Efficiency, Sequenceability and Deal-Optimality in Fair Division of Indivisible Goods
Aurélie Beynier, Sylvain Bouveret, Michel Lemaître, Nicolas Maudet, Simon Rey, Parham Shams

To cite this version:
Aurélie Beynier, Sylvain Bouveret, Michel Lemaître, Nicolas Maudet, Simon Rey, et al.. Efficiency, Sequenceability and Deal-Optimality in Fair Division of Indivisible Goods. 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), May 2019, Montreal, Canada. hal-02132742

HAL Id: hal-02132742
https://hal.archives-ouvertes.fr/hal-02132742
Submitted on 17 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Efficiency, Sequenceability and Deal-Optimality in Fair Division of Indivisible Goods

Aurélie Beynier  
Sorbonne Université, CNRS, LIP6  
F-75005 Paris, France  
aurelie.beynier@lip6.fr

Nicolas Maudet  
Sorbonne Université, CNRS, LIP6  
F-75005 Paris, France  
nicolas.maudet@lip6.fr

Sylvain Bouveret  
Univ. Grenoble Alpes, CNRS, LIG  
Grenoble, France  
sylvain.bouveret@imag.fr

Simon Rey  
Sorbonne Université, ENS P-S  
Paris, Cachan, France  
srey@ens-paris-saclay.fr

Michel Lemaître  
Formerly ONERA  
Toulouse, France  
michel.lemaitre.31@gmail.com

Parham Shams  
Sorbonne Université, CNRS, LIP6  
F-75005 Paris, France  
parham.shams@lip6.fr

ABSTRACT
In fair division of indivisible goods, using sequences of sincere choices (or picking sequences) is a natural way to allocate the objects. The idea is as follows: at each stage, a designated agent picks one object among those that remain. Another intuitive way to obtain an allocation is to give objects to agents in the first place, and to let agents exchange them as long as such “deals” are beneficial. This paper investigates these notions, when agents have additive preferences over objects, and with other efficiency and fairness notions. In particular, we show that an allocation is sequenceable if and only if it is optimal for a certain type of deals, namely cycle deals involving a single object. Furthermore, any Pareto-optimal allocation is sequenceable, but not the converse. Regarding fairness, we show that an allocation can be envy-free and non-sequenceable, but that every competitive equilibrium with equal incomes is sequenceable. To complete the picture, we show how some domain restrictions may affect the relations between these notions. Finally, we experimentally explore the links between the scales of efficiency and fairness.

KEYWORDS
Multiagent Resource Allocation; Fair Division; Efficiency


1 INTRODUCTION
In this paper, we investigate fair division of indivisible goods. In this problem, a set of indivisible objects or goods has to be allocated to a set of agents, taking into account the agents’ preferences about the objects. This classical collective decision making problem has plenty of practical applications, among which the allocation of space resources [9, 30], of tasks to workers in crowdsourcing market systems [34], papers to reviewers [26] or courses to students [19].

This problem can be tackled from two different perspectives. The first possibility is to resort to a benevolent entity in charge of collecting in a centralized way the preferences of all the agents. This entity then computes an allocation that takes into account these preferences and satisfies some fairness (e.g. envy-freeness) and efficiency (e.g. Pareto-optimality) criteria, or optimizes a well-chosen social welfare ordering. The second possibility is to have a distributed point of view, e.g. by starting from an initial allocation and letting the agents negotiate to swap their objects [23, 35]. A somewhat intermediate approach consists in allocating the objects to the agents using a protocol, which allows to build an allocation interactively by asking the agents a sequence of questions. Protocols are at the heart of works mainly concerning the allocation of divisible resources (cake-cutting) [16], but have also been studied in the context of indivisible goods [14, 16].

In this paper, we focus on a particular allocation protocol: sequences of sincere choices (also known as picking sequences). This very simple protocol works as follows. A central authority chooses a sequence of agents before the protocol starts, having as many agents as the number of objects (some agents may appear several times in the sequence). Then, each agent appearing in the sequence is asked to choose in turn one object among those that remain. For instance, according to the sequence (1, 2, 2, 1), agent 1 will choose first, then agent 2 will pick two objects in a row, and agent 1 will take the last object. This protocol, used in a lot of everyday situations, has been studied for the first time by Kohler and Chandrasekaran [29]. Later, Brams and Taylor [17] have studied a particular version of this protocol, namely alternating sequences, in which the sequence of agents is restricted to a balanced ((1,2,2,1,...)) or strict ((1,2,1,2,...)) alternation of agents. Bouveret and Lang [11] have further formalized this protocol, whose properties (especially related to game theoretic aspects) have later been characterized by Kalinowski et al. [27, 28]. Finally, Aziz et al. [4] have studied the complexity of problems related to finding whether a particular assignment (or bundle) is achievable by a particular class of picking sequences. Picking sequences have also been considered by Brams and King [15], that focus on a situation where the agents have ordinal preferences. They make an interesting link between this
protocol and Pareto-optimality, showing, among others, that picking sequences always result in a Pareto-optimal allocation, but also that every Pareto-optimal allocation can be obtained in this way.

In this paper, we elaborate on these ideas and analyze the links between sequences, certain types of deals among agents, and some efficiency and fairness properties, in a more general model in which the agents have numerical additive preferences on the objects. Our main contributions are the following. We give a formalization of the link between allocations and sequences of sincere choices, highlighting a simple characterization of the sequenceability of an allocation (Section 3). Then, we show that in this slightly more general framework than the one by Brams and King, Pareto-optimality and sequenceability are not equivalent anymore (Section 4). By unveiling the connection between sequenceability and cycle deals among agents (Section 5), we obtain a rich “scale of efficiency” that allows us to characterize the degree of efficiency of a given allocation. Interestingly, some domain restrictions have significant effects on this hierarchy (Section 6). We also highlight (Section 7) a link between sequenceability and another important economical concept: the competitive equilibrium from equal income (CEEI). Another contribution is the experimental exploration of the links between the scale of efficiency and fairness properties (Section 8).

2 MODEL AND DEFINITIONS

The aim of the fair division of indivisible goods, also called MultiAgent Resource Allocation (MARA), is to allocate a finite set of objects $O = \{o_1, \ldots, o_m\}$ to a finite set of agents $N = \{1, \ldots, n\}$.

A sub-allocation on $O' \subseteq O$ is a vector $\pi_{\pi'}^{O'} = \langle \pi^{O'}_1, \ldots, \pi^{O'}_{n}\rangle$ of bundles of objects, such that $\forall i, \forall j$ with $i \neq j: \pi^{O'}_i \cap \pi^{O'}_j = \emptyset$ (a given object cannot be allocated to more than one agent) and $\bigcup_{i \in N} \pi^{O'}_i = O'$. Let $\pi^{O'}_i$ be the share of agent $i$’s bundle $\pi^{O'}_i \subseteq O'$ when $\pi^{O'}_i \subseteq \pi^{O'}_j$ for each agent $i$. Any sub-allocation of $\pi^{O'}_i$ on the entire set of objects will be written $\pi$ and just called allocation.

Any satisfactory allocation must take into account the agents’ preferences on the objects. Here, we will make the classical assumption that these preferences are numerically additive. Each agent $i$ has a utility function $u_i : 2^O \rightarrow \mathbb{R}^+$ measuring her satisfaction $u_i(\pi)$ when she obtains share $\pi$, which is defined as follows:

$$u_i(\pi) \triangleq \sum_{o_k \in \pi} w(i, o_k),$$

where $w(i, o_k)$ is the weight given by agent $i$ to object $o_k$. This assumption, as restrictive as it may seem, is made by a lot of authors [8, 31, for instance] and is considered a good compromise between expressivity and conciseness.

**Definition 2.1.** An instance of the additive multiagent resource allocation problem (add-MARA instance) is a tuple $I = (N, O, \omega)$, where $N$ and $O$ are as defined above and $\omega : N \times O \rightarrow \mathbb{R}^+$ is a mapping. Here, $\omega(i, o_k)$ is the weight given by agent $i$ to object $o_k$.

We say that the agents’ preferences are **strict on objects** if, for each agent $i$ and each pair of objects $o_k \neq o_j$, we have $w(i, o_k) \neq w(i, o_j)$. Similarly, we say that the agents’ preferences are **strict on shares** if, for each agent $i$ and each pair of shares $\pi \neq \pi'$, we have $u_i(\pi) \neq u_i(\pi')$. Note that strict preferences on shares entail strict preferences on objects; the converse is false.

We will denote by $\mathcal{P}(I)$ the set of allocations for $I$. We will also use the notation $o_1o_2\ldots$ as a shorthand for bundle $(o_1, o_2, \ldots)$.

**Definition 2.2.** Given an agent $i$ and a set of objects $O'$, we will write $\text{best}(O', i) = \arg\max_{o_k \in O'} w(i, o_k)$ the objects from $O'$ having the highest weight for $i$ (they will be called top objects of $i$).

A (sub)-allocation $\pi \in O'$ is said **frustrating** if no agent receives one of her top objects in $\pi$ (formally: $\text{best}(O', i) \cap \pi = \emptyset$ for each agent $i$), and non-frustrating otherwise.

In the following, we will consider a particular way of allocating objects to agents: sequences of sincere choices.

**Definition 2.3.** Let $I = (N, O, w)$ be an add-MARA instance. A sequence of sincere choices (or simply sequence when the context is clear) is a vector of $N^m$. We will denote by $S(I)$ the set of possible sequences for the instance $I$.

Let $\bar{\sigma} \in S(I)$ be a sequence of agents and let $\sigma_t$ be the $t^{th}$ agent of the sequence. $\bar{\sigma}$ is said to generate allocation $\pi$ if and only if $\pi$ can be obtained as a possible result of the non-deterministic Algorithm 1 on input $I$ and $\bar{\sigma}$.

**Algorithm 1:** Execution of a sequence

| Input: | an instance $I = (N, O, w)$ and a sequence $\bar{\sigma} \in S(I)$ |
| Output: | an allocation $\pi \in \mathcal{P}(I)$ |
| 1 | $\pi \leftarrow$ empty allocation (such that $\forall i \in N : \pi_i = \emptyset$); |
| 2 | $O_1 \leftarrow O$; |
| 3 | for $t$ from $1$ to $m$ do |
| 4 | $i \leftarrow \sigma_t$; |
| 5 | choose object $o_t \in \text{best}(O_t, i)$; |
| 6 | $\pi_i \leftarrow \pi_i \cup \{o_t\}$; |
| 7 | $O_{t+1} \leftarrow O_t \setminus \{o_t\}$ |

**Definition 2.4.** An allocation $\pi$ is said to be **sequenceable** if there exists a sequence $\bar{\sigma}$ that generates $\pi$, and non-sequenceable otherwise. For a given instance $I$, we will write $s(I)$ the binary relation defined by $(\bar{\sigma}, \pi) \in s(I)$ if and only if $\pi$ can be generated by $\bar{\sigma}$.

**Example 2.5.** Let $I$ be the instance represented by the following weight matrix:

$$
\begin{bmatrix}
8 & 2 & 1 \\
5 & 1 & 5
\end{bmatrix}
$$

For instance, sequence $(2, 1, 2)$ generates two possible allocations: $(o_2, o_2o_3)$ and $(o_2, o_3o_2)$, depending on whether agent 2 chooses object $o_1$ or $o_3$ that she both prefers. Allocation $(o_2o_3, o_3)$ can be generated by three sequences. Allocations $(o_2o_3, o_2)$ and $(o_3, o_2o_3)$ are non-sequenceable.

For any instance $I$, $|S(I)| = |\mathcal{P}(I)| = n^m$. Also note that the number of objects allocated to an agent by a sequence is the number of times the agent appears in the sequence.

1The algorithm contains an instruction choose spliting the control flow into several branches, building all the allocations generated by $\bar{\sigma}$.

2In this example and the following ones, we represent instances by a matrix where the value at row $i$ and column $o_k$ represents the weight $w(i, o_k)$.
The notion of frustrating allocation and sequenceability were already implicitly present in the work by Brams and King [15], and sequenceability has been extensively studied by Aziz et al. [4] with a focus on sub-classes of sequences (e.g., alternating sequences). However, a fundamental difference is that in our setting, the preferences might be non strict on objects, which entails that the same sequence can yield different allocations (in the worst case, an exponential number), as Example 2.5 shows.

3 SEQUENCEABLE ALLOCATIONS

We have seen in Example 2.5 that some allocations are non-sequenceable. We will now formalize this and give a precise characterization of sequenceable allocations. That is, we will try to identify under which conditions an allocation is achievable by the execution of a sequence of sincere choices. We first start by noticing that in every sequenceable allocation, the first agent of the sequence gets a top object, so every frustrating allocation is non-sequenceable. However, being non-frustrating is not a sufficient condition for an allocation to be sequenceable, as the following example shows:

Example 3.1. Consider this instance:

\[
\begin{pmatrix}
0 & 8 & 2 & 1 \\
2 & 5 & 3 & 1 \\
1 & 4 & 3 & 1 \\
\end{pmatrix}
\]

In allocation \( \pi = (o_1, o_2, o_3) \), each agent receives her top object. However, after \( o_1 \) and \( o_2 \) have been allocated (they must be allocated first by all sequences generating \( \pi \)), the dotted sub-allocation remains. This sub-allocation is obviously non-sequenceable because it is frustrating. Hence \( \pi \) is not sequenceable either.

This property of containing a frustrating sub-allocation exactly characterizes the set of non-sequenceable allocations:

Proposition 1. Let \( I = (N, O, w) \) be an instance and \( \pi \) be an allocation of this instance. The two following statements are equivalent:

(A) \( \pi \) is sequenceable.

(B) No sub-allocation of \( \pi \) is frustrating (in every sub-allocation, at least one agent receives a top object).

Proof. (B) \( \Rightarrow \) (A). Let us suppose that for all subsets of objects \( O' \subseteq O \), at least one agent gets one of her top objects in \( \pi|O' \). We will show that \( \pi \) is sequenceable. Let \( \sigma \) be a sequence of agents and \( \overline{O} \in \{2^O\}^m \) be a sequence of bundles jointly defined as follows:

- \( O_1 = O \) and \( \sigma_1 \) is an agent that receives one of her top objects in \( \pi|O_1 \);
- \( O_{t+1} = O_t \setminus \{o_t\} \), where \( o_t \in \text{best}(O_t, \sigma_t) \) and \( \sigma_t \) is an agent that receives one of her top objects in \( \pi|O_t \), for \( t \geq 1 \).

From the assumption on \( \pi \), we can check that \( \sigma \) is well-defined. Moreover, \( \pi \) is one of the allocations generated by \( \sigma \).

(A) \( \Rightarrow \) (B) by contraposition. Let \( \pi \) be an allocation containing a frustrating sub-allocation \( \pi|O' \). Suppose that there exists a sequence \( \sigma \) generating \( \pi \). We can notice that in a sequence of sincere choices, when an object is allocated to an agent, all the objects that are strictly better for her have already been allocated at a previous step. Let \( o_k \in O' \), and let \( i \) be the agent that receives \( o_k \) in \( \pi \). Since \( \pi|O' \) is frustrating, there is another object \( o_l \in O' \) such that \( w(i, o_l) > w(i, o_k) \). As we have seen, \( o_l \) is necessarily allocated before \( o_k \) in the execution of \( \sigma \). Let \( j \) be the agent who receives \( o_l \). Using the same argument for \( j \) and \( o_l \) we find another object \( o_p \in O' \) allocated before \( o_l \) in the sequence. Iterating this argument, since \( O' \) is finite, we will eventually find an object which has been encountered before. This creates a cycle in the precedence relation of the objects in the execution of the sequence. Contradiction: no sequence can thus generate \( \pi \).

Besides characterizing a sequenceable allocation, the proof of Proposition 1 gives a practical way of checking if an allocation is sequenceable, and, if it is the case, of computing a sequence that generates this allocation.

Proposition 2. Let \( I = (N, O, w) \) be an instance and \( \pi \) be an allocation of this instance. We can decide in time \( O(n \times m^2) \) if \( \pi \) is sequenceable.

The proof is based on the execution of Algorithm 2. This algorithm is similar in spirit to the one proposed by Brams and King [15] but is more general because (i) it can involve non-strict preferences on objects, and (ii) it can conclude with non-sequenceability.

Algorithm 2: Sequencing an allocation

Input: \( I = (N, O, w) \) and \( \pi \in \mathcal{P}(I) \)

Output: a sequence \( \sigma \) generating \( \pi \) or NonSeq

1. \( (\sigma, O') \leftarrow (\varnothing, O) \);
2. for \( t \) from 1 to \( m \) do
3. if \( \exists i \) such that \( \text{best}(O', i) \cap \pi_i \neq \emptyset \) then
4. Append \( i \) to \( \sigma \);
5. let \( o_k \in \text{best}(O', i) \cap \pi_i \);
6. \( O' \leftarrow O' \setminus \{o_k\} \);
7. else return NonSeq;
8. return \( \sigma \);

Proof. We show that Algorithm 2 returns a sequence \( \sigma \) generating the input allocation \( \pi \) if and only if there is one. Suppose that the algorithm returns a sequence \( \sigma \). The sequence of (in the loop from line 2 to line 7), at each step \( t \), \( i = \sigma_t \) can choose an object in \( \pi_i \) that is one of her top objects. Conversely, suppose the algorithm returns NonSeq. Then, at a given step \( t \), \( \forall i \), \( \text{best}(O', i) \cap \pi_i = \emptyset \). By definition, \( \pi|O' \) is therefore, at this step, a frustrating sub-allocation of \( \pi \). By Proposition 1, \( \pi \) is thus non-sequenceable. The loop from line 2 to line 7 runs in time \( O(n \times m) \), because searching for the top objects in the preferences of each agent can be completed in time \( O(n \times m) \). This loop being executed \( m \) times, the algorithm runs in \( O(n \times m^2) \).

4 PARETO-OPTIMALITY

An allocation is Pareto-optimal if no other allocation dominates it. In our context, allocation \( \pi \) dominates allocation \( \pi' \) if for all agent \( i \), \( u_i(\pi') \geq u_i(\pi) \) and \( u_i(\pi') > u_i(\pi) \) for at least one agent \( j \). When an allocation is generated from a sequence, in some sense, a weak form of efficiency is applied to build the allocation: each successive (picking) choice is "locally" optimal. This raises a natural question: is every sequenceable allocation Pareto-optimal?
This question has already been extensively discussed independently by Aziz et al. [3] and Bouveret and Lemaître [13]. We complete the discussion here to give more insights about the implications of the previous results in our framework.

Brams and King [15, Proposition 1] prove the equivalence between sequenceability and Pareto-optimality. However, they have a different notion of Pareto-optimality, because the agents’ preferences are given as linear orders over objects. To be able to compare bundles, these preferences are lifted on subsets using the responsive set extension $\succ_{RS}$. This extension leaves many bundles incomparable and leads to define possible and necessary Pareto-optimality.

Brams and King’s notion is possible Pareto-optimality. Aziz et al. [2] show that, given a linear order $\succ$ on objects and two bundles $\pi$ and $\pi'$, $\pi \succ_{RS} \pi'$ if and only if $u(\pi) > u(\pi')$ for all additive utility functions $u$ compatible with $\succ$ (that is, such that $u(o_i) > u(o_j)$ if and only if $o_i \succ o_j$). This characterization of responsive dominance yields the following reinterpretation of Brams and King’s result: an allocation $\pi$ is sequenceable if and only if for each other allocation $\pi'$, there is a set $u_1, \ldots, u_n$ of additive utility functions, respectively compatible with $\succ_1, \ldots, \succ_n$ such that $u_i(\pi_i) > u_i(\pi'_i)$ for at least one agent $i$.

The latter notion of Pareto-optimality is very weak, because (unlike in our context) the set of additive utility functions is not fixed — we just have to find one that works. Under our stronger notion, the equivalence between sequenceability and Pareto-optimality no longer holds.\(^3\)

**Example 4.1.** Let us consider the following instance:

$$
\begin{pmatrix}
5 & 4 & 2 \\
8 & 2 & 1
\end{pmatrix}
$$

The sequence $(1, 2, 2)$ generates allocation $\pi = (o_1, o_2 o_3)$ giving utilities $(5, 3)$. $\pi$ is then sequenceable but it is dominated by $\pi' = (o_2 o_3, o_1)$, giving utilities $(6, 8)$ (and generated by $(2, 1, 1)$). Observe that, under ordinal linear preferences, $\pi'$ would not dominate $\pi$, but they would be incomparable.

The last example shows that a sequence of sincere choices does not necessarily generate a Pareto-optimal allocation. What about the converse? We can see, as a trivial corollary of the reinterpretation of Brams and King’s result in our terminology, that the answer is positive if the preferences are strict on shares. The following result is more general, because it holds even without this assumption:

**Proposition 3 ([3, 13]).** Every Pareto-optimal allocation is sequenceable.

As already noticed by Aziz et al. [3], the proof follows from an adaptation of Brams and King’s Proposition 1 (necessity part of the proof) [15]. However, we find useful to give the proof, because it is more general than the previous one, and will be reused in subsequent results of this paper. Before giving this proof, we illustrate it on a concrete example [12, Example 5].

**Example 4.2.** Let us consider the following instance:

$$
\begin{pmatrix}
\uparrow 12 & 15 \\
2 & 12 \\
15 & \uparrow 20
\end{pmatrix}
$$

The circled allocation $\pi$ is not sequenceable: indeed, every sequence that could generate it should start with $(3, 1, \ldots)$, leaving the frustrating sub-allocation $\pi'$ in a dotted box.

Let us consider agent 1 for instance. Since the suballocation is frustrating, she does not receive $o_1$ (which is her top object), but agent 2 does. This latter agent, however, does not get her top object, $o_4$, because agent 1 receives it. Obviously, if agent 1 gives $o_4$ to agent 2 and receives $o_3$ in return, we have built a cycle in which each agent gives a regular object and receives a top one. Doing this, we have built an allocation strictly dominating $\pi$.

**Proof.** As stated in the example, we will now prove the contra-position of the proposition: every non-sequenceable allocation is dominated. Let $\pi'$ be a non-sequenceable allocation. From Proposition 1, in a non-sequenceable allocation, there is at least one frustrating sub-allocation. Let $\pi'$ be such a sub-allocation (that can be $\pi$ itself). We will, from $\pi'$, build another sub-allocation dominating it. Let us choose an arbitrary agent $i$ involved in $\pi'$, receiving an object not among her top ones in $\pi'$. Let $o_i$ be a top object of $i$ in $\pi'$, and let $j (\neq i)$ be the unique agent receiving it in $\pi'$. Let $o_j$ be a top object of $j$. We can notice that $o_j \neq o_i$ (otherwise $j$ would obtain one of her top objects and $\pi'$ would not be frustrating). Let $k$ be the unique agent receiving $o_j$ in $\pi'$, and so on. Using this argument iteratively, we form a path starting from $i$ and alternating agents and objects, in which two successive agents and objects are distinct. Since the number of agents and objects is finite, we will eventually encounter an agent which has been encountered at a previous step of the path.

Let $i$ be the first such agent and $o_k$ be the last object seen before her in the sequence (i.e., the unique agent receiving $o_k$). We have built a cycle $i \rightarrow o_k \rightarrow k \rightarrow \cdots i+1 \rightarrow o_i \rightarrow i$ in which all the agents and objects are distinct, and that has at least two agents and two objects. From this cycle, we can modify $\pi'$ to build a new sub-allocation by giving to each agent in the cycle a top object instead of another less preferred object, all the agents not appearing in the cycle being left unchanged. More formally, the following additions in $\pi'$ (and hence in $\pi$) are made: $(i \leftarrow o_k)(i + 1 \leftarrow o_i) \cdots (k \leftarrow o_k)$ where $(i \leftarrow o_j)$ means that $o_j$ is attributed to $i$. The same substitutions operated in $\pi$ yield an allocation $\pi''$ that dominates $\pi$. \(\square\)

**Corollary 4.3.** No frustrating allocation can be Pareto-optimal (equivalently, in every Pareto-optimal allocation, at least one agent receives a top object).

Proposition 3 implies that there exists, for a given instance, three classes of allocations: (1) non-sequenceable (therefore non Pareto-optimal) allocations, (2) sequenceable but non Pareto-optimal allocations, and (3) Pareto-optimal (hence sequenceable) allocations. These three classes define a “scale of efficiency” that can be used to characterize the allocations. What is interesting and new here is the intermediate level. We will see that this scale can be further refined.

\(^3\)Actually, since it is known [1, 21] that testing Pareto-optimality with additive preferences is $\text{coNP}$-complete, and that testing sequenceability is in $P$ (Proposition 2), they cannot be equivalent unless $P = \text{coNP}$.
5 CYCLE DEALS-OPTIMALITY

Pareto-optimality can be thought as a reallocation of objects among agents using improving deals [35, 37], as we have seen, to some extent, in the proof of Proposition 3. Trading cycles or cycle deals constitute a sub-class of deals, which is classical and used, e.g., by Varian [39, page 79] and Lipton et al. [31, Lemma 2.2] in the context of envy-freeness. Trying to link efficiency concepts with various notions of deals is thus a natural idea.

Definition 5.1. Let \((N, O, w)\) be an add-MARMA instance and \(\pi\) be an allocation of this instance. A \((N, M)\)-cycle deal of \(\pi\) is a sequence of transfers of items \(\mu = (\langle \mu_1, O_1 \rangle, \ldots, \langle \mu_N, O_N \rangle)\), where, for each \(j \in \{1, \ldots, N\}\), \(\mu_j\) denotes the \(j\)th agent involved in the sequence and \(\mu_j \in N, O_j \subseteq \pi_j\), and \(|O_j| \leq M\). The allocation \(\pi[\leftarrow \mu]\) resulting from the application of \(\mu\) to \(\pi\) is defined as follows:

- \(\pi[\leftarrow \mu] = \pi_{\mu_1} \setminus O_1 \cup O_1^{-1}\) for \(j \in \{2, \ldots, N\}\);
- \(\pi[\leftarrow \mu] = \pi_{\mu_1} \setminus O_1 \cup O_1; \pi_{\mu_1} \setminus \pi_{\mu_i} = \pi_i \) if \(i \not\in \{\mu_1, \mu_i\}\).

A cycle deal \((\langle \mu_1, O_1 \rangle, \ldots, \langle \mu_N, O_N \rangle)\) will be written \(\pi_1 \rightarrow \pi_2 \rightarrow \ldots \rightarrow \pi_N \rightarrow \pi_1\).

In other words, in a cycle deal (we omit \(N\) and \(M\) when they are not necessary to understand the context), each agent gives a subset of at most \(M\) items from her share to the next agent in the sequence and receives in return a subset from the previous agent. ((\(N, 1\))-cycle deals will be denoted by \(N\)-cycle deals. 2-cycle deals will be called swap-deals. Among these cycle deals, some are more interesting: those where each agent improves her utility by trading objects. More formally, a deal \(d\) will be called weakly improving if \(u_i(\pi[\leftarrow d]) \geq u_i(\pi)\) \(\forall i \in N\) with at least one of these inequalities being strict, and strictly improving if all these inequalities are strict.

Intuitively, if it is possible to improve an allocation by applying an improving cycle deal, then it means that this allocation is inefficient. Reallocation of the items according to the deal will make everyone better-off. It is thus natural to derive a concept of efficiency from this notion of cycle-deal.

Definition 5.2. An allocation is said to be \(>(N, M)\)-cycle optimal (resp. \(\geq(N, M)\)-cycle optimal) if it does not admit any strictly (resp. weakly) improving \((K, M)\)-cycle deal for any \(K \leq N\).

We begin with easy observations. First, \(\geq\)-cycle optimality implies \(\geq\)-cycle optimality, and these two notions become equivalent when the preferences are strict on shares. Moreover, restricting the size of the cycles and the size of the bundles exchanged yield less possible deals and hence lead to weaker optimality notions.

Note that for \(N' \leq N\) and \(M' \leq M\) (at least one of these inequalities being strict), \(>(N', M')\)-cycle-optimality and \(\geq(N', M')\)-cycle-optimality are incomparable. These observations show that cycle-deal optimality notions form a (non-linear) hierarchy of efficiency concepts of diverse strengths. The natural question is whether they can be related to sequenceability and Pareto-optimality. Obviously, Pareto-optimality implies both \(\geq\)-cycle-optimality and \(\geq\)-cycle-optimality. An easy adaptation of the proof of Proposition 3 leads to the following stronger result:

Proposition 4. An allocation \(\pi\) is sequenceable if and only if it is \(\geq\)-cycle optimal (with \(n = |N|\)).

Proof. Let \(\pi\) be a non-sequenceable allocation. Then by Proposition 1, there is at least one frustrating sub-allocation in \(\pi\). Using the same line of arguments as in the proof of Proposition 3 we can build a strictly improving \(k\)-cycle. Hence \(\pi\) is not \(\geq\)-cycle optimal. Conversely, suppose that \(\pi\) admits a strictly improving \(k\)-cycle deal. Then obviously this cycle yields a sub-allocation that is frustrating.

The scale of efficiency introduced in Section 4 can then be refined with a hierarchy of \(\geq\)-cycle optimality notions below sequenceable allocations: Pareto-optimal \(\Rightarrow\) sequenceable \(\equiv \geq\)-\(n\)-cycle optimal \(\Rightarrow \geq\)-(\(n-1\))-cycle optimal \(\Rightarrow \ldots \Rightarrow \geq\)-swap optimal.

As for \(\geq\)-cycle optimality, it forms a parallel hierarchy between Pareto-optimal and non-sequenceable allocations. Note that sequenceability does not involve any \(\geq\)-\(n\)-cycle-optimality. Thus, as soon as \(k < n\), \(\geq-k\)-cycle optimality and sequenceability become incomparable notions.

For instance, for 3 agents, there exist allocations which are \(\geq\)-swap optimal but not sequenceable and the other way around:

\[
\begin{bmatrix}
2 & \dag & 1 \\
\dag & 3 & 1 \\
1 & \dag & 3 \\
\end{bmatrix}
\]

Here the circled allocation is \(\geq\)-swap optimal, but not sequenceable: there exists a strictly improving 3-cycle. At the same time, the dag allocation is sequenceable (by (2, 3, 1, 1)), but not even \(\geq\)-swap optimal, since 1 and 2 may agree on a weakly improving deal.

The observations previously made in this section suggest that, in some situations, the most complex cycle deals could be required to reach Pareto-optimal allocations. This is indeed the case—we now make this claim more precise. Observe that to be involved in a weakly improving cycle deal, each agent must pass at least one item, thus for a \((n, k)\)-cycle deal, we have that \(k \leq m - (n - 1)\) (i.e. the "largest bundle" circulating in this cycle deal can be at most \(m-n+1\)).

The following generic example shows that it is necessary to implement a \((n, m-n+1)\)-cycle to reach a Pareto-optimal allocation.

Here the realization of the deal comes to ensure that no agent holds bundles \(\beta_i\), while agent \(n\) holds bundles \(\beta_{m-n+1}\). Hence everyone enjoys utility 1. This allocation is dominated by the allocation where each agent holds items \(a_{i-1} \cdot \beta_{m-n+1}\). In this allocation, the utilities of agents are instead \((1, 1, \ldots, 2)\). But to obtain \(a_{n-1}\), agent \(n\) must get it from another agent who would only release it if she gets \(a_{1-2}\). The end, agent 1 will only release \(a_1\) if she gets the full bundle \(\beta_1 \cdot \beta_{m-n+1}\). Overall there are \(m\) agents involved in the deal, exchanging up to \(m-n+1\) items. By construction, it is easy to check that no simpler cycle deal (either in terms of number of items or number of agents) would allow to reach this allocation.
Furthermore, there is clearly no other allocation Pareto-dominating the initial allocation.

However, it is important to note that cycle-deals may not be sufficient to reach Pareto-optimal outcomes when there are more items than agents.

Example 5.3. Consider the following example:

\[
\begin{pmatrix}
3 & 6 & \uparrow6 & 0 & 6 & \downarrow4 \\
\downarrow2 & 0 & \downarrow6 & 3 & \uparrow7 & 0 \\
0 & \uparrow5 & 0 & \uparrow4 & 6 & 3
\end{pmatrix}
\]

Note that in the circled allocation, all agents enjoy the same utility, \((9, 9, 9)\), and that it is Pareto-dominated by the dag allocation which induces the vector of utilities \((10, 9, 9)\). We leave it to the reader to check that no swap deal, nor 3-cycle, would be weakly improving. In fact, the only way to reach the dag allocation from this initial allocation would require to implement simultaneously two \((3, 1)\)-cycle deals \((1 \rightarrow 2 \rightarrow 3 and 3 \rightarrow 2 \rightarrow 1)\).

Finally, a corollary of Propositions 2 and 4 is that checking whether an allocation is \(>\times n\)-cycle optimal can be made in polynomial time (by checking whether it is sequenceable).

More generally, we can observe that checking whether an allocation is \((k, k')\)-cycle optimal can be done by iterating over all \(k\)-uples of agents\(^4\), and for each one iterating over all possible transfers involving less than \(k'\) objects. In total, there are \(k!\binom{m}{k}\) \(k\)-uples of agents (which is upper-bounded by \(nk^{k+1}\)). For each \(k\)-uple, there are at most \(\sum_{k'=0}^{\min\{k, m\}} \binom{m}{k'} k^{k'}\) possible transfers, which is again upper-bounded by \((1 + m)k^{k'}\). Hence, in total, checking whether an allocation is \((k, k')\)-cycle optimal can be done in time \(O(nk^{k+1} \times (1 + m)k^{k'})\). This is polynomial in \(n\) and \(m\) if both \(k\) and \(k'\) are bounded (as for swap deals).

### 6 RESTRICTED DOMAINS

We now study the impact of several preference restrictions on the hierarchy of efficiency notions introduced in Section 5.

**Strict preferences on objects.** When the preferences are strict on objects, then obviously every sequence generates exactly one allocation. The following proposition is stronger and shows that the converse is also true:

**Proposition 5.** Preferences are strict on objects iff \(s(I)\) is a mapping from \(S(I)\) to \(P(I)\).

Proof. If preferences are strict on objects, each agent has only one possible choice at her turn in the sequence of sincere choices and hence every sequence generates one and only one allocation.

Conversely, if preferences are not strict on objects, at least one agent (suppose w.l.o.g. agent 1) gives the same weight to two different objects \(o_k, o_l\). Suppose that exactly \(t\) objects are ranked above \(o_k\) and \(o_l\). Then the sequence where agent 1 picks \(t + 1\) items in a row, and 2 picks the \(m - t - 1\) remaining ones obviously generates two allocations, depending on agent 1’s choice at step \(t + 1\). □

\(^4\)We do not need to also run through all cycles of strictly less than \(k\) agents: such a cycle can be simulated just by appending at the end some agents whose role is just to pass the objects they receive to the next agent.

**Same order preferences.** We say that the agents have same order preferences \([12]\) if there is a permutation \(\pi: O \mapsto O\) such that for each agent \(i\) and each pair of objects \(o_k, o_l\), if \(\pi(o_k) < \pi(o_l)\) then \(w(i, \pi(o_k)) \geq w(i, \pi(o_l))\).

**Proposition 6.** All the allocations of an instance with same order preferences are sequenceable (and actually cycle-deal optimal).

Conversely, if all the allocations of an instance are sequenceable, then this instance has same order preferences.

Proof. Suppose that the agents have same order preferences, and let \(\pi\) be an arbitrary allocation. In every sub-allocation of \(\pi\) at least one agent obtains a top object (because the preference order is the same among agents) and hence cannot be frustrating. By Proposition 1, \(\pi\) is sequenceable.

Conversely, let us assume for contradiction that there are two distinct objects \(o_k, o_l\) and two distinct agents \(i, j\) such that \(w(i, o_k) > w(i, o_l)\) and \(w(j, o_k) < w(j, o_l)\). The sub-allocation \(\pi' = (o_k, o_l)\) such that \(\pi'_i(o_k, o_l) = \{o_l\}\) and \(\pi'_j(o_k, o_l) = \{o_k\}\) is frustrating. By Proposition 1, every allocation \(\pi\) containing this frustrating sub-allocation is non-sequenceable. □

Let us now characterize the instances for which \(s(I)\) is a one-to-one correspondence.

**Proposition 7.** For a given instance \(I\), the following two statements are equivalent.

(A) Preferences are strict on objects and in the same order.

(B) The relation \(s(I)\) is a one-to-one correspondence.

The proof is a consequence of Propositions 5 and 6.

**Single-peaked preferences.** An interesting domain restriction are single-peaked preferences \([10, 22]\), which, beyond voting, is also relevant in resource allocation settings \([6, 20]\). Formally, in this context, single-peakedness can be defined as follows.

There exists a linear order \(\succ\) over the set of objects \(O\). Let \(top(i)\) be the preferred object of \(i\). An agent \(i\) has single-peaked preferences wrt. \(\succ\) if, for any two objects \((o_k, o_l) \in O\) such that either \(top(i) \succ o_k\) or \(o_l \succ o_k\), \(o_l \succ o_k\) (i.e., lying on the same “side” of the agent’s peak), it is the case that \(i\) prefers \(o_l\) over \(o_k\).

Interestingly, when preferences are single-peaked, the hierarchy of \(n\)-cycle optimality collapses at the second level:

**Proposition 8.** If the preferences are single-peaked and additive, then an allocation \(\pi\) is \(\geq n\)-cycle optimal iff it is swap-optimal.

Proof. ([20, revisited]) First, note that \(\geq n\)-cycle optimality trivially implies swap-optimality. Let us now show the converse.

Let us consider for the sake of contradiction an allocation \(\pi\) that is swap-optimal and such that there exists a \(\geq k\)-cycle \(\mu\), with \(k \leq n\). Without loss of generality, let us suppose that \(\mu = \langle (\mu_1, \{o_k\}), \ldots, (\mu_k, \{o_k\}) \rangle\). We show by induction on \(k\), the length of \(\mu\), that such a cycle can not exist.

**Base case:** \(k = 2\) A 1-cycle of length \(k = 2\) is a swap-deal but as \(\pi\) is swap-optimal, no improving swap-deal exists in \(\pi\) hence the contradiction.

**Induction step:** Let us assume that for each \(k'\) such that \(2 \leq k' \leq k - 1\), no \(\geq k'\)-cycle exists in \(\pi\) and let us show that no cycle of length \(k\) exists.
To exhibit a contradiction we will need to use the following necessary condition [7]: to be single-peaked, a profile \( U \) needs to be worst-restricted, i.e. for any triple of objects \( O = (o_a, o_b, o_c) \in O^3 \) there always exists an object \( o_j \in O \) such that there exists an agent \( i \) with \( o_j \notin \arg\min_{o_k \in O} w_i(o_k) \) [36].

Because \( \mu \) is a \( \geq-k \)-cycle, for all agent \( \mu_0 \neq \mu_1 \) involved in \( \mu \) we have \( o_{i-1} \succ_o \mu_0 \) and \( o_{j} \succ_o \mu_1 \). As no \( \geq-k' \)-cycle exists, with \( k' < k \), for all agents \( \mu_0 \neq \mu_1 \) involved in \( \mu \) and for all objects \( o_j \in \mu_0 \), \( o_j \neq o_{i-1} \), we have \( o_j \succ_o \mu_1 \). Moreover for all objects \( o_j \in \mu_0 \), \( o_j \neq o_1 \) and \( o_j \neq o_k \), we have \( o_j \succ_o \mu_1 \). If the preferences do not respect these conditions, a \( \geq-k \)-cycle exists with \( k' < k \).

Because the profile is worst-restricted, for all the triple of objects \( O \in \{o_1, \ldots, o_k\} \), at most two resources of \( O \) can be ranked last among \( O \) by the agents. Let us call \( o_o \) one of these objects ranked last by agent \( \mu_l \) and held by agent \( \mu_w \). Thanks to the previous paragraph, we know that best(\( O, \mu_w \)) = \( o_{w-1} \) and so, because her preferences are single-peaked, \( \mu_w \) puts \( o_{w+1} \) in last position among \( o_{w-1}, o_w, o_{w+1} \). The same holds for agent \( \mu_{w+1} \) who ranks \( o_w \) in last position among \( o_{w-1}, o_w, o_{w+1} \) (because top(\( \mu_{w+1} \)) = \( o_w \)).

Therefore when we focus only on the three objects \( o_{w-1}, o_w, o_{w+1} \), each of them is ranked last among them by one agent which violates the condition of worst-restriction. The contradiction is set, no \( \geq-k \)-cycle exists in \( \pi \). □

Together with Proposition 4, Proposition 8 gives another interpretation of sequenceability in this domain:

**Corollary 6.1.** If preferences are single-peaked (and additive), then an allocation \( \pi \) is sequenceable if and only if it is swap-optimal.

Proposition 1 by Damamme et al. [20] is much stronger than our Corollary 6.1, as it shows that swap-optimality is actually equivalent to Pareto-efficiency when each agent receives a single resource. Unfortunately, in our context where each agent can receive several items, this is no longer the case, as the following example shows:

**Example 6.2.** Consider this instance, single-peaked with respect to \( 1 \succ \cdots \succ 6 \):

\[
\begin{bmatrix}
1 & \dagger & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & \dagger & 4 & \dagger & 6 & 3
\end{bmatrix}
\]

The circled allocation is swap-optimal, but Pareto-dominated by the allocation marked with daggers.

**7 ENVY-FREEDOM AND CEEI**

The use of sequences of sincere choices can also be motivated by the search for a *fair* allocation protocol. Here, we will focus on two fairness properties and analyze their link with sequenceability.

The first of these notions is probably one of the most prominent fairness properties: envy-freeness [25, 38, 39].

**Definition 7.1.** Let \( I \) be an add-MARA instance and \( \pi \) be an allocation. \( \pi \) verifies the envy-freeness property (or is simply envy-free), when \( u_i(\pi) \geq u_j(\pi) \), \( \forall (i, j) \in N^2 \) (no agent strictly prefers the share of any other agent).

The notion of competitive equilibrium is an old and well-known concept in economics [24, 40]. If equal incomes are imposed among the stakeholders, this concept becomes the competitive equilibrium from equal incomes [32], yielding a very strong fairness concept that has been recently explored both in artificial intelligence and in economics [12, 19, 33].

**Definition 7.2.** Let \( I = (N, O, w) \) be an add-MARA instance, \( \pi \) an allocation, and \( p \in [0, 1]^m \) a vector of prices. A pair \((\pi, p)\) is said to form a competitive equilibrium from equal incomes (CEEI) if

\[
\forall i \in N : \pi_i \in \arg\max_{\pi \subseteq O} \left\{ u_i(\pi) : \sum_{o_k \in \pi} p_k \leq 1 \right\}.
\]

In other words, \( \pi_i \) is one of the maximal shares that \( i \) can buy with a budget of 1, given that the price of each object \( o_k \) is \( p_k \).

We will say that allocation \( \pi \) is a CEEI if there exists a vector \( p \) such that \((\pi, p)\) forms a CEEI.

As Bouvier and Lemaître [12] and Brânzei et al. [18] have shown, with additive preferences, every CEEI allocation is envy-free. In this section, we investigate the question of whether an envy-free or CEEI allocation is necessarily sequenceable. For envy-freeness, the answer is negative.

**Proposition 9.** There exist non-sequenceable envy-free allocations, even if the agents’ preferences are strict on shares.

**Proof.** A counterexample with strict preferences on shares is given in Example 4.2 above, for which we can check that the circled allocation \( \pi \) is envy-free and non-sequenceable. □

Concerning CEEI, it is already well known that any CEEI allocation is Pareto-optimal (hence sequenceable) if the preferences are strict on shares [12]. This is also a consequence of the First Welfare Theorem introduced by Babaioff et al. [5] for indivisible goods.

However, surprisingly, this result does not hold anymore if the preferences are not strict on shares, as the following example shows:

\[
\begin{bmatrix}
1 & \dagger & \dagger & \dagger & \dagger & \dagger & \dagger \\
2 & 3 & 4 & 1 & \dagger & \dagger & \dagger
\end{bmatrix}
\]

The circled allocation is CEEI (with prices 0.5, 1, 1, 0.5) but is ordinarily necessary (hence also additively) dominated by the allocation marked with \( \dagger \).

In spite of this negative result, we can still guarantee a certain level of efficiency for CEEI allocations:

**Proposition 10.** Every CEEI allocation is sequenceable.

**Proof.** We will show that no allocation can be at the same time non-sequenceable and CEEI. Let \( \pi \) be a non-sequenceable allocation. We can use the same terms and notations than in the proof of Proposition 3, especially concerning the dominance cycle.

Let \( C \) be the set of agents concerned by the cycle. \( \pi \) contains the following shares:

\[
\begin{align*}
\pi_i &= \{o_k\} \cup \tau_i \\
\pi_{i+1} &= \{o_l\} \cup \tau_{i+1} \\
&\vdots \\
\pi_k &= \{o_{k-1}\} \cup \tau_k
\end{align*}
\]

whereas the allocation \( \pi' \) that dominates it, contains:

\[
\begin{align*}
\pi_i' &= \{o_1\} \cup \tau_i \\
\pi_{i+1}' &= \{o_1\} \cup \tau_{i+1} \\
&\vdots \\
\pi_k' &= \{o_k\} \cup \tau_k
\end{align*}
\]

the other shares being unchanged from \( \pi \) to \( \pi' \).

Suppose that \( \pi \) is CEEI. This allocation must satisfy two kinds of constraints. First, \( \pi \) must satisfy the price constraint. If we write \( p(\pi) = \sum_{o_k \in \pi} p_k \), we have, \( \forall i \in C, p(\pi_i) \leq 1 \) (1).
We have exhibited in Sections 4 and 5 a “hierarchy of allocation efficiency” made of several steps: Pareto-optimal (PO), sequenceable (Seq), (cycle-deal-optimal), non-sequenceable (NS). A natural question is to know, for a given instance, which proportion of allocations are located at each level of the scale. We give a first answer by experimentally studying the distribution of allocations between the different levels. For cycle-deal optimality, we focus on the simplest type of deals, namely, >-swap-deals. We thus have a linear scale of efficiency concepts, from the strongest to the weakest: PO → Seq → Swap → NS. We also analyze the relation between efficiency and various notions of fairness by linking this latter scale with the 6-level scale of fairness introduced by Bouveret and Lemaître [12]:

\( \text{PO} \rightarrow \text{Envelope-Freeness (EF)} \rightarrow \text{min-max share (mMS)} \rightarrow \text{proportionality (PFS)} \rightarrow \text{max-min share (MMS)} \rightarrow \text{NS} \).

For both models, a set of weights are uniformly drawn in the interval [0, 100] and the instances are then normalized. For the second model, these weights are reordered afterwards to make the preferences single-peaked. For each instance, we generate all 6561 allocations, and identify for each of them the highest level of fairness and efficiency satisfied. The average number of allocations with min-max interval is plotted as a box for each level on a logarithmic scale in Figure 1.

Note that some fairness and efficiency tests require to solve NP-hard or coNP-hard problems (MMS, mMS, and PO tests). These tests are delegated to an external ILP solver. This is especially interesting for the CEEI test which is known to be NP-hard [18], and for which, to the best of our knowledge, no practical method had been described before. The implementation is available as a fully documented and tested Free Python library.\(^5\)

We note several interesting facts. First, a majority of allocations do not have any efficiency nor fairness property (first black bar on the left). Second, the distribution of allocations on the efficiency scale seems to be related to the fairness criteria: a higher proportion of swap-optimal or sequenceable allocations are found among envy-free allocations than among allocations that do not satisfy any fairness property, and for CEEI allocations, there are even more Pareto-optimal allocations than just sequenceable ones. Lastly, the absence of vertical bar for swap-optimality in the experiments concerning single-peaked preferences confirms the results of Corollary 6.1: in this context, no allocation can be swap-optimal but not sequenceable; hence, all the allocations that are swap-optimal are contained in the bars concerning sequenceable or Pareto-optimal allocations. Similarly, the absence of bars for swap-optimality and NS (non-sequenceable) in both graphs for the CEEI case confirms the result of Proposition 10.

8 EXPERIMENTS

We have exhibited in Sections 4 and 5 a “hierarchy of allocation efficiency” made of several steps: Pareto-optimal (PO), sequenceable (Seq), (cycle-deal-optimal), non-sequenceable (NS). A natural question is to know, for a given instance, which proportion of allocations are located at each level of the scale. We give a first answer by experimentally studying the distribution of allocations between the different levels. For cycle-deal optimality, we focus on the simplest type of deals, namely, >-swap-deals. We thus have a linear scale of efficiency concepts, from the strongest to the weakest: PO → Seq → Swap → NS. We also analyze the relation between efficiency and various notions of fairness by linking this latter scale with the 6-level scale of fairness introduced by Bouveret and Lemaître [12]:

\( \text{PO} \rightarrow \text{Envelope-Freeness (EF)} \rightarrow \text{min-max share (mMS)} \rightarrow \text{proportionality (PFS)} \rightarrow \text{max-min share (MMS)} \rightarrow \text{NS} \).

For both models, a set of weights are uniformly drawn in the interval [0, 100] and the instances are then normalized. For the second model, these weights are reordered afterwards to make the preferences single-peaked. For each instance, we generate all 6561 allocations, and identify for each of them the highest level of fairness and efficiency satisfied. The average number of allocations with min-max interval is plotted as a box for each level on a logarithmic scale in Figure 1.

Note that some fairness and efficiency tests require to solve NP-hard or coNP-hard problems (MMS, mMS, and PO tests). These tests are delegated to an external ILP solver. This is especially interesting for the CEEI test which is known to be NP-hard [18], and for which, to the best of our knowledge, no practical method had been described before. The implementation is available as a fully documented and tested Free Python library.\(^5\)

We note several interesting facts. First, a majority of allocations do not have any efficiency nor fairness property (first black bar on the left). Second, the distribution of allocations on the efficiency scale seems to be related to the fairness criteria: a higher proportion of swap-optimal or sequenceable allocations are found among envy-free allocations than among allocations that do not satisfy any fairness property, and for CEEI allocations, there are even more Pareto-optimal allocations than just sequenceable ones. Lastly, the absence of vertical bar for swap-optimality in the experiments concerning single-peaked preferences confirms the results of Corollary 6.1: in this context, no allocation can be swap-optimal but not Sequenceable; hence, all the allocations that are swap-optimal are contained in the bars concerning sequenceable or Pareto-optimal allocations. Similarly, the absence of bars for swap-optimality and NS (non-sequenceable) in both graphs for the CEEI case confirms the result of Proposition 10.

9 CONCLUSION

In this paper, we have shown that picking sequences and cycle-deals can be reinterpreted to form a rich hierarchy of efficiency concepts. Many interesting questions remain open, such as the complexity of computing cycle-deals or the link between efficiency concepts and social welfare. One could also think of further extending the efficiency hierarchy by studying restrictions on possible sequences (e.g. alternating) or extending the types of deals to non-cyclic ones.

\(^5\)Available at: https://gricad-gitlab.univ-grenoble-alpes.fr/bouveres/fairdiv.
ACKNOWLEDGMENTS
This work is partially supported by the ANR project 14-CE24-0007-01 - CoCoRiCo-CoDec.

REFERENCES