N$. Then, Theorem 1 tells us that the core is non-empty if $b \leq 15$ while Theorem 2 (i) guarantees non-emptiness of the core for $b \geq 50$. For $b \in [35, 50)$, by Theorem 2 (ii) the core is empty. We verify what happens when $b \in (15, 35)$.

For $b \in [20, 35)$, coalitions $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$ use 2 machines, all others use a single one. Using the fact that we must have $y(\{i, j\}) = b + w_i + 2w_j$ for all $i \in \{1, 2\}, j \in \{3, 4\}$ in any core allocation, we obtain maximal allocations of $(40, 35, 25, 15)$. Using $C(N) - C(N \setminus \{i\})$, we obtain minimal allocations of $(b + 15, b + 10, b, b - 10)$. Immediately, the core is empty for $b \in (25, 35)$. For $b \in [20, 25]$, we can verify that the allocation $(b + 15, b + 10, 25, 15)$ is in the core.

For $b \in (15, 20)$, coalition $\{1, 2, 4\}$ also uses 2 machines. Using the same technique as above to obtain minimal and maximal allocations, our only candidate for a core allocation is $(b + 15, b + 10, 25, 15)$. But then, coalition $\{3, 4\}$ pays 40, while its stand-alone cost is $b + 20$, and thus there are no core allocations.

For $b \in (10, 15]$, all coalitions of 3 or more agents use 2 machines, as well as coalition $\{1, 2\}$. In addition to $(b + 20, b + 15, 20, 10)$, the allocation $\left(\frac{b}{2} + 25, \frac{b}{2} + 20, \frac{b}{2} + 15, \frac{b}{2} + 5\right)$ is also in the core.

The example is summarized in Figure 1, with the results on the core and the description of m , the optimal number of machines, depending on machine cost b . In order to provide a clear illustration, Figure 1 does not respect the appropriate proportions.

When $b \leq w_\mu$, the game we obtain is reminiscent of assignments games: we need to match agents with a machine (those with $b \leq w_i$) to those without (with $b > w_i$), matching at most one agent from the second group to each agent in the first group. An agent from the second group generates value of $b - w_i$ when he matches with any agent from the second group. Notice that the first group contains at least half of the agents. If it is strictly more, the only core allocation allocates all gains to the short side of the market, the agents

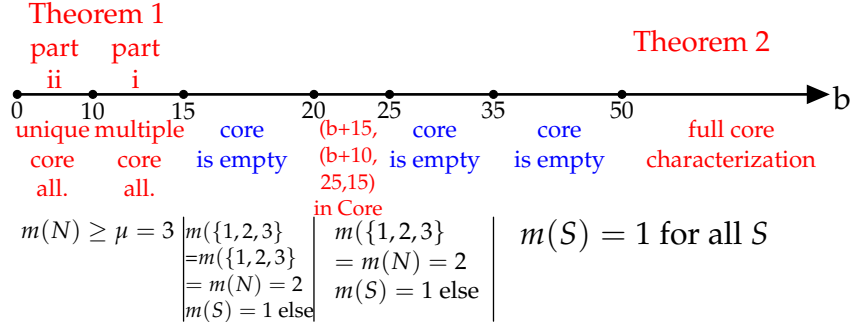


FIGURE 1. Summary of Example 1.

with $b > w_i$. If it is exactly half, as in our example above with $b \in (10, 15]$, the core contains multiple allocations. This is where the similarities with the assignment game ends, as members of the second group do create value when matched together (they share a machine), which constrains the allocation that is optimal for the first side (those with $b \leq w_i$).

4. Requeueing games with an endogenous number of machines

For the rest of the paper, we consider queueing problems with an existing queue, which study the problem from a different perspective: while queueing problems consider the minimal cost of organizing the queue for a set of players, starting from scratch, in the following, we consider requeueing problems where possible cost savings can be obtained when we reschedule a given queue. In our study of the problem with an endogeneous number of machines, this implies that we start with a given number of machines, and that the reorganization can include adding or removing machines.³

Then, a *requeueing problem with an endogenous number of machines* can be described by (N, m_0, σ_0, w, b) where m_0 is the initial number of machines and σ_0 is the initial (existing) queue. Our first aim is to find an optimal schedule that minimizes the total costs, as in Section 2. As for queueing games, we build a coalitional function from the requeueing games, now associating to each coalition $T \subseteq N$ the maximum cost savings $V(T)$ it can generate from the initially existing queue. We will distinguish between two cases based on whether new machines are exclusive for a set of agents (private) or available for all agents (public).

4.1. Private requeueing games. We consider requeueing problems in which if a coalition buys a new machine, it gains exclusive use of that machine and if a coalition sells a machine it recovers the full value of that machine. These two assumptions can be seen as “exclusive” use of machines for a coalition and hence they are “private” machines for a coalition.

³We use queueing problems with an existing queue and requeueing problems interchangeably.

In order to determine the maximal cost savings of a coalition $T \subseteq N$, we have to define which rearrangements are admissible. Various assumptions have been made on admissible rearrangements of the initial schedule, see [Curiel et al. \(1993\)](#), [Slikker \(2006b\)](#), [Musegaas et al. \(2015\)](#), and [Atay et al. \(2021\)](#), among others. Following the literature, we consider two approaches to define admissible rearrangements to study the non-emptiness of the core for requeueing games with an endogenous number of machines.

First, we do not allow agents in a coalition to jump over agents outside the coalition. Then, we say that a scheduling plan (m, σ) is *admissible* for a coalition T with respect to (m_0, σ_0) if for any agent outside coalition T there are no new agents in her set of predecessors. That is, for all $i \in N \setminus T$ it holds that $\varphi(i) = \varphi_0(i)$ and

$$(1) \quad \{l \in N^{\varphi(i)} : s(l) < s(i)\} \subseteq \{l \in N^{\varphi_0(i)} : s_0(l) < s_0(i)\}.$$

Notice that we do not require equality, as a predecessor of i might move to a new machine. For short, we call this assumption the “no swaps” assumption, and the set of admissible schedules for coalition T that satisfy (1) is denoted by Σ_T^{ns} .

Second, we relax the condition by allowing agents in a coalition to jump over agents outside the coalition. We say that a scheduling plan (m, σ) is *admissible* for a coalition T with respect to (m_0, σ_0) if the starting time for all agents outside the coalition does not increase. That is, for all $i \in N \setminus T$ it holds that $\varphi(i) = \varphi_0(i)$ and

$$(2) \quad s(i) \leq s_0(i).$$

Once again, we do not have equality, as predecessors are allowed to move to a new machine. By opposition, this is the “swaps” assumption, and the set of admissible schedules for coalition T that satisfy (2) is denoted by Σ_T^s .

In our setting, we also must consider the possibility for a coalition to sell a machine. We suppose that a coalition T can sell a machine only if all users of that machine are members of T . We then suppose that the agents that were on the removed machine move at the end of the queue on the remaining machines, a condition that is already covered by both (1) and (2).

For a set of admissible schedules, we can associate the corresponding cooperative TU-game called a private requeueing game with an endogenous number of machines. A *private requeueing problem with an endogenous number of machines* is a 5-tuple (N, m_0, σ_0, w, b) .⁴ The corresponding *private requeueing game with an endogenous number of machines* (N, V) is defined by

$$V(T) = c_{\sigma_0}(T) - c_{\sigma}(T) - (m - m_0)b,$$

⁴Since population and costs are fixed, with an abuse of notation, we denote a private requeueing problem also by the initial number of machines and the initial queue, (m_0, σ_0) .

where (m, σ) is an optimal admissible scheduling plan for coalition T . Furthermore, admissible schedules with and without swaps lead to different games. We denote private queueing games with swaps by V_s and private queueing games without swaps by V_{ns} .

While concavity of a cost game is a sufficient condition for its core to be non-empty, for value games the corresponding concept is that of convexity. The condition has been widely studied to prove balancedness of sequencing games associated with different problems (see for instance [Curiel et al., 1994](#); [Hamers et al., 2005](#); [Musegaas et al., 2018](#)). A game (N, v) is said to be *convex* if for all $i \in N$ and all $S \subseteq T \subseteq N \setminus \{i\}$, it holds $v(T \cup \{i\}) - v(T) \geq v(S \cup \{i\}) - v(S)$.

Unfortunately, [Example 2](#) shows that the associated game need not be balanced, regardless if admissible schedules allow or not to jump over players outside the coalition.

Example 2. Consider (N, m_0, σ_0, w, b) with $N = \{1, 2, 3, 4, 5\}$. The waiting costs per unit for agents are given by the weight vector $w = (w_i)_{i \in N} = (20, 15, 13, 13, 5)$, and the cost of a machine is $b = 18$.

First, we suppose that (m_0, σ_0) is such that we order agents in the queue on one machine according to their weights, in decreasing order:

$$m_1 \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}.$$

Notice that agent 1 is a dummy player since he is served first, moving to another machine is strictly worse for her. Thus, we can focus on the game (N, V_s) for the remaining agents. One can calculate that $V_s(\{2, 3, 4\}) = 36$, $V_s(\{2, 3, 5\}) = 25$, $V_s(\{3, 4, 5\}) = 44$, and $V_s(\{2, 3, 4, 5\}) = 46$. Next, let us consider the coalition $T = \{2, 4, 5\}$.

First, suppose that we allow players in the coalition to jump over players outside the coalition when we define admissible rearrangements. Take the coalition $T = \{2, 4, 5\}$. Then, an optimal scheduling plan for coalition T , (m_T, σ_T) , is

$$\begin{array}{l} m_1 \begin{array}{|c|c|c|} \hline 1 & 4 & 3 \\ \hline \end{array} \\ m_2 \begin{array}{|c|c|c|} \hline 2 & 5 & \\ \hline \end{array} \end{array}$$

and then the total waiting cost savings are $15+26+15=56$, but the coalition buys a machine at a cost 18, and hence the maximal total cost savings is $38 = V_s(\{2, 4, 5\})$. Then, $V_s(\{2, 3, 4\}) + V_s(\{2, 3, 5\}) + V_s(\{3, 4, 5\}) + V_s(\{2, 4, 5\}) = 143 > 138 = 3V_s(\{2, 3, 4, 5\})$, and hence the core is empty.

Second, suppose that we do not allow players in the coalition to jump over players outside the coalition when we define admissible rearrangements. Take the coalition $T = \{2, 4, 5\}$. Then, an optimal scheduling plan for coalition T , (m'_T, σ'_T) , is

$$\begin{array}{l} m_1 \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline \end{array} \\ m_2 \begin{array}{|c|c|c|} \hline 4 & 5 & \\ \hline \end{array} \end{array}$$

and then the total waiting cost savings are $39+15=54$, but the coalition buys a machine at a cost 18, and hence the maximal total cost savings is $36 = V_{ns}(\{2, 4, 5\})$. For all other

coalitions S , we have $V_{ns}(S) = V_s(S)$. Then, $V_{ns}(\{2, 3, 4\}) + V_{ns}(\{2, 3, 5\}) + V_{ns}(\{3, 4, 5\}) + V_{ns}(\{2, 4, 5\}) = 141 > 138 = 3V_{ns}(\{2, 3, 4, 5\})$, and hence the core is empty.

Although we have seen that the associated private requeueing game can have an empty core, there are sufficient conditions to guarantee the non-vacuity of the core. First, we provide a lower bound on the cost of a machine for the non-emptiness of the core.

Theorem 3. *Let (N, m_0, σ_0, w, b) be a private requeueing problem and let (N, V_{ns}) be the associated private requeueing game without swaps, and (N, V_s) be the associated private requeueing game with swaps. If $b \leq w_n$, then $\text{Core}(V_{ns}) \neq \emptyset$ and $\text{Core}(V_s) \neq \emptyset$.*

Proof. Since the cost a machine is at most equal to the smallest unit waiting cost, $b \leq w_n$, for any coalition $S \subseteq N$, it is optimal to have $|S|$ machines. Let $s_0(i)$ be the starting time of the job i under the schedule σ_0 . The first agents at each machine in the initial schedule, that is $i \in N$ such that $s_0(i) = 0$, need not change their position. For all $S \subseteq N$, let S_0 be the set of such agents. For all other agents $i \in N \setminus S_0$ such that $s_0(i) \geq 1$, buying a new machine is the option that maximizes the cost savings at any given coalition since $b \leq w_n$. Thus there exists a unique core allocation where $y_i = s_0(i)w_i - b$ for all $i \in N \setminus S_0$ and $y_i = 0$ for $i \in S_0$. Then, $\sum_{i \in N} y_i = \sum_{i \in S_0} 0 + \sum_{i \in N \setminus S_0} (s_0(i)w_i - b) = \sum_{i \in N \setminus S_0} s_0(i)w_i - (n - m_0)b = V_s(N) = V_{ns}(N)$, and efficiency holds. For any coalition $S \subset N$, $y(S) = \sum_{i \in S \setminus S_0} (s_0(i)w_i - b)$.

Since using a machine for each agent in the coalition is the optimal schedule in both cases $y(S) = V_s(S) = V_{ns}(S)$. Thus, y also satisfies coalitional rationality and hence it is a core allocation. \square

Following a lower-bound on the cost of a machine to guarantee the non-emptiness of the core, we provide an upper-bound on the cost of a machine for the non-emptiness of the core when the initial number of machines is 1, $m_0 = 1$. Intuitively, it consists in setting the machine cost so high that no coalition wants to buy a second machine. The problem then becomes one of reorganizing the queue on the existing machine, making the problem equivalent to one with a single machine and no possibility to add more.

Theorem 4. *Let (N, m_0, σ_0, w, b) be a private requeueing problem. If $m_0 = 1$ and $b \geq \max_{k=1}^n (n - k)w_i$, then $\text{Core}(V_s) \neq \emptyset$ and $\text{Core}(V_{ns}) \neq \emptyset$.*

Proof. We will show that for the private sequencing game without swaps where the agents are ordered in an increasing way with respect to their waiting costs at the initial order, making use of the only machine is better than buying a new machine for any coalition. Given that this is the worst case scenario in terms of ordering, and we still do not want to buy more than one machine, the result holds for all other orderings. Formally, we consider an initial schedule σ_0 such that $s_0(i) = n - i$ for all $i \in N$.

First, consider the last agent in the order, agent 1, which by definition is such that $w_1 \geq (w_i)_{i \in N \setminus \{1\}}$. If she buys a machine, the total cost savings are $(n-1)w_1 - b$. Since $b \geq \max_{k=1}^{\mu} (n-k)w_i \geq (n-1)w_1$, the total cost savings $(n-1)w_1 - b \leq 0$. Hence, agent 1 is worse off by buying a new machine. Note that for any other individual coalition $\{i\}$ such that $i \in N \setminus \{1\}$, since the gain by buying a new machine is $(n-i)w_i < (n-1)w_1$, the total cost savings $(n-i)w_i - b < (n-1)w_1 - b \leq 0$. Thus, no individual coalition $\{i\}$ such that $i \in N$ has incentives to buy a new machine.

We next show that it is true for a coalition containing $k \leq \mu$ agents. Consider the coalition of the last k agents in the order, agents 1 through k . Recall that $w_1 \geq \dots \geq w_k \geq w_i$ for all other $i \in N$ such that $s(i) < n-k$. Then, the gain for coalition $\{1, \dots, k\}$ is $(n-1)w_1 + \dots + (n-k)w_k$. Since $b \geq \max_{k=1}^{\mu} (n-k)w_i \geq (n-1)w_1 + \dots + (n-k)w_k$, the total cost savings $\{k-1, \dots, 1\}$ is $(n-1)w_1 + \dots + (n-k)w_k - b \leq 0$, and hence coalition $\{1, \dots, k\}$ prefers to use one machine. Notice that no other coalition S with $|S| = k$ can achieve higher total cost savings than coalition $\{1, \dots, k\}$, hence no k -agent coalition has incentives to buy a new machine.

Notice that for any coalition that consists of last k agents where $k > \mu$, only agents with a position in the initial order of more than μ get to use the new machine. Hence, their gains are less than the last μ agents, and they also prefer to use the only machine provided to them. Moreover, since we compare all possible coalitions with the same size coalition that consists of the last agents in the initial queue, our result under not allowing swaps subsumes if we allow swaps. Then, we deal with a 1-machine problem with an initial queue. Together with the result of [Curiel et al. \(1989\)](#) and [Hamers et al. \(1999\)](#) stating that the core is always non-empty for 1-machine problem, we guarantee the non-emptiness of the core whenever $b \geq \max_{k=1}^{\mu} (n-k)w_i$ with $m_0 = 1$. \square

Given that the previous result is obtained using the worst case scenario of a completely inefficient initial ordering, for a random initial ordering a lower bound guaranteeing a non-empty core could be found. However, a general expression for such a bound is difficult to obtain.

4.2. Public requeueing games. Implicit in the previous subsection was the assumption that if a coalition buys a new machine it would gain exclusive use of that machine. That does not have to be the case. We consider here the opposite assumption, in which new machines are available for all agents. To illustrate the differences between the two assumptions, suppose that we have a long queue of agents waiting to go through security/ticket control at a sporting event. If somehow agents waiting got hold of an additional employee who could, given appropriate compensation, open a new lane to speed up the process, who would have access to that lane? Up to now, we had supposed that this new lane would be a VIP lane, accessible only to agents who helped compensate this additional worker. But

another reasonable interpretation is that this new lane would be available to all, making this new machine a public good.

More precisely, if a new line opens up, the initial schedule σ_0 is split up in two: 1 and 2 are served first, 3 and 4 second, and 5 third and the new scheduling plan (m', σ') is

m_1	1	3	5
m_2	2	4	

Then, we can calculate the worth of the coalition $T = \{2, 4, 5\}$, as the waiting costs saved by its members only, net of the new machine cost. In other words, when we add machines a coalition receives the gains its members make in waiting costs, as the queue moves up, but must fully pay for the new machines.

To properly express how this requeueing occurs, we build from the initial schedule $\sigma_0 = (\varphi_0, s_0)$ a priority order π , which will allow us to determine, which agent moves up when new machines becomes available. Formally, for any $i, j \in N$

$$\pi(i) < \pi(j) \Leftrightarrow \{s_0(i) < s_0(j) \text{ or } \{s_0(i) = s_0(j) \text{ and } \varphi_0(i) < \varphi_0(j)\}\}.$$

In words, to rank agents we first look at the period in which they are served, and break ties by giving priority to agents served on machines identified with lower numbers.

Notice that in a public requeueing game a coalition T has much less ability to choose an alternative schedule. Once it has chosen a new number of machines, agents requeue automatically using the ordering π . Coalition T however can still reorder its members, under constraint. We assume that they can do so at two occasions, before and after adjusting the number of machines. We define as $\hat{\Sigma}^s$ and $\hat{\Sigma}^{ns}$ the admissible schedules under these constraints.

In a public requeueing game (public game for short), we suppose that a coalition T still can only sell a machine if only its members are using it, but we now further suppose that the revenues from the sale must be split equally among all agents in N , as the machines are public. Coalition T thus receives a fraction $\frac{|T|}{n}$ of the proceeds.

Let $\hat{V}_s(T, k)$ and $\hat{V}_{ns}(T, k)$ be the functions giving the value (possibly negative) that we obtain if we force coalition T to use k machines, in the public game with and without swaps, respectively. We then have that:

$$\hat{V}_s(T, k) = \begin{cases} \max_{\sigma \in \hat{\Sigma}_T^s(k)} (\sum_{i \in T} (s_0(i)) - s(i))w_i - (k - m_0)b & \text{if } k \geq m_0 \\ \max_{\sigma \in \hat{\Sigma}_T^s(k)} (\sum_{i \in T} (s_0(i)) - s(i))w_i - \frac{|T|}{n}(k - m_0)b & \text{if } k < m_0 \end{cases}$$

It is possible for $\hat{\Sigma}_T^s(k)$ to be empty if $k < m_0$, if T does not have exclusive use of $m_0 - k$ machines, in which case we simply let $\hat{V}_s(T, k) = 0$. We define $\hat{V}_{ns}(T, k)$ in the same manner.

We obtain the optimal cost savings for a coalition by maximizing over the number of machines: $\widehat{V}_s(T) \equiv \max_{k \in \{1, \dots, n\}} \widehat{V}_s(T, k)$ and $\widehat{V}_{ns}(T) \equiv \max_{k \in \{1, \dots, n\}} \widehat{V}_{ns}(T, k)$.

A *public requeueing problem* is a requeueing problem (N, m_0, σ_0, w, b) with an endogenous number of machines where the machines are public goods. If all machines are public goods, we call a requeueing game with an endogenous number of machines a *public requeueing game* denoted by (N, \widehat{V}_s) or (N, \widehat{V}_{ns}) , depending if we allow swaps or not.

We guarantee the non-emptiness of the core for public requeueing games when the initial queue is optimal.

Theorem 5. *Let (N, m_0, σ_0, w, b) be a public requeueing problem such that the ordering π induced by σ_0 is the optimal queue $\{1, 2, \dots, n\}$, and let (N, \widehat{V}_s) be the associated public requeueing game with swaps and (N, \widehat{V}_{ns}) be the associated public requeueing game without swaps. Then, $\text{Core}(\widehat{V}_s) \neq \emptyset$ and $\text{Core}(\widehat{V}_{ns}) \neq \emptyset$.*

Proof. Given that $\widehat{V}_{ns}(S) \leq \widehat{V}_s(S)$ for all $S \subset N$ and $\widehat{V}_{ns}(N) = \widehat{V}_s(N)$, it suffices to show that $\text{Core}(\widehat{V}_s) \neq \emptyset$. To ease on the notation in the proof, we use V for \widehat{V}_s .

Let $V(S, k)$ be the total gain (potentially negative) when we force coalition S to use k machines. If $k > m_0$, we have $V(S, k) = \sum_{l \in S} \left(\left\lceil \frac{l}{m_0} \right\rceil - \left\lceil \frac{l}{k} \right\rceil \right) w_l - b(k - m_0)$.

For $m_0 > k$, we have that $V(S, k) = \frac{|S|}{n} b(m_0 - k) - \sum_{l \in S} \left(\left\lceil \frac{l}{k} \right\rceil - \left\lceil \frac{l}{m_0} \right\rceil \right) w_l$ which is only feasible if coalition S contains all users of $m_0 - k$ machines. We consider the value function $\bar{V}(S, k)$ which allows S to sell any number of machines it wants and receive the full amount for it. That is, $\bar{V}(S, k) = b(m_0 - k) - \sum_{l \in S} \left(\left\lceil \frac{l}{k} \right\rceil - \left\lceil \frac{l}{m_0} \right\rceil \right) w_l$. On the other hand, for $k > m_0$, we have $\bar{V}(S, k) = V(S, k)$.

Notice that for all S and k , $\bar{V}(S, k) \geq V(S, k)$ and $\bar{V}(N, k) = V(N, k)$. Let \bar{V} be the maximum value function over the number of machines, i.e., $\bar{V}(S) := \max_{k \in \{1, \dots, n\}} \bar{V}(S, k)$. Then, $\bar{V}(S) \geq V(S)$ and $\bar{V}(N) = V(N)$. Let $\bar{m}(S)$ be the optimal number of machines for S .⁵ $\bar{m}(\cdot)$ has the same properties as $m(\cdot)$.

Let $S \subseteq N \setminus \{i, j\}$ and suppose that $i > j$. By Lemma 2 we have that $\bar{m}(S \cup \{i\}) \leq \bar{m}(S \cup \{j\})$.

By definition, $\bar{V}(S \cup \{i, j\}) \geq \bar{V}(S \cup \{i, j\}, \bar{m}(S \cup \{j\}))$, and hence

$$\begin{aligned} \bar{V}(S \cup \{i, j\}) &\geq \sum_{l \in S} \left(\left\lceil \frac{l}{m_0} \right\rceil - \left\lceil \frac{l}{\bar{m}(S \cup \{j\})} \right\rceil \right) w_l - b(\bar{m}(S \cup \{j\}) - m_0) \\ &\quad + \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\bar{m}(S \cup \{j\})} \right\rceil \right) w_i + \left(\left\lceil \frac{j}{m_0} \right\rceil - \left\lceil \frac{j}{\bar{m}(S \cup \{j\})} \right\rceil \right) w_j \\ &= \bar{V}(S \cup \{j\}) + \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\bar{m}(S \cup \{j\})} \right\rceil \right) w_i \end{aligned}$$

⁵There could be many, in which case we pick the lowest one.

and thus

$$(3) \quad \bar{V}(S \cup \{i, j\}) - \bar{V}(S \cup \{j\}) \geq \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\bar{m}(S \cup \{j\})} \right\rceil \right) w_i.$$

Similarly, we have that $\bar{V}(S) \geq \bar{V}(S, \bar{m}(S \cup \{i\}))$, and thus

$$\bar{V}(S) \geq \sum_{l \in S} \left(\left\lceil \frac{l}{m_0} \right\rceil - \left\lceil \frac{l}{\bar{m}(S \cup \{i\})} \right\rceil \right) w_l - b(\bar{m}(S \cup \{i\}) - m_0)$$

and

$$(4) \quad \bar{V}(S \cup \{i\}) - \bar{V}(S) \leq \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\bar{m}(S \cup \{i\})} \right\rceil \right) w_i.$$

Combining (3) and (4), we get

$$\begin{aligned} \bar{V}(S \cup \{i, j\}) - \bar{V}(S \cup \{j\}) &\geq \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\bar{m}(S \cup \{j\})} \right\rceil \right) w_i \\ &\geq \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\bar{m}(S \cup \{i\})} \right\rceil \right) w_i \\ &\geq \bar{V}(S \cup \{i\}) - \bar{V}(S), \end{aligned}$$

where the second inequality follows from $\bar{m}(S \cup \{i\}) \leq \bar{m}(S \cup \{j\})$, proving that \bar{V} is convex and hence $\text{Core}(\bar{V})$ is non-empty. Then, $\bar{V}(S) \geq V(S)$ and $\bar{V}(N) = V(N)$ imply that $\text{Core}(\bar{V}) \subseteq \text{Core}(V)$. Together with the non-emptiness of $\text{Core}(\bar{V})$, it implies that $\text{Core}(V)$ is also non-empty which completes the proof. \square

Interestingly, while the concept of public sequencing games leads naturally to the assumption that a coalition getting rid of machines would only collect part of the value of that machine, and would be severely limited in when they could sell a machine that is publicly owned, it turns out that these assumptions are not necessary to obtain core allocations. We always have core allocations, even if we eliminate all constraints on selling machines and allow coalitions to pocket to full value of the machines it sells. Even more, the game becomes convex with these assumptions.

The assumption that agents are ranked in an optimal way in the original schedule is crucial in our proof, allowing us to obtain the same structure for the optimal number of machines as when queueing without an initial allocation. For instance, if agent 1 has a particularly large waiting cost and is initially ranked last, when he is by himself he might prefer to buy many machines to be served earlier, while with other agents it might not be necessary, as switching spots with other members of the coalition might allow him to be served early without buying as many machines.

An important consequence of Theorem 5 is that whenever each agents own a machine at the initial schedule, then the core is always non-empty.

Corollary 1. *Given a public requeueing problem (N, m_0, σ_0, w, b) such that $m_0 = |N|$, the associated public requeueing game without swaps (N, \widehat{V}_{ns}) , and with swaps (N, \widehat{V}_s) , has a non-empty core.*

Theorem 5 establishes the non-emptiness of the core for public requeueing games under the assumption that the initial schedule is optimal. The next example shows that if we relax this assumption, the core of a public requeueing game can be empty.

Example 3. Consider (N, m_0, σ_0, w, b) with $N = \{1, 2, 3, 4\}$, $m_0 = 1$, and the ordering induced by σ_0 being $\pi = (4, 3, 2, 1)$. The waiting costs per unit for agents are $w = (w_i)_{i \in N} = (13, 7, 6, 1)$, and the cost of a machine is $b = 15$.

Notice first that the initial queue $(4, 3, 2, 1)$ is not optimal:

$$m_1 \begin{array}{|c|c|c|c|} \hline 4 & 3 & 2 & 1 \\ \hline \end{array}.$$

Suppose that we allow agents to jump over the agents not belonging to the coalition. Let (N, \widehat{V}_s) be the associated public requeueing game. Let us illustrate how to calculate the value of a coalition by doing it for $\{1, 4\}$. If agent 1 and agent 4 do not buy any machine, then they switch their positions:

$$m_1 \begin{array}{|c|c|c|c|} \hline 1 & 3 & 2 & 4 \\ \hline \end{array},$$

and hence the total savings for $\{1, 4\}$ is $(13 - 1) \times 3 = 36$.

If they buy a new machine, since it is a public requeueing game, the queue moves in a way such that agents 3, 4 are served first and agents 1, 2 are served second. Then, agents 1 and 4 change positions:

$$\begin{array}{|c|c|c|c|} \hline m_1 & 3 & 2 & \\ \hline m_2 & 1 & 4 & \\ \hline \end{array}$$

and hence the total savings for $\{1, 4\}$ is $13 \times 3 - 15 - 1 = 23$.

If they buy two new machines, the queue moves in a way such that agents 2, 3, 4 are served in the first position of a machine while agent 1 is served in the second position at a machine. Then, agents 1 and 4 switch positions:

$$\begin{array}{|c|c|c|c|} \hline m_1 & 1 & 4 & \\ \hline m_2 & 3 & & \\ \hline m_3 & 2 & & \\ \hline \end{array}$$

and hence the total savings for $\{1, 4\}$ is $13 \times 3 - 15 \times 2 - 1 = 8$. Finally, if they buy three machines, all agents' operations are processed at the first position on each machine:

$$\begin{array}{|c|c|c|c|} \hline m_1 & 1 & & \\ \hline m_2 & 2 & & \\ \hline m_3 & 3 & & \\ \hline m_4 & 4 & & \\ \hline \end{array}$$

and hence the total savings is $13 \times 3 - 15 \times 3 = -6$. Then, the maximum of possible total cost savings for $\{1, 4\}$ is achieved when they do not buy any new machine, $\widehat{V}_s(\{1, 4\}) = \max\{36, 23, 8, -6\} = 36$.

Consider now coalitions $\{1, 4\}$, $\{2, 4\}$, $\{3, 4\}$, $\{1, 2, 3\}$, and $\{1, 2, 3, 4\}$. Following our illustration, one can check that $\widehat{V}_s(\{1, 4\}) = 36$, $\widehat{V}_s(\{2, 4\}) = 12$, $\widehat{V}_s(\{3, 4\}) = 5$, $\widehat{V}_s(\{1, 2, 3\}) = 31$, and $\widehat{V}_s(N) = 37$. Now, suppose that there exists a core allocation $(y_1, y_2, y_3, y_4) \in \text{Core}(\widehat{V}_s)$. Then, it would satisfy the core constraints $y_1 + y_4 \geq 36$, $y_2 + y_4 \geq 12$, $y_3 + y_4 \geq 5$, $y_1 + y_2 + y_3 \geq 31$. Nevertheless, such a payoff vector (y_1, y_2, y_3, y_4) is not in the core since the core constraints $(y_1 + y_4) + (y_2 + y_4) + (y_3 + y_4) + 2(y_1 + y_2 + y_3) = 3(y_1 + y_2 + y_3 + y_4) \geq 115$ is not compatible with $3\widehat{V}_s(N) = 111$, and hence $\text{Core}(\widehat{V}_s) = \emptyset$.

5. Concluding remarks

This paper studies queueing problems from a game theoretical point of view. The novelty of this paper is that the number of machines is endogenous. For a given problem, agents are allowed to (de)activate as many machines they want, at a cost. We have distinguished two types of queueing problems: without and with an initial queue. For the first case, we have provided both a lower and an upper bound on the cost of machine to guarantee the non-emptiness of the core. Moreover, in some instances we have provided a full characterization of the core by means of concavity. For the second case, although we have shown that the core may be empty, we have guaranteed balancedness when all machines are accessible to all agents and the initial ordering correctly ranks agents in decreasing order of their waiting costs.

Compared to the earlier literature, our main innovations are (i) the existence of an endogenous number of machines at a given queueing problem, (ii) the cost associated with a machine to (de)activate it, (iii) the distinction between private and public queueing problems with an initial queue.

An interesting direction for future research is to study the existence of an allocation rule that always selects a stable allocation for balanced requeueing games. Furthermore, although we have a counterexample when swaps are allowed for public requeueing games with the non-optimal initial queue, it is still an open question whether it is also the case when swaps are not allowed.

References

- Atay, A., Calleja, P., and Soteras, S. (2021). Open shop scheduling games. *European Journal of Operational Research*, 295:12–21.
- Bahel, E. and Trudeau, C. (2019). Stability and fairness in the job scheduling problem. *Games and Economic Behavior*, 117:1–14.

- Calleja, P., Borm, P., Hamers, H., Klijn, F., and Slikker, M. (2002). On a new class of parallel sequencing situations and related games. *Annals of Operations Research*, 109:265–277.
- Chun, Y. (2006). No-envy in queueing problems. *Economic Theory*, 29:151–162.
- Chun, Y. (2016). *Fair queueing*. Springer.
- Chun, Y. and Heo, E. J. (2008). Queueing problems with two parallel servers. *International Journal of Economic Theory*, 4:299–315.
- Curiel, I., Pederzoli, G., and Tijs, S. (1989). Sequencing games. *European Journal of Operational Research*, 40:344–351.
- Curiel, I., Potters, J., Prasad, R., Tijs, S., and Veltman, B. (1993). Cooperation in one machine scheduling. *Zeitschrift für Operations Research*, 38:113–129.
- Curiel, I., Potters, J., Prasad, R., Tijs, S., and Veltman, B. (1994). Sequencing and cooperation. *Operations Research*, 42:566–568.
- Gillies, D. B. (1959). Solutions to general non-zero-sum games. In Tucker, A. W. and Luce, R. D., editors, *Contributions to the Theory of Games IV*, pages 47–85. Princeton University Press.
- Hamers, H., Klijn, F., and Suijs, J. (1999). On the balancedness of multiple machine sequencing games. *European Journal of Operational Research*, 119:678–691.
- Hamers, H., Klijn, F., and Van Velzen, B. (2005). On the convexity of precedence sequencing games. *Annals of Operations Research*, 137:161–175.
- Maniquet, F. (2003). A characterization of the shapley value in queueing problems. *Journal of Economic Theory*, 109:90–103.
- Musegaas, M., Borm, P., and Quant, M. (2015). Step out–Step in sequencing games. *European Journal of Operational Research*, 246:894–906.
- Musegaas, M., Borm, P., and Quant, M. (2018). On the convexity of step out–step in sequencing games. *TOP*, 26:68–109.
- Shapley, L. S. (1953). A value for n-person games. In W., K. H. and Tucker, A. W., editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press.
- Shapley, L. S. (1971). Cores of convex games. *International Journal of Game Theory*, 1:11–26.
- Slikker, M. (2006a). Balancedness of multiple machine sequencing games revisited. *European Journal of Operational Research*, 174:1944–1949.
- Slikker, M. (2006b). Relaxed sequencing games have a nonempty core. *Naval Research Logistics*, 53:235–242.
- Smith, W. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3:59–66.