Games and Economic Behavior 144 (2024) 104–125

Contents lists available at ScienceDirect

Games and Economic Behavior

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

Queueing games with an endogenous number of machines [☆]

Ata Atay [a], Christian Trudeau [b,*]

[a] Department of Mathematical Economics, Finance and Actuarial Sciences, and Barcelona Economic Analysis Team (BEAT), University of Barcelona, Spain
[b] Department of Economics, University of Windsor, Windsor, ON, Canada

ARTICLE INFO

JEL classification:
C44
C71
D61
D63

Keywords:
Queueing problems
Convexity
Cost sharing
Allocation problems

ABSTRACT

We study queueing problems with an endogenous number of machines, the novelty being that coalitions not only choose how to queue, but on how many machines. After minimizing the processing costs and machine costs, we share the proceeds of this cooperation, and study the existence of stable allocations. First, we study queueing problems, 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. Next, we study requeueing problems, 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 stable allocations may not exist, we guarantee their existence when all machines are considered public goods, and we start with an initial queue in which agents with larger waiting costs are processed first.

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 (requeueing) an existing queue optimally to minimize the total weighted makespan. That is, the sum of total waiting cost of agents together with the cost of machines. 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 (that is linear with respect to the moment it can leave the system). 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. It thus includes the trade-off that groups have between the cost of maintaining multiple machines and the savings of having their jobs processed faster on said machines. As an example, during the COVID pandemic, health authorities not only had to decide on the order of the queue for vaccines, but also on the speed of the vaccination operations. Similarly, research groups have to determine if they prefer to wait for access to highly-

☆ Ata Atay is a Serra Húnter Fellow (Professor Lector Serra Húnter). Ata Atay gratefully acknowledges the support from the Spanish Ministerio de Ciencia e Innovación research grant PID2020-113110GB-100/AEI/10.13039/501100011033, from the Generalitat de Catalunya research grant 2021-SGR-00306 and from the University of Barcelona research grant AS017672. Christian Trudeau gratefully acknowledges financial support by the Social Sciences and Humanities Research Council of Canada [grant number 435-2019-0141]. We thank two anonymous referees and an associate editor; Youngsub Chun, Herbert Hamers, Marina Núñez, Zhihao Wang, Nathalie Van Raemdonck, and participants at the 12th Conference on Economic Design.

* Corresponding author.
E-mail addresses: aatay@ub.edu (A. Atay), trudeauc@uwindsor.ca (C. Trudeau).

https://doi.org/10.1016/j.geb.2024.01.005
Received 2 December 2022
Available online 18 January 2024

specialized equipment or to buy new equipment for faster access. The concept of an endogenous number of machines is particularly relevant when studying, as we do, the problem using cooperative game theory; the concept of core stability now implies that when a coalition threatens to leave the group, it would do so by paying for the number of machines that minimizes its own cost.

Queueing problems have been widely studied from a normative and strategic point of view (see for instance Kayı and Ramaekers, 2010; Mitra and Mutuswami, 2011; Hashimoto and Saitoh, 2012; Chun and Yengin, 2017; Chun et al., 2019). In this approach, the main aim is to identify direct mechanisms that satisfy the desiderata. A second approach, in line with this paper, is applying game theoretical tools to queueing problems. In this case, a TU-game is associated to a given queuing problem. Cooperative game theory tools are then used to share the common queueing costs.

Maniquet (2003) studies the one machine queueing problem and the associated TU-game is obtained by defining the worth of a coalition as the minimum waiting cost incurred by its members when they are served before the non-coalitional members. This is dubbed the optimistic approach. Then, he takes a normative approach. He introduces the minimal transfer rule which assigns positions in the queue and a compensation. The compensation is equal to half of their unit waiting cost multiplied by the number of agents in front of them in the queue subtracted by half of the sum of the unit waiting costs of the people behind them in the queue. He shows that the Shapley value (Shapley, 1953) of the (optimistic) associated TU-game coincides with the minimal transfer rule.

By contrast, Chun (2006) adopts a pessimistic approach to the queueing problem. The associated TU-game is obtained by defining the worth of a coalition as the minimum waiting cost incurred by its members when they are served after the non-coalitional members. He defines the maximal transfer rule that assigns a position in the queue and a compensation to each agent. The compensation is equal to a half of the sum of the unit waiting costs of her predecessors minus a half of her unit waiting cost multiplied by the number of her followers. He shows that the Shapley value (Shapley, 1953) of the (pessimistic) associated TU-game coincides with the maximal transfer rule.

Towards a generalization to multiple machines, Mitra (2005) and Mukherjee (2013) study fairness properties. Mitra and Mutuswami (2021) focuses on the no-envy property when there are more than one queue to serve agents.

Chun and Heo (2008) investigates queueing problems with two parallel machines from an optimistic and a pessimistic approach. They generalize both the maximal and the minimal transfer rules to the case with two machines serving agents. Han and Chun (2022) generalizes this result to an arbitrary number of machines.

The need to consider pessimistic and optimistic approaches comes from the fact that agents impose negative externalities on each other when there are a fixed number of machines: for distinct groups S and T, the waiting cost of $S \cup T$ is weakly larger than the sum of waiting costs for S and T. We thus need to make assumptions on the behavior outside of a given coalition. In our case with an endogenous number of machines, this is not true anymore. S and T can duplicate their individual costs by buying enough machines to act independently of each other. If any other arrangement is taken by $S \cup T$, it must be because it is less costly. Thus, our coalitional cost game is subadditive. This generates a more traditional interpretation of our coalitional cost function, which represents how much a coalition would have to pay if it stood alone. The concept of the core can therefore be applied, and it is desirable to propose stable allocations in which no coalition pay more than this stand-alone cost.

There is a scarce literature that studies the existence of stable allocations for queueing problems. González and Herrero (2004) studies the existence of stable allocations for Markov queueing models with one single machine. García-Sanz et al. (2008), Özen et al. (2011), Zeng et al. (2018) provide stable allocation rules. However, in these cases, the number of machines is fixed.

Starting with the problem of optimally queueing players (before they arrive) on an endogenous number of machines, we identify the structure of the core whenever the optimal number of machines is at least half of the number of agents (Theorem 1). We then proceed to present a comprehensive description of the core when the optimal number of machines is equal to 1 (Theorem 2). Then, we illustrate the gap between Theorems 1, 2 by Example 4.

We then examine the problem when we start with an existing queue/number of machines. In this problem, that we call the requeueing problem, we assign to each coalition the cost savings it can generate, either by changing the order of its members in the existing queue, or by purchasing new machines or selling some existing ones. Many assumptions can be made on what a coalition is allowed to do in this situation, and what happens when the number of machines is changed.

First, what happens if a coalition S buys new machines? Two possibilities seem reasonable: i) S gains exclusive access to these machines, or ii) the whole queue moves up, benefiting not only S. Given the link with private and public goods (see for instance Suijs, 1996; De and Mitra, 2017; De and Mitra, 2019), we call these approaches respectively the private and public approaches. In the second case, the decision of a coalition S to add a new machine is similar to the decision of a group to make a donation towards a public good that maximizes the benefit of the coalition.

As an example of the public approach, take for instance internet access around the globe. The flow of internet traffic crosses oceans primarily through submarine communication cables. Internet Service Providers (ISPs) provide connectivity to the internet. Consortiums formed by ISPs from countries with landing points build different segments of these cables. The contributing ISPs are responsible for paying for the submarine telecommunications cable. Whenever a new cable is constructed, other regions can also use it for internet traffic. Essentially, the reason to have new routes (cables) is for not making the users queue on existing ones. For example, SEA-ME-WE4 has 17 landing points from France to Singapore. It was constructed as a complement to SEA-ME-WE3 (which has 39 landing points from Germany to Australia). The ISP providers from 17 landing points on SEA-ME-WE4 share the costs, while any data transfer from other regions (e.g. data transfer from the United States to New Zealand) can also use SEA-ME-WE4.

After briefly discussing the private requeueing game, we focus on the public version. In the spirit of machines being public goods, we also suppose that if a coalition sells a machine, the proceeds are equally shared among all agents, including non-coalitional members. We complete the set of assumptions by supposing that a coalition i) can reposition any agent, even non-coalitional members, if it offers to compensate them for the extra waiting cost if their job is processed later, and ii) can sell any machine, as long, once again, that non-coalitional members are properly compensated if they wait longer. We call this game the public requeueing game with side-payments, and show that if the initial queue efficiently ranks the agents but on a possibly non-optimal number of machines, then the resulting game is always convex, and thus its core is non-empty (Theorem 3).

The result follows 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) that examine conditions guaranteeing the existence of stable allocations.

While the literature has made many additional restrictions on what constitutes an eligible requeue, and while we can add additional constraints on when a coalition can sell a machine, these restrictions will simply (weakly) reduce the value that a subset of agents can obtain, without changing the value obtainable by the grand coalition. Thus, adding restrictions (weakly) enlarges the core and its non-emptiness carries regardless of what is an acceptable rearrangement of a queue (Corollary 1).

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 requeueing problems (and games), and mostly focus on the approach where added machines are public goods, and where a coalition can move non-coalitional members as desired, as long as it properly compensates them. We discuss the non-emptiness of the core, before extending to games with stricter constraints. Finally, we draw conclusions in Section 5. We consign formal proofs in Appendix A.

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 with unit processing time 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 suppose that $w_1 \geq w_2 \geq \dots \geq w_n$. We refer to the vector of weights by $w := (w_i)_{i \in N}$. Let a subset of agents $S \subseteq N$ be ordered according to the set $N = \{1, 2, \dots, n\}$, $\rho : S \to \{1, \dots, |S|\}$ such that $\rho(i) < \rho(j)$ if $i < j$. Then, we denote the waiting cost of the k^{th} agent in S by w^S_k and $w^{-S}_k \equiv w^{N \setminus S}_k$ be the waiting cost of the k^{th} agent outside the coalition S. In words, if we partition N in S and $N \setminus S$, and rank agents in each element in decreasing order of their waiting cost, w^S_k and w^{-S}_k are the k^{th} largest waiting costs in respectively S and $N \setminus S$.

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.

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 $m \in \{1, \dots, n\}$, the assignment of agents to machines $\varphi : N \to \{1, \dots, m\}$ and the starting time of all agents $s : N \to \mathbb{N} \cup \{0\}$. Given m, a queue $\sigma = (\varphi, s)$ 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 admissible queues with m machines is denoted by $\Sigma(m)$. A queue system is (m, σ), with $\sigma \in \Sigma(m)$.

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; Curiel 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. $w_i < w_j \Rightarrow s(i) \geq s(j)$.

Given this result, if we install m machines, it is optimal to queue 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 queued in the next period, and so on. We call such orderings efficient orderings. Thus, the queueing problem reduces to finding the number of machines that solves[1]

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

[1] For all $x \in \mathbb{R}$, $\lceil x \rceil := \min\{k \in \mathbb{Z} | x \leq k\}$ while $\lfloor x \rfloor := \max\{k \in \mathbb{Z} | x \geq k\}$.

The problem of finding the optimal number of machines is simple. If the cost of a machine is very high, we buy a single one. As b, the cost of a machine, decreases we buy more machines, up to the point where all agents have their own machines, and there is no more gain to add additional machines. More precisely, let $r^w(k)$ be the critical value (for coalition N) to buy k machines: if b is weakly above that critical value, we buy less than k machines. If it is less than that critical value, we buy at least k machines. Let $m(S)$ be the optimal number of machines for coalition $S \in N$.[2]

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

In what follows, we use Lemma 1 to quickly identify how many machines are bought. In particular, the cases in which a single machine or n machines are bought are studied.

We illustrate the concepts by an example. Suppose that we have 3 agents, with $w = (15, 10, 5)$. We only need to check efficient orderings for different number of machines. If we buy a single machine, agent 1 waits 1 period, agent 2 waits 2 periods, and agent 3 waits 3 periods, for a total cost of $b + 15 + 2 \times 10 + 3 \times 5 = b + 50$. If we buy 2 machines, agent 2 now waits a single period, and agent 3 for 2 periods, for a cost of $2b + 15 + 10 + 2 \times 5 = 2b + 35$. Finally, if we buy 3 machines, all agents wait a single period, for a cost of $3b + 15 + 10 + 5 = 3b + 30$. Thus, we have that $r^w(2) = 15$ and $r^w(3) = 5$. Also notice that if $b = 8$, the grand coalition buys 2 machines and so does coalition $\{1, 2\}$. However, coalition $\{1, 3\}$ prefers to buy a single machine. It is a general result that adding a player to a coalition (moving here from $\{1, 2\}$ or $\{1, 3\}$ to the grand coalition) or replacing an agent with a low waiting cost to one with a large one (moving here from $\{1, 3\}$ to $\{1, 2\}$) cannot result in less machines being bought.

We can similarly define a non-increasing function $r^w_S : \{2, \dots, |S|\} \to \mathbb{R}_+$ 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 $i \in N$.

The following lemma allows to formalize some intuitions about the optimal number of machines. If a coalition grows, then the optimal number of machines cannot decrease, as the marginal benefit (decrease in waiting costs, net of the machine cost) of the k^{th} machine cannot decrease. In a similar manner, if we replace an agent with a low waiting cost by one with a high waiting cost, it cannot be optimal to use fewer machines.

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 $i > j$.

Even though the problem to determine the optimal number of machines is simple and well-behaved, the resulting cooperative game does not behave as well, in particular when studied under the lens of the (non-)vacuity of the core.

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 characteristic function C assigns to each coalition $T \subseteq N$ its cost $C(T) \in \mathbb{R}$, with $C(\emptyset) = 0$. Thus, $C(T)$ is the stand-alone cost of a coalition T and is calculated before the queue starts. Hence, when calculating $C(T)$, we suppose that T selects the optimal number of machines to serve only its own members.

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:

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

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

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

Concave TU-games 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 the core of queueing games with an endogenous number of machines

We look for conditions under which the core is empty or non-empty. It turns out that for queueing games with an endogenous number of machines, the core can alternate between being empty and non-empty depending on the cost of a machine. Two particular cases are easier to analyze. If the cost of a machine is sufficiently small so that $m(N) \geq \frac{n}{2}$, then agents wait at most 2 periods for their job to be processed. If the cost of a machine is sufficiently large, then $m(N) = 1$ and the dynamic is simpler to analyze as all coalitions use the same number of machines.

We first examine the cases when the cost of a machine is low enough for $m(N) \geq \frac{n}{2}$, generating Theorem 1, before examining the case when the cost is high enough for $m(N) = 1$, obtaining Theorem 2. We conclude the section with an example illustrating Theorems 1 and 2, how the core varies with the cost of machines, including in the gap not covered in Theorems 1 and 2.

For the sake of comprehensiveness, let us introduce some notation: Let $\mu \equiv \left\lceil \frac{n}{2} \right\rceil$. If n is even, then $\{1, \ldots, \mu\}$ and $\{\mu + 1, \ldots, n\}$ both contain μ agents, while if n is odd, then $\{1, \ldots, \mu\}$ contains μ agents and $\{\mu + 1, \ldots, n\}$ contains μ-1 agents. We start with a Lemma describing when the grand coalition buys at least μ machines.

Lemma 3. *Let (N, w, b) be a queueing problem with an endogenous number of machines. We have that $m(N) \geq \mu$ if and only if one of the following conditions is satisfied:*

(i) n is odd and $b \leq w_\mu + w_n$,
(ii) $n = 2$,
(iii) $n = 4$ and $b \leq w_2 + w_3 + 2w_4$,
(iv) n is even, $n \geq 6$ and $b \leq w_\mu + w_{n-1} + w_n$.

When $m(N) \geq \mu$ we can partition agents into at most 2 groups: the set $N_1 = \{1, \ldots, m(N)\}$ who are served in the first period, and the set $N_2 = \{m(N) + 1, \ldots, n\}$ who are served in the second period. The game is then very much similar to an assignment game (Shapley and Shubik, 1971), in which we must match agents from different sides of the market, here agents in N_1 to agents in N_2. Our game has the characteristic that an agent $i \in N_2$ creates $b - w_i \geq 0$ with any agent in N_1.[3] This is a characteristic of a particular case of assignment games, namely Böhm-Bawerk horse market (Böhm-Bawerk, 1923) games. For instance, if $N_1 = \{1, 2\}$, $N_2 = \{3, 4\}$ and $b > w_3 \geq w_4$, then agent 3 creates the value $b - w_3$ with an agent in N_1, whereas agent 4 creates $b - w_4$.

If all agents from one side, say N_1, create the same value with agents from N_2, then our assignment game is a glove market game (Shapley, 1959). In this case, if $|N_1| > |N_2|$, then the only candidate for a core allocation is for the short side of the market to extract all of the cooperation surplus. In our case, this corresponds to an allocation where agents in N_1 pay their stand-alone cost of $b + w_i$ while agents in N_2 pay $2w_i$, the cost to wait for 2 periods, which is no larger than their stand-alone cost of $b + w_i$. While in assignment games a coalition of agents from the same side never generates any cooperation gain, this is not necessarily true in our case. For instance, a coalition of agents in $S \subset N_1$ would generate no benefit if and only if it would be optimal for them to buy $|S|$ machines. If b is large enough, this is not the case, and the core is empty. The game behaves in a slightly different manner depending if the number of agents is odd or even, so we illustrate with the following examples.

Example 1. Suppose first that we have three agents, with $w = (15, 10, 5)$. We buy at least 2 machines if $b \leq 15$. If $b \leq 5$, then any coalition S buys $|S|$ machines, and $\left\{\left(b + w_i\right)_{i \in N}\right\}$ is the only core allocation. If $5 < b \leq 10$, then the grand coalition buys two machines and coalitions $\{1, 3\}$ and $\{2, 3\}$ buy a single one. It is easy to check that the allocation $(b + 15, b + 10, 10)$ is the only core allocation. Notice that coalition $\{1, 2\}$ still buys two machines, meaning that members of N_1 have no benefit to cooperate. However, if $10 < b \leq 15$, coalition $\{1, 2\}$ buys a single machine, and our only core candidate, $(b + 15, b + 10, 10)$, is no longer in the core.

Now add a fourth agent with a valuation of 20, so that $w' = (20, 15, 10, 5)$. We buy at least 3 machines if $b \leq 10$. Because the condition for coalition $N_1 = \{1, 2, 3\}$ to buy 3 machines is also $b \leq 10$, then agents in N_1 never generate any benefit when cooperating among themselves, and $\left(\min\left(b + w_i, 2w_i\right)\right)_{i \in N}$ is the only core allocation. Notice that agent 3 together with agent 1 or agent 2, can create the value $b - 10$ whereas agent 4 together with either of these agents can create the value $b - 5$. Hence, our model bear a resemblance to Böhm-Bawerk horse markets.

Parts (i) and (ii) of Theorem 1 generalize the results of Example 1.

It is already known from Shapley and Shubik (1971) that the core of the Böhm-Bawerk horse market game consists of a line segment, with extreme points being two side-optimal core allocations. In glove market games, we also know that if both sides of the market are of the same size (which is only possible if n is even) there are multiple core allocations. For instance, we can assign all cooperation gains to one side of the market or the other, or take any convex combination of these side-specific optimal allocations.

[3] The value created is the cost savings compared to the sum of their stand-alone costs. Separately, they each pay for a machine, and wait for a single period. Together, they pay for a single machine, the agent in N_1 waits for a period and the agent in N_2 waits for two periods.

Once again, this result depends crucially on the fact that agents on the same side of the market do not generate any cooperation gain, which is not necessarily true in our setting.

Example 2. Reconsider the above example with 4 players, and suppose that $10 < b \leq 35$. Then $m(N) = 2$, $N_1 = \{1,2\}$ and $N_2 = \{3,4\}$. As long as $b \leq 15$, coalition $N_1 = \{1,2\}$ still buys 2 machines, and have no cooperation gain. Then, the allocation $\left\{\left(\min\left(b + w_i, 2w_i\right)\right)_{i\in N}\right\}$ is in the core, which also includes allocations more beneficial to N_1. When $b > 15$, coalition $N_1 = \{1,2\}$ now buys a single machine and generates cooperation gains. The allocation $\left(\min\left(b + w_i, 2w_i\right)\right)_{i\in N}$ is no longer in the core. To remain in the core, we must reduce the allocations of agents in N_1, which also require increasing the allocations of agents in N_2. For b not too large, it is feasible to construct such a core allocation, but when $b > 25$, all coalitions except N use a single machine, and it's impossible to find a core allocation anymore.

Parts (iii) and (iv) of Theorem 1 formalize these arguments in Example 2. All together, Theorem 1 covers all cases in which $m(N) \geq \frac{n}{2}$. Recall that Lemma 3 provides the upper bound on b, as a function of n for which this is verified. Notice that in the statement of the theorem we use $\left\lfloor \frac{n}{2} \right\rfloor$. Observe that if n is even, then $\mu = \left\lceil \frac{n}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor$, but if n is odd, $\mu = \left\lceil \frac{n}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor + 1$.[4] The proof of the Theorem is in Appendix A.

Theorem 1. *Let (N, w, b) be a queueing problem with an endogenous number of machines, (N, C) be the associated TU-game and suppose that $m(N) \geq \mu = \left\lceil \frac{n}{2} \right\rceil$. Then we have the following:*

(i) If $b \leq w_{\left\lfloor \frac{n}{2} \right\rfloor + 1}$, then $Core(C) = \left\{\left(\min\left(b + w_i, 2w_i\right)\right)_{i\in N}\right\}$.
(ii) If n is odd and $b > w_{\left\lfloor \frac{n}{2} \right\rfloor + 1}$, then $Core(C)$ is empty.
(iii) if n is even and $b \in \left(w_{\left\lfloor \frac{n}{2} \right\rfloor + 1}, w_{\left\lfloor \frac{n}{2} \right\rfloor} + 2w_n\right]$, then $Core(C)$ is non-empty.
(iv) if n is even and $b > w_{\left\lfloor \frac{n}{2} \right\rfloor} + 2w_n$, then $Core(C)$ is empty.

Having fully studied the case where $m(N) \geq \mu$, we now provide a full characterization of the core when $m(N) = 1$. To do so, we use a technical approach and introduce another game $(N, \hat{C})$ related to the cost game (N, C). Before proving the results, let us provide an interpretation for the game $\hat{C}$.

When the cost of a machine is large enough that all coalitions use a single machine, it is still not enough to guarantee that the core is empty. We in fact show that as the cost of a machine goes through the threshold for all coalitions to use a single machine, the core is empty. As the cost of a machine keeps increasing, we reach another threshold for which the core becomes non-empty. In that case, we are able to provide a full description of the core. We do so by observing that there are more strict core constraints than the stand-alone costs. We define these constraints in a function $\hat{C}$ that has the same core as our coalitional game and that is a one-bound core game (Gong et al. (2023)). We illustrate this function in the following example.

Example 3. Reconsider the 4-player example from above, such that $w = (20, 15, 10, 5)$. Suppose that $b = 60$. It is easy to verify that all coalitions use a single machine. Table 1 describes the core constraints for characteristic function C and the corresponding function $\hat{C}$.

Consider coalition $\{3,4\}$. While we have $C(\{3,4\}) = 80$, we have the core constraints $y(\{1,3,4\}) \leq 115$ and $y(\{2,3,4\}) \leq 110$. Adding these up, we obtain $y(N) + y(\{3,4\}) \leq 225$. Since $y(N) = 160$, this simplifies to $y(\{3,4\}) \leq 65$. Where does this new upper bound on the shares of agents 3 and 4 come from? If we add agent 1 to $\{3,4\}$, there is an additional cost of $w_1 + w_3 + w_4 = 35$. If we add agent 2 to $\{3,4\}$, there is an additional cost of $w_2 + w_3 + w_4 = 30$. But if we add $\{1,2\}$ to $\{3,4\}$, the additional cost is larger than the sum of these costs:

$$w_1 + 2w_2 + 2w_3 + 2w_4 = 80 > 65 = w_1 + w_2 + 2w_3 + 2w_4.$$

It is by this difference of w_2 that the upper bound for $\{3,4\}$ can be reduced. The same logic can be applied to all pairs $\{i,j\}$, reducing the upper bound by the smallest waiting cost in $N \setminus \{i,j\}$.

Consider now singleton $\{4\}$. We now know that the upper bound for coalition $\{1,4\}$ is 80. For $\{2,3,4\}$ it remains at 110. The corresponding core constraints are thus $y(\{1,4\}) \leq 80$ and $y(\{2,3,4\}) \leq 110$. Adding them up and simplifying as above, we obtain $y_4 \leq 30$. Once again, the additional cost of adding agent 1 (considering the new upper bound for $\{1,4\}$) is $w_1 + w_4 - w_3 = 15$, while the additional cost of adding agents 2 and 3 is $w_2 + 2w_3 + 2w_4 = 45$. But if we add $\{1,2,3\}$ to $\{4\}$, the additional cost is larger than the sum of these costs:

$$w_1 + 2w_2 + 3w_3 + 3w_4 = 95 > 60 = w_1 + w_2 + w_3 + 3w_4.$$

[4] We thank an anonymous referee for suggesting to use both floor and ceiling functions.

Table 1
Core constraints for characteristic function C and the corresponding function $\hat{C}$.

y_1	y_2	y_3	y_4	C	$\hat{C}$
1	.	.	.	80	60
	1	.	.	75	55
.	.	1	.	70	45
.	.	.	1	65	30
1	1	.	.	110	105
1	.	1	.	100	95
1	.	.	1	90	80
.	1	1	.	95	90
.	1	.	1	85	75
.	.	1	1	80	65
1	1	1	.	140	140
1	1	.	1	125	125
1	.	1	1	115	115
.	1	1	1	110	110
1	1	1	1	160	160

It is by this difference of $w_2 + 2w_3$ that the upper bound can be reduced. The same logic can be applied for all agent i, with the upper bound reduced by one time the second smallest waiting cost and two times the smallest waiting cost of agents in $N \setminus \{i\}$.

The resulting upper bounds are represented as function $\hat{C}$ in the table. Notice that for any agent, there are only two possible incremental costs: a large one when he joins the empty set, and a smaller one when he joins any non-empty set. Moreover, for all agents the difference between the large and small incremental costs is the same. In our example it is 10. This naturally leads us to conclude that agents must pay at least their small incremental costs, which are respectively 50, 45, 35 and 20. See that it allows to share 150 out of the cost of 160. We have 10 left to distribute, which is the difference between the large and small incremental costs. We can distribute that 10 in any (non-negative) way in the core. We show in part i) of the theorem below that it is no accident and holds in general, if b is large enough.

Suppose now that $b = 40$. Then, we can calculate $\hat{C}$ in the same manner: for all coalitions the cost is reduced by 20. It is then easy to check that $\hat{C}$ is no longer subadditive, and its core is empty. It then follows that the core of C is empty.

Intuitively, when b is large enough for all coalitions to use a single machine but not large enough for $\hat{C}$ to be subadditive, the gains obtained from saving the cost of additional machines are not large enough to compensate for the cost generated by the additional congestion on the unique machine, resulting in an empty core.

In what follows we formalize the findings of Example 3.[5] We need the following notation. For $x \in \mathbb{R}_+$, let $\Delta(N, x)$ be the set of vectors $y \in \mathbb{R}^N_+$ such that $y(N) = x$. For all $k \in N$, let $A_k = \sum\limits_{i>k} w_i + k w_k$. Let $A = (A_1, \ldots, A_n)$. We can see A_k as the incremental cost when k joins the grand coalition: given that a single machine is used, k has to wait k periods to be served, and it delays all agents with higher rank (and thus no larger waiting costs) by one period. The necessary condition for $Core(\hat{C})$ to be non-empty is $b \geq \sum\limits_{i=1}^{n} (i-1) w_i$ and can be interpreted as the benefit of staying with one machine if the alternative was to buy a (hypothetical) machine, at cost b, that can serve $n-1$ agents concurrently. The core then consists in each agent paying his A_k, and any division of the surplus $b - \sum\limits_{i=1}^{n} (i-1) w_i$.

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

The formal proof is consigned in Appendix A. While Theorem 1 covers the cases in which $m(N) \geq \mu$, Theorem 2 covers instances where $m(N) = 1$. Notice that if $n \leq 4$ all cases are covered. However, for $n > 4$, a gap exists, for cases such that $1 < m(N) < \mu$, for which we have no results. In such cases, there is at least one agent that waits three periods or more to have his job processed. We conclude this section by illustrating with an example, showing that the core is sometimes empty, sometimes not, in that interval.

[5] The cost game $\hat{C}$ can be rewritten as a value game $\hat{V} = (|S| - 1)(b - \sum_{i=1}^{n} (i-1) w_i)$. This game is an upper-bound core game (Gong et al., 2023), and hence a 1-convex game (Driessen, 1985). Moreover, Dehez (2021) remarks that a cost game is 1-concave if and only if the associated value game is 1-convex.

Example 4. We extend the above examples by adding a fifth agent with a valuation of 20, so that $w = (20, 20, 15, 10, 5)$. Theorem 1i) tells us that if $b \leq 15$, then $(b + 20, b + 20, b + 15, 20, 10)$ is the only core allocation. Theorem 1ii) tells us that the core is empty for $b \in]15, 20]$. Theorem 2ii) tells us that if $b \in [65, 100[$, the core is also empty. Finally, Theorem 2i) tells us that for $b \geq 100$ the core is $(70, 70, 60, 45, 25) + \Delta(N, b - 100)$.

It remains to check for $b \in (20, 65)$, over which $m(N) = 2$, meaning that $C(N) = 2b + 105$. Suppose first that $b \in (20, 25)$. Then, coalition $\{2, 3, 4, 5\}$ uses 2 machines, at a cost of $2b + 60$, which implies that we must have $y_1 \geq 45$. Since $y_1 \leq C(\{1\}) = b + 20$, we have that the core is empty if $b < 25$.

Suppose next that $b \in [25, 35]$. One can verify that the allocation $\left(\frac{b}{2} + 27.5, \frac{b}{2} + 27.5, \frac{b}{2} + 22.5, \frac{b}{2} + 12.5, 15\right)$ is in the core. For $b \in [35, 40]$, the allocation $\left(\frac{b}{2} + 30, \frac{b}{2} + 30, \frac{b}{2} + 20, \frac{b}{2} + 10, 15\right)$ is in the core. Thus, the core is non-empty for $b \in [25, 40]$.

Finally, for $b \in (40, 65)$, notice first that we have $C(\{1, 4\}) + C(\{2, 3, 5\}) = C(N)$, so we must have $y(\{1, 4\})) = C(\{1, 4\}) = b + 40$ in all core allocations. Since $C(\{1, 3, 4\}) = b + 80$, it implies that we must have $y_3 \leq 40$. But, since $C(N) = 2b + 105$ and $C(\{1, 2, 4, 5\}) = b + 105$, it implies that we must have $y_3 \geq b$. But since $b > 40$, we have no core allocations.

Overall, Example 4 shows that in the gap between Theorems 1 and 2, in which $1 < m(N) < \mu$, we typically have ranges of b for which the core is empty and some others where the core is non-empty. Given that in these cases the resulting cost game does not have as nice of a structure as in Theorems 1 and 2, we are unable to provide general conditions for the (non-) vacuity of the core.

4. Requeueing games with an endogenous number of machines

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 rearrange a given queue. In our study of the problem with an endogenous number of machines, this implies that we start with a given number of machines, and that the reorganization can include adding or removing machines.

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 queue system that minimizes the total costs, as in Section 2. As for queueing games, we build a characteristic function for 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 system. As in the previous section we are interested in core allocations; as we have moved from a cost to a value function, core inequalities are reversed.

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

In the private case, 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.

Two intuitive bounds for a non-empty core occur. First, setting the machine cost so high makes every coalition to prefer not buying a new machine. If there is only one initial machine, which cannot be sold, we are left with the problem of reorganizing the queue on the existing machine, making the problem equivalent to one with a single machine and no possibility to add more. This problem has been studied extensively since Curiel et al. (1989). For this case, it is proved that the core is always non-empty. Second, if the cost of a machine is so low that every coalition wants to buy a machine for each of its members, then the only core allocation consists of allocating to each agent the net benefit of moving to a new machine if that agent is not already served in the first period.

4.2. Public requeueing games with side payments

Given the lack of general results in between these trivial cases, we focus on the case where machines are “public”, and for which we are able to offer more results. To illustrate, imagine a large employer in a small, isolated community. If it makes a donation to the local health system to buy medical equipment, the extra equipment is a public good that can be used by the whole community. However, the waiting cost of the donor’s employees to receive treatment will be reduced, yielding a private gain for the employer that might make the donation worthwhile even from a selfish perspective.

We illustrate this by an example.

Example 5. 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	1	2	3	4	5

.

Notice that agent 1 is a dummy player since she is served first, moving to another machine is strictly worse for her. If a new line opens up, the initial queue σ_0 is split up in two: 1 and 2 are served first, 3 and 4 second, and 5 third and the new queue system (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 queue $\sigma_0 = (\varphi_0, s_0)$ a priority order π, which allows us to determine which agent moves up when new machines become available. Formally, for any $i, j \in N$

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

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. Note that adding machines will lead to efficient orderings if and only if the order π is $(1, 2, 3, \dots, n)$.

Notice that in a requeueing game a coalition T has much less ability to choose an alternative queue system. Once it has chosen a new number of machines, agents requeue automatically using the ordering π. In fact, conceptually, while the private game supposes the existence of separate queues at each machine, the public game makes the implicit assumption of a single queue, with agents simply going to the first free available machine when their turn comes up.

For a fixed number of machines, requeueing problems have considered different assumptions on what is an acceptable requeueing, for instance limiting the possibility for a coalition S to move its members in front of agents in $N \setminus S$. See Curiel et al. (1993) for a discussion of the various alternatives. This analysis becomes more complicated with an endogenous number of machines as the queues - and thus one's neighbors - change with the number of machines.

We initially consider a simple variant, that we call the *public game with side payments*: a coalition S can change the queue as desired, even moving down agents in $N \setminus S$, as long as they provide them with side payments that cover their additional waiting costs. This approach has two advantages. First, it is easily tractable. In particular, if the initial queue is an efficient ordering, then we are able to describe the requeueing decisions, which have a structure comparable to the queueing problems of the previous section. In particular, we are able to relate the decisions of S and T, if $S \subset T$, which allows us to show that the resulting game is convex, and thus has a non-empty core. Secondly, since the assumption of side payments is the least constraining, any game with stricter constraints will generate less value for all coalitions and the same value for the grand coalition, and thus a core allocation for the game with side payments is also a core allocation for these games with stricter constraints.

As for selling some of the initial machines, we make two assumptions in the public game with side payments. First, we suppose that a coalition S can sell as many machines as it wants (one machine must always be kept) and requeue agents as desired on the remaining machines, as long as it fairly compensates agents in $N \setminus S$ if they have to wait a longer time. Second, we suppose, since these machines are public goods, that the revenues from the sale of machines must be split equally among all agents in N. Coalition T thus receives a fraction $\frac{|T|}{n}$ of the proceeds.

Let $\hat{V}_{sp}(T, k)$ be the function giving the value (possibly negative) that we obtain if we force coalition T to use k machines in the public game with side payments. We then have that $\hat{V}_{sp}(T) = \max_{k=1,\dots,n} \hat{V}_{sp}(T, k)$.

Let $\hat{m}$ be the function assigning to each coalition the optimal number of machines to use in the public game with side-payments.[6] We offer results on the structure of $\hat{m}$ when we start with an efficient initial ordering.

Lemma 4. *Let (N, m_0, σ_0, w, b) be a public requeueing problem such that σ_0 is an efficient ordering. Then, we have:*
i) for all $S, T \subseteq N$, $(\hat{m}(S) - m_0)(\hat{m}(T) - m_0) \geq 0$.
ii) if $S \subset T \subseteq N$, then $\left|\hat{m}(S) - m_0\right| \leq \left|\hat{m}(T) - m_0\right|$.

In words, part i) confirms that we cannot have some coalition buying machines while others sell machines. Either all coalitions buy machines (or stay put) or all coalitions sell machines (or stay put). Part ii) says that if S is a subset of T, T will make at least as many transactions as S: if S buys some machines, T will buy at least as many, and if S sells some machines, T will sell at least as many.

This structure allows us to guarantee the non-emptiness of the core for public requeueing games with side-payments when the initial queue is an efficient ordering. We show that the game with side-payments, $(N, \hat{V}_{sp})$, is convex which guarantees the existence of stable allocations, $Core(\hat{V}_{sp}) \neq \emptyset$. We consign the formal proof to Appendix A.

[6] There could be many, in which case we pick the lowest one.

Theorem 3. *Let (N, m_0, σ_0, w, b) be a public requeueing problem such that σ_0 is an efficient ordering, and let $(N, \hat{V}_{sp})$ be the associated public requeueing game with side payments. Then, $\hat{V}_{sp}$ is convex. Hence, $Core(\hat{V}_{sp}) \neq \emptyset$.*

We conclude this subsection by showing that the initial queue being an efficient ordering is crucial to the result. Without the assumption the core might be empty.

Example 6. 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 $w = (13,7,6,1)$, and the cost of a machine is $b = 15$.

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

m_1	4	3	2	1

We start by looking at coalition $\{1\}$. Agent 1 can offer to switch place with agent 4, offering 3 as compensation, for a net gain of $3 \times 13 - 3 = 36$. Given the limited gain there is no appetite to buy additional machines. Thus, $\hat{V}_{sp}(\{1\}) = 36$.

Consider next coalition $\{2\}$. In the same way, we obtain $\hat{V}_{sp}(\{2\}) = 12$ by having agent 2 compensate agent 4 with a payment of 2 to switch place.

Next, consider coalition $\{1,2\}$. The best they can do is to move to the efficient ordering, obtained by having agent 1 switch place with agent 4 (with a compensation of 3) and agent 2 switch place with agent 3 (with a compensation of 6). Notice that this is also the surplus obtained by coalition N. We thus obtain that $\hat{V}_{sp}(\{1,2\}) = \hat{V}_{sp}(N) = 3 \times 13 + 7 - 6 - 3 = 37$.

Thus, $\hat{V}_{sp}(\{1\}) + \hat{V}_{sp}(\{2\}) > \hat{V}_{sp}(\{1,2\}) = \hat{V}_{sp}(N)$, and $\hat{V}_{sp}$ is not superadditive, much less convex. In fact, its core is empty.

4.3. Extending to constrained requeueing

As discussed in the previous subsection, the literature on requeueing with a fixed number of machines has considered various constraints on eligible changes to queues. See Curiel et al. (1993) and Slikker (2006). Such analysis is particularly complex with an endogenous number of machines because changing the number of machines changes the queues and an agent's neighbors, which typically affect eligible changes. One such restriction (which we summarize as the "no swap" assumption) supposes that members in a coalition can only changes spots with a neighbor, i.e. it cannot jump over a non-coalitional member. A less constraining assumption (the "swap" assumption) allows to jump over a non-coalitional member, as long as they do not have to wait longer than in the initial queue. We thus need to consider an agent's neighbors before and after the change in the number of machines, opening the door for strategic changes simply to increase the set of possibilities.

Another difficulty is how to manage sales of machines. A strict assumption consists in supposing that a coalition can sell a machine only if all users of said machine are part of the coalition. Users are then moved at the end of the queues of other machines, possibly swapping with other coalitional members. A more lenient assumption allows a coalition to sell a machine that has non-coalitional members among its users, as long as these agents can be relocated on other machines without seeing their wait times increase. Still, in both cases, it might be impossible for a coalition to sell $k < m_0$ machines.

The next example illustrates the differences between the assumptions.

Example 7. Consider six agents on two machines, with the initial queue system as follows:

m_1	5	3	1
m_2	4	2	6

Consider coalition $\{1,2\}$ and its possibilities to requeue without changing the number of machines. Under the "side-payment" assumption, agent 1 swaps with agent 5, offering a compensation of $2w_5$, and agent 2 swaps with agent 4, offering a compensation of w_4, for a net gain of $2w_1 + w_2 - w_4 - 2w_5$. Under the "swap" assumption, agents 1 and 2 can swap spots with each other, as it leaves other agents unaffected. Gains are $w_1 - w_2$. Under the "no swap" assumption, this swap is blocked by agent 6, who would see agent 1 move in front of him. Thus, no requeueing is possible.

To illustrate different possibilities when adding a machine, consider coalition $\{1,2,3\}$. After adding a machine, the queue moves up and the queue on machine 1 is 5-2, on machine 2 it is 4-1 and on machine 3 it is 3-6. Under the "side payment" assumption, agents 1 and 2 move to the first position on their machine, compensating agents 4 and 5 with, respectively, w_4 and w_5, for a net gain of $2w_1 + w_2 + w_3 - w_4 - w_5 - b$. Under the "swap" assumption, agents 1 and 3 can swap spots either before the machine was added (they were neighbors on machine 1) or after (they are not on the same machine, but the swap leaves other agents unaffected). The gain is $2w_1 - b$. Under the "no swap" assumption, these swaps are blocked by agent 6, who would see someone move in front of him. Thus, the gain is $w_1 + w_3 - b$.

Finally, to illustrate the different possibilities when selling a machine, consider coalition $\{1,2,4\}$. Under the "side payment" assumption, the coalition can sell a machine and move to the optimal queue 1-2-3-4-5-6, paying agents 3,5,6 respective compensations of w_3, $4w_5$ and $3w_6$, for a net gain of $\frac{b}{2} + 2w_1 - w_3 - 3w_4 - 4w_5 - 3w_6$. Under the more lenient assumption for sales, the only way to not hurt the non-coalitional members is to put them at the front of the new queue. The best way to proceed is to pick queue

5-3-6-1-2-4, for a gain of $\frac{b}{2} - w_1 - 3w_2 - 5w_4$. Under the stricter assumption for sales, it is impossible for coalition $\{1,2,4\}$ to sell a machine, as they are not the sole users of a machine.

We consider the requeueing game with side payments to have minimal constraints – loosening constraints further would allow for a coalition S to move non-coalitional members further down the queue without proper compensation.

While there are multiple games to consider by combining the various constraints, we can provide some results without looking precisely at these games, by using the following fact. Adding any constraints on requeueing (the "swap" or "no swap" assumptions) or on selling machines (the "lenient" and the "strict" assumptions) cannot increase the value created by a coalition, and leaves unchanged the value created by the grand coalition, for which the constraints do not apply. Thus, adding constraints (weakly) enlarges the core.

Proposition 1. *Let (N, m_0, σ_0, w, b) be a public requeueing problem and $\hat{V}_c$ and $\hat{V}_{c'}$ be public requeueing games with different constraints on eligible requeueing. If the constraints in $\hat{V}_{c'}$ are stricter than in $\hat{V}_c$, then, for all $S \subset N$ we have $\hat{V}_c(S) \geq \hat{V}_{c'}(S)$, $\hat{V}_c(N) = \hat{V}_{c'}(N)$ and $Core\left(\hat{V}_c\right) \subseteq Core\left(\hat{V}_{c'}\right)$.*

Thus, for instance, moving from the "swap" assumption to the "no swap" assumption (weakly) enlarges the core, as does moving from the "lenient" to the "strict" assumption on machine sales.

Given Theorem 3 and Proposition 1, the following corollary is immediate.

Corollary 1. *Let (N, m_0, σ_0, w, b) be a public requeueing problem such that σ_0 is an efficient ordering and let $(N, \hat{V}_c)$ be an associated public requeueing game with weakly stricter constraints on eligible requeueing than the public requeueing game with side-payments. Then, $\emptyset \neq Core(\hat{V}_{sp}) \subseteq Core(\hat{V}_c)$.*

An important consequence of Theorem 3 is that whenever each agent owns a machine at the initial queue, then the core is always non-empty, regardless of the assumptions on what constitutes an eligible requeueing.

Corollary 2. *Given a public requeueing problem (N, m_0, σ_0, w, b) such that $m_0 = |N|$, and let $(N, \hat{V}_c)$ be an associated public requeueing game with weakly stricter constraints on eligible requeueing than the public requeueing game with side payments. Then, $\emptyset \neq Core(\hat{V}_{sp}) \subseteq Core(\hat{V}_c)$.*

Recall that we guarantee the existence of stable allocations by showing that the game with side-payment is convex when the order of agents in the original queue is optimal. The following example shows that even if the initial queue is an efficient ordering, the assumption of side payments is crucial for the convexity result.

Example 8. Consider (N, m_0, σ_0, w, b) with $N = \{1,2,3,4\}$, $m_0 = 4$, and σ_0 being the efficient ordering. The waiting costs per unit for agents are $w = (w_i)_{i \in N} = (52, 28, 24, 4)$, and the cost of a machine is $b = 60$.

Notice that at the initial queue there are four machines $m_0 = 4$ and hence each agent's job is processed at a different machine:

m_1	1			
m_2	2			
m_3	3			
m_4	4			

Since each agent starts at a different machine, there is no distinction between the assumption that all users must be part of the coalition selling a machine or that the coalition must simply guarantee that non-coalitional members are not moved to a later spot. In what follows we use $\hat{V}_c(R)$ for any $R \subseteq N$ to denote the value created by R.

For singleton coalitions $\{i\}_{i \in N \setminus \{4\}}$, they would not sell their machine: $\hat{V}_c(\{i\}) = 0$ for $i = \{1,2,3\}$, whereas coalition $\{4\}$ sells her machine, $\hat{V}_c(\{4\}) = 11$.

For two-player coalitions $\{1,2\}$, $\{1,3\}$, and $\{1,4\}$ selling one machine is optimal whereas selling two machines is optimal for $\{2,3\}$, $\{2,4\}$, and $\{3,4\}$ with $\hat{V}_c(\{1,2\}) = 2$, $\hat{V}_c(\{1,3\}) = 6$, $\hat{V}_c(\{1,4\}) = 26$, $\hat{V}_c(\{2,3\}) = 8$, $\hat{V}_c(\{2,4\}) = 28$, $\hat{V}_c(\{3,4\}) = 32$.

For all three-player coalitions selling two machines is optimal with the worth $\hat{V}_c(\{1,2,3\}) = 38$, $\hat{V}_c(\{1,2,4\}) = 58$, $\hat{V}_c(\{1,3,4\}) = 62$, and $\hat{V}_c(\{2,3,4\}) = 62$.

Finally, for the grand coalition it is best to sell two or three machines with the worth $\hat{V}_c(\{1,2,3,4\}) = 92$.

Then, we obtain for $S = \{1,3\}$, $T = \{1,3,4\}$, and $i = 2$ that

$$\hat{V}_c(\{1,2,3\}) - \hat{V}_c(\{1,3\}) > \hat{V}_c(\{1,2,3,4\}) - \hat{V}_c(\{1,3,4\}).$$

That is, $\hat{V}_c(S \cup \{i\}) - \hat{V}_c(S) > \hat{V}_c(T \cup \{i\}) - \hat{V}_c(T)$ which contradicts convexity. Hence, $\hat{V}_c$ is not convex.

When selling machines, using the side-payments game is an effective way to obtain a core allocation. However, as it creates a symmetric game, it eliminates all differences between agents, offering few alternatives to an equal division of the value created. Example 8 shows that without side-payments agent 4 has a bigger role than others in creating value, which should imply a larger share. While a thorough analysis of a fair and stable allocation in these games is left for further research, a quick fix would be to assign the marginal value vector in the decreasing order of waiting cost, resulting in Example 8 in an allocation of $(11, 21, 30, 30)$, which at least approximates the fact that agents with lower waiting costs create more value.

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 as 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 and (iii) the introduction of public queueing problems with an initial queue.

An interesting direction for future research is to characterize axiomatically an allocation rule that always selects a stable allocation for balanced requeueing games. Furthermore, although we have a counterexample showing that stable allocations may not exist when swaps are allowed for public requeueing games with the initial queue not an efficient ordering (Example 6),[7] it is still an open question whether it is also the case when swaps are not allowed.

Declaration of competing interest

None.

Data availability

No data was used for the research described in the article.

Appendix A

We consign to this Appendix formal proofs. We organize them based on the sections in which they are presented.

A.1. Proofs of Section 2

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

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

which simplifies to

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

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 $r^w : \{2, \dots, n\} \to \mathbb{R}_+$.

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

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

[7] It is left to the reader to verify that coalition $\{1, 4\}$ can obtain a value of 36, coalition $\{2, 4\}$ a value of 12, coalition $\{3, 4\}$ a value of 5 and coalition $\{1, 2, 3\}$ a value of 31. It is then impossible to satisfy all these core constraints and budget balance.

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

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

which together with $w_k \le w_{k-1}$ 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

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

and we see that removing a machine is costlier in terms of waiting costs if there are fewer 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 $b \ge r^w(2)$. Then, since r^w is non-increasing, $b \ge r^w(k)$ for all $k \in \{2,\dots,n\}$. This implies that $C(N,k) \le C(N,k+1)$ for all $k = 1,\dots,n-1$. By transitivity, $C(N,1) \le C(N,k)$ for all $k \in \{2,\dots,n\}$ and thus $m(N) = 1$.

Suppose next that $r^w(k) > b \ge r^w(k+1)$ for some $1 < k < n$. By the same argument as above, $b \ge r^w(k+1)$ implies that $C(N,k) \le C(N,l)$ 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 $m(N) = n$. □

Proof of Lemma 2. Let $r^w_S(k)$ be the equivalent of $r^w(k)$ for coalition S.

i) If $m(T) \ge |S|$, 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 $r^w_S(k) \le r^w_T(k)$. That is,

$$\begin{aligned}
r^w_S(k) &= w^S_k + \sum_{l=k+1}^{|S|} \left(\left\lceil \frac{l}{k-1} \right\rceil - \left\lceil \frac{l}{k} \right\rceil \right) w^S_l \\
&\le w^T_k + \sum_{l=k+1}^{|S|} \left(\left\lceil \frac{l}{k-1} \right\rceil - \left\lceil \frac{l}{k} \right\rceil \right) w^T_l \\
&\le w^T_k + \sum_{l=k+1}^{|T|} \left(\left\lceil \frac{l}{k-1} \right\rceil - \left\lceil \frac{l}{k} \right\rceil \right) w^T_l \\
&= r^w_T(k),
\end{aligned}$$

where the first inequality comes from the fact that $w^S_k \le w^T_k$ for all k.

Then, if $b \ge r^w_T(2)$, $b \ge r^w_S(2)$ and $m(S) = m(T) = 1$. Otherwise, $m(S)$ is the highest integer such that $b < r^w_S(m(S))$. But since $r^w_S(m(S)) \le r^w_T(m(S))$, we have $b < r^w_T(m(S))$, and thus $m(S) \le m(T)$, as desired.

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

A.2. Proofs of Section 3

Proof of Lemma 3. **(i)** If n is odd, and N uses μ machines, agents $1,\dots,\mu$ are served in the first period, and others in the second period, for a cost of $\mu b + \sum_{i=1}^{\mu} w_i + \sum_{i=\mu+1}^{n} 2w_i$. If we remove one machine, agent μ is now served in the second period, and agent n in the third period. Thus, we prefer to use μ machines if $b \le w_\mu + w_n$.

(ii) When $n = 2$, $\mu = 1$.

(iii) When $n = 4$, $\mu = 2$. If we use 2 machines, the cost is $2b + w_1 + w_2 + 2w_3 + 2w_4$. If we use a single machine, the cost is $b + w_1 + 2w_2 + 3w_3 + 4w_4$. Thus, we use at least 2 machines if $b \le w_2 + w_3 + 2w_4$.

(iv) When n is even, $\mu = \frac{n}{2}$. If N uses μ machines, agents $1,\dots,\mu$ are served in the first period, and others in the second period, for a cost of $\mu b + \sum_{i=1}^{\mu} w_i + \sum_{i=\mu+1}^{n} 2w_i$. If $n \ge 6$ and we remove one machine, agent μ is now served in the second period, and agents $n-1$ and n in the third period. Thus, we prefer to use μ machines if $b \le w_\mu + w_{n-1} + w_n$. □

Now, we can provide the proof of Theorem 1.

Proof of Theorem 1. Notice that when n is odd $\mu = \left\lfloor \frac{n}{2} \right\rfloor + 1$ and when n is even $\mu = \frac{n}{2} = \left\lfloor \frac{n}{2} \right\rfloor$.

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

First, notice that if we use $k > \mu$ machines, then agents in $\{1, \dots, k\}$ are served in the first period and agents in $\{k+1, \dots, n\}$ are served in the second period. Removing a machine moves agent k from the first group to the second, with all other agents served as before. Thus, the k^{th} machine generates waiting cost savings of w_k and we have $r^w(k) = w_k$ for all $k > \mu$. Also, by Lemma 3, $m(N) \geq \mu$.

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,

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

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 if $b + w_k \leq 2w_k$. While $r^w(\mu) \geq w_\mu$, by assumption $b \leq w_{\left\lfloor \frac{n}{2} \right\rfloor + 1} \leq w_\mu$. The fourth equality also comes from the fact that by assumption, $b \leq w_\mu$.

It remains to prove that the 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 $\kappa \geq \frac{|T|}{2}$, with the cost no smaller otherwise. Thus the core constraint is satisfied. Since T is arbitrarily chosen, the proof is complete.

Next, we show that this is the unique core allocation.

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 $k \in \left\{ \left\lfloor \frac{n}{2} \right\rfloor + 1, \dots, n-1 \right\}$. 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 $k - 1 \geq \left\lfloor \frac{n}{2} \right\rfloor \geq \frac{n-1}{2}$. 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 $b < w_k$ 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 $j \in \{k+1, \dots, n\}$.

It remains to show the result for $b = \left\lfloor \frac{n}{2} \right\rfloor + 1$. It follows immediately using the same procedure as above, with the non-consequential difference that coalition N is indifferent between using $\left\lfloor \frac{n}{2} \right\rfloor + 1$ and $\left\lfloor \frac{n}{2} \right\rfloor$ machines.

Given that we have shown that the allocation is a core allocation, our proof of **(i)** is complete.

(ii) From Lemma 3 we know that $m(N) \geq \mu$. We show that $m(N) = \mu$. If we use $\mu + 1$ machines, we must pay an extra b, but agent $\mu + 1$ moves from being served in the second period to the first period, and thus the net marginal savings are $w_{\mu+1} - b < 0$ as $b > w_\mu \geq w_{\mu+1}$ and n being odd implies that $\mu = \left\lfloor \frac{n}{2} \right\rfloor + 1$.

Next, we show that for all $j \in \{1, \dots, \mu\}$, coalition $N \setminus \{j\}$ uses $\mu - 1$ machines. If we use an extra machine, we must pay an extra b, but agent $\mu + 1$ moves from being served in the second period to the first period, and thus net marginal savings are $w_{\mu+1} - b < 0$. If we use one less machine, the last agent in $\{1, \dots, \mu\} \setminus \{j\}$ moves from the first to the second period, and (at least) agent n moves from the second to the third period. Thus, the net savings are at most $b - w_\mu - w_n < 0$. Thus, the cost for the coalition is $(\mu - 1)b + \sum_{i=1}^{\mu} w_i - w_j + \sum_{i=\mu+1}^{n} 2w_i$.

Together, the two results above show that $y_j \geq C(N) - C(N \setminus \{j\}) = b + w_j$ in any core allocation.

Since we also have that $y_j \leq C(\{j\}) = b + w_j$, we must have $y_j = b + w_j$ for all $j \in \{1, \dots, \mu\}$ in any core allocation.

Consider coalition $\{\mu - 1, \mu\}$. We have that

$$\begin{aligned} y_{\mu-1} + y_\mu &= 2b + w_{\mu-1} + w_\mu \\ &> b + w_{\mu-1} + 2w_\mu \\ &= C(\{\mu - 1, \mu\}) \end{aligned}$$

where the inequality comes from $b > w_\mu$. Thus, the core is empty.

(iii) We divide in three intervals: $b \in \left[w_{\mu+1}, w_{\mu+1} + w_n\right]$, $b \in \left[w_{\mu+1} + w_n, w_\mu + w_n\right]$ and $b \in \left[w_\mu + w_n, w_\mu + 2w_n\right]$. For each interval, we provide an allocation and show that it lies in the core.

Case 1: $b \in \left[w_{\mu+1}, w_{\mu+1} + w_n\right]$.
We show that the following allocation $y \in \mathbb{R}^n$ is in the core: $y_i = \frac{b}{2} + w_i + \frac{w_{\mu+1}}{2}$ for all $i \in \{1, \dots, \mu\}$, and $y_j = \frac{b}{2} + 2w_j - \frac{w_{\mu+1}}{2}$ for all $j \in \{\mu + 1, \dots, n\}$.

First, from Lemma 3 we know that $m(N) \geq \mu$. We show that $m(N) = \mu$. If N adds a machine, the net gain is $w_{\mu+1} - b < 0$. This implies that our allocation is budget-balanced.

Next, we pick $S \subseteq \{1, \dots, \mu\}$ and $T \subseteq \{\mu + 1, \dots, n\}$ and show that $y(S \cup T) \leq C(S \cup T)$. Let $s = |S|$ and $t = |T|$.

We have that

$$y(S \cup T) = \frac{s+t}{2} b + \sum_{i \in S} w_i + \sum_{i \in T} 2w_i + \frac{s-t}{2} w_{\mu+1},$$

and

$$C(S \cup T) \leq mb + \sum_{k=1}^{s} \left\lceil \frac{s_k}{m} \right\rceil w_{s_k} + \sum_{k=1}^{t} \left\lceil \frac{s + t_k}{m} \right\rceil w_{t_k},$$

where s_k and t_k are the k^{th} agent in S and T, respectively. Thus, we need to show that

$$\frac{s+t}{2} b + \sum_{i \in S} w_i + \sum_{i \in T} 2w_i + \frac{s-t}{2} w_{\mu+1} \leq mb + \sum_{k=1}^{s} \left\lceil \frac{s_k}{m} \right\rceil w_{s_k} + \sum_{k=1}^{t} \left\lceil \frac{s + t_k}{m} \right\rceil w_{t_k}.$$

Case 1.1: $s + t \leq 2m$ and $s \leq m$.
$s + t \leq 2m$ guarantees that no agent wait for more than 2 periods, and $s \leq m$ guarantees that all agents in S are served in the first period. Thus, the inequality simplifies to

$$\sum_{k=1}^{m-s} 2w_{t_k} + (s - t)w_{\mu+1} \leq (2m - s - t)b.$$

Since $b \geq w_{\mu+1}$, a sufficient condition is

$$\sum_{k=1}^{m-s} 2w_{t_k} + (s - t)w_{\mu+1} \leq (2m - s - t)w_{\mu+1}$$

$$\sum_{k=1}^{m-s} 2w_{t_k} \leq (m - s)2w_{\mu+1}.$$

On the left-hand side, we have $m - s$ terms, all no larger than $2w_{\mu+1}$, and thus the sufficient condition is verified.

Case 1.2: $s + t \leq 2m$ and $s > m$.
$s + t \leq 2m$ guarantees that no agent wait for more than 2 periods, and $s > m$ guarantees that all agents in T are served in the second period. Thus, the inequality simplifies to

$$-\sum_{k=m+1}^{s} 2w_{s_k} + (s-t)w_{\mu+1} \le (2m-s-t)b.$$

Since $b \ge w_{\mu+1}$, a sufficient condition is

$$-\sum_{k=m+1}^{s} 2w_{s_k} + (s-t)w_{\mu+1} \le (2m-s-t)w_{\mu+1}$$

$$(s-m)2w_{\mu+1} \le \sum_{k=m+1}^{s} 2w_{s_k}.$$

On the right-hand side, we have $s-m$ terms, all no smaller than $2w_{\mu+1}$, and thus the sufficient condition is verified.

Case 1.3: $s+t > 2m$ and $s \le m$.
$s+t > 2m$ implies that some agent will wait for more than two periods, and $s \le m$ implies that all agents in S are served in the first period. Thus, the inequality simplifies to

$$(s+t-2m)b \le \sum_{k=1}^{t} \left(\left\lceil \frac{s+t_k}{m} \right\rceil - 2 \right) 2w_{t_k} + (t-s)w_{\mu+1}.$$

Notice that agents t_k with $k = 1, \ldots, 2m-s$ are served in the second period, while others are served in the third period or later. Thus, we have that

$$\sum_{k=1}^{t} \left(\left\lceil \frac{s+t_k}{m} \right\rceil - 2 \right) 2w_{t_k} \ge \sum_{k=2m-s+1}^{t} 2w_{t_k}.$$

Therefore, a sufficient condition, using $b \le w_{\mu+1} + w_n$, is

$$(s+t-2m)\left(w_{\mu+1} + w_n\right) \le \sum_{k=2m-s+1}^{t} 2w_{t_k} + (t-s)w_{\mu+1}$$

which simplifies to

$$(s+t-2m)w_n \le (m-s)2w_{\mu+1} + \sum_{k=2m-s+1}^{t} 2w_{t_k}.$$

The summation on the right-hand side contains $s+t-2m$ terms all at least as large as w_n. Since $m \ge s$, the first term on the right-hand side is non-negative, and thus the sufficient condition is satisfied.

Case 1.4: $s+t > 2m$ and $s > m$.
$s+t > 2m$ implies that some agent will wait for more than two periods, and $s > m$ implies that some agents in S are served in the second period. Notice also that it is never optimal to have $m < \frac{s}{2}$, so that at worst, agents in S wait two periods. Thus, the inequality simplifies to

$$(s+t-2m)b \le \sum_{k=m+1}^{s} 2w_{s_k} + \sum_{k=1}^{t} \left(\left\lceil \frac{s+t_k}{m} \right\rceil - 2 \right) 2w_{t_k} + (t-s)w_{\mu+1}.$$

As in the case above, we have that

$$\sum_{k=1}^{t} \left(\left\lceil \frac{s+t_k}{m} \right\rceil - 2 \right) 2w_{t_k} \ge \sum_{k=2m-s+1}^{t} 2w_{t_k}.$$

Therefore, a sufficient condition, using $b \le w_{\mu+1} + w_n$, is

$$(s+t-2m)\left(w_{\mu+1} + w_n\right) \le \sum_{k=m+1}^{s} 2w_{s_k} + \sum_{k=2m-s+1}^{t} 2w_{t_k} + (t-s)w_{\mu+1},$$

which simplifies to

$$(s-m)2w_{\mu+1} + (s+t-2m)w_n \le \sum_{k=m+1}^{s} 2w_{s_k} + \sum_{k=2m-s+1}^{t} 2w_{t_k}.$$

The first summation on the right-hand side contains $(s-m)$ terms, all no smaller than $2w_{\mu+1}$. The second summation contains $s+t-2m$ terms, all no smaller than w_n. Thus, the sufficient condition holds.

All together, we have verified all combinations, and Case 1 is complete.

Case 2: $b \in [w_{\mu+1} + w_n, w_\mu + w_n]$.
Using the same technique as for Case 1, we can show that the following allocation $y \in \mathbb{R}^n$ is in the core: $y_i = b + w_i - \frac{w_n}{2}$ for all $i \in \{1, \ldots, \mu\}$, $y_j = 2w_j + \frac{w_n}{2}$ for all $j \in \{\mu+1, \ldots, n\}$.

Case 3: $b \in [w_\mu + w_n, w_\mu + 2w_n]$.
Using the same technique as for Case 1, we can show that the following allocation $y \in \mathbb{R}^n$ is in the core: $y_i = \frac{b}{2} + w_i + \frac{w_\mu}{2}$ for all $i \in \{1, \ldots, \mu\}$, $y_j = \frac{b}{2} + 2w_j - \frac{w_\mu}{2}$ for all $j \in \{\mu+1, \ldots, n\}$.

(iv) From Lemma 3 we know that $m(N) \geq \mu$. We show that $m(N) = \mu$, and thus $C(N) = b\mu + \sum_{i \in N_1} w_i + \sum_{i \in N_2} 2w_i$. If it were to add a machine, the net gain would be $w_{\mu+1} - b < 0$, confirming that $m(N) = \mu$.

Second, we show that $C(N \setminus \{\mu+1\}) = (\mu-1)b + \sum_{j=1}^{\mu-1} w_j + 2w_\mu + \sum_{j=\mu+2}^{n-1} 2w_j + 3w_n$. If it were to add a machine, the net gain would be $w_\mu + w_n - b < 0$. If it were to remove a machine, the net gain would be $b - w_{\mu-1} - w_{n-2} - w_{n-1} < 0$.

Since in any core allocation we must have that $y_{\mu+1} \geq C(N) - C(N \setminus \{\mu+1\})$, we obtain

$$y_{\mu+1} \geq b - w_\mu + 2w_{\mu+1} - w_n.$$

Recall that since $\mu = m(N)$, $N_1 = \{1, \ldots, \mu\}$ and $N_2 = \{\mu+1, \ldots, n\}$. Now, consider $S \subset N_1$ and $T \subset N_2 \setminus \{\mu+1\}$ such that $n \in T$ and that $|S| = |T| = k$.

We have that $C(S \cup T) \leq kb + \sum_{i \in S} w_i + \sum_{i \in T} 2w_i$. We also have that $C(N \setminus (S \cup T)) \leq (\mu-k)b + \sum_{i \in N_1 \setminus S} w_i + \sum_{i \in N_2 \setminus T} 2w_i$. This, along with the value of $C(N)$, shows that if any of these inequalities is strict, the core is empty, completing the proof. Suppose otherwise. Then, in any core allocation we have $y(S \cup T) = C(S \cup T) = kb + \sum_{i \in S} w_i + \sum_{i \in T} 2w_i$.

Consider coalition $S \cup T \cup \{\mu+1\}$. We have that $C(S \cup T \cup \{\mu+1\}) \leq kb + \sum_{i \in S} w_i + \sum_{i \in T \setminus \{n\}} 2w_i + 2w_{\mu+1} + 3w_n$. Since $y(S \cup T) = C(S \cup T)$, we have that in any core allocation we must have

$$\begin{aligned} y_{\mu+1} &\leq C(S \cup T \cup \{\mu+1\}) - C(S \cup T) \\ &= 2w_{\mu+1} + w_n. \end{aligned}$$

Therefore, a necessary condition for the existence of a core allocation is

$$b - w_\mu + 2w_{\mu+1} - w_n \leq 2w_{\mu+1} + w_n$$

or

$$b \leq w_\mu + 2w_n$$

which is not satisfied. Therefore, the core is empty. □

Making use of Lemmata 1, 2 we can provide the proof of Theorem 2.

Proof of Theorem 2. Notice that $w_2 + \sum_{i=3}^{n} \left(i - \left\lceil \frac{i}{2} \right\rceil\right) w_i = r^w(2)$, 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. Recall that w_k^T denotes the waiting cost of the k^{th} agent in T, according to the order in N and $w_k^{-T} \equiv w_k^{N \setminus T}$. Thus, for all $\emptyset \neq T \subseteq N$, $C(T) = b + \sum_{i=1}^{|T|} i w_i^T$ and let $\hat{C}(T) := C(T) - \sum_{i=1}^{n-|T|-1} i w_{i+1}^{-T}$.

The proof consists in showing that $Core(\hat{C}) = Core(C)$, then for part (i), to show that $\hat{C}$ as a very specific structure yielding the given core,[8] and for part (ii), in the given interval, that $\hat{C}$ is not subadditive.

Notice first that $\hat{C}(T) = C(T)$ if $|T| \geq n-1$. 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 $|T| = m$ and fix $k \in N \setminus T$. We consider the core constraints for $T \cup \{k\}$ and $N \setminus \{k\}$. Since both contain at least $m+1$ agents, by the recursive argument we must have $y(T \cup \{k\}) \leq \hat{C}(T \cup \{k\})$ and $y(N \setminus \{k\}) \leq \hat{C}(N \setminus \{k\})$. By summing them up and using the fact that $y(N) = C(N) = \hat{C}(N)$ we obtain $y(T) \leq \hat{C}(T \cup \{k\}) + \hat{C}(N \setminus \{k\}) - \hat{C}(N)$.

We have that $C(T \cup \{k\}) = b + \sum_{i=1}^{|T|+1} i w_i^{T \cup \{k\}}$, $\hat{C}(T \cup \{k\}) = C(T) - \sum_{i=1}^{n-|T|-2} i w_{i+1}^{-(T \cup \{k\})}$ and $\hat{C}(N \setminus \{k\}) = C(N \setminus \{k\}) = b + \sum_{i=1}^{n-1} i w_i^{N \setminus \{k\}}$ and $\hat{C}(N) = C(N) = b + \sum_{i=1}^{n} i w_i$.

Thus,

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

[8] While the resulting game has a simple enough structure to analyze its core directly, we can provide an alternative proof by showing that the resulting game, written as a value sharing game, is a symmetric one-bound core game (Driessen, 1985; Gong et al., 2023).

We simplify this expression by looking at the terms associated to w_j, for a given $j \in N$. Before proceeding, we observe that for all $k \in N$ and all $T \in N \setminus \{k\}$, the rank of agent k in $T \cup \{k\}$ plus its rank in $N \setminus T$ is exactly $k+1$. Agent k's waiting cost appears in the second and fifth terms. In the second term, it appears with a weight equal to its rank in $T \cup \{k\}$. In the fifth term, the weight is $-k$. Using the equality established above, this simplifies to its rank in $N \setminus T$, minus one.

Next, we consider agent $j \neq k$, and distinguish several cases.

Case 1: $j \in T$ such that $j < k$. Then, the waiting cost of agent j appears in the second, fourth and fifth terms. Since it is the j^{th} agent in $N \setminus \{k\}$ and in N, the fourth and fifth terms cancel out. It remains in the second term, where it appears with a weight equals to its rank in $T \cup \{k\}$, which is the same as its rank in T.

Case 2: $j \in T$ such that $j > k$. Then, the waiting cost of agent j appears in the second, fourth and fifth terms. Since it is the $(j-1)^{th}$ agent in $N \setminus \{k\}$ and the j^{th} in N, the fourth and fifth terms simplify to $-w_j$. In the second term, the waiting cost of agent j appears with a weight equal to its rank in $T \cup \{k\}$. Since $j > k$, the rank in $T \cup \{k\}$ minus one is equal to the rank in T.

Case 3: $j \notin T$ such that $j < k$. Then, the waiting cost of agent j appears in the third, fourth and fifth terms. Since it is the j^{th} agent in $N \setminus \{k\}$ and in N, the fourth and fifth terms cancel out. It remains in the third term, where it appears with a weight equals to its rank in $N \setminus (T \cup \{k\})$ minus one. Since j has the same rank in $N \setminus T$, this is equal to its rank in $N \setminus T$ minus one.

Case 4: $j \notin T$ such that $j > k$. Then, the waiting cost of agent j appears in the third, fourth and fifth terms. Since it is the $(j-1)^{th}$ agent in $N \setminus \{k\}$ and the j^{th} in N, the fourth and fifth terms simplify to $-w_j$. In the third term it appears with a weight equals to its rank in $N \setminus (T \cup \{k\})$ minus one. Adding the simplification of the fourth and fifth terms, we obtain its rank in $N \setminus (T \cup \{k\})$ minus two. Since $j > k$, this is equal to its rank in $N \setminus T$ minus one.

Putting everything together, we can simplify the expression, differentiating if an agent belongs to T or not. If it does, the agent is in Cases 1 or 2. If not, it is in Cases 3 or 4, or it is agent k. We obtain:

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

Thus, we have the core constraint $y(T) \leq \hat{C}(T)$. This completes the recursive argument, and $y \in Core(C)$ implies that $y \in Core(\hat{C})$. Given that $\hat{C} \leq C$ and $\hat{C}(N) = C(N)$, this completes the proof that $Core(\hat{C}) = Core(C)$.

We now establish the value of incremental costs for the function $\hat{C}$.

Fix $\emptyset \neq T \subseteq N \setminus \{k\}$. Then, we have that

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

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 $T \neq \emptyset$. For $T = \emptyset$, notice that $\hat{C}(\{k\}) = b + w_k - \sum_{i=1}^{n-2} i w_{i+1}^{-\{k\}} = b + w_k - \sum_{i<k}(i-1)w_i - \sum_{i>k}(i-2)w_i$.

A necessary condition for $Core(\hat{C})$ to be non-empty is $\hat{C}(\{k\}) + \hat{C}(N \setminus \{k\}) \geq \hat{C}(N)$, or equivalently

$$\hat{C}(N) - \hat{C}(N \setminus \{k\}) = \sum_{i>k} w_i + k w_k \leq b + w_k - \sum_{i<k}(i-1)w_i - \sum_{i>k}(i-2)w_i = \hat{C}(\{k\})$$

which simplifies to

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

Therefore, in the interval given in statement (*ii*), the core is empty, as desired.

To complete the proof of statement (*i*), notice that the inequality above, in the given interval is satisfied, which also implies that $\hat{C}$ is concave. Thus, the extreme points of its core are given by the allocations y^k, for all $k \in N$, such that

$$y_j^k = \begin{cases} b + w_k - \sum_{i<k}(i-1)w_i - \sum_{i>k}(i-2)w_i \text{ if } k = j \\ A_j \text{ otherwise} \end{cases}$$

as it only matters if an agent k is picked first, and pays $\hat{C}(\{k\})$, or not, and pays A_k.

As established above, we have that $b + w_k - \sum_{i<k}(i-1)w_i - \sum_{i>k}(i-2)w_i - A_k = b - \sum_{i=1}^{n}(i-1)w_i$, allowing us to rewrite the allocation as

$$y_j^k = \begin{cases} A_k + b - \sum_{i=1}^{n}(i-1)w_i & \text{if } k = j \\ A_j & \text{otherwise} \end{cases}.$$

Since $Core(\hat{C})$ is the convex combination of these extreme points, and since $b - \sum_{i=1}^{n}(i-1)w_i$ is non-negative and independent of k, it is immediate that $Core(C) = Core(\hat{C}) = A + \Delta\left(N, b - \sum_{i=1}^{n}(i-1)w_i\right)$, as desired. □

A.3. Proofs of Section 4

Proof of Lemma 4. i) First, it is immediate that if a coalition prefers to buy $k > 1$ machines than use m_0 machines, it also prefers to buy one machine to using m_0 machines. In the same way, if a coalition prefers to sell $k > 1$ machines to using m_0 machines, it also prefers to sell one machine to using m_0 machines. Thus, we only need to show that there cannot be $S, T \subseteq N$ such that S prefers to buy a machine to using m_0 machines and T prefers to sell a machine to using m_0 machines.

Suppose first that S prefers to buy a machine to using m_0 machines. Thus, $\sum_{i\in S}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i - b > 0$. But, we have that

$$\begin{aligned}
\sum_{i\in S}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i - b &\leq \sum_{i\in N}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i - b \\
&\leq \sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i - b \\
&\leq \sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i - \frac{|T|}{n} b
\end{aligned}$$

and thus $\sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i - \frac{|T|}{n} b > 0$, which can be rewritten as $\frac{|T|}{n} b - \sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i < 0$ which indicates that T does not prefer to sell 1 machine to using m_0 machines.

Suppose next that S prefers to sell a machine to using m_0 machines. Thus, $\frac{|S|}{n} b - \sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i > 0$. But, we have that

$$\begin{aligned}
\frac{|S|}{n} b - \sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i &\leq b - \sum_{i\in N}\left(\left\lceil \frac{i}{m_0-1}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i \\
&\leq b - \sum_{i\in N}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i \\
&\leq b - \sum_{i\in T}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i
\end{aligned}$$

and thus $b - \sum_{i\in T}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i > 0$, which can be rewritten as $\sum_{i\in T}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{m_0+1}\right\rceil\right) w_i - b < 0$, which indicates that T does not prefer to buy 1 machine to using m_0 machines.

ii) Suppose that S buys machines. Then, by part i), so does T. We have that $\sum_{i\in S}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S)}\right\rceil\right) w_i - b\left(\hat{m}(S) - m_0\right) \geq \sum_{i\in S}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{k}\right\rceil\right) w_i - b\left(k - m_0\right)$ for all $k = m_0, \ldots, \hat{m}(S)$. Add $\sum_{i\in T\setminus S}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S)}\right\rceil\right) w_i$ on both sides to obtain

$$\sum_{i\in T}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S)}\right\rceil\right) w_i - b\left(\hat{m}(S) - m_0\right) \geq \sum_{i\in T}\left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{k}\right\rceil\right) w_i - b\left(k - m_0\right)$$

for all $k = m_0, \ldots, \hat{m}(S)$, and thus T buys at least as many machines as S.

Suppose next that S sells machines. Then, by part i), so does T. We have that $\frac{|S|}{n} b(m_0 - \hat{m}(S)) - \sum_{i\in N}\left(\left\lceil \frac{i}{\hat{m}(S)}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i \geq \frac{|S|}{n} b(m_0 - k) - \sum_{i\in N}\left(\left\lceil \frac{i}{k}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i$ for all $k = \hat{m}(S), \ldots, m_0$. We then have that

$$\frac{|T|}{n} b(m_0 - \hat{m}(S)) - \sum_{i\in N}\left(\left\lceil \frac{i}{\hat{m}(S)}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i \geq \frac{|T|}{n} b(m_0 - k) - \sum_{i\in N}\left(\left\lceil \frac{i}{k}\right\rceil - \left\lceil \frac{i}{m_0}\right\rceil\right) w_i$$

for all $k = \hat{m}(S), \ldots, m_0$, and thus T sells at least as many machines as S. □

Next, we provide the formal proof of Theorem 3 in Section 4 showing that the game with side-payments for requeueing problems, $(N, \hat{V}_{sp})$, is convex and hence its core is non-empty, $Core(\hat{V}_{sp}) \neq \emptyset$.

Proof of Theorem 3. To ease on the notation, we use V instead of $\hat{V}_{sp}$. Fix $S \subset T \subseteq N \setminus \{i\}$. We show that $V(S) + V(T \cup \{i\}) \geq V(S \cup \{i\}) + V(T)$.

By Lemma 4, either all coalitions buy machines (or stay put) or all coalitions sell machines (or stay put). We consider these two cases separately.

Case 1: All coalitions sell machines.

We have that

$$V(S) \geq \left(m_0 - \hat{m}(S \cup \{i\})\right) b \frac{|S|}{n} - \sum_{j \in N} \left(\left\lceil \frac{j}{\hat{m}(S \cup \{i\})} \right\rceil - \left\lceil \frac{j}{m_0} \right\rceil \right) w_j$$

since S has the option to pick $\hat{m}(S \cup \{i\})$. In the same way, we have

$$V(T \cup \{i\}) \geq \left(m_0 - \hat{m}(T)\right) b \frac{|T|+1}{n} - \sum_{j \in N} \left(\left\lceil \frac{j}{\hat{m}(T)} \right\rceil - \left\lceil \frac{j}{m_0} \right\rceil \right) w_j$$

Summing these inequalities and rearranging, we obtain

$$\begin{aligned} V(S) + V(T \cup \{i\}) &\geq \left(m_0 - \hat{m}(S \cup \{i\})\right) b \frac{|S|}{n} - \sum_{j \in N} \left(\left\lceil \frac{j}{\hat{m}(S \cup \{i\})} \right\rceil - \left\lceil \frac{j}{m_0} \right\rceil \right) w_j \\ &+ \left(m_0 - \hat{m}(T)\right) b \frac{|T|+1}{n} - \sum_{j \in N} \left(\left\lceil \frac{j}{\hat{m}(T)} \right\rceil - \left\lceil \frac{j}{m_0} \right\rceil \right) w_j \\ &\geq \left(m_0 - \hat{m}(S \cup \{i\})\right) b \frac{|S|+1}{n} - \sum_{j \in N} \left(\left\lceil \frac{j}{\hat{m}(S \cup \{i\})} \right\rceil - \left\lceil \frac{j}{m_0} \right\rceil \right) w_j \\ &+ \left(m_0 - \hat{m}(T)\right) b \frac{|T|}{n} - \sum_{j \in N} \left(\left\lceil \frac{j}{\hat{m}(T)} \right\rceil - \left\lceil \frac{j}{m_0} \right\rceil \right) w_j \\ &= V(S \cup \{i\}) + V(T), \end{aligned}$$

where the second inequality comes from the following observation: When coalitions are selling machines, V is symmetric. Thus, Lemma 4 part ii) implies that if $|R| \geq |R'|$, $\hat{m}(R) \leq \hat{m}(R')$. In our case, this implies that $\hat{m}(T) \leq \hat{m}(S \cup \{i\})$.

Case 2: All coalitions buy machines.

Since in this case V is no longer symmetric, we need to distinguish two subcases: a) $\hat{m}(T) \geq \hat{m}(S \cup \{i\})$ and b) $\hat{m}(T) < \hat{m}(S \cup \{i\})$.

a) $\hat{m}(T) \geq \hat{m}(S \cup \{i\})$.

We have that

$$V(S) \geq \sum_{j \in S} \left(\left\lceil \frac{j}{m_0} \right\rceil - \left\lceil \frac{j}{\hat{m}(S \cup \{i\})} \right\rceil \right) w_j - \left(\hat{m}(S \cup \{i\}) - m_0\right) b$$

as S had the option to pick $\hat{m}(S \cup \{i\})$. In the same way, we have

$$V(T \cup \{i\}) \geq \sum_{j \in T} \left(\left\lceil \frac{j}{m_0} \right\rceil - \left\lceil \frac{j}{\hat{m}(T)} \right\rceil \right) w_j + \left(\left\lceil \frac{i}{m_0} \right\rceil - \left\lceil \frac{i}{\hat{m}(T)} \right\rceil \right) w_i - \left(\hat{m}(T) - m_0\right) b$$

as $T \cup \{i\}$ had the option to pick $\hat{m}(T)$.

Summing these inequalities and rearranging, we obtain

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

where the second inequality comes from the fact that $\hat{m}(T) \geq \hat{m}(S \cup \{i\})$.

b) $\hat{m}(T) < \hat{m}(S \cup \{i\})$.

We have that

$$V(S) \geq \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(T)}\right\rceil\right) w_j - \left(\hat{m}(T) - m_0\right) b$$

as S had the option to pick $\hat{m}(T)$. In the same way, we have

$$V(T\cup\{i\}) \geq \sum_{j\in T}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_j + \left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_i - \left(\hat{m}(S\cup\{i\}) - m_0\right) b$$

as $T\cup\{i\}$ had the option to pick $\hat{m}(S\cup\{i\})$.

Summing these inequalities and rearranging, we obtain

$$\begin{aligned}
V(S) + V(T\cup\{i\}) \geq\ & \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(T)}\right\rceil\right) w_j - \left(\hat{m}(T) - m_0\right) b \\
& + \sum_{j\in T}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_j + \left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_i \\
& - \left(\hat{m}(S\cup\{i\}) - m_0\right) b \\
& = \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(T)}\right\rceil\right) w_j - \left(\hat{m}(T) - m_0\right) b + \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_j \\
& + \sum_{j\in T\setminus S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_j + \left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_i \\
& - \left(\hat{m}(S\cup\{i\}) - m_0\right) b \\
& \geq \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(T)}\right\rceil\right) w_j - \left(\hat{m}(T) - m_0\right) b \\
& + \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_j + \sum_{j\in T\setminus S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(T)}\right\rceil\right) w_j \\
& + \left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_i - \left(\hat{m}(S\cup\{i\}) - m_0\right) b \\
& = \sum_{j\in S}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_j + \left(\left\lceil \frac{i}{m_0}\right\rceil - \left\lceil \frac{i}{\hat{m}(S\cup\{i\})}\right\rceil\right) w_i \\
& - \left(\hat{m}(S\cup\{i\}) - m_0\right) b \\
& + \sum_{j\in T}\left(\left\lceil \frac{j}{m_0}\right\rceil - \left\lceil \frac{j}{\hat{m}(T)}\right\rceil\right) w_j - \left(\hat{m}(T) - m_0\right) b \\
& = V(S\cup\{i\}) + V(T)
\end{aligned}$$

where the second inequality comes from the fact that $\hat{m}(S\cup\{i\}) > \hat{m}(T)$. □

References

Atay, A., Calleja, P., Soteras, S., 2021. Open shop scheduling games. Eur. J. Oper. Res. 295, 12–21.

Bahel, E., Trudeau, C., 2019. Stability and fairness in the job scheduling problem. Games Econ. Behav. 117, 1–14.

Böhm-Bawerk, E., 1923. Positive Theory of Capital. Translated by W. Smart. G.E. Steckert, original publication 1891.

Calleja, P., Borm, P., Hamers, H., Klijn, F., Slikker, M., 2002. On a new class of parallel sequencing situations and related games. Ann. Oper. Res. 109, 265–277.

Chun, Y., 2006. A pessimistic approach to the queueing problem. Math. Soc. Sci. 51, 171–181.

Chun, Y., 2016. Fair Queueing. Springer.

Chun, Y., Heo, E.J., 2008. Queueing problems with two parallel servers. Int. J. Econ. Theory 4, 299–315.

Chun, Y., Yengin, D., 2017. Welfare lower bounds and strategy-proofness in the queueing problem. Games Econ. Behav. 102, 462–476.

Chun, Y., Mitra, M., Mutuswami, S., 2019. A characterization of the symmetrically balanced VCG rule in the queueing problem. Games Econ. Behav. 118, 486–490.

Curiel, I., Pederzoli, G., Tijs, S., 1989. Sequencing games. Eur. J. Oper. Res. 40, 344–351.

Curiel, I., Potters, J., Prasad, R., Tijs, S., Veltman, B., 1993. Cooperation in one machine scheduling. Z. Oper.-Res. 38, 113–129.

De, P., Mitra, M., 2017. Incentives and justice for sequencing problems. Econ. Theory 64, 239–264.

De, P., Mitra, M., 2019. Balanced implementability of sequencing rules. Games Econ. Behav. 118, 342–353.

Dehez, P., 2021. 1-convex transferable utility games, a reappraisal. Mimeo.

Driessen, T.S.H., 1985. Properties of 1-convex n-person games. OR Spektrum 7, 19–26.

García-Sanz, M.D., Fernández, F.R., Fiestras-Janeiro, M.G., García-Jurado, I., Puerto, J., 2008. Cooperation in Markovian queueing models. Eur. J. Oper. Res. 188, 485–495.

Gillies, D.B., 1959. Solutions to general non-zero-sum games. In: Tucker, A.W., Luce, R.D. (Eds.), Contributions to the Theory of Games IV. Princeton University Press, pp. 47–85.
Gong, D., Dietzenbacher, B., Peters, H., 2023. One-bound core games. Mimeo.
González, P., Herrero, C., 2004. Optimal sharing of surgical costs in the presence of queues. Math. Methods Oper. Res. 59, 435–446.
Han, C., Chun, Y., 2022. The Shapley value in positional queueing problems. Mimeo.
Hashimoto, K., Saitoh, H., 2012. Strategy-proof and anonymous rule in queueing problems: a relationship between equity and efficiency. Soc. Choice Welf., 473–480.
Kayı, Ç., Ramaekers, E., 2010. Characterizations of Pareto-efficient, fair, and strategy-proof allocation rules in queueing problems. Games Econ. Behav. 68, 220–232.
Maniquet, F., 2003. A characterization of the Shapley value in queueing problems. J. Econ. Theory 109, 90–103.
Mitra, M., 2005. Incomplete information and multiple machine queueing problems. Eur. J. Oper. Res. 165, 251–266.
Mitra, M., Mutuswami, S., 2011. Group strategyproofness in queueing models. Games Econ. Behav. 72, 242–254.
Mitra, M., Mutuswami, S., 2021. No-envy in the queueing problem with multiple identical machines. In: Borkotokey, S., Kumar, R., Mukherjee, D., Rao, K.S.M., Sarangi, S. (Eds.), Game Theory and Networks: New Perspectives and Directions. In: Indian Statistical Institute Series. Springer, pp. 161–176.
Mukherjee, C., 2013. Weak group strategy-proof and queue-efficient mechanisms for the queueing problem with multiple machines. Int. J. Game Theory 42, 131–163.
Musegaas, M., Borm, P., Quant, M., 2015. Step out–step in sequencing games. Eur. J. Oper. Res. 246, 894–906.
Özen, U., Reiman, M.I., Wang, Q., 2011. On the core of cooperative queueing games. Oper. Res. Lett. 39, 385–389.
Shapley, L.S., 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, pp. 307–317.
Shapley, L.S., 1959. The solutions of a symmetric market game. Ann. Math. Study 40, 145–162.
Shapley, L.S., 1971. Cores of convex games. Int. J. Game Theory 1, 11–26.
Shapley, L.S., Shubik, M., 1971. The assignment game I: the core. Int. J. Game Theory 1, 111–130.
Slikker, M., 2006. Relaxed sequencing games have a nonempty core. Nav. Res. Logist. 53, 235–242.
Smith, W., 1956. Various optimizers for single-stage production. Nav. Res. Logist. Q. 3, 59–66.
Suijs, J., 1996. On incentive compatibility and budget balancedness in public decision making. Econ. Des. 2, 193–209.
Zeng, Y., Zhang, L., Cai, X., Li, J., 2018. Cost sharing for capacity transfer in cooperating queueing systems. Prod. Oper. Manag. 27, 644–662.