

Envy-Freeness Relaxations for the Fair Division of Indivisible Resources

Master Thesis of

Simon Höfer

At the Department of Informatics and Mathematics
Chair of Theoretical Computer Science



Reviewers: Prof. Dr. Ignaz Rutter
Prof. Dr. Tobias Harks
Advisors: Prof. Dr. Ignaz Rutter
Prof. Dr. Tobias Harks

Time Period: 15 December 2025 – 15 June 2026

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text. The use of Large Language Models (LLMs) has been fully disclosed and documented in a dedicated chapter.

Passau, October 11, 2026

Abstract

Envy-free (EF) allocations of a set of indivisible resources do not generally exist. That is, there are instances where, for any distribution of the resources among the agents, there is an agent i that values their own bundle as strictly less than the bundle some other agent j has received. Therefore, we study relaxations of EF where envy is permitted. Two of the most prominent relaxations permit that some agent i may envy the bundle X_j of another agent j as long as there is a subset $T \subseteq X_j$ such that $X_j \setminus \{x\}$ is not envied by i for all $x \in T$. The more permissive of the two relaxations is *EF1*, which only requires $|T| \geq 1$. There are multiple algorithms known that compute such EF1 allocations under different restrictions, including instances only using additive valuations. The more demanding relaxation is *EFX*, which requires that $T = X_j$, so removing any resource eliminates the envy towards X_j . There are a few instances known for which EFX does not exist, as well as a few restrictions under which the existence of EFX is guaranteed. In this thesis, we restrict the fair division instances to additive valuations, for which the question of whether EFX allocations always exist is open. There is, in a sense, a large gap between results for EF1 and EFX, but also in their definitions. In this thesis, we investigate this gap by parameterizing $|T|$ but mainly $|X_j \setminus T|$. To this end, a set X_j is *t-tolerated* if for every envying agent there are at most t resources that do not eliminate envy when removed. A *t-tolerant* allocation is an allocation where all bundles are *t-tolerated*. We extend two algorithms that produce EF1 allocations and investigate for what t they are guaranteed to produce *t-tolerant* allocations. The first algorithm, envy-cycle elimination, has already been partially extended to the EFX setting through the use of *g*-champion edges. We further generalize the concept of *g*-champion edges to allow for more general Pareto improvements using *t*-tolerance. However, we are only able to show that we are guaranteed to find $\lceil \frac{m-2}{n} \rceil$ -tolerant EFX allocations while also presenting an instance where a *t*-tolerance of $t \geq \lceil \frac{m-2}{n} \rceil$ is required for Pareto improvements using this extended envy-cycle elimination. The second algorithm we consider is round-robin. We adapt it by allowing agents to choose their resources in a different order each round, in a manner such that envious agents always get to choose before the agents they envy. This upper-bounds envy between two agents with the difference in value between two resources chosen in the same round r_g . This is a better fairness guarantee than that given by regular round-robin, where envy is merely bounded by the value of the highest-valued resource. We spend a significant part of this work trying to find a better guarantee for *t*-tolerance for the improved round-robin on permutation instances where all valuations are permutations of $[m]$. Unfortunately, we are only able to find a sufficient bound on the number of resources which do not eliminate envy that are chosen after round r_g . To highlight that this is a limitation of the improved round-robin, we construct a counterexample where a tolerance of $t \geq \lceil \frac{m}{n} \rceil - 2$ is required.

Deutsche Zusammenfassung

Neidfreie (EF) Allokationen einer Menge unteilbarer Ressourcen existieren im Allgemeinen nicht. Folglich gibt es Instanzen, in denen für jede Verteilung der Ressourcen unter den Agenten ein Agent i existiert, der sein eigenes Bündel strikt schlechter bewertet als das Bündel eines anderen Agenten j . Daher untersuchen wir Relaxierungen von EF, bei denen Neid zulässig ist. Zwei der bekanntesten Relaxierungen erlauben es einem Agenten i , das Bündel X_j eines anderen Agenten j zu beneiden, solange eine Teilmenge $T \subseteq X_j$ existiert, sodass für alle Güter $x \in T$ der Agent i die Menge $X_j \setminus \{x\}$ nicht beneidet. Die weniger restriktive der beiden Relaxierungen ist EF1, welche lediglich $|T| \geq 1$ erfordert. Es sind mehrere Algorithmen bekannt, die solche EF1-Allokationen unter verschiedenen Einschränkungen berechnen – einschließlich Instanzen, die ausschließlich additive Bewertungsfunktionen nutzen. Die restriktivere Relaxierung ist EFX. Sie setzt $T = X_j$ voraus, sodass das Entfernen jeder beliebigen einzelnen Ressource den Neid gegenüber X_j eliminiert. Während für einige Instanzen bekannt ist, dass keine EFX-Allokation existiert, ist die Existenzfrage unter der Einschränkung rein additiver Bewertungsfunktionen nach wie vor offen. Zwischen den Resultaten für EF1 und EFX, aber auch in deren Definitionen, klafft somit eine erhebliche Lücke. In dieser Arbeit untersuchen wir diese Lücke, indem wir $|T|$, hauptsächlich jedoch $|X_j \setminus T|$, parametrisieren. Genauer gesagt nennen wir eine Menge X_j dann t -toleriert, wenn es für jeden Agenten, welcher X_j beneidet, höchstens t Ressourcen gibt, deren individuelles Entfernen den Neid nicht eliminiert. Eine t -tolerante Allokation ist dementsprechend eine Allokation, bei der alle Bündel t -toleriert sind. Wir erweitern zwei Algorithmen zur Berechnung von EF1-Allokationen und untersuchen, für welches t sie stets t -tolerante Allokationen ausgeben. Der erste Algorithmus, die Envy-Cycle-Elimination, wurde in der Literatur bereits durch die Verwendung von g -Champion-Kanten erweitert, um einige der bekannteren Resultate für EFX-Allokationen zu beweisen. Wir verallgemeinern das Konzept der g -Champion-Kanten mithilfe der t -Toleranz weiter, um zusätzliche Pareto-Verbesserungen zu ermöglichen. Wir zeigen jedoch nur, dass es auf diese Weise möglich ist, fast triviale $\lceil \frac{m-2}{n} \rceil$ -tolerante Allokationen zu finden. Darüber hinaus präsentieren wir ein Gegenbeispiel, welches demonstriert, dass in manchen Fällen eine t -Toleranz von $t \geq \lceil \frac{m-2}{n} \rceil$ notwendig ist, um die erweiterten g -Champion-Kanten für Pareto-Verbesserungen nutzen zu können. Als zweiten Algorithmus betrachten wir Round-Robin. Wir passen diesen so an, dass die Agenten ihre Ressourcen in jeder Runde in einer variierenden Reihenfolge wählen dürfen – und zwar so, dass neidische Agenten immer vor denjenigen Agenten wählen dürfen, die sie beneiden. Dies beschränkt den Neid von oben durch die Differenz der Werte zweier in derselben Runde r_g gewählter Ressourcen. Dies stellt eine verbesserte Schranke zum regulären Round-Robin-Algorithmus dar, bei dem Neid lediglich durch den Wert der höchstbewerteten Resource beschränkt ist. Ein signifikanter Teil der Arbeit ist der Versuch, diese bessere Neidschranke auf eine verbesserte Toleranzgarantie für Permutationsinstanzen zu übertragen, in denen die Bewertungsfunktionen Permutationen von $[m]$ sind. Leider ist es nur möglich, die Anzahl der Ressourcen, die Neid nicht eliminieren, zu beschränken, wenn diese nach der Runde r_g gewählt werden. Um zu zeigen, dass dies eine Beschränkung des Algorithmus ist, geben wir anschließend eine Konstruktionsbeschreibung für ein Gegenbeispiel, in welchem eine Toleranz von $t \geq \lceil \frac{m}{n} \rceil - 2$ benötigt wird.

Contents

1	Introduction	1
1.1	EFX vs EFX_0	4
1.2	Related Work	4
1.3	Relaxations in Between EFX and EF1	6
1.4	Contributions and Outline	8
2	Pareto improving toward t-Tolerant EFX Using g-championing	11
2.1	Envy-Cycle Elimination	11
2.2	g -Championship	13
2.2.1	Results Using g -Championing	15
2.3	t -Tolerated g -Championship	16
2.4	Limitations of the approach	19
2.4.1	An Initial Bound	20
2.4.2	Counterexample Background	23
2.4.3	$(\delta - 1)$ -Tolerance Does Not Suffice for Pareto Improvements Using PI Cycles	27
2.4.4	Non-Degenerate Instance	29
2.4.5	Conclusion and Further Questions	31
3	Improving the Round-Robin Algorithm	35
3.1	Classic Round-Robin	36
3.2	Order-Varying Round-Robin	40
3.2.1	Intuition: Bounding Tolerance in Permutation Instances	46
3.3	Envy-Reducing Round-Robin	50
3.3.1	Redefining the gap	51
3.3.2	Bounding Envy and Envy Potential	55
3.3.2.1	Bounding Envy and Envy Potential over Multiple Rounds	57
3.3.3	Attempting to Bound t -Tolerance	62
3.3.3.1	Case: $ \mathcal{R} = 0$	66
3.3.3.2	Case: $ \mathcal{R} > 0$	67
3.3.4	Constructing an Example for $t \geq \lceil \frac{m}{n} \rceil - 2$	72
3.4	Conclusion and Further Questions	73
4	Conclusion	75
	Bibliography	77
	Declaration of LLM Usage	81
	Appendix	83
A	Example: 4-Tolerant EFX Allocation Computed by Envy-Reducing Round- Robin on a Permutation Instance	83

1. Introduction

We live in a world of limited resources. Resources whose values people generally do not agree on. Therefore, it is interesting to research if and how we can distribute any set of resources in a manner that is „fair“. The two most prominent notions of fairness are *proportionality* and *envy-freeness* [ABFRV22]. Given a set of resources, we partition them into bundles and allocate them to the people, or as we call them, *agents*. For the notion of proportionality, the name already implies that it is fulfilled only if a proportion is given. The proportion in question is the ratio between the value that every agent assigns to the set it was assigned and the value it assigns to the set of all resources. For an allocation to be fair under this notion, this proportion is at least $\frac{1}{n}$, where n is the amount of agents, as one may intuitively expect. Note that under this notion, the question of whether an agent perceives an allocation as fair depends only on their allocated set. However, under proportionality, an agent i perceives an allocation as fair even if they believe other agents are assigned higher-valued bundles. For example, agent i might value its own bundle at just a $\frac{1}{n}$ of the total value while there might be some other agent j whose bundle i values at $\frac{2}{n}$ of the total value. If a person, who is not completely magnanimous, believes that another person has received something worth twice as much as what they have, we expect this person to experience envy and to believe that the given allocation is unfair. This motivates the strictly stronger fairness notion of envy-freeness (EF). An allocation is EF if no agent believes that any other agent is assigned a bundle strictly better than their own. It is easy to see that all envy-free allocations are proportional, but not vice versa. In this thesis, we approach fairness from the notion of envy-freeness.

A classic example of these kinds of problems is the cake-cutting problem. Here, the goal is to distribute a single divisible resource (a cake) between two or more agents in a way that is fair or, for our purposes, EF. Keep in mind that agents may prefer different parts of the resource. E.g., one may like the chocolate, another may like fruit, and a third might like everything quite the same except for the pieces with too much whipped cream. For two agents, it is not only always possible to find an EF allocation, but there is also a protocol to find such an allocation [LR24] that people who grew up with siblings may know. The so-called Cut & Choose Protocol only has two steps: 1. The first agent divides/cuts the resource into two pieces that they value equally. 2. The second agent then chooses the piece they prefer and gives the other to the first agent. The second agent does not experience envy because they, by definition, take the piece they prefer. The first agent does not experience envy because neither piece is valued strictly better than the other. Notice here that this depends on the fact that the resource is arbitrarily divisible.

In reality, there are some resources that cannot be divided, at least not without significantly reducing their value (e.g., a rare book). This forms the setting with which we will work. To formalize this, we are given a set of resources $[m]$ and agents $[n]$, where every agent i is equipped with a valuation function $v_i : \mathcal{P}([m]) \rightarrow \mathbb{R}$. We use $[k]$ with $k \in \mathbb{N}_+$ as a shorthand for the set $\{n \in \mathbb{N}_+ | n \leq k\}$. Before we can define our notion of what a fair allocation is, we need to define what an allocation is.

Definition 1.1 (Allocation). *Given resources $[m]$ and agents $[n]$, an allocation is a tuple $\mathbf{X} = (X_1, \dots, X_n, P)$ where for all $i, j \in [n], i \neq j$, $X_i \cap X_j = \emptyset$ and $X_i \cap P = \emptyset$ and $\bigcup_{i \in [n]} X_i \cup P = [m]$. The allocation $\mathbf{X}$ is total if $P = \emptyset$ and it is partial if $P \neq \emptyset$.*

For all $i \in [n]$, we refer to X_i of an allocation $\mathbf{X}$ as a bundle. When explicitly writing our allocations, we omit the unallocated resources P as they are implicitly defined by the bundles. Note that we do not always specify whether an allocation is total or partial. In definitions we are usually fine with either, while for explicit results and their proofs, assume any allocation is total if not specified otherwise. We can now define our first fairness notion:

Definition 1.2 (Envy-free — EF). *Given an allocation $\mathbf{X}$, an agent i envies the bundle $S \subseteq [m]$ if $v_i(X_i) < v_i(S)$. An allocation is EF if and only if for all $i, j \in [n]$, i does not envy X_j .*

EF allocations are known to exist when we are able to divide the resources arbitrarily, as mentioned with the cake-cutting protocol above. For indivisible resources, however, there are instances where no such allocation exists. For example:

Example 1.3.

$$m = 1, n = 2, v_i(S) = \begin{cases} 1 & \text{if } 1 \in S \\ 0 & \text{otherwise} \end{cases}, \quad i \in [2]$$

No matter which agent gets the only resource, the other agent is always envious. If we want to always be able to find a fair allocation, we cannot use EF as a fairness notion. Instead, we may try to find an allocation that is, in some sense, close to being envy-free.

Definition 1.4 (Envy-free up to one good — EF1). *An allocation $\mathbf{X}$ is envy-free up to one good if and only if for all agents i, j where i envies j , there exists $g \in X_j$ such that*

$$v_i(X_i) \geq v_i(X_j \setminus \{g\}).$$

For simplicity, we often do not write braces for singleton sets. So we will denote the union $S \cup \{g\}$ of a set S with a singular resource g as $S + g$, the difference $S \setminus \{g\}$ as $S - g$ and for a valuation v_i , we write $v_i(g)$ for $v_i(\{g\})$.

When discussing existence results for EF1, we restrict the valuations as follows if not explicitly stated otherwise:

Definition 1.5 (General valuations). *A general valuation v is a valuation that is both*

- monotone, i.e., $v(S) \leq v(T)$ for all $S \subseteq T \subseteq [m]$,
- and normalized, i.e., $v(\emptyset) = 0$.

For the existence of EF1 allocations, monotone valuations are sufficient ([CKM⁺19]). We restrict the valuations even further for our own result. There, we use *additive valuations* where for every $i \in [n], g \in [m]$ there is a fixed value $v_i(g)$, such that for all $S \subseteq [m]$, $v_i(S) = \sum_{g \in S} v_i(g)$.

One may argue that it is fairly easy to find fair allocations. Simply assign every agent an empty bundle. This motivates the fact that we are not only interested in fairness, but also in some form of economic efficiency. In other words, we want to know if our fair allocations, in some sense, waste the value of resources. The two notions of efficiency commonly referenced in the work of EF and its variants are *Pareto Optimality* and *Nash welfare*. We now define both, starting with Pareto optimality, which states that there is no way to increase the value of an agent without decreasing the value of another.

Definition 1.6 (Pareto Domination). *An allocation $\mathbf{X}$ Pareto dominates an allocation $\mathbf{Y}$ if for all agents $i \in [n]$, $v_i(X_i) \geq v_i(Y_i)$, and for at least one agent $j \in [n]$, $v_j(X_j) > v_j(Y_j)$. An allocation $\mathbf{X}$ is Pareto optimal if, for the given resources, agents, and valuations, there is no allocation Pareto dominating $\mathbf{X}$.*

The second notion is Nash welfare or, as it is sometimes called, Nash social welfare. Social welfare is more commonly associated with Bentham’s Utilitarianism. There we consider the utility of objects and actions¹ against interest of a community defined as the sum of the interests of the individuals that compose it [Ben89, Chapter I, Sections III-V]. Nash welfare takes the same idea, but replaces the sum with the product of the valuations.

Definition 1.7 (Nash welfare). *The Nash welfare of an allocation $\mathbf{X}$ is given as $\prod_{i \in [n]} v_i(X_i)$.*

The use of the product somewhat biases Nash welfare towards fairness. This can intuitively be seen when considering an allocation where an agent evaluates their bundle as 0. The low value of one agent has more of an impact on Nash welfare. Caragiannis et al. [CKM⁺19] express this by stating: „Informally, it hits a sweet spot between Bentham’s utilitarian notion of social welfare — maximize the sum of utilities — and the egalitarian notion of Rawls — maximize the minimum utility.“ Furthermore, they showed that maximum Nash welfare allocations (MNW) are both Pareto optimal and EF1 for additive valuations [CKM⁺19, Theorem 3.2]. Thus, for these valuations, there always exists an EF1 allocation that is both fair and efficient by these notions.

On the other hand, we can also find cases where on an intuitive level EF1 allocations are clearly less fair than some others allocations. Consider the following example of a fair division problem with additive valuations:

Example 1.8.

$$n = 2, m = 3, v_1(r) = v_2(r) = \begin{cases} 1 & \text{if } r = 1, \\ 2 & \text{if } r = 2, \\ 10 & \text{if } r = 3 \end{cases}$$

The allocation $(\{1, 3\}, \{2\})$ is EF1 because only agent 2 envies agent 1, and removing resource 3 from the bundle of agent 1 annuls the envy. Our intuition, however, tells us that there is an allocation with even higher fairness: $(\{3\}, \{1, 2\})$. Since agent 1 already has the resource that causes the envy, why give them a second one? In this allocation, agent 2 still envies agent 1, but less. In this case, it is not just that there exists a resource that, when removed, annuls the envy. Rather, if we remove any resource from agent 1’s bundle, the envy disappears. This stronger fairness is formalized in the following definition.

¹*utility* is a common synonym for valuations in papers about fair division

Definition 1.9 (Envy-free up to any good — EFX₀). *An allocation $\mathbf{X}$ is envy-free up to any good if for all i, j*

$$\forall g \in X_j : v_i(X_i) \geq v_i(X_j - g).$$

Another perspective which characterizes the difference between EF1 and EFX₀ uses the *marginal utility* $v_i(X_j) - v_i(X_j - r)$ of a resource $r \in X_j$. For EF1 it is permissible that only the removal of the resource with the highest marginal utility annuls the envy, while for EFX₀ it must be *any* resource or in other words, even the one with the lowest marginal utility.

1.1 EFX vs EFX₀

Before we further expand on what is known about EFX₀, it is important to differentiate two slight variants of EFX. Our definition of EFX₀ has this 0 subscript because it differs from the original definition of EFX by Caragiannis et al. [CKM⁺19].

Definition 1.10 (Envy-free up to the least positively valued good — EFX). *An allocation $\mathbf{X}$ is Envy-free up to the least positively valued good if for all i, j , where i envies j , we have*

$$\forall g \in X_j : v_i(g) > 0 \Rightarrow v_i(X_i) \geq v_i(X_j \setminus g).$$

Although Kyropoulou et al. [KSV20] distinguish between EFX and EFX₀, we adopt the more standard EFX terminology. This is justified because, for most of our results, we do not have 0-valued resources. Furthermore, for additive valuations — which is the only type of valuation we use for our results — there is a polynomial-time reduction from finding EFX₀ allocations to finding EFX allocations [ABFR⁺21, Proposition 2.3]. Since every EFX₀ allocation is an EFX allocation, there is an EFX₀ allocation if and only if there is an EFX allocation, in instances which we consider.

1.2 Related Work

In this thesis, we investigate the „space“ between EF1 and EFX. To motivate the relevance of this space, we now reiterate some of the most important results regarding these two relaxations of EF, not only in regard to existence but also efficiency (PO and Nash welfare).

Let us start with EF1. As we have already seen, for a fair division problem with indivisible resources and monotone valuations, there exists an EF1 allocation. Furthermore, there is an EF1 allocation that is both MNW and PO [CKM⁺19]. For EF, we have seen with Example 1.3 that the existence of such an allocation is not guaranteed for indivisible resources. Thus, the existence of an allocation that is additionally PO or MNW is also not guaranteed. However, when we consider divisible resources, we find again that EF allocations are known to exist, as Caragiannis et al. [CKM⁺19] point out. This is because maximum Nash welfare allocations coincide with *Competitive Equilibrium from Equal Income (CEEI)* solutions [AI82, Chapter 14], and CEEI solutions are both EF and PO [Var74, Theorem 2.2].

With the efficiency and existence results summarized for EF and EF1, we now turn to these questions for EFX which is more permissive than EF but less so than EF1. These results are in the realm of indivisible resources, as is the rest of this thesis. Unlike EF1 (and EF for divisible resources), MNW allocations are in general not EFX [ABFR⁺21, Section 3.1]. As MNW implies PO, this rules out this simple path to showing efficiency. Chaudhury et al. [CGM24, Theorem 21] were able to show that there are partial EFX allocations

that are not Pareto dominated by any total EFX allocation. Using this, they were even able to disprove the conjecture by Caragiannis et al. [CGH19] that adding a resource to a fair division problem instance does not decrease the maximum Nash welfare of the EFX allocations. Hence, we find that EF1 allocations have better efficiency guarantees than EFX allocations. In Chapter 2, we further discuss how the fact that there are allocations where we cannot use Pareto domination as a progress measure is a challenge for the adaptation of EF1 algorithms to EFX.

With the question of efficiency now answered, what remains is the question of whether there is an EFX allocation for every fair division problem instance. For some time, this question remained an „enigmatic open problem“, as Caragiannis et al. [AAB⁺23] put it, even under restrictions such as agents having only additive valuations. However, during the writing of this thesis, Akrami et al. [AMM⁺26] proved the existence of a fair division instance using submodular valuations for which no EFX allocation exists by providing a large SAT-generated monotone counterexample. Mackenzie et al. [MS26] simplified and extended this to a counterexample using subadditive valuations. For now, the question of existence still remains open for additive valuations, which we work with in this thesis. We now present some of the specific cases and relaxations of EFX where existence is guaranteed.

Existence results

One of the most prominent existence results was given by Chaudhury et al. [CGM24]. They proved that for three agents with additive valuations, there always exists an EFX allocation. To do so, they introduced champion edges and graphs, which we extend upon in Chapter 2. This technique formed the basis for future techniques and results. For instance, a generalization of the result by Chaudhury et al. to *nice cancelable valuations* instead of additive valuations [BCFF22]. By *cancelable* valuation v , they mean that for any $S, T \subseteq [m]$ and $g \in [m]$, it holds that $v(S \cup \{g\}) > v(T \cup \{g\}) \Rightarrow v(S) > v(T)$. *Nice* allows the valuation itself to not be cancelable, but for there to be a valuation v' that is *non-degenerate* i.e., for all $S, T \subseteq [m]$, $S = T \Leftrightarrow v'(S) = v'(T)$ and that *respects* v i.e., $v(S) > v(T) \Leftrightarrow v'(S) > v'(T)$. The existence of EFX for three agents was simplified and further extended by Akrami et al. [AAC⁺22]. They were able to show that only one of the valuations needs to be additive, while the other two only need to be monotone. Instead of restricting the number of agents, one can also restrict the number of resources to at most $n + 3$ to ensure the existence of an EFX allocation on general valuations, as shown by Mahara [Mah24].

A substantial body of existence results has been established for fair division problems where the valuations are restricted even more tightly than in the instances above:

- EFX exists for one [PR20, Theorem 4.2], two [Mah23], and three types of agents [HGNV25], meaning that every agent has one of three additive valuation functions.
- Amanatidis et al. [ABFR⁺21] showed that for 2-valued instances, there always exists an EFX allocation. In 2-valued instances, every valuation is additive, there is a set V with $|V| = 2$, and for every resource $r \in [m]$ and agent $i \in [n]$, $v_i(r) \in V$.
- Another notable result was given by Christodoulou et al. [CFKS23]. They have shown that EFX allocations always exist for additive valuations that can be represented by a graph. That is, every resource has a non-zero value for at most two agents, and the set of resources two distinct agents value as non-zero intersect in at most one resource.

Approximations, relaxations and variants

The other research direction is the relaxation and approximation of EFX. One approach here is to bound the value $0 \leq c \leq 1$ so that it is always possible to find an allocation $\mathbf{X}$ so that for every two agents $i, j \in [n]$,

$$\forall x \in X_j : v_i(X_i) \geq c \cdot v_i(X_j \setminus x).$$

Here, the first lower bound $c = \frac{1}{2}$ was proposed by Plaut and Roughgarden [PR20, Theorem 6.1] for subadditive valuations. For additive valuations, it was later improved by Amanatidis et al. [AMN20, Theorem 4.4] to $\phi - 1$, where ϕ is the golden ratio. However, there are critiques of this kind of approximation, as there are no non-trivial approximation guarantees for general valuations for maximum Nash welfare allocations [ABFR⁺21, Chapter 5], which we have seen are both efficient and at least EF1. Instead, they propose the approximation α -vEFX, which is based on how much value would need to be transferred between two agents so that EFX holds again. Here, maximum Nash welfare allocations have a guaranteed $\alpha \geq \frac{1}{2}$, where $\alpha = 1$ is equivalent to the allocation being EFX.

Another approach is to allow the discarding or „donation“ of unallocated resources. In other words, we permit partial allocations. Discarding all resources gives us a trivial empty EFX allocation, so it is interesting to minimize the number of items we need to discard. This direction was initiated by Caragiannis et al. [CGH19]. They showed that there is a way of removing at most $n - 1$ items from a maximum Nash welfare allocation to find an EFX allocation. This was later decreased to $n - 2$ resources in instances with general valuations by Mahara [Mah24], where the bundle of unallocated resources is not envied by any agent. For the special case of four agents and nice cancelable valuations, Berger et al. [BCFF22] were able to push it down to just one unallocated resource. One very notable aspect of this EFX with charity is its relation to Nash welfare. The original paper by Caragiannis et al. [CGH19] approaches this donation by removing some items from the bundles of the agents in the maximum Nash welfare allocation. Using this, they were able to prove that the resulting allocation decreases the Nash welfare by at most half. Furthermore, they showed that this forms the upper bound an EFX allocation may have even when the items are not donated. This might suggest that donation may not decrease the efficiency of EFX.

As we use our own relaxations in this thesis, we bring up two other relaxations we purposefully want to distinguish ours from. EF^kX allows envy between two agents if the removal of any k items annuls said envy. Note that every EF1X allocation is exactly an EFX allocation and vice versa. Currently, it is known that EF2X exists for four agents and cancelable valuations [AGS25], as well as for any number of agents equipped with *restrictive additive valuations* [ARS22]. Here, restrictive additive valuations lie somewhere between additive, identical, and 2-valued valuations, as every resource g is assigned a value $v(g)$ and for every agent $i \in [n]$, $v_i(g) \in \{0, v(g)\}$. Another relaxation of interest is *envy-freeness up to a random good (EFR)*, where for any two agents i, j , after removing a random good of j , it is expected that i does not envy j . There is a polynomial-time algorithm returning a 0.73-EFR approximation, as shown by Farhadi et al. [FHL⁺21].

1.3 Relaxations in Between EFX and EF1

In the last section, we saw that it is fairly easy to compute EF1 allocations, meanwhile EFX allocations may not exist in the general case, while for additive valuations the existence is still unknown. So, in a sense, there is a significant gap between EFX and EF1. While there are relaxations of EFX, they are not necessarily related to EF1. In this section, we discuss two relaxations that are closer to EF1 or EFX depending on a parameter. Both

parameterize either the amount of resources that are guaranteed to eliminate envy or the amount of resources that do not. Both have slight advantages and drawbacks, and below we discuss the relation between the two relaxations. Let $\mathbf{X}$ be an allocation in which an agent i envies an agent j . We consider the set

$$T = \{x \in X_j \mid v_i(X_i) < v_i(X_j - x)\}.$$

If $\mathbf{X}$ is EFX, it holds that $|T| = 0$. However, if $\mathbf{X}$ is EF1, we are only guaranteed that $|T| \leq |X_j| - 1$. The following idea parameterizes this concept:

Definition 1.11 (*t-tolerance*). *A set $S \subseteq [m]$ is t -tolerated by an agent $i \in [n]$ if there is a set $T \subseteq S$, $|T| \leq t$, for which*

$$v_i(X_i) \geq v_i(S - x), \forall x \in S \setminus T.$$

For any $x \in S$ where it holds that $v_i(X_i) < v(S - x)$, we say that x is *required to be tolerated*.

Definition 1.12 (*t-tolerant EFX allocation*). *A t -tolerant EFX allocation $\mathbf{X}$ is an allocation where every bundle X_j is t -tolerated by every agent.*

Note that we refer to an allocation, a concept including all agents, as being „tolerant“, while bundles and other concepts we introduce later on are referred to as being „tolerated“, as these objects do usually not include all agents.

Observe that any 0-tolerant EFX allocation is EFX, while every 1-tolerant EFX allocation is EF1. This holds because any envied bundle X_j either has size $|X_j| > 1$, in which case there is one resource that eliminates envy when removed, or it has size $|X_j| = 1$, in which case removing the singular resource also eliminates envy. However, for $t \geq 2$, a t -tolerant EFX allocation is not necessarily EF1. As long as an envied bundle X_j is of size larger than t , it is required to have at least one resource eliminating envy. For any bundle of size t or smaller, this does not hold. Thus, a 2-tolerant EFX allocation might have an envied bundle of size two where removing either of the two resources does not eliminate envy for some agent. Consequently, even though t -tolerance offers a new way to approach EFX allocations by first considering t -tolerant EFX allocations with high t , these allocations are not necessarily fairer than EF1. For this reason, we use another relaxation.

Definition 1.13 (*Envy-freeness up to any of k goods — EFX k*). *An allocation $\mathbf{X}$ is envy-free up to any of k goods if for all i, j where i envies j , there is a set $S \subseteq [X_j]$ with $|S| \geq \min\{k, |X_j|\}$ so that*

$$\forall g \in S : v_i(X_i) \geq v_i(X_j \setminus g).$$

Let $\mathbf{X}$ be an EFX k allocation. For $k \geq 1$, $\mathbf{X}$ is EF1. For $k \geq \mu$, where $\mu = \max_{i \in [n]} |X_i|$, the allocation $\mathbf{X}$ is EFX. Thus, it is better suited for exploring the space between EF1 and EFX. For example, since EF1 allocations are known to always exist for additive valuations while the existence of EFX allocations remains an open question, it is interesting to first consider whether or not EFX2 allocations always exist.

We also want to highlight that EF k X, as defined by Akrami et al. [ARS22], and EFX k might initially look similar, but EF k X allows envy if removing a set of resources of size at most k eliminates the envy.

The relaxation $\text{EFX}k$ was proposed by Professor Harks in personal communication and served as the core inspiration for introducing t -tolerance. Given that $\text{EFX}k$ offers more meaningful fairness guarantees in relation to EF1 , it might initially seem surprising that t -tolerance remains our primary relaxation throughout this thesis. The main reason for this choice is that in Chapter 2, utilizing t -tolerance facilitated simpler, cleaner generalizations of existing concepts. We highlight this again inside that chapter. For Chapter 3, we continue to employ t -tolerance primarily because we had established more intuition and useful phrasing for it at that time. Another important factor is that it is possible to transfer some of the results found for t -tolerant EFX allocations to $\text{EFX}k$ allocations and vice versa using the next lemma. In this thesis, we regrettably find no result where this transfer is meaningful.

Lemma 1.14. *Let $\mathbf{X}$ be a total allocation with the maximum bundle size $\mu = \max_{i \in [n]} |X_i|$ and the minimum bundle size $\nu = \min_{i \in [n]} |X_i|$. If $\mathbf{X}$ is a t -tolerant EFX allocation with $t \in [\mu]$ and an $\text{EFX}k$ allocation for $k \in [\nu]$, then it holds that*

$$t \leq \mu - k \text{ and } k \geq \nu - t.$$

This lemma is the second reason for using t -tolerance in Chapter 3, as we deal with allocations there for which $\nu = \lceil \frac{m}{n} \rceil - 1$ and $\mu = \lceil \frac{m}{n} \rceil$. Thus, any positive existence result for a sufficiently small t can be applied to $\text{EFX}k$ without much loss in precision.

Proof. Let X_j be some bundle envied by an agent i . Partition X_j into the bundles $S = \{x \in X_j \mid v_i(X_j - x) \leq v_i(X_i)\}$ and $T = \{x \in X_j \mid v_i(X_j - x) > v_i(X_i)\}$. It holds that $|S| \geq k$ and $|T| \leq t$. Using this, we find that both

$$|T| = |X_j \setminus S| = |X_j| - |S| \leq |X_j| - k \leq \mu - k$$

and

$$S = |X_j \setminus T| = |X_j| - |T| \geq |X_j| - t \geq \nu - t$$

hold. Since i and j were two arbitrary agents, this holds for all pairs, and thus upper-bounds t and lower-bounds k . $\square$

By rearranging the equations of Lemma 1.14, we find that $k + t$ is bounded, which is an interesting observation on its own that we do not further use in this thesis.

Corollary 1.15. *Let $\mathbf{X}$ be a total allocation. If $\mathbf{X}$ is a t -tolerant EFX allocation with $t \in [\mu]$ and an $\text{EFX}k$ allocation for $k \in [\nu]$, then it holds that*

$$\nu \leq k + t \leq \mu.$$

Observe that if $\mathbf{X}$ is EFX , then $k = \mu$ and $t = 0$, and thus $k + t = \mu$. The converse does not hold in general.

1.4 Contributions and Outline

This thesis is split into two chapters, in both of which we extend algorithms computing EF1 allocations and analyze what guarantee for t is given for the t -tolerant EFX allocations they compute.

In Chapter 2, we consider the envy-cycle elimination algorithm. We introduce it along with its existing extension to EFX , where g -champion edges are used to form Pareto

improvable cycles. This allows for the traversal between partial EFX allocations using Pareto dominance as a progress measure. While envy-cycle elimination only requires monotone valuations, we restrict our focus to additive valuations, as we do throughout this thesis. We discuss existing literature showing that Pareto dominance is limited as a progress measure for traversing between partial EFX allocations. Specifically, as soon as there are fewer than $n - 1$ unallocated resources, the existence of Pareto improvable cycles is no longer guaranteed. To circumvent this limitation, we extend g -champion edges to the t -tolerant setting, which enables traversal between t -tolerant EFX allocations instead. We then show that for $t \geq \lceil \frac{m-2}{n} \rceil$, it is possible to find t -tolerant EFX allocations using t -tolerant Pareto-improvement cycles. However, we also construct and discuss an example of a partial EFX allocation where $t \geq \frac{m-2}{n}$ is required to find a t -tolerant Pareto-improvement cycle, demonstrating that this simple extension of the envy-cycle elimination algorithm does not yield allocations with stronger fairness guarantees. Indeed, for a parameter t of this magnitude, the allocation is not even guaranteed to be EF1.

In Chapter 3, we shift our focus to the round-robin algorithm, which provides a stricter bound for t -tolerance, as it returns EF1 allocations with nearly equal bundle sizes. We discuss the structural limitations of the round-robin mechanism with respect to computing fair allocations. One primary drawback is the potential occurrence of rounds where envious agents must choose items after the agents they envy. For a significant portion of this chapter, we work towards a modified round-robin algorithm without this issue and prove that this it computes allocation where envy is bound by the difference in value between two resources chosen within the same round r_g . This constitutes an improvement over the standard round-robin algorithm, where the envy can be as large as the value of the single highest-valued resource. The largest part of this chapter is dedicated to an attempt of translating this improved envy bound into an improved bound for the required t -tolerance. Due to the fixed bundle sizes inherent to round-robin algorithms, such an approach is meaningful only for instances in which the valuations do not assign values to some resources, that are magnitudes larger than the value of other resources. To this end, we introduce permutation instances, where every valuation is a permutation of the set $[m]$. Under this setting, let i be an agent envying bundle X_j . We show that the number of resources in X_j chosen after round r_g that do not eliminate envy is bounded from above by $\lceil \frac{1}{2} \lceil \frac{m}{n} \rceil \rceil + 1$. We further demonstrate that this quantity decreases if there are resources chosen before round r_g that do not eliminate envy when removed. However, we are unable to bound the number of resources chosen before round r_g that fail to eliminate envy. This limitation is surprising, and we prove it is an inherent bottleneck of our modified round-robin algorithm. Utilizing the insights gathered during our bounding attempts, we give a counterexample construction demonstrating that a parameter of at least $t \geq \lceil \frac{m}{n} \rceil - 2$ is required in the worst case.

2. Pareto improving toward t -Tolerant EFX Using g -championing

In the work toward answering more questions about EFX allocations, many techniques have been used. One of the most common techniques is *g-champion edges*, which have been applied both to show that EFX allocations exist for three agents [CGM24] and to demonstrate that, for four agents, it is possible to find a partial EFX allocation with only one unallocated resource [BCFF22]. In this chapter, we construct a version of the g -champion edges by relaxing them using t -tolerance. We use them to traverse the space of partial t -tolerant EFX allocations, employing Pareto improvements as our measure of progress. In Section 2.4, we then examine how much we need to relax t to guarantee that we can find a t -tolerant EFX allocation. We find that it is necessary to relax t up to a near-trivial bound. We begin by motivating the idea behind g -champion edges through an introduction to the *envy-cycle elimination* algorithm of Lipton et al. [LMMS04], which is used to find EF1 allocations. We do so because g -champion edges can be viewed as a way partially to adapt this algorithm to find EFX allocations.

2.1 Envy-Cycle Elimination

In this section, we will discuss how we can use the *envy-cycle elimination* algorithm to find EF1 allocations as a basis for the algorithm we introduce in Section 2.3. Initially, Lipton et al. [LMMS04] neither called it the envy-cycle elimination algorithm nor was the goal to find an EF1 allocation, as this fairness notion was formally defined only a few years later by Budish [Bud11]. This initial implicit usage of EF1 was pointed out by Amanatidis et al. [ABFRV22]. Instead, they wanted to show that there always exists an allocation with its maximum envy $\max_{i,j \in [n]} \max\{v_i(X_j) - v_i(X_i), 0\}$ upper bounded by the maximum marginal utility [LMMS04, Theorem 2.1], which is defined as $\max_{r \in [m], i \in [n]} \max_{X \subset [m]} v_i(X + r) - v_i(X)$. Incidentally, this is always true for all EF1 allocations, because for any envy we can—by definition—find a resource causing it, and hence the difference in valuations is bounded by the marginal value of that resource. We will now give a quick overview of how and why this algorithm works, while introducing concepts which we need later on.

Throughout the algorithm, we will always maintain a partial EF1 allocation $\mathbf{X}$. We start with the empty allocation, which is EF. In every round, we assign an unassigned resource g to some agent. The idea is that if we assign the resource to an unenvied agent i , then we again get an EF1 allocation, because the removal of g will annul any envy towards $X_i + g$.

Naturally, this is only possible as long as we have unenvied agents. When we are in the case where all agents are envied, we shuffle around the current bundles to create a new EF1 allocation with at least one unenvied agent. This operation is what gives the algorithm its name. To eliminate an *envy-cycle*, we need to first define it. The definition follows naturally from the definition of a cycle in graph theory, when we consider the *envy-graph*.

Definition 2.1 (Envy-graph). *The envy-graph $E_{\mathbf{X}} = ([n], E)$ of an allocation $\mathbf{X}$ is the directed graph where there is an envy edge $(i, j) \in E$ if and only if i envies j .*

An envy-cycle is now simply a cycle of the envy-graph. Observe that as long as there are no unenvied agents, we can always find such a cycle. For the elimination of such a cycle $C = (a_0, \dots, a_k)$, we will rotate the bundles in the counter direction of the envy edges, i.e., we assign every agent of the cycle a bundle it envies. Formally, we find a new allocation $\mathbf{X}'$ by setting

$$X'_i = \begin{cases} X_i & \text{if } i \notin C \\ X_{a_{j+1 \bmod k}} & \text{if } i = a_j \in C \end{cases}$$

To see that it works, we now need to check the following two points:

1. We get an allocation with an unenvied agent after applying the operation a finite amount of times and
2. if we apply this operation on an EF1 allocation, we again get an EF1 allocation.

The first point follows from the fact that the valuations of the agents along the cycle strictly improve while the valuations of the other agents stay the same, i.e., $\mathbf{X}'$ Pareto dominates $\mathbf{X}$. Because we can only Pareto improve a finite amount of times, after a finite amount of cycle-eliminations, we will have no more cycles and hence at least one unenvied agent. The second point, that $\mathbf{X}'$ is also an EF1 allocation, follows from the fact that we do not change the bundles. If i envies j in $\mathbf{X}'$, we know there was some agent $l \in [n]$ such that in the prior allocation $X_l = X'_j$, and there is some $x \in X_l$ such that $v_i(X_i) \geq v_i(X_l - x)$. Using $v_i(X'_i) \geq v_i(X_i)$, we find that this is also true for X_j . Thus $\mathbf{X}'$ is EF1. Putting all of this together, we find that we can always find an EF1 allocation for fair division problems where the only restriction on the valuations is that they are monotone.

However, this does not work for EFX. Consider Example 1.8. We first allocate resource 1 to agent 1, making agent 2 envious and the only unenvied agent. We then assign resource 2 to agent 2, making agent 1 envious and increasing the valuation of agent 2 sufficiently to resolve its envy towards agent 1, making agent 1 the only unenvied agent. Lastly, we assign agent 1 resource 3 and get the exact same EF1 allocation as described in the introduction or again here.

Example 2.2.

$$X_1 = \{1, 3\}, X_2 = \{2\} \text{ with the valuations } v_1(r) = v_2(r) = \begin{cases} 1 & \text{if } r = 1, \\ 2 & \text{if } r = 2, \\ 10 & \text{if } r = 3 \end{cases}$$

Note that this is not an EFX allocation. One way to describe the issue is that we added a highly valued resource to agent 1 even though it still had a resource that was valued poorly. If we instead removed resource 1 when we added resource 3, we might have progressed a further.

2.2 *g-Championship*

We know that we cannot use the envy-cycle elimination algorithm to find EFX allocations. However, this approach of looking for cycles and canceling around them is used to find relaxed variants of EFX (f.e. by donating [CGH19]) or even prove the existence of EFX allocations for restricted cases (three agents [CGM24]). Although it is generally less used as an algorithm and more as a general-purpose technique, it is called *g-championship*. Using it, we no longer navigate through the space of EF1 allocations but of EFX allocations. Note that these ideas are mainly based on the work of Chaudhury et al. [CGM24].

Let $\mathbf{X}$ be our current EFX allocation with the unallocated resource g and some unenvied agent j . Let us now consider what happens when we add g to X_j as we did in the section above. Assume for now that j does not have marginal value 0 for g , then j envies $X_j + g$. There also might be some other agent i envying $X_j + g$. This in itself is not yet a problem. However it stops being a permissible EFX allocation as soon as i envies some proper subset $S \subsetneq X_j + g$, as then for any $x \in (X_j + g) \setminus S$

$$v_i(X_i) < v_i((X_j + g) - x).$$

So when we assign a new bundle X'_j to some agent j using this technique, we want this set to have two properties

1. That X'_j does not contain any proper subsets envied by any agent.
2. And that i envies X'_j .

We will later see why we need this. We call this kind of set a *minimal envied set*.

Definition 2.3 (minimal envied set). *For a given allocation $\mathbf{X}$, resources $[n]$, and valuations $(v_i)_{i \in [n]}$, $S \subseteq [m]$ is a minimal envied set of agent $j \in [n]$ if j envies S and for all $S' \subsetneq S$*

$$v_i(X_i) \geq v_i(S'), \forall i \in [n].$$

When we assign such a minimal envied set to an agent j envying it, then no agent envies j , and if j did not experience strong envy before, it does not do so after assigning the set as well, as its valuation increased. The challenge is finding these sorts of sets. Simply adding g might lead to some other agent envying a proper subset, as it is with Example 2.2. However, we can fix this specific case by removing resource 1 from the bundle, as resource 3 alone is envied and also trivially a minimal envied set. But even this is not always possible. Consider a partial allocation with valuations such that $v_j(x) > v_j(g) \forall x \in X_j$. Then the only subset that j envies of $X_j + g$ is $X_j + g$ itself. If now some other agent envies a subset of this, then there is no minimal envied subset of $X_j + g$ for j . However, we can still work with this, as we will show that it suffices to work on valuations that guarantee that for every agent j there exists an agent i that has a minimally envied subset of $X_j + g$. We can then use this relation as a new edge, a sort of envy edge, that is only realized by adding g to X_j and potentially removing the elements not in the envied set. Before we formalize this, let us deal with the second issue of this approach: termination.

In the envy-cycle elimination algorithm, we were able to prove that there is only a finite amount of rounds, because there we reduce the amount of unassigned resources in every round. When we add g to a bundle and then deallocate resources to create a minimal envied set, we can no longer rely on this. However, we may use the same technique we used to show that after a finite amount of envy-cycle cancels there will once again be an unenvied agent: Pareto domination. This is the reason we use minimal *envied* sets and not minimal „non strong-envy creating“ sets. It would be fine to assign a lower-valued bundle

X'_i to i as long as $v_i(X'_i)$ is not so low that i experiences strong envy towards another agent. This would also give us a new EFX allocation, but then we would need to find a new way of measuring progress.

Let us now formally define the new edges for our cycles.

Definition 2.4 (g -championship). *For a partial allocation $\mathbf{X}$ with the unallocated resource g , an agent $i \in [n]$ g -champions the agent $j \in [n]$ if there is a minimal envied subset of $X_j + g$ for agent i .*

The champion graph $M_{\mathbf{X}} = ([n], E)$ extends the envy graph with labeled champion edges where $(i, g, j) \in E$ if i g -champions j .

Note that some agents may self-champion. In general, if no agent envies $X_j + g$ then there will be no g -championship of j . For our approaches we try to avoid this. We can ensure that j is always championed by adding a second property to the valuations alongside additivity.

Definition 2.5 (non-degenerate instance). *A valuation v is non-degenerate if and only if for all $S, T \subseteq [m]$, $v(S) = v(T) \Leftrightarrow S = T$. A non-degenerate instance consists of a set of agents $[n]$, resources $[m]$, and valuations $(v_i)_{i \in [n]}$, where each valuation v_i is non-degenerate.*

With this, for every agent j and unassigned resource g , the set $X_j + g$ is always envied, at least by j , because of the monotonicity. Hence we can always find a minimal envied subset for some agent and thus a g -champion. We can also perturb any additive valuation into a non-degenerate valuation by adding small weights to the values of the resources. These small weights are different for every resource but small enough that, for every combination of resources into bundles, the strict order does not change under the new valuations. Chaudhury et al. [CGM24] showed that for three agents, if we can find an EFX allocation in an instance in which all valuations are perturbed, then the resulting allocation is also EFX for the unperturbed valuations. This was further extended to any number of agents, for example in [Mah23, Lemma 5]. Hence, we will often assume that our valuations are non-degenerate.

Now, if we know that for every unallocated resource g we can find a champion for any agent, we can always find a cycle of these edges. Of course, we cannot always eliminate this cycle like we did with the envy-cycles, because we only get a Pareto improvement if we assign the bundles around the cycle and add g to *every* one of these bundles. But this is possible if we find a cycle containing only a single g -champion edge.

Definition 2.6 (Pareto improvable cycle). *A cycle $C \subseteq M_{\mathbf{X}}$ is Pareto improvable (PI) if for all champion edges (i, r, j) the resource r is distinct.*

Note that such a cycle may also include envy edges without any further restrictions.

This definition is based on Definition 3.3 of Berger et al. [BCFF22]. Although they use a more general definition there, we still use the same terminology. We eliminate such a cycle by assigning new sets along the cycle. For an envy edge, we assign the envied set to the envying agent. For a g -champion edge between agents i and j , we assign the minimal envied set of $X_j + g$ which i envies to i . This naturally extends the envy-cycle elimination to EFX and, as was the case with this approach for EF1, results in a Pareto improvement. This is formally shown by Berger et al. [BCFF22, Lemma 3.4]. We now demonstrate some uses of this technique.

For Example 2.2, the g -champion graph is shown in Figure 2.1. In the literature, it is common to have only one g -champion per agent. For our purposes, this restriction is

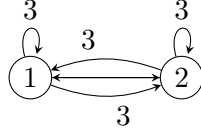


Figure 2.1: Champion Graph of Example 2.2. Unlabeled edges are envy edges. Label edges are champion edges with respect to the labeled resource.

not necessary. We can now see that there are three PI cycles to choose from: the two self-improvement cycles and the cycle between agent 1 and agent 2, where we use the envy edge from agent 1 to agent 2 and the g -champion edge back. If we choose to Pareto improve using this cycle, we assign agent 1 the bundle $X_2 = \{2\}$ and agent 2 the set $\{3\}$, as this is its minimal envied subset of $X_1 + 3$.

2.2.1 Results Using g -Championing

Lemma 2.7. *For every fair division instance using only additive valuations, there is an EFX allocation with at most $n - 1$ unallocated resources, where additionally the bundle of unassigned resources P is not envied by any agent.*

The argumentation is based on Lemma 2.4 of the paper of Berger et al. [BCFF21].

Proof. Given any additive fair division instance, we only need to show that we can find a partial EFX allocation for the perturbed instance, where for every unassigned resource g , every agent has a g -champion. We will now show that there is always a PI cycle as long as there are at least n unallocated resources. Consider for this some agent i and find its g -champion j , then for j we choose a different unallocated resource h that we did not use before and find j 's h -champion k , $\dots$. After continuing this with at most n distinct resources, there must be at least one champion appearing twice in the sequence of champions, forming a PI cycle. Therefore, we can always find a Pareto-dominant EFX allocation as long as there are at least n unallocated goods. And since we can at most finitely Pareto improve, there must be a partial EFX allocation with at most $n - 1$ resources. Now, if some agent envies these $n - 1$ resources, we may simply assign a minimal envied subset of them to a most envious agent. Hence, in this case we can again always Pareto improve. $\square$

This argumentation can even be extended to the nice cancellable valuations and down to $n - 2$ unallocated resources. Here, Berger et al. [BCFF21, Theorem 4.1] base their result on the structure that $M_{\mathbf{X}}$ has if $\mathbf{X}$ has exactly $n - 1$ unallocated resources. They use the fact that for any unallocated resource g , the g -champion edges form a Hamiltonian cycle. In fact, for all unallocated resources, these cycles are parallel. They then use this observation along with a generalized version of the g -champion edges, which we only consider briefly in Section 2.4.5, to show that a Pareto improvement is possible in this case.

And this is a common way to proceed. Both Chaudhury et al. and Berger et al. base the existence result for three agents and the almost-existence result for four agents on extensive case distinctions over the structure of the champion graphs: the existence of envy in that graph, the number of sources, the length of cycles, etc. This is also the reason why this approach might not simply extend to more agents. It is important to note that Pareto improvements in general are not a sufficient measure of progress. Both use, in some of their cases, lexicographic improvements, and there are even partial EFX allocations where we cannot Pareto improve to a total EFX allocation [CGM24].

This motivates the question, whether by relaxing EFX to t -tolerant EFX allocations, can find new ways to Pareto improve.

2.3 t -Tolerated g -Championship

In the proof of Lemma 2.7 we used that for at least n unallocated goods there always exists a Pareto improvable cycle. Consider the g -champion edges that we use in such PI cycles. When we assign a minimally envied set along such an edge, we find that this set is 0-tolerated by all agents.

Observation 2.8. *A set S is minimally envied by an agent $i \in [n]$ if and only if it is envied by agent i and 0-tolerated by all others.*

Proof. Let S be a minimally envied set envied by some agent i . Then i envies S , and for any other agent j it is true that

$$v_j(S - x) \leq v_j(X_j) \quad \forall x \in S.$$

Hence no element of S must be tolerated, so S is 0-tolerated. Conversely, let S be a 0-tolerated set envied by i . Then i envies the set, and by the definition of 0-tolerated there is no $x \in S$ for which

$$v_j(S - x) > v_j(X_j).$$

Thus S is minimally envied by i . □

When we have fewer than $n - 1$ unallocated resources we may find that the present g -champion edges do not form PI-cycles. Recall that in a minimally envied set, removing any resource eliminates any envy. We have also seen t -tolerated sets, where for every agent i there are at most t resources which when removed do not eliminate envy from i . We use this to relax the idea of g -championship to the t -tolerated sets, to investigate if this results in new edges. Intuitively eliminating a cycle with such an edge, does no longer result in EFX allocation, but we will see that the amount of resource which do not eliminate envy when removed, is bounded by t .

As hinted at in Section 1.3, this observation explains why we use t -tolerance instead of EFX k . It would have been possible to reframe this observation for EFX k as well by introducing a term like t -tolerated, which implements the ideas of EFX k on an inter-agent basis. One could then consider that a bundle S is minimally envied by an agent i if agent i envies S and $k \geq |S|$. However, pulling the bundle size into the equivalence makes the generalization we consider in the rest of the chapter significantly more complicated.

Definition 2.9 (t -tolerated g -championship). *For a partial allocation $\mathbf{X}$ with an unallocated resource g , the agent $i \in [n]$ is a t -tolerated g -champion of agent $j \in [n]$ if there is a set $S \subseteq X_j + g$ such that agent i envies S and all agents $k \in [n]$ t -tolerate it.*

The t -tolerant champion graph $M_{\mathbf{X}}^t = ([n], E)$ is a supergraph of the envy graph with labeled t -tolerated champion edges, where $(i, g, j) \in E$ if i is a t -tolerated g -champion of j .

Note that we changed the wording compared to Definition 2.4. Using the same sentence structure would have resulted in a phrasing such as „For a partial allocation $\mathbf{X}$ the agent $i \in [n]$ t -toleratedly g -champions $j \in [n]$...“. However, „toleratedly“ is not an English word, so we need to be a little more cautious about our phrasing.

We can also extend Definition 2.6 to t -tolerated PI cycles, where we allow the use of t -tolerated g -champion edges. When it is clear that we are arguing about the t -tolerant champion graph, we continue to use the term PI cycles. As this is a generalization of the standard PI cycles, we will now reprove some lemmas that also hold for the generalization. The original variants were provided by Berger et al. [BCFF22].

We eliminate t -tolerated PI cycles by assigning the envied and t -tolerated sets around them just as we did with the regular PI cycles in Definition 2.6. However, we need to make sure that this elimination always produces a t -tolerant EFX allocation.

Lemma 2.10. *Let $\mathbf{X}$ be a (partial) t -tolerant EFX allocation. Eliminating a t -tolerated PI cycle produces a t -tolerant EFX allocation $\mathbf{Y}$ that Pareto dominates $\mathbf{X}$.*

Proof. We need to show two properties: $\mathbf{Y}$ Pareto dominates $\mathbf{X}$, and every bundle in $\mathbf{Y}$ is t -tolerated. To see that $\mathbf{Y}$ Pareto dominates $\mathbf{X}$, observe that any bundle assigned around the cycle is envied, so the value of every agent on the cycle strictly improves. The bundles of the agents not on the cycle are not changed, hence their value stays the same. Thus no agent's value decreases, and the value of at least one agent increases. Hence $\mathbf{Y}$ Pareto dominates $\mathbf{X}$.

Next, we show that $\mathbf{Y}$ is a t -tolerant EFX allocation. To do this, we recall that every bundle of $\mathbf{Y}$ is t -tolerated in $\mathbf{X}$. We then use the Pareto dominance of $\mathbf{Y}$ to show that this property holds for $\mathbf{Y}$ as well. In $\mathbf{Y}$, there are two types of bundles: those that remain unchanged from $\mathbf{X}$ to $\mathbf{Y}$, and those that were created by the elimination of a PI cycle. The unchanged bundles are t -tolerated in $\mathbf{X}$ because $\mathbf{X}$ is a t -tolerant EFX allocation. For the changed bundles, they were created by assigning a t -tolerated set along a t -tolerated champion edge. Therefore, these bundles are t -tolerated in $\mathbf{X}$. To see why these bundles are also t -tolerated in $\mathbf{Y}$, consider any agent $i \in [n]$ along with some set S containing x , for which $v_i(S - x) < v_i(X_i)$. Since $\mathbf{Y}$ Pareto dominates $\mathbf{X}$, it holds that $v_i(X_i) \leq v_i(Y_i)$. Consequently, $v_i(S - x) < v_i(Y_i)$. Because the amount of resources that do not eliminate envy does not increase when moved from $\mathbf{X}$ to $\mathbf{Y}$, if S is t -tolerated in $\mathbf{X}$, then it is also t -tolerated in $\mathbf{Y}$. In conclusion, $\mathbf{Y}$ is a t -tolerant EFX allocation. $\square$

We now show that the existence of a t -tolerated EFX allocation is guaranteed if we can show that t -tolerated PI cycles always exist in partial allocations. The argumentation is very similar to that used to show Lemma 2.7.

Corollary 2.11. *For a fixed $t \in \mathbb{N}_0$, if for every instance and partial t -tolerant EFX allocation $\mathbf{X}$ there is a t -tolerated PI cycle in the champion graph $M_{\mathbf{X}}^t$, then there is a t -tolerated EFX allocation for every instance.*

Proof. Assume that we can always find a t -tolerated PI cycle in a partial t -tolerant EFX allocation. By Lemma 2.10, if we eliminate a t -tolerated PI cycle, we get a new t -tolerant and Pareto dominating EFX allocation. Since we can always find these cycles, we can always Pareto improve any partial t -tolerant EFX allocation. Since we can only Pareto improve a finite number of times, there must exist a t -tolerant EFX allocation. $\square$

The goal of this chapter is to address the questions for which t it is always possible to construct a t -tolerant EFX allocation through the elimination of t -tolerated PI cycles. To do this, for the rest of this chapter, we focus on the special case where there is only one unallocated resource g , as this is arguably the hardest case to Pareto improve. If we have k unallocated resources, every agent has at least k , possibly not distinct, champions, and our

PI cycles may contain up to k champion edges. With only one unallocated good, there are considerably fewer options. On the other hand, if it is always possible to Pareto improve using only a single (t -tolerated) champion edge, along with the envy edge, then it is always possible to Pareto improve an allocation with any positive number of unallocated resources, by essentially adding them one by one to the instance and taking all but one away if a Pareto improvement frees multiple resources.

With the setup now clear, let us study some of the simplest cases where it is always possible to find a PI cycle. To start out, we reframe the definition of the t -tolerated PI cycle for our special case with a focus on the set $T \subseteq X_j + g$ whose elements do not eliminate envy when removed.

Observation 2.12. *Given a t -tolerant partial EFX allocation $\mathbf{X}$ with an unallocated good g , if there is an agent i that is reachable in the envy graph from an agent j and there is a subset $T \subseteq X_j + g$ for which $v_i(T) > v_i(X_i)$ and*

$$\forall l \in [n] - i : |\{x \in T \mid v_l(T - x) > v_l(X_l)\}| \leq t$$

then there is a t -tolerated PI cycle in $M_{\mathbf{X}}^t$.

Note here that $i = j$ is possible and a common case in this chapter.

Proof. If there is a t -tolerated g -champion edge from agent i to agent j , then it forms a PI cycle together with a path of envy edges from agent j to agent i , as this is a cycle with only one t -tolerate champion edge. We mention that there is only one champion edge, as if there would be two we would also need to show, that they use distinct unallocated resources to form a PI cycle. To see that this t -tolerated champion edge exists, consider that i envies the subset $S \subseteq X_j + g$ and all other agents, by definition, tolerate it. $\square$

Note that the other direction also holds provided that $M_{\mathbf{X}}$ does not contain any envy-cycle and that g is the only unallocated resource. Let us now show three common cases where Observation 2.12 enables self-improvement.

Corollary 2.13. *Given a partial t -tolerant EFX allocation $\mathbf{X}$ where g is unallocated, there is a t -tolerated PI cycle in $M_{\mathbf{X}}^t$ if at least one of the following statements is true.*

- (i) *There is an agent i for which there exists a subset $R \subseteq X_i$ with $v_i(R) < v_i(g)$ and $|R| \geq |X_i| - t + 1$.*
- (ii) *There is an agent i with $|X_i| < t$ and $v(g) > 0$.*
- (iii) *There is an unenvied agent i with $|X_i| \leq t$ and $v(g) > 0$.*

Note that $v(g) > 0$ is always true for non-degenerate instances.

Proof. (i) We use Observation 2.12 with $i = j$. For $T = (X_i \setminus R) + g$ it is true that $|T| \leq |X_i| - |R| + t - 1 + 1 = t$. Because of its size T is trivially t -tolerated. It is also envied by i , as $v_i(T) = v_i(X_i) - v_i(R) + v_i(g) > v_i(X_i)$.

(ii) This is a special case of Claim i where $R = \emptyset$.

(iii) This extends Claim ii, using that the removal of g from $X_i + g$ also eliminates any envy, as X_i is unenvied.

$\square$

Using this we can find an almost trivial bound for t , where it is always possible to improve using PI cycles.

Lemma 2.14. *For any partial t -tolerant EFX allocation $\mathbf{X}$ in a non-degenerate instance, there is a PI cycle if $t \geq \lceil \frac{m-2}{n} \rceil$.*

Proof. We show this by fixing the amount of agent n and the tolerance t . We then show that there is a PI cycle for every valid choice of $m \leq tn + 2$. We start out with lower m and gradually increase it to introduce new arguments. For $m \leq tn$ there are at most $tn - 1$ resources allocated, as we have a partial allocation. Thus there always exists an agent i with $|X_i| < t$. Applying Corollary 2.13, Claim ii, we find that there is a self-improvement cycle. For $m = tn + 1$ the argument is almost the same. Now there are at most tn resources allocated. If there is an agent i with $|X_i| < t$, we are in the previous case. Otherwise, all agents have exactly t resources. Here we either have the case that there is an unenvied agent, in which case we apply Corollary 2.13, Claim iii to the unenvied agent to find a self-improvement cycle. Alternatively all agents are envied, in which case there must be a PI cycle consisting of envy edges as described after Definition 2.1. We finally extend this argument to $m = tn + 2$ by considering whether we have at least two unenvied agents. We first consider the case where there are at least two unenvied agents and no PI cycle, then every envied agent has at least t resources and every unenvied agent has at least $t + 1$ by Claim ii and Claim iii. Hence there are at least $(n - 2)t + 2(t + 1) = tn + 2$ allocated resources. Thus there would not be an unallocated resource, which is a contradiction. Let us now discuss the case that there are fewer than two unenvied agents. Specifically, consider the case where there is exactly one unenvied agent j , as we have already covered the case where there are zero unenvied agents. Since we are in a non-degenerate instance, every agent, including agent j , has a g -champion, where g is the unallocated resource. Let agent i be the g -champion of agent j . Since there is only one unenvied agent and no envy-cycle (otherwise we could take that envy-cycle), there must be a path from agent j to agent i using only envy edges. The champion edge between the two agents thus closes a PI cycle. Hence in any case there is a PI cycle. $\square$

Thus, we know that if we relax the tolerance sufficiently, we can always make progress using Pareto domination until we find a total t -tolerant EFX allocation. However, this bound is arguably not very impressive. For $t = \lceil \frac{m}{n} \rceil$, it suffices to partition the set of resources $[m]$ into n bundles of size $\lceil \frac{m}{n} \rceil$ or $\lceil \frac{m}{n} \rceil - 1$ when there are not enough resources. Every bundle is then trivially tolerated, and thus the allocation forms a valid t -tolerant EFX allocation without even considering the valuations.

One might therefore attempt to improve this bound. In the next section, we show that this may not be possible.

2.4 Limitations of the approach

In this section, we further investigate the near trivial tolerance $\lceil \frac{m-2}{n} \rceil = \delta$ as defined in Lemma 2.14. We have seen that we can always find a PI cycle if $t \geq \delta$. We now show that this bound is tight, that is, it does not guarantee the existence of a PI cycle for $t < \delta$. Thus, the final goal of this chapter is to construct an example in which there is no PI cycle for $t = \delta - 1$.

We begin this search by constructing Example 2.16 as an initial attempt in which there is no PI cycle for $t = \delta - 2$ in Section 2.4.1. We then discuss its shortcomings in Section 2.4.2 and show that for all non-degenerate instances and partial allocation with the same champion graph of Example 2.16, that there is always a PI cycle in $M_{\mathbf{X}}^t$ for $t = \delta - 1$ if either at least one of the following two cases is given.

1. The bundle sizes are as they are in Example 2.16.
2. We use the minimal number of resources for a fixed δ .

By the second point, we refer to the fact that there is rounding in the definition of δ . Thus, for fixed n and δ , there are several possible choices for m . Larger values of m make it easier to construct examples in which there is no Pareto improvable cycle. To see why, consider any instance with any allocation. We can always add new resources that all agents value at 0 (or at very small values in the case of non-degenerate instances). If such resources are added to the bundle of an agent who is already envied, removing one of these resources will not eliminate the envy and may therefore increase the required tolerance. We find the smallest m by considering for what m it holds that $\delta - 1 < \frac{m-2}{n} \leq \delta$. Thus, the smallest and, for a counterexample, most restrictive m is $m = (\delta - 1)n + 3$. We now consider what it means if we find a non-degenerate instance along with an allocation where, for a fixed δ and m defined this way, there is no $(\delta - 1)$ -tolerated PI cycle. Recall that $\delta = \lceil \frac{m-2}{n} \rceil$ is a lower bound for t where we know that for all $t \geq \delta$, there is a t -tolerated PI cycle in $M_{\mathbf{X}}^t$. However, it may be possible that there is an even better lower bound, for example, $\delta' = \lceil \frac{m-3}{n} \rceil$, for which there are also always tolerated PI cycles for the tolerance $t \geq \delta'$. Due to the rounding, for many choices of m and n , $\delta = \delta'$. However, considering fixed δ and n , for $m = (\delta - 1)n + 3$ it holds that

$$\delta' = \lceil \frac{m-3}{n} \rceil = \lceil \frac{(\delta-1)n}{n} \rceil < \delta.$$

Thus, if we find a non-degenerate instance with $m = (\delta - 1)n + 3$ resources and a partial allocation where there is no PI cycle for $t < \delta$, this shows that δ , as it is currently defined, is already tight in the sense that there are instances for which no PI cycles exist for any smaller value. If we aim to work toward constructing t -tolerant EFX allocations by eliminating arbitrary t -tolerated PI cycles, then the best t for which there is always a t -tolerant EFX allocation is δ .

However, to conclude this, we have to find such an instance and partial allocation. This constitutes the main result of this chapter.

Theorem 2.15. *For any $\delta > 2$, there exists a non-degenerate instance with $(\delta - 1)n + 3$ resources and a partial EFX allocation $\mathbf{X}$ such that there is no Pareto improvable cycle in $M_{\mathbf{X}}^t$ for $t < \delta$.*

In Section 2.4.3, we carry out the main step in proving Theorem 2.15 by constructing an instance with $(\delta - 1)n + 3$ resources and a partial EFX allocation without a $(\delta - 1)$ -tolerated PI cycle. However, for the sake of simplicity, the instance is highly degenerate. Finding such a degenerate instance is comparatively easy. However, finding one that we can transform into a non-degenerate instance while maintaining the absence of a PI cycle is the challenging part. This forms the content of Section 2.4.4, thereby completing the proof of Theorem 2.15. With Section 2.4.5, we conclude this chapter by discussing how these results demonstrate that relaxing the tolerance in order to make the envy-cycle-elimination algorithm work for t -tolerant EFX has only limited potential. We then present how a more generalized notion of championing may offer greater potential, as it performs well on the counterexamples we provide. We use this observation to motivate two questions for future research.

2.4.1 An Initial Bound

In this section, we construct an instance and an allocation that approach the near-trivial bound $\delta = \lceil \frac{m-2}{n} \rceil$ from Lemma 2.14. We begin with a partial EFX allocation that requires a tolerance of at least $t = \delta - 1$ for any $\delta > 1$ in order to obtain a Pareto improvement

using t -tolerated g -champion edges. A more practical way to phrase this is that there are no PI cycles for $t = \delta - 2$. Note that if there is no PI cycle for $t = a$, for some value a , then there are also no PI cycles for $t < a$.

It is important to note that we focus exclusively on PI cycles. Pareto improvements, more generally, are often possible in ways that cannot be represented as the elimination of PI cycles. Therefore, whenever we refer to a PI improvement in the remainder of this chapter, we mean the elimination of PI cycles unless explicitly stated otherwise.

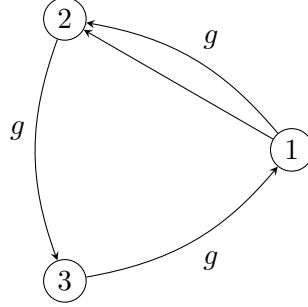


Figure 2.2: Envy and Champion Graph
Unlabeled edges are envy edges.

The counterexample we now construct involves three agents and has the g -champion graph shown in Figure 2.2. We provide some justification for this choice in Section 2.4.2. A main flaw of this example is that each agent is assigned the same number, namely δ , of resources, as one may assume that this simplifies the example construction. There is only one unallocated resource g . To conform to the definition of δ , the total number of resources m is restricted to the range $\{(\delta - 1)n + 3, \dots, \delta n + 2\}$. For our choice of $m = 3\delta + 1$, this condition is satisfied. We could even use one additional resource to simplify the construction, or one fewer resource, which would make it more challenging. Together with the valuations defined below, this yields the full initial counterexample.

Example 2.16.

$$n = 3, m = \delta n + 1, |X_1| = |X_2| = |X_3| = \delta, P = \{g\}.$$

For the valuation, there are two fixed resources $x'_3 \in X_3$ and $x'_1 \in X_1$ such that

$$v_1(x) = \begin{cases} \frac{1-\frac{1}{\delta}}{\delta} & \text{if } x \in X_1 + g \\ \frac{1}{\delta} & \text{if } x \in X_2 \\ 0 & \text{otherwise} \end{cases}, v_2(x) = \begin{cases} \frac{1}{\delta} & \text{if } x \in X_2 + g \\ 0 & \text{if } x \in X_3 \\ 1 & \text{if } x = x'_3 \\ 0 & \text{otherwise} \end{cases},$$

$$v_3(x) = \begin{cases} \frac{1}{\delta} & \text{if } x \in X_1 + g \\ 0 & \text{if } x \in X_1 \\ 1 & \text{if } x = x'_1 \\ 0 & \text{otherwise} \end{cases}.$$

We now give a partial proof that there is no $(\delta - 2)$ -tolerated PI cycle. We omit some of the details, as a full proof for the higher tolerance $t = \delta - 1$ is provided in Section 2.4.3.

First, observe that envy from agent 1 to agent 2 exists, as shown in Figure 2.2:

$$v_1(X_1) = \delta \frac{(1 - \frac{1}{\delta+1})}{\delta} = 1 - \frac{1}{\delta+1} < 1 = \frac{\delta}{\delta} = v_1(X_2).$$

We also sketch that the champion graph is as shown in Figure 2.2.

For g -championship, every agent i has only two candidates, since the third agent j always has $v_j(X_i) = 0$ and $v_j(g)$ is not high enough to cause any envy. If the g -champion were i itself, a self-improvement would be possible. Hence, we need to rule this out for a valid counterexample. We will show that these self-improvements are not t -tolerated even for $t = \delta - 2$, which implies that they are also not tolerated for $t = 0$. Thus, no agent champions itself. Even though the example is clearly a degenerate instance, every agent i still has a g -champion as there is always at least one agent envying $X_i + g$. Since we ruled out two of the agents, the structure of the graph must be as is in Figure 2.2.

For this example, we will only show that self-improvements are not $(\delta - 2)$ -tolerated. There are also no PI cycles involving more than one agent. A full explanation is provided for Example 2.19. The rough argument is the same. Such cycles would need to include envy edges. Since there is only one envy edge, all further edges in the cycle would need to be g -champion edges. If we verify that these do not change for $t = \delta - 2$, the graph shown in Figure 2.2 is also the t -tolerant g -champion graph. Here, one can visually verify that there is no valid PI cycle. Recall that the champion edges in PI cycles must all use distinct unallocated resources.

We now show that there are no self-improvement cycles in $M_{\mathbf{X}}^t$. Since the other possible g -champion edges either already exist or do not exist for any t due to a lack of envy, this shows that the t -tolerant champion graph is the same as the 0-tolerant champion graph. First, observe that for any $i \in [3]$, the only subset of $X_i + g$ that agent i envies is $X_i + g$ itself, because $v_i(g) \leq v_i(x)$ for all $x \in X_i$. Thus, self-improvement is restricted to this set, simplifying our task.

For agent 2, self-improvement is not tolerated, since removing any resource from $X_2 + g$ does not eliminate agent 1's envy. As $|X_2 + g| = \delta + 1$, a tolerance of $\delta + 1$ is necessary to permit this self-improvement.

For agent 3, we need to show that there is a subset $S \subseteq X_3 + g$ of size $|S| > t = \delta - 2$ such that removing any resource does not eliminate envy from agent 2. Here, we define this set as $S = X_3 - x'_3$ with $|S| = \delta - 1 > t$. And by the definition of the valuation $v_2(x_3) = 0$ for all $x_3 \in S$. Thus removing them, has no impact upon the envy and so

$$v_2(X_2) = 1 < 1 - 0 + \frac{1}{\delta} = v_2(X_3) - v_2(x_3) + v_2(g), \quad \forall x_3 \in S. \quad (2.1)$$

Hence, $X_3 + g$ is not t -tolerated. The exact same argument applies to agent 1, showing that agent 3 does not t -tolerate self-improvement.

Note that in this example, it clearly makes a difference whether we consider EFX or EFX_0 , as discussed in Section 1.1. Using the original definition of EFX, self-improvements are tolerated: for example, consider agent 3 were to self improve using g . The only envying agent is agent 2 and by the original definition of EFX we ignore all resources agent 2 values as 0. Thus the only resources which actually need to eliminate envy when removed are g and x'_3 . It is easy to see that they do. However this does not mean, that we have to start differentiating between EFX and EFX_0 as there is a simple way of transforming Example 2.16 into a version which also works for the original definition of EFX. To do so, we need to use a slightly more complex valuation where for example $v_2(x_3) = \varepsilon$ for all $x_3 \in X_3 - x'_3$ and $v_2(x'_3) = 1 - (\delta - 1)\varepsilon$, with $\varepsilon < \frac{1}{\delta}$. For this valuation, Equation (2.1) still holds, with ε in place of the 0 in that equation. Similarly we can also adapt the other valuations to construct a counterexample. In Section 2.4.4 we apply a similar operation to this to create a non-degenerate instance of Example 2.19.

Our goal in this chapter is to construct a partial allocation for which $t = \delta$ is required to Pareto improve. Example 2.16 is not a valid counterexample. To see this, observe that for

$t = \delta - 1$, there may not be a self-improvement cycle for agent 2, but there is one for both agent 1 and agent 3. For $i \in \{1, 3\}$, the bundles each agent uses to self-improve contain at least two resources whose removal eliminates any envy. For g , this is clear, since X_i is unenvied and the removal of x'_i (that is, x'_1 or x'_3) also eliminates envy. Thus, an agent needs to tolerate at most $|X_i + g| - 2 = \delta - 1$ resources, so $t = \delta - 1$ suffices for these self-improvements.

Let us now consider how to modify the example so that no self-improvement cycle exists. We have seen that for X_2 , $t = \delta + 1$ is required to permit self-improvement. One idea might be to remove one of its resources and, if the valuations are adapted properly, add it to either X_1 or X_3 as another resource that must be tolerated under self-improvement. However, this still leaves the other bundle, to which no resource has been added, with self-improvement still $\delta - 1$ tolerated. We could, of course, add one new resource, as permitted for a fixed choice of δ , but if our goal is to find an example with a tight m , there is no way around adapting the valuations so that no bundles contain resources x'_i (as in x'_1 or x'_3) that eliminate envy when removed.

In the next section, we will show that this is not possible for the champion graph in Figure 2.2, after explaining some ideas that went into the construction of these examples.

2.4.2 Counterexample Background

In this section, we do two things. First, we motivate why Example 2.16 has a champion graph as shown in Figure 2.2. We then discuss why we will always find PI cycles for $m = \delta n$ and $t = \delta - 1$. We also touch upon the case where $m = \delta n + 1$, as it is in Example 2.16, and see why we were not able to improve it without changing the bundle sizes. All of these results should serve as background to highlight the relative simplicity of Example 2.19.

We start our discussion of the choice of the champion graph in Figure 2.2 by taking a look at the number of agents n . When creating an example, one might initially try to make it as simple as possible. Thus, one might first consider $n = 2$. There are two arguments against this choice. Firstly, EFX allocations for two agents can generally be found in a quite simple manner without champion or even envy edges. We pretend as if agent 2 has the same valuation as agent 1. Then we can apply Plaut and Roughgarden’s algorithm for finding an EFX allocation when all agents have identical valuations [PR20, Proof of Theorem 4.2]. In this allocation, agent 1 does not experience strong envy. Furthermore, agent 1 does not experience strong envy even if we swap the bundles of the two agents. Hence, agent 2 may simply pick its favorite bundle of the two, and agent 1 gets the other. It is essentially a cut-and-choose protocol for EFX. Note that this concept was expanded to three agents by Akrami et al. [AAC⁺22], but this requires significantly more effort. Secondly, even if we choose to approach this instance from the perspective of eliminating cycles to produce Pareto improvements, this is rather simple without any tolerance relaxation being required for non-degenerate instances¹. If there is envy between the agents, say from agent 1 to agent 2, we can always find a Pareto improvement because agent 1 necessarily has a g -champion. If this champion is agent 1 itself, we have a self-champion. If this agent is agent 2, we have a PI cycle if we use the g -champion edge and the envy edge. Thus, the only case that might be of interest is the one in which there is no envy. For this special case, one can apply a procedure proposed by Chaudhury et al. [CGM24, Remark 11] to produce a Pareto improvement. Note here that this procedure actually works on three agents, while in our case we have only two. However, it applies to a case in which there is a two-cycle in the champion graph — as is given in our case — and leaves the bundle of the third agent unchanged. To see that we can apply this procedure here, imagine

¹Our examples will not initially be non-degenerate. We will look at a way to perturb them in Section 2.4.4

we have a third completely uninvolved agent, i.e., an agent who assigns all bundles a zero value and is assigned an empty bundle.

Therefore, we will use at least three agents and at least one envy edge. As mentioned in Section 1.2, EFX allocations always exist for nice-cancellable valuations, which are a superset of the additive valuations we use. The difference also extends to the ways one may approach finding EFX allocations as for $n = 2$ it is possible to Pareto improve any partial EFX allocation to another EFX allocation as we have just described. On the other hand, for three agents, it is known that there are partial EFX allocations from which we cannot Pareto improve to other EFX allocations [CGM24, Theorem 21]. Note that this is strictly stronger than saying that our simple PI cycles do not exist in these cases.

For our champion graph, we chose the one shown in Figure 2.2. There are other configurations we could have chosen, such as one in which two envy edges point toward the same agent and the envious agents each champion each other. When constructing the examples, the goal is to not have PI cycles even for high tolerances. Avoiding the PI cycles that use the single envy edge can be fairly simple, as we will see in the next section. The main challenge is avoiding the self-improvement cycles. To avoid them, we need an agent that does not tolerate the set used to self-improve. As g -champions are already close to envying the set of an agent, they are a natural choice for this role in our example.

Lemma 2.17. *Given a partial allocation $\mathbf{X}$ with an unallocated resource g where, for some agent j , the set $S \subseteq X_j + g$ is not t -tolerated. If j has only a single g -champion i , then i does not t -tolerate S .*

Proof. Let $S \subseteq X_j + g$ be a set that is not t -tolerated by some agent $l \in [n]$. If $l = i$, then i does not tolerate S and the proof is complete. Therefore assume that $l \neq i$. Since agent l does not tolerate S , l envies S as well as at least $t + 1$ distinct proper subsets of S . If there were some subset $S' \subsetneq S$ envied by l but not by i , then there would be a minimal envied subset of S' . The agent envying this minimal envied set, who is an agent different from i , would by definition be a g -champion of j . Thus there would be more than one g -champion of j , which is a contradiction. Thus agent i envies every proper subset of S which j envies. And as there are at least $t + 1$ of these sets, i does not t -tolerate S . $\square$

Since we have only one g -champion for every agent in our example, if we can show that the g -champion of agent j t -tolerates a subset of $X_j + g$, then using Lemma 2.17 we can conclude that all agents do. We use this to disprove that an example with the champion graph of Figure 2.2 can be used to show the tightness of δ . That is, for $m = \delta n$, we show that, for all additive valuations and partial allocations with a champion graph isomorphic to Figure 2.2, it is never required to set $t = \delta$ to Pareto improve. So, we show that for all such valuations there exists a PI cycle for $t = \delta - 1$. Specifically, there always exists a self-improvement cycle, which we show by the existence of an agent j envying a subset $S \subseteq X_j + g$ that is tolerated by its champion and thus by all agents.

To do so, let us first consider the size of the bundles. Using Corollary 2.13, we need at least $|X_1|, |X_3| \geq t + 1 = \delta$, as these two are unenvied, and $|X_2| \geq t = \delta - 1$, so that there is no trivial self-improvement possible. Again, this is only necessary if the instance is non-degenerate. For degenerate instances, we may find that $v_i(X_i + g) = v_i(X_i)$, i.e., that self-improvements are generally not possible, so bundle sizes become somewhat irrelevant. The examples we use are degenerate for simplicity. However we apply Corollary 2.13 as if they are non-degenerate as it is possible to perturb them to create a non-degenerate version with the same properties. We discuss how this is done in Section 2.4.4. As we currently have one resource unallocated, we already have $m = \delta n$ resources if all the

inequalities for the bundles are tight. Note that if any agent j can self-improve with a set $S \subsetneq X_j + g$, then this would be tolerated by Corollary 2.13 Claim i. Thus, we only consider self-improvements where g is added to the bundle of an agent and nothing is removed. We now want to use this to highlight a challenge in constructing the counterexample. Even though our examples are far from non-degenerate instances, they all have the property that every agent has a g -champion. However, not every agent is envied. Hence, there are cases where the g -champion i of an agent j does not envy j . If i is the only champion of j , it is required for our counterexample that i does not t -tolerate any self-improvement of j by Lemma 2.17. Thus, it must fulfill

$$v_i(X_i) < v_i(X_j) - v_i(x_j) + v_i(g) \quad \forall x_j \in X_j.$$

We can now rearrange this to form a bound for $v_i(g)$, and together with agent i not envying any strict subset of $X_i + g$, we get

$$v_i(X_i) - v_i(X_j) + v_i(x_j) < v_i(g) \leq v_i(x_i) \quad \forall x_i \in X_i, x_j \in X_j.$$

Thus, the valuation of agent i for the bundles X_i, X_j imposes potentially tight bounds for $v_i(g)$. This is especially true once we consider that $v_i(X_i) \geq v_i(X_j)$ since i does not envy j . The „farther“ i is from envying j , the larger $v_i(g)$ is and in turn the larger $v_i(x_i)$ is. For certain bundle sizes, we find that it is even impossible to find a valuation that satisfies the inequality. We formalize this in the following lemma.

Lemma 2.18. *Given an allocation $\mathbf{X}$ and a fixed $t \geq 2$: For every two agents $i, j \in [n]$ and $S \subseteq X_j + g$ envied by j , at least one of the following properties is false:*

- (i) i envies $S - x$ for all $x \in S - g$.
- (ii) i does not envy j .
- (iii) $|X_i| \geq |S - g|$.
- (iv) i envies no strict subset $R \subsetneq X_i + g$.

We will later mainly use this lemma by showing that for some allocation Properties ii to iv hold, leading us to the conclusion that Property i does not. For the cases we consider, finding only one resource that eliminates envy when removed is enough to show that the bundle is tolerated. Property iv is another way to state that $v_i(g) \leq v_i(x)$ for all $x \in X_i$. Also, note that Property iii hints that making all bundles the same size, as we did with our initial example, might make the construction more challenging.

Before we apply the lemma to our example, let us confirm it is true.

Proof. We assume that all properties hold. By Property i, we have

$$\forall x \in S - g : v_i(X_i) < v_i(S) - v_i(x). \quad (2.2)$$

Observe that $g \in S$, as agent i does not envy X_j by Property ii but does envy $S \subseteq X_j + g$ by Property i and the monotonicity of v_j . Using this, we can remove g from S and reformulate Equation (2.2) as:

$$\forall x \in S - g : v_i(g) > v_i(X_i) - v_i(S - g - x). \quad (2.3)$$

We use this inequality to find a contradiction by working with the individual items of the bundles. To do so, we first show that all elements of X_i are valued higher than those from X_j , or formally:

$$\forall x_i \in X_i, x_j \in S - g : v_i(x_i) > v_i(x_j). \quad (2.4)$$

We start by fixing $x_i \in X_i, x_j \in S - g$. Observe that $v_i(g) \leq v_i(x_i)$ by Property iv. We now only need to show that $v_i(x_j) < v_i(g)$. Since $x_j \in S - g$, we apply Equation (2.3) and extend it to:

$$\begin{aligned} v_i(g) &> v_i(X_i) - v_i(S - g) + v_i(x_j) \\ &\geq v_i(X_i) - v_i(X_j) + v_i(x_j) \geq v_i(x_j). \end{aligned}$$

We used the fact that $S - g \subseteq X_j$ and the monotonicity of the valuations, as well as the fact that $v_i(X_i) - v_i(X_j) \geq 0$, by Property ii.

We finish the proof by returning to Equation (2.3) and fixing some resource $y \in X_i$. To create a contradiction to Property iv, we use:

$$\begin{aligned} \forall x \in S - g : v_i(g) &> v_i(X_i) - v_i(S - g - x) \\ &= v_i(y) + v_i(X_i - y) - v_i(S - g - x) > v_i(y). \end{aligned}$$

Here, $v_i(X_i - y) - v_i(S - g - x) > 0$ holds, since Property iii implies $|X_i - y| \geq |S - g - x|$, which, combined with Equation (2.4), yields the result. Thus, at least one of Properties i to iv is false. $\square$

We now apply Lemma 2.18 to an instance with the champion graph in Figure 2.2 and $m = \delta n$ resources to show that in such an instance never there are always PI cycles for $t = \delta - 1$. Note that $m = \delta n$ is one fewer resource than in Example 2.16. As before, by Corollary 2.13, we have fixed bundle sizes $|X_1| = |X_3| = \delta$ and $|X_2| = \delta - 1$. We apply Lemma 2.18 to $i = 3$ and $j = 1$ with $S = X_1 + g$, where g is the unallocated resource. All properties that we refer to for the rest of this section are found in Lemma 2.18. We now analyze which properties are true by the structure of the champion graph and the bundle sizes. By Figure 2.2, Property ii is true. Property iii and Property iv are both true because of the bundle sizes that are fixed by Corollary 2.13. By Lemma 2.18, this leaves two options. Either Property i is false, which results in the self-improvement of agent 1 to be t -tolerated, as $|X_1| = \delta$ and at most $\delta - 1$ resources do not eliminate envy by agent 3. The alternative is that some other condition needed to apply Lemma 2.18 is false. Assuming that $t \geq 2$, this only leaves the possibility that $j = 1$ does not envy $S = X_1 + g$. This is only possible in degenerate instances. We consider a way to circumvent these in Section 2.4.4. Thus, at least for non-degenerate instances, there is no example with the champion graph Figure 2.2 and $m = \delta n$ where $t = \delta$ is required for the existence of PI cycles.

The next step is to consider whether this still holds for $m = \delta n + 1$. We want to avoid PI cycles, so to avoid the simple cases of Corollary 2.13, we have the following restriction on the bundle sizes:

$$|X_1| \geq \delta, |X_3| \geq \delta \text{ and } |X_2| \geq \delta - 1.$$

As we have a partial allocation, we need to distribute δn resources over these bundles. Before, all of these inequalities were tight, but now we may have one that is strict. We now consider these three cases. If $|X_2| > \delta - 1$, the bundle sizes of X_1 and X_3 do not change, so for non-degenerate instances Property i is false and the self-improvement of agent 1 using $X_1 + g$ is tolerated by agent 3 and thus, by Lemma 2.17, by all. If one of the other two inequalities is strict, the bundle sizes of X_1 and X_3 are different from their previous sizes, so we cannot use the same proof. In fact, we will now show that for no choice of i, j, S Lemma 2.18 guarantees that Property i is false. We start by considering the choices for i, j, S such that Properties ii to iv are true. As there is envy between agent 1 and agent 2, we only consider $(i, j) \in \{(3, 1), (2, 3)\}$ to apply the lemma because of Property ii.

For $i = 2, j = 3$, it is not guaranteed that Property i holds as Property iii is not necessarily given. To see this, recall that $|X_i| = |X_2| = \delta - 1$. For Property iii to be false, it suffices that $|S - g| = |S| - 1 > |X_i| = \delta - 1$. There is no immediate reason why $S \subset X_3 + g$ should not contain more than δ elements as $|X_3| \geq \delta$, and there is also no immediate reason why such an S would lead to self-improvement. Thus, for this i and j , this is where the argumentation fails.

This leaves only $i = 3, j = 1$, for which we will now consider the different bundle sizes. First, let $|X_1| > \delta$; then we have bundle sizes $|X_1| = \delta + 1, |X_3| = \delta, |X_2| = \delta - 1$. Property iii is false if $S = X_1 + g$. For $|X_3| > \delta$, we have $|X_1| = \delta, |X_3| = \delta + 1, |X_2| = \delta - 1$. Property iv is false if there exists $x \in X_3$ such that agent 3 envies $S = X_3 + g - x$. This does not show that there is an instance and partial allocation with $m = \delta n + 1$ where there is no PI cycle for $t = \delta - 1$. We do not give an example of this here, as this m is not the minimum amount of resources for a given δ . Instead, we move on to the case where we actually use the minimum amount of resources for a fixed δ on another g -champion graph.

2.4.3 $(\delta - 1)$ -Tolerance Does Not Suffice for Pareto Improvements Using PI Cycles

In this section, we present the main part of the proof of Theorem 2.15 by constructing Example 2.19, which is an instance along with an allocation that has the properties mentioned in the theorem, except for the fact that it is highly degenerate. We show that it is possible to perturb the example to a non-degenerate version in Section 2.4.4 and also explain why it is so important that this is possible. Thus, we construct an example for which it is necessary to relax the tolerance to $t \geq \delta = \lceil \frac{m-2}{n} \rceil$ so that any Pareto improvable cycles exist. Furthermore, the example uses the minimum amount of resources $m = (\delta - 1)n + 3$. We do this while keeping in mind that we will have to revisit the proof in the next section to construct the non-degenerate version.

We use $n = 4$ agents with additive valuations and an allocation $\mathbf{X}$ whose champion graph is shown in Figure 2.3. As the example should be as tight as possible, we use the lowest possible resource amount for a fixed δ , which is $m = 4\delta - 1$. These resources are distributed in the only way so that there is no trivial PI improvement using Corollary 2.13 if the instance were non-degenerate: allocating δ resources to every unenvied agent and $\delta - 1$ to every envied one, keeping only one resource g unallocated. Combined with the valuations, this gives us the following example.

Example 2.19.

$$n = 4, m = \delta n - 1, |X_1| = |X_3| = \delta - 1, |X_2| = |X_4| = \delta, P = \{g\}$$

$$\begin{aligned} v_1(x) &= \begin{cases} \frac{1}{\delta-1} & \text{if } x \in X_1 + g \\ \frac{1}{\delta} & \text{if } x \in X_2 \\ 0 & \text{otherwise} \end{cases} & v_2(x) &= \begin{cases} \frac{1}{\delta} & \text{if } x \in X_2 + g \\ \frac{1}{\delta-2} & \text{if } x \in X_3 \\ 0 & \text{otherwise} \end{cases} \\ v_3(x) &= \begin{cases} \frac{1}{\delta-1} & \text{if } x \in X_3 + g \\ \frac{1}{\delta} & \text{if } x \in X_4 \\ 0 & \text{otherwise} \end{cases} & v_4(x) &= \begin{cases} \frac{1}{\delta} & \text{if } x \in X_4 + g \\ \frac{1}{\delta-2} & \text{if } x \in X_1 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Observe that they partially satisfy the criteria for graph instances analyzed by Christodoulou et al. [CFKS23], as most resources are relevant for only two agents. However, g is relevant for all agents, and for any two agents, there are multiple resources relevant for both.

$i \backslash j$	1	2	3	4
1	1	1	0	0
2	0	1	$1 + \frac{1}{\delta-2}$	0
3	0	0	1	1
4	$1 + \frac{1}{\delta-2}$	0	0	1

 Table 2.1: Values of $v_i(X_j)$ for $i, j \in [4]$

Lemma 2.20. *For any $\delta > 2$, Example 2.19 is an instance with $(\delta - 1)n + 3$ resources and a partial EFX allocation $\mathbf{X}$ where there is no Pareto improvable cycle in $M_{\mathbf{X}}^t$ for $t < \delta$.*

Proof. Consider the champion graph $M_{\mathbf{X}}^t$. We want to show that there are no cycles. For this, let us differentiate between self-improvement cycles and cycles of length greater than 1. Let us start with the second group of cycles. For a PI cycle to use at least two edges, it includes an envy edge, as using two champion edges labeled with the same resource does not conform with the definition of a PI cycle. Since we have only a single unallocated resource, we may only use a single champion edge and thus rely on the envy edges. We are going to show that the envy edges are as shown in Figure 2.3, as this makes it clear that for a PI cycle of length greater than 1, there are only two candidates, which we can then rule out. Let us begin by verifying the two envy edges: one from agent 2 to agent 3 and one from agent 4 to agent 1. To do so, observe that in the table of valuations Table 2.1 of $v_i(X_j)$, the only i for which there exists a j such that $v_i(X_j) > v_i(X_i)$ is $(i, j) \in \{(2, 3), (4, 1)\}$. Note that these edges are in two different connected components of the graph. Thus, there are only two candidates for g -champion edges that have the potential to close a cycle, namely $(3, 2)$ and $(1, 4)$. However, observe that neither agent 3 envies $X_2 + g$ nor does agent 1 envy $X_4 + g$, since

$$v_1(X_4 + g) = 0 + \frac{1}{\delta - 2} \leq 1 = v_1(X_1).$$

Thus, no matter how large we set t , these g -champion edges never exist, and consequently neither do the PI cycles that use envy edges.

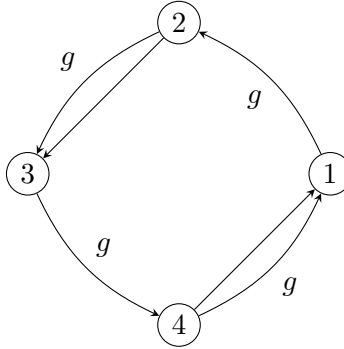


Figure 2.3: Envy and Champion Graph

This leaves us with the self-improvement cycles. To see why these do not exist, observe that for every $j \in [4]$, the only minimal set that j envies is $X_j + g$, since all resources in $X_j + g$ are valued equally. Hence, we only need to show that for every j there is an agent i and a set $T \subsetneq X_j + g$ with $|T| > \delta - 1$ such that for all $x \in T$,

$$v_i(X_i) < v_i(X_j) - v_i(x) + v_i(g). \quad (2.5)$$

Then, for every j , the agent i is always the g -champion, as shown in Figure 2.3. We will now verify that we can find a set T for every j such that Equation (2.5) is fulfilled for some agent i . Starting with $i = 1, j = 2$, and $T = X_2$, it is true that

$$1 < 1 - \frac{1}{\delta} + \frac{1}{\delta - 1}.$$

The same is true for $j = 4, i = 3, T = X_4$ with

$$1 < 1 - \frac{1}{\delta} + \frac{1}{\delta - 1},$$

and $j = 3, i = 2, T = X_3 + g$ with

$$1 < (1 + \frac{1}{\delta - 2}) - \frac{1}{\delta - 2} + \frac{1}{\delta} < (1 + \frac{1}{\delta - 2}) - \frac{1}{\delta} + \frac{1}{\delta},$$

and finally $j = 1, i = 4, T = X_1 + g$ with

$$1 < (1 + \frac{1}{\delta - 2}) - \frac{1}{\delta - 2} + \frac{1}{\delta} < (1 + \frac{1}{\delta - 2}) - \frac{1}{\delta} + \frac{1}{\delta}.$$

Thus, there exists no PI cycle for $t < \delta$. □

2.4.4 Non-Degenerate Instance

One may argue that Example 2.19 does not have any PI cycles simply because it is a degenerate instance. Furthermore, when we consider degenerate instances, finding an example for which there is no PI cycle becomes trivial. Recall that we argued at the beginning of Section 2.4.2 that examples with two agents are not interesting to us, since there is always a PI cycle if there is at least one envy edge, provided that the instance is non-degenerate. Now consider that both agents have a valuation of $v(g) = 0$, where g is the unallocated resource. Then there is only one g -champion edge, from the envious agent to the envied one. In any instance where g is not valued positively by any agent, all g -champion edges are only the envy edges labeled with g . So the problem is mainly reduced to finding an allocation with an acyclic envy graph and suitable bundle sizes. Why then go through the trouble of creating Example 2.19? In their results, Chaudhury et al. [CGM24] showed that for three agents, it is possible to perturb every degenerate instance into a non-degenerate instance. If it is possible to find an EFX allocation for said instance, then it is also EFX in the degenerate instance. Thus, it may be possible to apply a similar argument to our examples. What we want to show is that there are cases where we have to relax t to a near-trivial value to find a t -tolerant EFX allocation. However, if it were possible to circumvent this by applying the technique of Chaudhury et al., it would not be a noteworthy result. Theorem 2.15, which we are going to prove in this section, states that there is a non-degenerate instance where it is necessary to relax t to δ , as perturbing a non-degenerate instance does not result in a lower t being sufficient.

We will do this by perturbing Example 2.16 in just the right way. Before we do, we will first look at why exactly non-degenerate instances are helpful and motivate why the argument that „Example 2.16 only has no PI cycles because it is not degenerate“ may not hold for our instances. Chaudhury et al. [CGM24] mainly use non-degenerate instances because, in such an instance, every agent has a g -champion. The trivial two-agent example sketch we gave also only works because the envious agent has no g -champion. If we make this instance non-degenerate, it will once again have one and therefore also a PI cycle. With this in mind, we constructed both Example 2.16 and Example 2.19 in a way that g -champions are already in place and such that increasing the tolerance does not modify them in any way. When we worked on Example 2.16 to turn it into a counterexample for $t = \delta$, we saw that

this is not possible for some bundle and resource sizes. Here, we did not assume that the instance is non-degenerate but relied on Lemma 2.18 and Lemma 2.17, which in turn rely on the existence of g -champions and the fact that an agent i envies X_j even after removing any resource from $X_j + g$. Both of these are given if the instance is non-degenerate but do not require it to be. With this, one may already have an intuition that degeneracy is not a problem for our examples. To fully prove this, we will now construct a non-degenerate instance from Example 2.19 for which the proof of Lemma 2.20 also works, in turn showing Theorem 2.15.

Proof of Theorem 2.15. We construct the instance and allocation from Example 2.19. We use the allocation without any changes. The instance, however, is a perturbed variant of the instance in the example. To create this instance, we first consider how Chaudhury et al. [CGM24] and Mahara [Mah23, Lemma 5] perturb degenerate instances. We then adapt their approach for our case and show that Lemma 2.20 also holds for the perturbed instance in place of Example 2.19, concluding the proof.

Let v_i be the valuation of an agent $i \in [n]$. We perturb it, creating v'_i to break the cases where $v_i(S) = v_i(T)$ for $S \neq T$ where $S, T \subseteq [m]$. This is done in a way that ensures strict inequalities are preserved, i.e., $v_i(S) < v_i(T)$ implies $v'_i(S) < v'_i(T)$. This allows us to maintain parts of the proof of Lemma 2.20 where only strict inequalities are used. To create a perturbation with these constraints, we define the boundary up to which it is permissible to increase the value of a bundle:

$$\kappa = \min_{i \in [n]} \min_{S, T \subseteq [m], v_i(S) \neq v_i(T)} |v_i(S) - v_i(T)|.$$

Using this, if $v'_i(S) - v_i(S) < \kappa$, we have $v_i(S) < v_i(T) \Rightarrow v'_i(S) < v'_i(T)$. We can now add small values to the valuation of every resource in a way that ensures that for every bundle, its valuation increases by less than κ . Simultaneously, we ensure each distinct bundle has its own distinct value. To do so, we chose $\varepsilon > 0$ such that

$$\varepsilon 2^{m+1} < \kappa.$$

Now we define the perturbed valuations as

$$v'_i(x) = v_i(x) + 2^{\pi_i(x)} \varepsilon \quad \forall x \in [m],$$

where $\pi_i : [m] \rightarrow [m]$ is a permutation used for the convenience of this proof. Mahara [Mah23] instead uses the index of x directly, adding $2^x \varepsilon$ to each resource. Observe that by monotonicity, $v'_i(S) - v_i(S)$ is maximal for

$$v'_i([m]) - v_i([m]) = \sum_{x \in [m]} 2^{\pi_i(x)} \varepsilon = (2^{m+1} - 2) \varepsilon < \kappa.$$

Hence, two distinct bundles with different values under v_i still have different values under v'_i . However, for two distinct bundles S, T where $v_i(S) = v_i(T)$, we have

$$|v'_i(S) - v'_i(T)| = \left| \sum_{x \in S} 2^{\pi_i(x)} - \sum_{x \in T} 2^{\pi_i(x)} \right| \varepsilon > 0 \text{ as } S \neq T.$$

Thus, the bundles are valued differently.

We now verify that this does not interfere with our proof of Lemma 2.20, provided we choose appropriate permutations π_i . We do this by going through the proof from start to finish, highlighting the critical parts. For each part, we impose a constraint on π_i . Finally, we show that it is possible to satisfy all constraints simultaneously. We began the proof of

Lemma 2.20 by verifying the structure of the champion graph. To this end, we showed there are exactly two envy edges using Table 2.1. Since envy requires a strict inequality, these envy edges also exist in the perturbed instance. However, for $(i, j) \in \{(1, 2), (3, 4)\}$, we have $v_i(X_i) = v_i(X_j)$. To prevent a new envy edge (and potential PI cycle), we ensure $v'_i(X_i) > v'_i(X_j)$ by setting $\pi_i(x_i) > \pi_i(x_j)$ for all $x_i \in X_i, x_j \in X_j$ as our first constraint. We then show that no champion edges close a cycle with the envy edges. This is done by showing $v_i(X_j + g) \leq v_i(X_i)$ for $(i, j) \in \{(1, 4), (3, 2)\}$. For $\delta = 3$, we have $v_i(X_j + g) = v_i(X_i)$. Here we could have changed Theorem 2.15 so that $t > 3$ but instead we show this works by setting the permutations such that $v'_i(X_j + g) < v'_i(X_i)$. We achieve this by imposing the constraint $\sum_{x \in X_j} 2^{\pi_i(x)} + 2^{\pi_i(g)} < \sum_{x \in X_i} 2^{\pi_i(x)}$.

To rule out self-envy, we showed that for every agent $i \in [n]$, the only candidate for self-improvement is $X_i + g$. This relies on $v_i(g) \leq v_i(x_i)$ for all $x_i \in X_i$. We make this inequality strict by requiring $\pi_i(g) < \pi_i(x_i)$ for all $x_i \in X_i$ as our last constraint. Because the argument that these sets are not tolerated by the g -champion relies on strict inequalities, we have to do no further work for the perturbed instance.

We now show that there are permutations satisfying all constraints. For $i \in \{2, 4\}$, we only require $\pi_i(g) < \pi_i(x_i)$ for all $x_i \in X_i$. This is satisfied by any permutation where $\pi_i(g) = 1$. For $i \in \{1, 3\}$, we define the tuple $(i, j, k) \in \{(1, 2, 4), (3, 4, 2)\}$ for which the following must hold:

$$\pi_i(x_i) > \pi_i(x_j) \quad \forall x_i \in X_i, x_j \in X_j, \quad (2.6)$$

$$\sum_{x \in X_k} 2^{\pi_i(x)} + 2^{\pi_i(g)} < \sum_{x \in X_i} 2^{\pi_i(x)}, \quad (2.7)$$

$$\pi_i(g) < \pi_i(x_i) \quad \forall x_i \in X_i. \quad (2.8)$$

Note that for these constraints we are less concerned about the valuations of the individual resources and more about where the bundles of the agents lie on the range of the permutation. With this idea in mind we find that they are satisfied for all permutations satisfying

$$\pi_i(g) = 1, \quad \pi_i(x_j) \in \{2, \dots, \delta + 1\} \quad \forall x_j \in X_j$$

and

$$\pi_i(x_k) \in \{\delta + 2, \dots, 2\delta + 1\}, \quad \pi_i(x_i) \in \{3\delta + 1, \dots, 4\delta - 1\} \quad \forall x_k \in X_k, x_i \in X_i.$$

Such permutations exist because $|X_1| = |X_3| = \delta - 1$ and $|X_2| = |X_4| = \delta$. By setting specific ranges for the bundles, any permutation within these ranges forms a valid π_i . This clearly satisfies Equation (2.6) and Equation (2.8). For Equation (2.7) observe that

$$\sum_{x \in X_k} 2^{\pi_i(x)} + 2^{\pi_i(g)} < 2^{2\delta+2} + 2^1 \quad (2.9)$$

$$< 2^{3\delta+1} < \sum_{x \in X_i} 2^{\pi_i(x)} \quad (2.10)$$

Thus there are permutations satisfying all constraints and perturbing Example 2.19 with them creates an instance and allocation satisfying Theorem 2.15. $\square$

2.4.5 Conclusion and Further Questions

In this chapter, we have extended the envy-cycle elimination algorithm to t -tolerant EFX by relaxing the technique of g -champion edges. However, we have found that we need to relax t to the near-trivial bound $\lceil \frac{m-2}{n} \rceil$ to guarantee that we can always find a Pareto improvement with this approach. Note that this does not constitute a strong positive result.

In fact, for $m \geq n + 3$, this only guarantees 2-tolerant EFX allocations, which we have seen in Section 1.3 are not even guaranteed to be EF1. Furthermore, the counterexample in Example 2.19 hinges upon having bundle sizes in $\{\lceil \frac{m}{n} \rceil - 1, \lceil \frac{m}{n} \rceil\}$. If it is possible to find an EF1 allocation with these sorts of bundle sizes, this implies the existence of $(\lceil \frac{m}{n} \rceil - 1)$ -tolerant EFX allocations, which is a strictly better fairness guarantee as long as there is no $\delta \in \mathbb{N}$ such that $\delta n + 1 \leq m \leq \delta n + 2$. In Chapter 3, we find that this is achieved by the round-robin algorithm. Since using t -tolerated PI cycles seems to be an approach with limited merit, for this last section, we discuss potential improvements to this approach. We place the main focus on a generalization of the champion edges. We show that it is possible to Pareto improve Example 2.19 with a PI cycle using these edges. We then briefly mention their limitations and drawbacks, and use them to motivate two questions for future research.

A straightforward solution to the examples (Example 2.19, Example 2.16) might be to avoid them altogether. Note that along with the fair division problem instance, we always gave an allocation. We never explicitly showed that it is possible to reach such an allocation using the elimination of PI cycles. Even if this is possible, there may be approaches where one can avoid reaching such allocations by canceling PI cycles in the right order. This may be aided if one uses t -tolerated PI cycles from the start. Our examples are all EFX allocations, and by expanding the domain to t -tolerated EFX allocations from the start, it may be easier to avoid these dead-end allocations.

Generally it is important to mention that our use of the champion edges is rather simple. They offer more than just PI cycles. Another property used often is that if an agent i g -champions an agent j without envying him, this partitions X_j into an upper-half bundle and a lower-half bundle where the lower half-bundle consists of all the resources that need to be removed from X_j so that the resulting set and g are minimally envied by i . Using these half bundles further PI improvements are possible in some cases, without involving PI cycles [CGM24][Section 4.1]. Hence better results may be achieved by generalizing this approach with t -tolerance as well.

A different way to extend this approach is to use *generalized championship* as introduced by Berger et al. [BCFF22]. Consider an agent i g -championing an agent j . To do so, it envies a minimal envied set $S \subseteq X_j + g$. When we assign S to agent i , for example when eliminating a cycle, we deallocate resources if $S \subsetneq X_j + g$. These newly unallocated resources then have their own champion edges. Thus, when we look for cycles using a specific g -champion edge, we can also use the champion edges that exist after eliminating this g -champion edge. Furthermore, when we have more than just one unallocated resource, we can also consider champion edges with respect to not only one unallocated resource g but a set H of resources. We simply adapt the definition so that agent i envies a minimal envied subset of $X_j \cup H$.

Definition 2.21 (Generalized championship). *For an allocation $\mathbf{X}$, an agent i champions agent j with respect to H_j and S_j where $H_j \subseteq [m] \setminus X_j, S_j \subseteq X_j$ if agent i envies a minimal envied subset of $(X_j \setminus S_j) \cup H_j$.*

We can again use this to define the generalized champion edges and the champion graph, as we have done before with g -championship and t -tolerated g -championship. A difference is that H_j is not restricted to the unallocated resources. Thus, we can use them to extend the definition of PI cycles in a significantly more powerful way. A cycle in the generalized champion graph forms a valid PI cycle if the sets H_j are non-intersecting for all j appearing in the cycle and if every resource in such a set H_j either is unallocated or appears in some S_k , where $k \in [n]$ is also an agent who is part of the cycle. Using generalized champion

edges, it is possible to Pareto improve in Example 2.19. We construct such a case for $\delta = 3$. Similar approaches work for other δ , but we do not offer any proof that it works for all $\delta > 2$.

Lemma 2.22. *There is a PI cycle using generalized champion edges for Example 2.19 for $\delta = 3$.*

Proof. We show that there is such a PI cycle by first constructing an allocation $\mathbf{Y}$ Pareto dominating the allocation $\mathbf{X}$ given in the example. We then show that we can construct sets H_i, S_i for all $i \in [n]$ that conform to the ones mentioned in Definition 2.21. Finally, we show that every bundle of $\mathbf{Y}$ is only envied by a single agent in $\mathbf{X}$ and hence that it is (formally, we actually only show it contains) a minimal envied set envied by the agent to which it is assigned in $\mathbf{Y}$. Once $\mathbf{Y}$ is constructed, this is mainly manual verification of the definitions. To decrease this work slightly, we work with a representative of the two types of agents in Example 2.19. We have one of the envied agents $j \in \{1, 3\}$ and one of the envious agents $k \in \{2, 4\}$. Every agent $i \in [n]$ only positively values g , its own resources, and the resources of agent $\text{succ}(i) = (i + 3 \bmod 4) + 1$, which i g -champions. Note that j champions an agent in $\{2, 4\}$ and vice versa. For convenience sake, we also define the predecessor of agent j (same for k) as $\text{pred}(j) = i \in [n]$ for which $\text{succ}(i) = j$. The main part of the proof is the observation:

$$2v_j(x_k) = \frac{2}{3} > \frac{1}{2} = v_j(x_j) \text{ and } v_k(x_j) = 1 > \frac{2}{3} = 2v_k(x_k).$$

Thus, it is clear that $\mathbf{Y}$ Pareto dominates $\mathbf{X}$ if we construct it by removing two resources from every agent in $\{2, 4\}$ and one from every agent in $\{1, 3\}$ and passing them to the respective agents championing them. Formally, we write

$$Y_j = X_j - x_j + 2x_{\text{succ}(j)}, \quad Y_k = X_k - 2x_k + x_{\text{succ}(k)},$$

here $x_j, x_k, x_{\text{succ}(j)}, x_{\text{succ}(k)}$ are any fixed resources from $X_j, X_k, X_{\text{succ}(j)}, X_{\text{succ}(k)}$ and $2x_i$ is a shorthand for any two resources in X_i . For this to be formally correct, the resources $2x_{\text{succ}(j)}$ we add to X_j are exactly the ones which are removed from $X_{\text{succ}(j)}$ and vice versa for the resource $x_{\text{succ}(k)}$ added to k . We now verify that this operation of reassigning resources can be expressed as even a 0-tolerated PI cycle using generalized champion edges. To do so, we show that every bundle in $\mathbf{Y}$ is minimally envied in $\mathbf{X}$ and that for every $i \in [n]$ there is a bundle $H_i \subseteq [m] \setminus X_i$ and $S_i \subseteq X_i$ such that $Y_i = (X_i \setminus S_i) \cup H_i$. We start out with H_i by setting $H_i = Y_i \setminus X_i$ and $S_i = X_i \setminus Y_i$, as

$$(X_i \setminus S_i) \cup H_i = (X_i \setminus (X_i \setminus Y_i)) \cup (Y_i \setminus X_i) = (X_i \cap Y_i) \cup (Y_i \setminus X_i) = Y_i.$$

This actually works for every $\mathbf{Y}$, not only our fixed one. For $\mathbf{X}$ in Example 2.19 and $\mathbf{Y}$ as defined, this results in $H_j = 2x_{\text{succ}(j)}, S_j = \{x_j\}$ and $H_k = \{x_{\text{succ}(k)}\}, S_k = 2x_k$.

The only thing left to show is that every set Y_i is envied only by agent i in $\mathbf{X}$. To verify that $\mathbf{Y}$ Pareto dominates $\mathbf{X}$, we have already seen that $v_i(Y_i) > v_i(X_i)$. To see that no other agent envies Y_i , this can quickly be done by considering which agents even positively value the resources in Y_i . For the sake of completion, we now explicitly verify this for Y_j and Y_k . Consider $Y_j = x_j + 2x_{\text{succ}(j)}$. The only agents valuing x_j positively are agent j and the $\text{pred}(j)$. Meanwhile, the only agents valuing $x_{\text{succ}(j)}$ are agent j and agent $\text{succ}(j)$. Here we find $v_{\text{pred}(j)}(X_{\text{pred}(j)}) = v_{\text{succ}(j)}(X_{\text{succ}(j)}) = v_k(X_k) = 1$ while $v_{\text{pred}(j)}(Y_j) = v_k(x_j) = 1$ and $v_{\text{succ}(j)}(Y_j) = v_k(2x_k) = \frac{2}{3}$. Hence, there is no envy towards Y_j in $\mathbf{X}$. We do the same for $Y_k = x_k + x_{\text{succ}(k)}$. Here, note that $\text{pred}(k)$ is the one other agent positively valuing x_k , while agent $\text{succ}(k)$ positively values $x_{\text{succ}(k)}$. We find $v_{\text{pred}(k)}(X_{\text{pred}(k)}) = v_{\text{succ}(k)}(X_{\text{succ}(k)}) = 1$

while $v_{\text{pred}(k)}(Y_k) = v_j(x_k) = \frac{1}{3}$ and $v_{\text{succ}(k)}(Y_k) = v_k(x_{\text{succ}(k)}) = \frac{1}{2}$. Hence, no other agent envies Y_k . Thus, with sets H_i, S_i as defined for all $i \in [n]$, there are four generalized champion edges forming a PI cycle for Example 2.19 which can be used to find a Pareto-dominating EFX allocation without even utilizing the unallocated resource g or relaxing the tolerance. $\square$

While generalized champion edges extend the reach of classical g -champion edges and can be strictly more powerful, they also exhibit structural and practical limitations. A primary drawback is that there is no general approach known to discover them and, consequently, it is harder to reason about them. For instance, Berger et al. [BCFF21, Section 3.1] work only with a subset of the generalized champion edges. By structural limitations, we refer to the following phenomenon: there exist (partial) EFX allocations $\mathbf{X}$ and $\mathbf{Y}$ such that $\mathbf{Y}$ Pareto dominates $\mathbf{X}$, yet there is no Pareto-improvement (PI) cycle consisting of generalized champion edges whose elimination yields $\mathbf{Y}$. The reason is that the bundles of $\mathbf{Y}$ are minimally envied with respect to $\mathbf{X}$. For example, suppose $\mathbf{X}$ consists only of empty bundles, and $\mathbf{Y}$ contains at least one bundle Y_i with at least two items that are each positively valued by some agent $j \neq i$. The allocation $\mathbf{Y}$ Pareto dominates $\mathbf{X}$, but Y_i is strongly envied by j , and thus there is no PI cycle producing $\mathbf{Y}$ by eliminating it. However, there may be something like an intermediate allocation. Here we then might once more consider using t -tolerance. We can relax the generalized champion as we did before and ask: for what t is it guaranteed that for any t -tolerated EFX allocations $\mathbf{X}, \mathbf{Y}$, there exists a generalized PI cycle using t -tolerated champion edges for $\mathbf{X}$ that, when eliminated, produces $\mathbf{Y}$? Away from PI cycles, this might bring us closer to the overarching question: For what t is it guaranteed that for every partial (0-tolerated) EFX allocation $\mathbf{X}$ there is a total t -tolerated EFX allocation Pareto dominating $\mathbf{X}$?

3. Improving the Round-Robin Algorithm

In computer science *round-robin* is a common approach used to distribute all kinds of objects, ranging from requests in load balancing [Luo97, Chapter 19] to access to physical hardware by operating systems [Sha87, Chapter 3]. This rarely fits neatly into our model of resources and valuations. Furthermore, round-robin is used because of its practicality and with the goal of efficiency, not fairness per se. However, round-robin, adapted to the fair division setting, is known to have some fairness guarantees. A version of it, as shown in Algorithm 3.1, takes as input a fair division problem instance along with a strict total order $\prec \subset [n] \times [n]$ on the agents, usually the natural order of $[n]$. Then over $R = \lceil \frac{m}{n} \rceil$ rounds, the agents in order of $\prec$ each select the unallocated resource that they value highest and add it to their bundles.

As we are going to show, this simple algorithm is guaranteed to return an EF1 allocation if all valuations are additive. However, the allocation does not necessarily maximize Nash welfare, nor is it necessarily Pareto optimal. A counterexample for the former is any instance with two agents and one resource, where the agent who gets to choose first values the resource less. Furthermore, any allocation returned may not be EFX. In this chapter, we discuss two reasons for this:

- (i) One fundamental restriction of Algorithm 3.1 is that for all $i \in [n]$, the final bundle X_i is either of size R or $R - 1$. To see how this interferes with an allocation being EFX, consider an instance with some fixed resource $x \in [m]$ that is highly valued by every agent i . If x is valued higher than all other resources combined, i.e., $v_i(x) > v_i([m] - x)$, then the set $\{x\}$ is a minimal envied set. Recall that in an EFX allocation, if a bundle is envied it must be a minimal envied sets. So no allocation where the bundle sizes are either $R - 1$ or R is EFX (for $R > 2$). We are not going to modify Algorithm 3.1 in a way that changes this property of the allocations. Instead, we are going to restrict the valuations in a way that limits the impact of this property and analyze the improvements we make to round-robin on instances that only use these valuations.
- (ii) The second issue is an inherent „unfairness“ of round-robin: the fact that for two agents i, j with $j \prec i$, j is inherently better off. This is not only because it can choose first in the first round, but in any round, even if i already envies it.

In this chapter, we seek to remedy this inherent unfairness by modifying Algorithm 3.1 to the *envy-reducing round-robin* algorithm as shown in Algorithm 3.3. First, however, as a warm-up and as a baseline, we analyze Algorithm 3.1. We show the claims that it

Algorithm 3.1: CLASSICROUNDROBIN

Input: Agent amount n , resource amount m and valuations $(v_i)_{i \in [n]}$, order of agents $\prec$

Output: An EF1 allocation $\mathbf{X}$

```

// Initialization
1  $R \leftarrow \lceil \frac{m}{n} \rceil$ 
2  $M \leftarrow [m]$ 
3  $N \leftarrow [n]$  ordered by  $\prec$ 
4 forall  $i \in [n]$  do
5    $X_i \leftarrow \emptyset$ 

// Main loop
6 forall  $r \in [R]$  do
7   forall  $i \in N$  do
8     if  $M \neq \emptyset$  then
9        $x \leftarrow \arg \max_{x \in M} v_i(x)$ 
10       $M \leftarrow M - x$ 
11       $X_i \leftarrow X_i + x$ 

```

produces EF1 allocations that are not necessarily PO. We also propose a lower bound for the $t \in \mathbb{N}$ necessary so that any allocation returned by Algorithm 3.1 is guaranteed to be a t -tolerant EFX allocation. In Section 3.2, we then discuss how to remedy the issue in Algorithm 3.1 by using more than one order in Algorithm 3.2. Here we also introduce the permutation instance and valuations that we have already hinted at in Constraint i. We give an intuitive idea for the bounds that we then formally show in Section 3.3. For Section 3.3 we introduce the final envy-reducing round-robin algorithm. In this section, we prove that this algorithm gives an improved bound for envy in Section 3.3.1. We extend this result to the permutation instances in Section 3.3.2. For Section 3.3.3, the original goal was to use the envy bound to show that any allocation returned by envy-reducing round-robin for a permutation instance is a t -tolerant EFX allocation for any $t \geq \lceil \frac{R}{2} \rceil$. We attempt this proof and find some partial results in Section 3.3.3.1, but we find why this approach does not work in Section 3.3.3.2. We use these results to motivate the construction of a counterexample that requires a tolerance of $t \geq R - 2$ in Section 3.3.4. We close this chapter in Section 3.4 with further thoughts on this envy-reducing round-robin algorithm.

3.1 Classic Round-Robin

The goal of this chapter is to analyze Algorithm 3.1 and to use this as a baseline for our improvements. The main result of this chapter is an upper bound for the envy between two agents, which consists of the value of the highest-valued resource in the worst case.

Lemma 3.1. *For any allocation $\mathbf{X}$ produced by Algorithm 3.1 on an instance with additive valuations, it holds that*

$$v_i(X_j) - v_i(X_i) \leq \max_{x \in X_j} v_i(x) - \varepsilon(R - 1) \leq \max_{x \in [m]} v_i(x) - \varepsilon(R - 1)$$

where $\varepsilon = \min_{x, y \in [m], x \neq y} |v_i(x) - v_i(y)|$ or even

$$v_i(X_j) - v_i(X_i) \leq \max_{x \in X_j} v_i(x) - \min_{x \in X_i} v_i(x) - \varepsilon(R - 1) \leq \max_{x \in [m]} v_i(x) - \min_{x \in [m]} v_i(x) - \varepsilon(R - 1)$$

if $m = Rn$.

Note that in the term $\max_{x \in X_j} v_i(x) - \varepsilon(R-1)$, only $\max_{x \in X_j} v_i(x)$ is positive. Thus, for additive resources, if we remove this resource $x = \arg \max_{x \in X_j} v_i(x)$ from X_j , then i does not envy $X_j - x$.

Corollary 3.2. *For any allocation $\mathbf{X}$ produced by Algorithm 3.1 on an instance with additive valuations, $\mathbf{X}$ is EF1.*

This corollary is often proven directly, for example by Caragiannis et al. [CKM⁺19], without establishing the envy bound of Lemma 3.1. A large part of this chapter is dedicated to improving this envy bound, hence we showed that it holds as a baseline. Note that if i envies j , then the first resource $x \in [m]$ chosen by agent j eliminates this envy. This does not directly follow from j choosing greedily, as x is maximal under v_j , not v_i . However, it will become clear that this holds during the proof of Lemma 3.1. Thinking about this claim may motivate that it may be helpful to have some notation to express „the first resource chosen by agent j “ in a more concise manner.

Notation 3.3. *Let $\mathbf{X}$ be the result of a round-robin algorithm (Algorithm 3.1, Algorithm 3.2, Algorithm 3.3).*

- *The history function $\eta(x) : [m] \rightarrow [R]$ is a shorthand used to express the round $r = \eta(x) \in [R]$ in which a resource $x \in [m]$ was chosen by some agent.*
- *For an agent i and a round $r \in [R+1]$, we define $X_i^r = \{x \in X_i \mid \eta(x) < r\}$. For $r \in [R]$, this is the bundle X_i restricted to the resources allocated before round r , where $X_i^1 = \emptyset$. We include $X_i^{R+1} = X_i$ for formal purposes.*
- *For an agent i and a round $r \in [R]$, $x_i^r \in [m] + \perp$ is the resource $x \in X_i$ for which $\eta(x) = r$ holds, if it exists. If $R < \frac{m}{n}$, then in the last round there is an agent j that does not receive a resource in this round. Hence, we use $\perp$ for x_j^R in this case.*

Note that we often set $v_i(\perp) = 0$ as then for every bundle $S \subseteq [m]$ it holds that $v(S) = v(S + \perp)$, which sometimes allows us to avoid case distinctions.

Throughout this section and Section 3.2, it suffices to think of X_j^r as the bundle agent j had at the start of round r and x_j^r as the resource j chose in round r . The reason why we used a more general wording in the definition is that this changes for the final algorithm, Algorithm 3.3 in Section 3.3. We now prove Lemma 3.1. This proof is intentionally presented in a detailed and introductory manner, as it illustrates some ideas and techniques used throughout this chapter. A shorter and more concise proof will later be given for the more general Lemma 3.8.

Proof of Lemma 3.1. Let $\mathbf{X}$ be an allocation produced by Algorithm 3.1 given the order $\prec$. Consider two agents, agent i and agent j . If $i \prec j$, then i chooses a resource before j in every round. Thus, under the valuation v_i , the resource x_i^r that it chooses in some round r is always valued greater than or equal to the resource x_j^r that j chooses in the same round r . The same holds if j or both do not allocate a resource in the last round, provided that we consider $v_i(\perp) = 0$. Formally, $v_i(x_i^r) \geq v_i(x_j^r)$ for every $r \in [R]$, and thus $v_i(X_i^{r+1}) \geq v_i(X_j^{r+1})$. Therefore, in this case, i never envies j in any round. However, if $j \prec i$, it may happen that $v_i(x_j^r) > v_i(x_i^r)$ since j may highly value the same resources as i and thus, by choosing first, „coincidentally“ pick a resource that i values highly as well. To bound the envy, we use the fact that for every $r > 1$, $v_i(x_i^{r-1}) \geq v_i(x_j^r)$ because i greedily picks its highest-valued, unallocated resource in round $r-1$, leaving j to pick a resource in

round r with at most the same value. Once again we set $v_i(\perp) = 0$, which allows us to split the envy as

$$v_i(X_j) - v_i(X_i) = v_i(x_j^1) - v_i(x_i^R) + \sum_{r=2}^R (v_i(x_j^r) - v_i(x_i^{r-1})) \quad (3.1)$$

into a positive part $v_i(x_j^1) - v_i(x_i^R)$ and a non-positive part $\sum_{r=2}^R (v_i(x_j^r) - v_i(x_i^{r-1}))$. It holds that $v_i(x_j^1) - v_i(x_i^R) \leq \max_{x \in X_j} v_i(x)$, and if $x_i^R \neq \perp$, then it is even upper bounded by $\max_{x \in X_j} v_i(x) - \min_{x \in X_i} v_i(x)$, which is guaranteed if $nR = m$. For the non-positive part, every summand is non-positive. Hence, if ε is the smallest difference between the values of two distinct resources, then

$$\sum_{r=2}^R (v_i(x_j^r) - v_i(x_i^{r-1})) \leq (R-1)\varepsilon.$$

Putting both the positive and non-positive parts together yields the upper bound on the envy. $\square$

Note that in this proof, for the case where $i \prec j$, we showed that i does not envy j . This allows the following observation:

Observation 3.4. *Let $\mathbf{X}$ be the allocation at any time during the execution of Algorithm 3.1 with some order $\prec$. Then the envy-graph $E_{\mathbf{X}}$ is acyclic, and $\prec^{-1}$ is a topological order for $E_{\mathbf{X}}$, where $\prec^{-1}$ is the reverse order of $\prec$ defined by $i \prec^{-1} j$ if and only if $j \prec i$ for all $i, j \in [n]$.*

We now turn toward the limitations of the algorithm. We first show that there are instances where the allocation resulting from Algorithm 3.1 is not PO.¹

Example 3.5. $n = 2$, $m = 6$ and two fixed values h, l , with the valuations:

$x \in [m]$	1	2	3	4	5	6
$v_1(x)$	$h + 3$	$h + 2$	$h + 1$	h	l	0
$v_2(x)$	h	$l + 4$	$l + 3$	$l + 2$	$l + 1$	l

The idea here is that h is a high value and l is a low value, so that under Algorithm 3.1 with $1 \prec 2$, the resources are selected in the natural order of $[6]$. However, there is an allocation that is better for both agents, as both receive one more resource the valued at least at h .

Lemma 3.6. *Let $\mathbf{X}$ be the allocation returned from Algorithm 3.1 given the input Example 3.5 along with $1 \prec 2$. Then for $h = 100$ and $l = 1$, $\mathbf{X}$ is not Pareto optimal.*

Proof. The allocation returned from Algorithm 3.1 given the input Example 3.5 along with $1 \prec 2$ is $\mathbf{X} = (\{1, 3, 5\}, \{2, 4, 6\})$, with $v_1(X_1) = 2h + 4 + l$ and $v_2(X_2) = 3l + 6$. Consider the allocation $\mathbf{Y} = (\{2, 3, 4\}, \{1, 5, 6\})$ with $v_1(Y_1) = 3h + 3$, $v_2(Y_2) = h + 2l + 1$. We find that $\mathbf{Y}$ Pareto dominates $\mathbf{X}$ if we use sufficiently large values for h . E.g., for $h = 100$, $l = 1$, it holds that

$$v_1(X_1) = 200 + 4 + 1 < 300 + 3 = v_1(Y_1)$$

¹The example we use is loosely based on one found in Wikipedia (visited 13.05.2026)

and

$$v_2(X_2) = 3 + 6 < 100 + 2 + 1 = v_2(Y_2).$$

□

Now for the final part of this section, we want to consider for what t any allocation $\mathbf{X}$ created by Algorithm 3.1 is a t -tolerant EFX allocation. One trivial lower bound for t is $t \leq R - 1$. This holds because every bundle has a size of at most $R = \lceil \frac{m}{n} \rceil$ and, in the case of envy, every bundle contains at least one resource that eliminates envy when removed, as the allocation is EF1 by Corollary 3.2. We now present a simple example where t is required to be close to this trivial bound.

Example 3.7. $m = n\delta$ for a fixed $\delta \geq 1$ and n , with the valuations

$$v_i(x) = x, \forall i \in [n], x \in [m].$$

Let $\mathbf{X}$ be the allocation produced by Algorithm 3.1 on Example 3.7 with the natural order on $[n]$ as input. Consider the agents 1 and n . For every $r \in [R]$ where $R = \delta$, $v_n(x_1^r) - v_n(x_n^r) = n - 1$. Thus, for these two agents, the envy reaches its maximal value given by Lemma 3.8,

$$v_n(X_1) - v_n(X_n) = (n - 1)\delta = n\delta - 1 + (\delta - 1) = \max_{x \in [m]} v_n(x) - \min_{x \in [m]} v_n(x) - 1(R - 1).$$

Let $r \in [R]$ be a round for which the removal of x_1^r does not resolve envy, i.e.,

$$v_n(x_1^r) = m - n(r - 1) = nR - n(r - 1) < R(n - 1) = v_n(X_1) - v_n(X_n).$$

Note here that $v_n(x_1^r) = m - n(r - 1)$ holds as agent 1 chooses greedily and first in every round while in each round the n highest-valued resources are allocated. We solve this for r :

$$nR - n(r - 1) < R(n - 1) \tag{3.2}$$

$$\Leftrightarrow nR + n - R(n - 1) < nr \tag{3.3}$$

$$\Leftrightarrow r > R + 1 - \frac{n - 1}{n}R = \frac{R}{n} + 1. \tag{3.4}$$

Since our goal is to find the number of resources that do not eliminate envy when removed, we can do this by considering the number of rounds $r \in \mathbb{N}$ for which both $r \leq R$ and Equation (3.4) hold. Thus, for $\mathbf{X}$ to be a t -tolerant EFX allocation, it is required that

$$t \geq R - \frac{R}{n} - 1 \geq R \frac{n - 1}{n} - 1.$$

Therefore, in general, the larger n is, the closer t approaches $R - 1$. Practically, there are instances for which $t \geq R - 2$, as this holds for $n = R$. In conclusion, the minimal t for which every allocation returned by Algorithm 3.1 is a t -tolerant EFX allocation satisfies $R - 1 \geq t \geq R - 2$.

In the next section, we aim to reduce this by eliminating the issue that some agents always get to choose a resource before others. However, for this, we need a more dynamic approach.

Algorithm 3.2: ORDER-VARYING ROUND-ROBIN

Input: Agent amount n , resource amount m and valuations $(v_i)_{i \in [n]}$, an order of agents for every round $(\prec_r)_{r \in [R]}$ where $R = \lceil \frac{m}{n} \rceil$

Output: An EF1 allocation $\mathbf{X}$

```

// Initialization
1  $M \leftarrow [m]$ 
2 forall  $i \in [n]$  do
3    $X_i \leftarrow \emptyset$ 

// Main loop
4 forall  $r \in [R]$  do
5    $N \leftarrow [n]$  ordered by  $\prec_r$ 
6   forall  $i \in N$  do
7     if  $M \neq \emptyset$  then
8        $x \leftarrow \arg \max_{x \in M} v_i(x)$ 
9        $M \leftarrow M - x$ 
10       $X_i \leftarrow X_i + x$ 

```

3.2 Order-Varying Round-Robin

At the start of this chapter, we observed that the fairness of round-robin is limited by Constraint ii, since it allows envied agents to choose resources before the agents who envy them. In this section, we discuss how to modify Algorithm 3.1 and lay an intuitive basis for how this improvement works and what its limitations are. There are instances, where no matter what order is used as an input for Algorithm 3.1, the allocation produced suffers from Constraint ii. E.g. Example 3.7, as no matter which agent starts, all other agents become and remain envious of the agent choosing first. Hence, instead of one fixed order $\prec$, we use a sequence of orders as an additional input for the algorithm. In every round $r \in [R]$, the order $\prec_r$ is used to determine the sequence in which the agents get to chose. The rest of this algorithm remains the same as Algorithm 3.1 and is shown in Algorithm 3.2.

We now show that using multiple orders, does not change the fairness guarantees we had by Lemma 3.1 and Corollary 3.2.

Lemma 3.8. *For any allocation $\mathbf{X}$ produced by Algorithm 3.2 on an instance with additive valuation with any sequence of strict total orders $(\prec_r)_{r \in [R]}$ of $[n]$, it holds that*

$$v_i(X_j) - v_i(X_i) \leq \max_{x \in X_j} v_i(x) - \varepsilon(R-1) \leq \max_{x \in [m]} v_i(x) - \varepsilon(R-1)$$

where $\varepsilon = \min_{x, y \in [m], x \neq y} |v_i(x) - v_i(y)|$, or even

$$v_i(X_j) - v_i(X_i) \leq \max_{x \in X_j} v_i(x) - \min_{x \in X_i} v_i(x) - \varepsilon(R-1) \leq \max_{x \in [m]} v_i(x) - \min_{x \in [m]} v_i(x) - \varepsilon(R-1)$$

if $m = Rn$.

The proof is essentially the same as the proof for Lemma 3.1 and thus is kept a bit more concise. Also note, that throughout this chapter, we mainly make arguments about some agent i envying an agent j . Starting with this proof, we use v instead v_i for these argument, if we are working on only one valuation.

Proof. Consider an agent i envying the bundle of an agent j . Let x_j^1 be the item agent j was allocated in the first round. As before, it holds for every $r > 1$ that $v(x_i^{r-1}) \geq v(x_j^r)$. Thus, we can split the sum of the resource valuations into

$$v(X_j) - v(X_i) = v(x_j^1) - v(x_i^R) + \sum_{r=2}^R (v(x_j^r) - v(x_i^{r-1})). \quad (3.5)$$

With $\sum_{r=2}^R (v(x_j^r) - v(x_i^{r-1})) \leq -\varepsilon(R-1)$ and $v(x_j^1) \leq \max_{x \in [m]} v(x)$ as well as

$$v(x_i^R) \leq \begin{cases} 0 & \text{if } x_i^R = \perp \\ \min_{x \in [m]} v(x) & \text{otherwise} \end{cases},$$

the positive and non-positive summands together upperbound the envy. $\square$

Using more than one order to extend the capabilities of round-robin has already been done for different purposes. Amanatidis et al. [ABFR⁺21] used such an approach to show that EFX allocations exist for instances using only additive valuations which are additionally restricted, in the following manner. For every $i \in [n]$, there is a $c_i \in \mathbb{R}_{>0}$ such that for all $x \in [m]$ it holds that $v_i(x) \in [c_i, 2c_i]$. Using Lemma 3.1, this already holds for Algorithm 3.1 if $m = Rn$. To see why, consider that envy is bounded by $\max_{x \in [m]} v_i(x) - \min_{x \in [m]} v_i(x) \leq c_i$ and removing any resource reduces the envy by at least c_i . However, if $m < nR$, then it may be that an envious agent does not get to choose a resource in the last round, which resulting in envy up to $2c_i$. To circumvent this, they use two different orders. For the rounds $r \in [R-1]$, they use an arbitrary strict total order $\prec$. Then, in the last round R , they use the reversed order $\prec^{-1}$. Recall that by Observation 3.4, $\prec^{-1}$ is a topological order of the envy-graph for the allocation at the start of the last round. Thus, if an agent j chooses a resource in this last round, then every agent i envious of j has already chosen a resource in this round. So if we have this case where $j \prec i$ and i envies j , then either neither agent selects a resource in the last round, or at least i does. If $x_i^R \neq \perp$, then using Equation (3.5), this again bounds the envy at c_i . If neither of the two agents does, then we can do something quite similar to Equation (3.5) by not considering the last round:

$$v(X_j) - v(X_i) = v(x_j^1) - v(x_i^{R-1}) + \sum_{r=2}^{R-1} (v(x_j^r) - v(x_i^{r-1})) \leq 2c_i - c_i.$$

This works because $x_i^{R-1} \neq \perp$. The last case to cover is if i envies j and $i \prec j$. By Observation 3.4, this envy can only be created during the last round. We show that any resource $x_j^r \in X_j$ eliminates envy by splitting the difference between the value of the two bundles as follows:

$$v(X_j) - v(X_i) = \sum_{r' < r} (v(x_j^{r'}) - v(x_i^{r'})) + v(x_j^r) + \sum_{r'=r}^{R-1} (v(x_j^{r'+1}) - v(x_i^{r'})) - v(x_i^R).$$

Here, for all $r' < r \leq R$, it holds that $v(x_j^{r'}) - v(x_i^{r'}) \leq 0$, as in these rounds agent i chooses first and greedily. And for all rounds $r' \geq r$ where $r' < R$, it holds that $v(x_j^{r'+1}) \leq v(x_i^{r'})$ since agent i chooses first here as well. Thus, the only positive summand is $v(x_j^r)$, and thus removing x_j^r eliminates envy. Note, that this envy created in this last round is eliminated by the removal of any resource, independent of the constraint $v_i(x) \in [c_i, 2c_i]$ for all $x \in [m]$ and $i \in [n]$.

Another result where round-robin with varying orders is used is in an extended setting where there are both goods and chores. In this model, additive valuations are not restricted to be greater than or equal to zero. Thus, there may even be resources that no agent prefers to have in its bundle. The definition of EF1 is then extended by allowing envy between two agents i and j if there is a resource in $X_j \cup X_i$ that eliminates the envy if it is removed from the bundle in which it is contained. This setting is more challenging, and a simple round-robin approach does not suffice for creating EF1 allocations, as shown by Aziz et al. [ACIW21, Proposition 3]. They propose the *double round robin algorithm* as the solution to this problem. It works by splitting the resources into two sets. The first set C consists of all resources that every agent values non-positively, and the set G consists of those resources where there is at least one agent valuing them strictly positively. They then run round-robin on the resource in C using a strict total order $\prec$, distributing the resources no agent wants. Then, they do another run of round-robin using the reverse order $\prec^{-1}$ and only the resources G . Combining the bundles returned from both runs then results in an EF1 allocation [ACIW21, Theorem 1]. Note that the second instance of round-robin is an extension of Algorithm 3.1, as they introduce dummy items for agents, which agents may choose if they have no resources left that they value positively. After termination of the second round-robin, these dummy resources are removed. This may lead to different bundle sizes.

Observe that both approaches use some fixed order which is reversed at a fixed time. Reversing the order is already quite potent, as it allows envious agents to choose before the agents they envy. However, as soon as new envy is created, this is no longer guaranteed when reversing the order again. So instead, we take a more dynamic approach.

Definition 3.9. An order $\prec_r$ is envy-reducing in round r of an execution of Algorithm 3.2 or Algorithm 3.3 if

$$v_i(X_i^r) < v_i(X_j^r) \Rightarrow i \prec_r j$$

where X_i^r, X_j^r are the bundles of agents i, j at the start of round r .

Note that, in general, a single round of round-robin with an envy-reducing order does not necessarily reduce envy. The precise guarantee is merely that envy does not increase. However, for non-degenerate instances, such as those we start to consider in Section 3.2.1, a reduction in envy is guaranteed.

In this section, we study the guarantees satisfied by an allocation returned by Algorithm 3.2 when provided with a sequence of envy-reducing orders as input. We then consider the primary challenge in constructing such sequences, as identified by Lemma 3.11. Namely, if an envy-cycle exists at the start of a round, then no envy-reducing order exists for that round and the current partial allocation. This issue is addressed in Section 3.3. For the time being, we assume the existence of such a sequence in order to derive an upper bound on the envy of an allocation produced by Algorithm 3.2.

Observation 3.10. Let $\mathbf{X}$ be the allocation created by Algorithm 3.2 on some instance with a sequence of envy-reducing orders $(\prec_r)_{r \in [R]}$. Then for two agents i, j where i envies j , it holds that

$$v_i(X_j) - v(X_i) \leq v_i(X_j^{r_g}) - v_i(X_i^{r_g}) + v_i(x_j^{r_g}) - v_i(x_i^{r_g}) + \sum_{r=r_g+1}^R (v_i(x_j^r) - v_i(x_i^r)),$$

where $r_g = \max\{r \in [R] \mid v_i(X_j^r) - v(X_i^r) \leq 0\}$ is the last round at the start of which i does not envy X_j^r . Here, $v_i(X_j^{r_g}) - v_i(X_i^{r_g}) \leq 0$ by definition, and $v_i(x_j^r) - v_i(x_i^r) \leq 0$ for all $r > r_g$ because of the envy-reducing order.

This is inherently similar to the results in Lemma 3.1 and Lemma 3.8 by limiting the envy to the difference of the values of two resources. For Lemma 3.8, this difference is $v(x_j^1) - v(x_i^R)$, while for Observation 3.10, it is $v(x_j^{r_g}) - v(x_i^{r_g})$. So for arbitrary orders, it may be the difference between the highest-valued resource to the lowest, while for an envy-reducing order, it is the difference between two resources selected in the same round, where the lower-valued resource $x_i^{r_g}$ is chosen greedily. In other words, the envy is bounded by the difference between the highest-valued unallocated resource at the start of round r_g and the $(n - 1)$ -th highest-valued, unallocated resource in the same round. Though without further restriction of the additive valuations, the difference between the highest-valued resource and even the second-highest-valued resource can be arbitrarily large. In such cases, using envy-reducing orders may not produce fairer allocations. To avoid these cases, we introduce a restriction on the valuations in Section 3.2.1.

We now discuss the problem with the assumption that such a sequence of envy-reducing orders exists for every instance. An initial approach to compute such a sequence might be, to run Algorithm 3.2 without a sequence of orders as input and then compute an envy-reducing order in every round. However, as soon as there is an envy-cycle at the start of a round, no strict total order exists for which every envious agent is allowed to choose before the envied agent. Note that envy-cycles do not appear during the execution of Algorithm 3.1 because of Observation 3.4. By varying the order, we enable this, and by using an envy-reducing order, we may even promote this. Think of some path of length l in the envy-graph with a source s and a sink t . By the envy-reducing order and by the transitivity of that order, s chooses before t and at least $l - 1$ agents chose between them. If s and potentially the other agents chose resources valued highly under v_t , then t may be forced to choose resource x_t valued significantly lower than x_s , the resource chosen by agent s . With a sufficiently high difference $v_t(x_s) - v_t(x_t)$, agent t starts envying s , creating a new edge from t to s in the envy-graph. Of course, because of the envy-reducing order, the envy between the other agents does not increase, or even decreases for non-degenerate instances. However, this decrease in envy is not bounded, so if it is 0 or close to it, this may result in little to no change in the envy between any agent and its successor in the path. With no envy edges disappearing and a new edge forming from t to s , a new envy-cycle is closed.

Lemma 3.11. *Let $\mathbf{X}$ be the allocation created by Algorithm 3.2 on the input of Example 3.12 along with the envy-reducing orders $1 \prec_1 2, 2 \prec_2 1$. Then for $h = 100, l = 1$, $E_{\mathbf{X}}$ contains an envy-cycle.*

Note that the envy-cycle does not matter for this execution of the algorithm, as it only appears at the end of the algorithm. However, we can modify Example 3.12 by adding two more resources valued lower than all other resources by both agents. Then, the envy-cycle exists after the second round, as it does for the original instance, resulting in no envy-reducing order existing at the start of the third and last round.

Example 3.12. $n = 2, m = 4$ and valuations:

$v_i \backslash x$	1	2	3	4
$v_1(x)$	$h + 1$	h	h	l
$v_2(x)$	h	$l + 2$	$l + 1$	l

Proof. We start by considering the allocation after the first round, where agent 1 chose first, resulting in agent 2 being envious at the start of round 2 as

$$v_2(x_2^1) = h > l + 2 = v_2(x_2^2).$$

This way, $2 \prec_2 1$ is the only possible envy-reducing order, resulting in

$$v_1(X_2) = h + h > h + 1 + l = v_1(X_1) \quad \text{and} \quad v_2(X_1) = h + l > l + 2 + l + 1 = v_2(X_2)$$

for the final allocation. Thus, there is an envy-cycle. $\square$

Most of our round-robin examples hinge on a „bad“ order being chosen. This is the case here again. The choice of the first order determines the second and, thus, the existence of the envy-cycle. So, to have the guarantees of Observation 3.10 in general, we initially tried to find some rule or heuristic to avoid a sequence of envy-reducing orders that determines the existence of an envy-cycle as $1 \prec_1 2$ did for Example 3.12. Another way of thinking about this is to use any sequence of envy-reducing orders until there is an envy-cycle in some round, go back to the round where this envy-cycle could last have been avoided, and use a different order there. However, the round where a „bad“ order is chosen can be arbitrarily far away from the envy-cycle. The next lemma shows the idea behind this, namely, that an order in the first round may determine the order in the last round. It also shows that this is possible while having only very little envy, which is helpful for combining such an instance with other instances e.g. for creating envy-cycles whose existence are determined by the order in earlier rounds.

Lemma 3.13. *For Example 3.14, there is only one sequence of envy-reducing orders starting with $1 \prec_1 2$. Furthermore, let $\mathbf{X}$ be the allocation created by Algorithm 3.2 from Example 3.14 and this sequence. As δ increases, the envy in the allocation approaches 0.*

Example 3.14. $m = \delta n$ for some fixed $\delta \in \mathbb{N}$ and $n = 2$ with the valuation

$$v(x) = \begin{cases} 7 - x & \text{if } x \in [3] \\ 4 - \sum_{k=2}^{r-1} (2^{2-k} 3) - 2^{1-r} 3 & \text{if } x = 2r \text{ for some } r > 1 \\ 4 - \sum_{k=2}^r (2^{2-k} 3) & \text{if } x = 2r + 1 \text{ for some } r > 1 \end{cases}$$

for both agents.

Note that Example 3.14 uses an identical valuation for both agents.

Proof. To prove Lemma 3.13, we show that after every round of Example 3.14 there is envy and that the direction of this envy alternates in every round. As there are only two agents, this already uniquely determines the sequence of envy-reducing orders. We demonstrate that this is the case for $1 \prec_1 2$, but it holds for $2 \prec_1 1$ analogously.

Formally, we show that the following equation holds for every $r \geq 1$:

$$v(X_1^{r+1}) - v(X_2^{r+1}) = (-1)^{r+1} 2^{1-r} \quad (3.6)$$

This implies that the sign of $v(X_1^{r+1}) - v(X_2^{r+1})$ alternates every round, which corresponds to the envy edge alternating direction because the agents have identical valuations. Furthermore, 2^{1-r} approaches 0 as r approaches infinity, which satisfies the second part of Lemma 3.13.

To do this, we first consider the difference between any two resources that agent 1 and agent 2 choose in the same round $r \in \mathbb{N}$. Observe that v is strictly increasing on $[m]$ (with the natural order on $[m]$). By the greediness of both agents, the first resource chosen

in a round r is $2r - 1$ and the second $2r$. The difference in their values depends on the round. For $r = 1$, it holds that $v(1) - v(2) = 1$, while for $r = 2$, it holds that

$$v(3) - v(4) = 4 - \left(4 - \sum_{k=2}^{r-1} (2^{2-k} 3) - 2^{1-r} 3 \right) = 2^{1-r} 3.$$

The latter difference is identical to the difference in all other rounds $r > 2$ as well, by the following chain of equations:

$$v(2r - 1) - v(2r) = 4 - \sum_{k=2}^{r-1} (2^{2-k} 3) - 2^{1-r} 3 - \left(4 - \sum_{k=2}^r (2^{2-k} 3) \right) \quad (3.7)$$

$$= 2^{2-r} 3 - 2^{1-r} 3 = 2^{1-r} 3 \quad (3.8)$$

We now have the necessary components to prove Equation (3.6) by induction. For $r = 1$, observe that

$$\begin{aligned} v(X_1^{r+1}) - v(X_2^{r+1}) &= v(1) - v(2) \\ &= 1 = (-1)^{r+1} 2^{1-r} \end{aligned}$$

For $r > 1$, we use the following chain of equations and explain the relevant steps below:

$$v(X_1^{r+1}) - v(X_2^{r+1}) = v(X_1^r) - v(X_2^r) + v(x_1^r) - v(x_2^r) \quad (3.9)$$

$$= (-1)^r 2^{1-(r-1)} + (-1)^{r+1} 2^{1-r} 3 \quad (3.10)$$

$$= (-1)^{r+1} 2^{1-r} (3 - 2^1). \quad (3.11)$$

Note that in Equation (3.10), we used both the induction hypothesis and Equation (3.7). Additionally, we added the factor $(-1)^{r+1}$ to account for whether agent 1 or agent 2 chooses first. To see why this holds, consider the case where r is even. Then $v(X_1^r) - v(X_2^r) > 0$ by the induction hypothesis. Thus, in round r , agent 2 chooses first, so the value from Equation (3.7) is prepended with a factor of -1 . Vice versa if r is odd. $\square$

It is even possible to combine both Example 3.12 and Example 3.14 such that the envy-cycle appears in the last round and is determined by the choice of $1 \prec_1 2$. To do this, we take the union of the resources from both examples. The high-level idea is to ensure that the resources of Example 3.14 are the ones chosen first. For this, we can add a constant to the valuation of each resource from that example. This does not influence the envy, provided that the valuations increase by the same amount. By adding a sufficiently large value, all resources of Example 3.14 are chosen first while the envy still approaches zero as stated in Lemma 3.13. In round $\delta + 1$, the agent choosing first is determined by the agent choosing first in round 1. Now, as described in the proof of Lemma 3.11, an envy-cycle is created in round $\delta + 2$. The residual envy of Example 3.14 does not interfere with this process, by either using a sufficiently large δ or by scaling the valuations of the resources of Example 3.12. This effectively scales the envy created in the last two rounds, overriding any prior envy.

This does not answer the question of whether sequences of envy-reducing orders always exist. It might be possible, especially with more agents, to combine Example 3.14 and Example 3.12 in a way such that Example 3.14 already limits the amount of candidates for an sequence of envy-reducing order and then there is an instance of Example 3.12 creating an envy-cycle for each of these candidates. However, this remains pure speculation. Rather than further exploring, whether sequences of envy-reducing orders always exist because of envy-cycles, we continue in Section 3.3 by eliminating them, as have done in Section 2.1. However,

this increases the complexity of the algorithm. Therefore, to gain an understanding of our goal, its proof and the challenges with it, we continue to assume for the next section that sequences of envy-reducing orders exist, allowing us to build an intuition for what this implies for the allocations produced by Algorithm 3.2.

3.2.1 Intuition: Bounding Tolerance in Permutation Instances

With Constraint i from the beginning of this chapter, we have seen how large differences between the values of two resources can pose a challenge for finding t -tolerated EFX allocations with non-near-trivial t . This has not changed with Observation 3.10. In arbitrary additive valuations, the gap between $v(x_j^{r_g})$ and $v(x_i^{r_g})$ can still be arbitrarily large, leading to arbitrarily large envy and in turn enabling many resources to not eliminate that envy when they are removed.

To demonstrate that Algorithm 3.2 has the potential to be fairer in some regards, we consider the following restriction on the valuations.

Definition 3.15 (Permutation instance). *A permutation valuation is an additive valuation where the values of the resources are a permutation of $[m]$. A permutation instance is a fair division problem instance where every valuation is a permutation valuation.*

It is currently unknown whether EFX allocations always exist for permutation instances. Researching this existence might be a fruitful direction because permutation valuations provide more structure to value of the bundles. Furthermore, if we were to find that EFX allocations always exist for permutation instances, this might open up interesting directions and interpretations for EFX on general non-degenerate additive instances. The reason for this is that every non-degenerate additive valuation has exactly one permutation valuation ordering the resources in the same order. Thus, studying allocations for corresponding permutation instances might open up new perspectives on EFX. Furthermore, for some non-degenerate additive instances, it might be possible to transform them into permutation instances. One idea here would be to scale and translate the valuations so that all resources are valued with natural numbers, and then fill up all „gaps“ to a permutation valuation using dummy resources, which in turn are highly valued by dummy agents. Such constructions are most likely not helpful for all non-degenerate instances, but they might help to generalize some results found on permutation instances. On the other hand, if we were to find that EFX allocations do not in general exist for non-degenerate additive instances, then we can interpret that the existence of fair allocations may hinge upon the valuations being uniformly spaced. In conclusion, studying the existence of EFX on permutation instances may be easier than in the more general additive case, while offering many directions to extend or interpret these results.

We now discuss how large the difference between $v(x_j^{r_g})$ and $v(x_i^{r_g})$ can become at most. Here and for the rest of this section, we argue about the bundles and rounds as defined in Observation 3.10. For permutation valuations, we find that this difference is bounded by the amount of resources that are valued between the two. We refer to these resources $G = \{x \in [m] \mid v(x_i^{r_g}) < v(x) < v(x_j^{r_g})\}$ as the *gap*. In Section 3.3, we formally redefine the gap, taking into account the envy-cycle elimination, but for the case where we have no envy-cycle elimination, both definitions are the same.

Lemma 3.16. *Let $\mathbf{X}$ be the allocation returned from Algorithm 3.2 given a permutation instance and a sequence of envy-reducing orders. The envy between an agent i and some other agent j is bounded by*

$$v(X_j) - v(X_i) \leq |G| + 1 - (R - r_g).$$

$r \in [R]$	1	2	3	4	5	6	7
$v(x_j^r)$	21	18	17	14	13	4	1
$v(x_i^r)$	20	19	16	15	7	5	2
$v(x_k^r)$	12	11	10	9	8	6	3

Table 3.1: Example execution of Algorithm 3.3. Resources (identified by v_i) mapped to the agent that chose them over seven rounds.

Proof. This follows from Observation 3.10 and the fact that $v(x_j^{r_g}) - v(x_i^{r_g}) = |G| + 1$, along with $v(x_j^r) - v(x_i^r) \leq -1$ for $r > r_g$. $\square$

Thus, to bound the envy, we first consider how large the gap might become. For this, observe that $|G| \leq (n-1)r_g$, as all resources in G are allocated by the time i chooses a resource in round r_g . There are at most $nr_g - 1$ resources chosen in total beforehand. However, $r_g - 1$ of these are chosen by agent i , and for every resource x_i^r with $r < r_g$, it holds that

$$v(x_i^r) > v(x_j^{r_g}),$$

by the greediness of agent i . By definition, these $r_g - 1$ resources are not in the gap, and hence $|G| \leq nr_g - 1 - (r_g - 1) = (n-1)r_g$. Another matter are the resources selected by agent j in the rounds before r_g . Of course, these can contribute to the size of the gap. However, for a round $r < r_g$ where $v(x_j^r) < v(x_j^{r_g})$, it holds that $v(x_j^r) \leq v(x_i^r) - 2$, since $v(x_i^r) > v(x_j^{r_g})$. So even though x_j^r contributes one to the size of the gap, it negatively contributes to the envy $v(X_j^r) - v(X_i^r)$ with a value of two. Of course, these negative contributions may be balanced out by other rounds $r' < r_g$ where $v(x_j^{r'}) > v(x_i^{r'})$. However, this is only possible if $x_j^{r'} \notin G$. Whether this has enough of an impact, so that there is an instance for which the allocation produced by Algorithm 3.2 as an envy after round r_g with $v(X_j^{r_g+1}) - v(X_i^{r_g+1}) > (n-2)r_g + 1$ constitutes one of the main questions of this chapter.

To get an intuition for why this holds, we consider an example for which this is exactly the case in Table 3.1. This table displays the resources that the agents j , i , and k chose over 7 rounds. Note that the resources are identified via their value to agent i . This works since $v_i = v$ is a permutation of $[m]$ and is thus bijective. In this example, we are only concerned with the envy from i towards j and thus do not specify the other valuations, for which there are multiple choices. One of the simplest approaches is to consider that the resources which agents j and k chose are simply the ones valued highest in order. For example: $v_j(v_i^{-1}(21)) = 21$, $v_j(v_i^{-1}(18)) = 20$, $v_j(v_i^{-1}(17)) = 19$, and so on. Another aspect that we have not specified directly is the sequence of orders $(\prec_r)_{r \in [7]}$, though for most rounds it is implicit. We will now go through the rounds $r \in [7]$ highlighting the key aspects of this example, and why it might be a candidate for an instance and execution with the maximum envy and most resources not eliminating this envy. One perspective which may help and which we have already started using a bit with the gap, is thinking of the resources $[m]$ as a sequence ordered descending by v . Specifically we consider which agent chooses what resource from this sequence. For Table 3.1 we do this, by looking at parts of the sequence S , which is this sequence of resources mapped to the agent which chose them.

We start with the rounds $r < r_g$ while keeping in mind that our goal is to maximize envy. The approach we choose here is to minimize $v(X_i^{r_g}) - v(X_j^{r_g})$ while maximizing the gap. To maximize the gap, agent k only chooses resources that will later be in the gap. To minimize $v(X_i^{r_g}) - v(X_j^{r_g})$, j always chooses first if i is not currently envious of j , and the

resources that j chooses are the ones i currently values highest. So if we consider S up to $x_j^{r_g}$, the sequence alternates between two i 's and j 's like so:

$$j, i, i, j, j, i, i, j \dots$$

Furthermore, $v(X_j^r) - v(X_i^r)$ alternates between 1 and 0.

In round r_g , agent j chooses first, then agent k , to make the gap larger by one, and finally agent i . For S , the agents choosing $x_j^{r_g}$, $x_i^{r_g}$, and all the resources between them form the sequence:

$$\dots, j, k, k, k, k, k, k, i \dots$$

For the final rounds $r > r_g$, we seek to minimize $v(x_i^r) - v(x_j^r)$ for every round, as this difference directly decreases the envy. Here we also want to consider one more aspect. Recall that our final goal is to find small bound for t so that Algorithm 3.2 returns a t -tolerant EFX allocation any permutation instance and sequence of envy-reducing order as input. To get an intuition for this as well, as a secondary goal we think about how v values the resources in $X_j \setminus X_j^{r_g+1}$ such that the maximum amount of resources do not eliminate envy. Consider a resource x_j^r that does not eliminate envy when removed. Formally:

$$v(x_j^r) < v(X_j) - v(X_i).$$

We can reformulate this as

$$v(x_i^r) < \sum_{r' \in [R] - r} (v(x_j^{r'}) - v(x_i^{r'})).$$

In other words, whether a resource eliminates envy does not directly depend on its own value, but on the value of x_i^r and the envy created in the other rounds. The value of x_i^r only depends upon the other agents who choose resources i values highly before i chooses in round r . Observe that in Table 3.1, all agents up to round r when i chooses have chosen the $n(r-1)$ resources that i values most. Furthermore, since $k \prec_r i$, k also allocates one of the resource which i values highly, decreasing the value of $v(x_i^r)$ to its smallest possible value. The other aspect, influencing whether x_j^r eliminates envy, is the envy created in the other rounds. With Table 3.1 we have thus far tried to intuitively maximize the envy. So there is nothing obvious to be improved in that aspect

To continue with this for $r > r_g$, we Table 3.1 uses $v(x_i^r) - v(x_j^r) = 1$. This is the smallest possible difference because i chooses before j due to the envy-reducing order. Together, this results in the order $k \prec_r i \prec_r j$, and the final part of the sequence S being:

$$\dots, k, i, j, k, i, j.$$

In total, the envy after round $r_g = 5$ is $v(X_j^{r_g+1}) - v(X_i^{r_g+1}) = 13 - 7 = 6$, but because of the last two rounds, it decrease this it to $v(X_j) - v(X_i) = 4$. Thus, there is only one resource in X_j that does not eliminate envy, which is $v^{-1}(1)$.

This type of example can be extended to more agents and resources, with the generalized upper bound for envy being

$$v(X_j) - v(X_i) \leq (n-2)r_g + 1 - (R - r_g) = (n-1)r_g - R + 1. \quad (3.12)$$

For the rest of this section, we describe the intuition behind the assumption that Algorithm 3.2 returns $\lceil \frac{R}{2} \rceil$ -tolerant EFX allocations when given permutation instances along with sequences of envy-reducing order. It is also possible to skip this last part of the section, as all arguments are formalized in Section 3.3, except for Lemma 3.17. Nonetheless, it sets

up the motivation behind some of the proofs of Section 3.3 and establishes the assumptions we make and why, one of which is the fatal one for this approach.

To consider which resource may not eliminate envy, observe that for Table 3.1, all resources in $X_j^{r_g+1}$ are not considered as they are valued at least at $v(x_j^{r_g})$. For those x_j^r with $r > r_g$, consider that for this example $v(x_j^r) = v(x_i^r) - 1$. We can use this to bound $v(x_j^r)$ using the value of the resource chosen by agent i in round r .

Lemma 3.17. *Given any execution of Algorithm 3.2 on a permutation instance with a sequence of envy-reducing orders, let x_i^r be any resource chosen by an agent i in a round r at the start of which agent i envies another agent. It holds that*

$$v(x_i^r) \geq m - nr + 2.$$

Proof. Recall that agent i greedily chooses the highest-valued resource under v in round r . With m being the highest value in total, we consider how many of the highest-valued resources may be chosen until agent i chooses in round r . For the first $r - 1$ rounds, every agent may choose the current highest-valued resource under v , which totals $n(r - 1)$ resources. For round r , as agent i envies another agent, at most $n - 2$ agents choose before agent i . Together by the time agent i chooses in round r , at most the $(nr - 2)$ -highest-valued resources may be chosen. Since i chooses greedily it holds that $v(x_i^r) \geq m - (nr - 2)$. $\square$

If x_j^r does not eliminate envy, then we can string these equations together with Equation (3.12) to get:

$$m - nr + 1 \leq v(x_i^r) - 1 = v(x_j^r) < v(X_j) - v(X_i) \leq (n - 1)r_g - R + 1.$$

Now, using $r_g < r$, this simplifies to

$$m - nr + 1 < (n - 1)r - R + 1,$$

which we can rearrange to lower bound r :

$$\frac{m + R}{2n - 1} < r.$$

Since this equation holds for all rounds r in which x_j^r does not eliminate envy, there can be at most roughly $\frac{1}{2}R$ such rounds. At least for these examples.

To generalize these results, we tried to reduce any allocation $\mathbf{X}$ created by Algorithm 3.2 given a permutation instance and a sequence of envy-reducing orders to Table 3.1. With the abstract goal being to showing the following two claims which holds for the Table 3.1 for any such allocation $\mathbf{X}$:

- (i) That Equation (3.12) gives an upper bound for any envy.
- (ii) That all resources valued higher or equal than $x_j^{r_g}$ eliminate envy.

Just as we did above, the idea would then be to leverage these claims, alongside an argument showing it suffices to consider $v(x_j^r) = v(x_i^r) - 1$, to prove that $\mathbf{X}$ is roughly a $\left(\frac{1}{2}\frac{m}{n}\right)$ -tolerant EFX allocation.

This would improve the worst-case guarantee by showing that any permutation instance has a t -tolerant EFX allocation for a smaller t . We already laid down the base case for this with Example 3.7, which we used to show that Algorithm 3.1 may produce allocations that are not even $\lceil \frac{m}{n} \rceil - 3$ -tolerant. Note that Example 3.7 is a permutation instance. Thus,

by showing that Algorithm 3.2 produces roughly $\frac{1}{2} \frac{m}{n}$ -tolerant EFX allocations, we would achieve a significantly better fairness guarantee compared to regular round-robin.

We have now set up what we wanted to show, and roughly how we planned on going about it. Let us now consider whether Claims i to ii are intuitive and reasonable. For Claim i, we have already built an intuition using Table 3.1. Here, it may be worth considering whether there is some easy way to modify the example to increase the envy. The only way to achieve this is to increase the gap by adding a resource of $X_j^{r_g}$ into it. This then hinges on the question of whether this is doable without decreasing $v(X_j^{r_g}) - v(X_i^{r_g})$. By considering Table 3.1 again, one finds that this is not easily achieved. For Claim ii not to hold, consider that we require a relatively small value in $X_j^{r_g}$, which drastically decreases the envy. More formally, let x_j^r be a resource that does not eliminate envy when removed, with $r < r_g$. Let a be the difference between x_j^g in this case and an example like Table 3.1. By decreasing the value of x_j^g by a , we also decrease the envy after that round by about a . The decrease in envy might be less than a because after round r , agent i may not envy agent j for a larger number of rounds. Thus, agent j may choose first for multiple rounds in a row, potentially building envy up again a little quicker. Nonetheless, this strong impact on envy is the reason why we assumed that this claim may hold.

This Claim ii is false, and in its place we initially used another claim: that there can be resources in $X_j^{r_g}$ that do not eliminate envy but only lead to a significant reduction in the envy, which in turn leads to significantly fewer resources in $X_j \setminus X_j^{r_g}$. This claim is actually correct, and we will show that it holds with Lemma 3.45. However, it no longer suffices to show the claim that there are at most roughly $\frac{R}{2}$ resources that do not eliminate envy, as for this we would need to bound these kinds of resources in $X_j^{r_g}$. And this, surprisingly, is not possible to a degree that would be an improvement upon round robin. We give an example construction where $|X_j^{r_g}| - 1$ resources do not eliminate envy in Section 3.3.4. The main issue with why this was not more obvious was the implicit assumption that we need to greedily maximize the size of the gap, as any envy created earlier disappears before it, leading to a reduction in the final envy and fewer resources not eliminating that envy. But if resources before the gap do not eliminate envy, we can push the gap-round to be in the last round, which in turn increases the envy again.

We formalize the idea behind this in Section 3.3.3.2. As a consolation, we are at least able to bound the amount of resources that do not eliminate envy after r_g . To do this, we first show that Claim i holds in Section 3.3.2. Before we do all of that, however, we first have to deal with the issue that we have been working on the assumption that we can find a sequence of envy-decreasing orders. We begin the next section by circumventing this issue.

3.3 Envy-Reducing Round-Robin

In the last section we have motivated our assumption that using a sequence of envy-reducing orders as input for Algorithm 3.2 may produce roughly $(\frac{m}{2n})$ -tolerant EFX allocations for permutation instances. For this section we formalize these ideas and see in Section 3.3.3.2 where they break.

The first issue we take care of is having to find a sequence of envy-reducing orders as input for Algorithm 3.2. We circumvent this by running envy-cycle elimination as discussed in Section 2.1 at the end of each round, until no envy-cycle remains. This way, at the start of each round, the envy-graph of the current allocation is a directed acyclic graph. Hence, we can use any algorithm computing a topological order to find an envy-reducing

Algorithm 3.3: ENVY-REDUCING ROUND-ROBIN**Input:** Number of agents n , number of resources m , and valuations $(v_i)_{i \in [n]}$ **Output:** An EF1 allocation $\mathbf{X}$

```

// Initialization
1  $R \leftarrow \lceil \frac{m}{n} \rceil$ 
2  $M \leftarrow [m]$ 
3 forall  $i \in [n]$  do
4    $X_i \leftarrow \emptyset$ 

// Main loop
5 forall  $r \in [R]$  do
6   Compute an envy-reducing order  $\prec_r$  for  $(X_i)_{i \in [n]}$ 
7    $N \leftarrow [n]$  ordered by  $\prec_r$ 
8   forall  $i \in N$  do
9     if  $M \neq \emptyset$  then
10       $x \leftarrow \arg \max_{x \in M} v_i(x)$ 
11       $M \leftarrow M - x$ 
12       $X_i \leftarrow X_i + x$ 
13   Eliminate envy-cycles as long as they exist

```

order for each round². With this last augmentation, Algorithm 3.3 is the final version of envy-reducing round-robin that we discuss for the rest of this chapter.

Using envy-cycle elimination at the end of each round opens up a new issue. By running it even once, the envy bound of Observation 3.10 no longer holds in general, at least for the current definition of r_g and the gap. Without Observation 3.10, the envy is not bounded by the difference of two resources, and hence we do not know whether an allocation produced by Algorithm 3.3 is guaranteed to be even EF1. In Section 3.3.1, we show that we can extend the definition of r_g and the gap for this final algorithm Algorithm 3.3. We show that a bound similar to Observation 3.10 exists for Algorithm 3.3 by an induction that follows naturally from the augmented definitions we introduce. Since this approach works well, we continue to use it to show some of the ideas of Section 3.2.1. We start with Claim i, which we prove in Section 3.3.2. Using Claim i we then prove in Section 3.3.3.1 that there is an upper bound $\lceil \frac{R}{2} \rceil + 1$ for the amount of resources which do not eliminate envy, but only for the resources chosen after the gap-round. In Section 3.3.3.2 we even tighten this bound for the cases where there are resources chosen before round r_g which do not eliminate envy.

3.3.1 Redefining the gap

Let $\mathbf{X}$ be an allocation returned by Algorithm 3.3. We do not yet restrict the setting to permutation instances. Let agent i be an agent who envies agent j . In Section 3.2, we defined r_g as the last round $r \in [R]$ for which $v(X_j^r) \leq v(X_i^r)$, where $X_i^r = \{x \in X_i \mid \eta(x) < r\}$ by Notation 3.3. Using this, we split the envy $v(X_j) - v(X_i)$ into

$$v(X_j) - v(X_i) = v(X_j^{r_g}) - v(X_i^{r_g}) + v(x_j^{r_g}) - v(x_i^{r_g}) + \sum_{r=r_g+1}^R (v(x_j^r) - v(x_i^r)),$$

in Observation 3.10. For Algorithm 3.3 under these definitions, it still holds that $v(X_j^{r_g}) - v(X_i^{r_g}) \leq 0$. We now explore why $v(x_j^r) - v(x_i^r) < 0$ does not hold in general. To this end,

²Observe that we have seen in Section 3.2 that the reverse order is used in a few round-robin variants. We have also seen that this reverse order is a topological order with Observation 3.4, and in this way an envy-reducing order before we even introduced it.

consider Example 3.18. In this example, there are three agents along with six resources distributed across two rounds. After the first round, agent i envies both agent j and agent k . Hence, the only restriction on $\prec_2$ is that i chooses first, while j and k can be ordered arbitrarily. For this example, $i \prec_2 j \prec_2 k$ is the order in the final round. The crucial part is that even though i chooses greedily before j (and thus receives a higher valued resource), at the end of the round, an envy-cycle forms between k and i by the same mechanism as in Example 3.12. By Algorithm 3.3, we eliminate this cycle at the end of the second round by swapping the bundles of agent k and agent i . However, k , who chose after i , did not choose the highest-valued resource under v_i , and chose a resource valued even lower than the one chosen by j in this round. Formally, agent k chooses the resource $\{2, 6\}$, though through envy-cycle elimination, they end up forming the bundle X_i , valued at $v(X_i) = m + 0$, which is valued less than $v(X_j) = h + l + 2$, where $h \gg m$. With the definition from Observation 3.10, $r_g = 1$ while $v(x_j^2) = l + 1 > 0 = v(x_i^2)$. Thus, the argument of Observation 3.10 stating that the envy is bounded by the difference of only two resources does not hold for Example 3.18 where Algorithm 3.3 is used.

Example 3.18. Agents i, j, k , resources $[6]$, and the envy-reducing orders $j \prec_1 k \prec_1 i$ and $j \prec_2 i \prec_2 k$, along with the valuations and allocations given by the following table:

$x \in [m]$	1	2	3	4	5	6
$\eta(x)$	1	1	1	2	2	2
agent choosing x	j	k	i	i	j	k
agent ending up with x	j	i	k	k	j	i
$v_i(x)$	h	m	$l + 3$	$l + 2$	$l + 1$	0
$v_k(x)$	0	$h + 2$	$h + 1$	h	0	0
$v_j(x)$	2	0	0	0	1	0

for $h \gg m \gg l$.

When an envy-cycle is eliminated, the value assigned to their own bundle increases for all agents participating in the cycle. Consequently, the envy of all agents involved in the envy-cycle elimination decreases, provided that the agents they envy are not part of the cycle. In the this case, envy may actually increase, but not the maximum envy over the instance, as every bundle received during the envy-cycle elimination was previously owned by another agent. So even if the new bundle is envied more, this is only because some other agent was like this before. Intuitively, we should expect an envy bound similar to Observation 3.10 to exist.

To see why this holds, we look closer at the limitation in the current definition of r_g . Intuitively, this round should be the last round in which the envy of i increases. However, the bundle X_i^r might be entirely different from the bundle that agent i actually possessed at the beginning of round r . Furthermore, as demonstrated in Example 3.18, x_i^r might not be chosen greedily under v_i . Hence, the defining property of the envy-reducing order fails to provide a bound for x_i^r . It is more intuitive to instead consider the actual bundles held by agent i in round r along with their actual choices.

We have implicitly relied on this perspective before: in Section 3.1 and Section 3.2, X_i^r was formally defined as $\{x \in X_i \mid \eta(x) < r\}$, yet we frequently interpreted it in prose as “the bundle agent i held at the start of round r .” This ambiguity was harmless for Algorithm 3.1 and Algorithm 3.2, where both notions coincide. However, for definitions to conform to the setting we are now in with Algorithm 3.3, we have intentionally delayed the formally defining concepts where it is necessary to differentiate the notions, such as r_g or the gap gap. Alternatively we have explicitly referenced what type of bundle is required.

For instance, in Definition 3.9, we utilize the symbol X_i^r but specify in the text that it refers to the bundle type we formalize below.

Definition 3.19 (Historical Bundles). *For execution of Algorithm 3.3, the historical bundle $H_i^r \subseteq [m]$ is the bundle that agent i held at the start of round r . The historical choice $h_i^r \in [m] \cup \{\perp\}$ is the resource that agent i chose in round r , if such a resource exists.*

For formal consistency, we set $H_i^{R+1} = X_i$.

Let us now consider how to properly replace the definition of r_g established in Observation 3.10. The round r_g serves as the final round in which envy is generated, essentially bounding the envy by the value of two resources chosen during that round. Thus far, we have achieved this by selecting the maximum round $r \in [R]$ for which $v(X_j^r) \leq v(X_i^r)$ holds. Through Example 3.18, we have seen that using the round-restricted bundles as defined by Notation 3.3 for agent i no longer suffices under Algorithm 3.3, which motivated the introduction of historical bundles. Consequently, we have two contenders for redefining r_g . We can either retain the round-restricted bundle for agent j , meaning r_g is the maximum round r for which $v(X_j^r) \leq v(H_i^r)$ holds, or we can use historical bundles for both agents, meaning it is the maximum round r for which $v(H_j^r) \leq v(H_i^r)$ holds.

To determine the appropriate bundle type for agent j , we must consider which concept is more critical: the actual bundle that j holds during a specific round, or the bundle that j holds at the termination of the algorithm (which may have been passed between different agents during execution). The historical bundles for j can vary widely in value. If r_g were defined as the last round at the start of which i does not envy j ($v(H_j^r) \leq v(H_i^r)$), then even after an envy-cycle elimination after r_g the new bundle received by j is highly valued by i . However, it would remain ambiguous whether this high valuation stems from resources chosen before or after r_g . While it might be possible to derive an alternative bound using this formulation, we do not pursue it further because retaining the round-restricted bundles provides a clearer approach. Specifically, if $r_g = \max\{r \in [R] \mid v(X_j^r) \leq v(H_i^r)\}$, we maintain the invariant the X_j restricted to round r is higher valued than H_i^r . Even if the bundle is not owned by agent j , who ever does own it, chooses after agent i for all round after r_g , by the envy-reducing order³. With these structural choices explained, we formalize the definitions of the gap round and the gap.

Definition 3.20 (Gap). *Let $\mathbf{X}$ be an allocation returned by Algorithm 3.3, and let agent i be an agent who envies agent j . The gap round $r_g \in [R]$ is the maximum $r \in [R]$ for which $v(X_j^r) \leq v(H_i^r)$. The gap is the set*

$$G = \{x \in [m] \mid v(h_i^{r_g}) < v(x) < v(x_j^{r_g})\}.$$

With these definitions established, the goal of this subsection is to prove an analogue of Observation 3.10 for Algorithm 3.3. The first step towards such a bound is the following lemma.

Lemma 3.21. *Let $\mathbf{X}$ be an allocation returned by Algorithm 3.3, with historical bundles $(H_i^r)_{(i,r) \in [n] \times [R]}$ and historical choices $(h_i^r)_{(i,r) \in [n] \times [R]}$. For any agent i who envies an agent j , and with r_g defined as in Definition 3.20, it holds that*

$$v(X_j) - v(X_i) \leq v(X_j^{r_g+1}) - v(H_i^{r_g+1}) + \sum_{r=r_g+1}^R (v(x_j^r) - v(h_i^r)),$$

where $v(x_j^r) - v(h_i^r) \leq -\varepsilon$ for all $r > r_g$, where $\varepsilon = \min_{x,y \in [m], x \neq y} |v(x) - v(y)|$.

³This definition has the added benefit, that the original definition of X_i^r has merit, as otherwise it would have made more sense to use the historical bundles throughout Section 3.1 and Section 3.2 as well.

To prove Lemma 3.21, we work with both round-restricted and historical bundles. The following are some simple but useful observations regarding both bundle types.

Observation 3.22. *Let $\mathbf{X}$ be the allocation returned by Algorithm 3.3, along with the historical bundles and choices $(H_i^r)_{(i,r) \in [n] \times [R]}$ and $(h_i^r)_{(i,r) \in [n] \times [R]}$. Then for every $i, j \in [n]$ and $r \in [R]$, it holds that:*

- (i) *If there exists no envy-cycle in and after some round r , then $X_i^r = H_i^r$ and $x_i^r = h_i^r$.*
- (ii) $X_i^{R+1} = X_i = H_i^{R+1}$
- (iii) $X_i^1 = \emptyset = H_i^1$
- (iv) $X_i^r + x_i^r = X_i^{r+1}$
- (v) $v_i(H_i^r) + v_i(h_i^r) \leq v_i(H_i^{r+1})$
- (vi) $v_i(X_j^{r+1}) - v_i(H_i^{r+1}) \leq v_i(X_j^r) - v_i(H_i^r) + v_i(x_j^r) - v_i(h_i^r)$.

Note that Observation iv does not hold for historical bundles as $H_i^r + h_i^r$ is not H_i^{r+1} if there is an envy-cycle elimination involving agent i . Observation v is in a sense the replacement we use for Observation iv on historical bundles, but it only work for the valuation of the owner of the bundles. So $v_j(H_i^r + h_i^r) \leq v_j(H_i^{r+1})$ in general does not holds, which a more formal expression of the reason we gave for using X_j^r in the definition of the gap round in Definition 3.20.

Proof. Observations i to iv follow directly from Notation 3.3 and Definition 3.19. To see why Observation v holds, consider whether an envy-cycle elimination occurs at the end of round r that involves agent i . If not, then $H_i^r + h_i^r = H_i^{r+1}$. If such an elimination does occur, then because every agent involved in an envy-cycle strictly improves their allocation, it holds that $v_i(H_i^r + h_i^r) < v_i(H_i^{r+1})$. The claim then follows from the additivity of the valuations. Finally, Observation vi follows directly by combining the two preceding claims. $\square$

Observation vi constitutes the primary argument used in most inductive proofs throughout the remainder of this thesis, beginning here with the proof of Lemma 3.21.

Proof of Lemma 3.21. We proceed by induction on r using the following induction hypothesis:

$$v(X_i^{r+1}) - v(H_i^{r+1}) \leq v(X_j^{r_g+1}) - v(H_i^{r_g+1}) + \sum_{r'=r_g+1}^r (v(x_j^{r'}) - v(h_i^{r'})) \quad (3.13)$$

for all $r \geq r_g$. To prove Lemma 3.21, we mainly show that Equation (3.13) holds. Furthermore, we need to prove that $v_i(x_j^{r'}) \leq v_i(h_i^{r'})$ for all $r' > r_g$. This follows from the use of envy-reducing order as long as i is envious of the agent which has $X_j^{r'}$ at the start of the round r' . This is given by the definition of r_g .

We now show that Equation (3.13) holds. For the base case $r = r_g$ it holds that:

$$v(X_j^{r_g+1}) - v(H_i^{r_g+1}) \leq v(X_j^{r_g+1}) - v(H_i^{r_g+1}) + 0.$$

For the inductive step, let $r > r_g$. Applying the induction hypothesis Equation (3.13) alongside Observation vi yields:

$$\begin{aligned} v(X_j^{r+1}) - v(H_i^{r+1}) &\leq v(X_j^r) - v(H_i^r) + v(x_j^r) - v(h_i^r) \\ &\leq v(X_j^{r_g+1}) - v(H_i^{r_g+1}) + \sum_{r'=r_g+1}^{r-1} (v(x_j^{r'}) - v(h_i^{r'})) + v(x_j^r) - v(h_i^r) \\ &= v(X_j^{r_g+1}) - v(H_i^{r_g+1}) + \sum_{r'=r_g+1}^r (v(x_j^{r'}) - v(h_i^{r'})). \end{aligned}$$

□

To find the analogue to Observation 3.10, we further extend Lemma 3.21 using Observation vi along with the fact that $v(X_j^{r_g}) - v(H_i^{r_g}) \leq 0$ by definition of r_g .

Corollary 3.23. *Let $\mathbf{X}$ be an allocation returned by Algorithm 3.3, with historical bundles $(H_i^r)_{(i,r) \in [n] \times [R]}$ and historical choices $(h_i^r)_{(i,r) \in [n] \times [R]}$. For any agent i who envies an agent j , and with r_g defined as in Definition 3.20, it holds that*

$$v(X_j) - v(X_i) \leq v(X_j^{r_g}) - v(H_i^{r_g}) + v(x_j^{r_g}) - v(h_i^{r_g}) + \sum_{r=r_g+1}^R (v(x_j^r) - v(h_i^r)) \leq v(x_j^{r_g}) - v(h_i^{r_g}).$$

Note that the two resources $x_j^{r_g}$ and $h_i^{r_g}$ are chosen in the same round, with $h_i^{r_g}$ selected greedily under v . Thus, we have obtained an analogous bound to Observation 3.10. The key differences are that $X_j^{r_g}$ and $x_j^{r_g}$ do not necessarily correspond to the bundle and choice of agent j in round r_g , and that we utilize an inequality ($\leq$) instead of an equality ($=$) to bound the envy due to potential envy-cycle eliminations. In the absence of envy-cycles, Lemma 3.21 reduces exactly to Observation 3.10. As with Observation 3.10, this result applies directly to permutation instances.

Corollary 3.24. *Let $\mathbf{X}$ be an allocation returned by Algorithm 3.3 given a permutation instance. For any agent i who envies an agent j , and with r_g defined as in Definition 3.20, it holds that:*

$$v(X_j) - v(X_i) \leq |G| + 1 - (R - r_g).$$

3.3.2 Bounding Envy and Envy Potential

For permutation instances, the envy between agents i and j mainly depends upon the gap. The gap is created by other agents choosing a continuous sequence of resources from $[m]$ ordered decreasingly under v_i , and some agent which choosing the last resource $x_j^{r_g}$ from before this sequence. We have also seen that using induction combined with Observation vi was a simple way to show Lemma 3.21. In this section, we aim to bound the envy for permutation instances and show that Claim i holds in a similar manner. To do this using induction, we use an invariance which bounds both envy and the maximum size of the gap in every round. The following definition is what we use to bound the latter.

Definition 3.25 (Envy Potential). *Let $\mathbf{X}$ be an allocation created by Algorithm 3.3 with the history function η . Let i be an agent which envies j . Then we define the envy potential at the beginning of a round r as:*

$$|P_E^r| = |\{x \in [m] \mid \eta(x) < r \wedge v_i(x) < \tau\}|$$

where $\tau = \max_{y \in [m], \eta(y) \geq r} v_i(y)$ is the value of the unallocated resource that i values highest.

One way to think about P_E^r is to consider $[m]$ as a sequence ordered descending by v_i . Going from left to right, the resource with value τ is the leftmost unallocated resource, partitioning the sequence into those left and right of it. All allocated resources to the right of it are part of the set that forms the potential. Essentially, $v^{-1}(\tau)$ is a candidate for $x_j^{r_g}$. So the gap is never larger than the amount of resources to the right of it, plus the resources chosen in round r_g by the agents $[n] - i - k$, where k is the agent for which $x_j^{r_g} = h_k^{r_g}$. Formally, this gives the bound $G \leq |P_E^{r_g}| + (n - 2)$.

Envy and envy potential influence each other. In any round $r < r_g$, it might also happen that $v(x_j^r) - 1 > v(h_i^r)$. As we had it with the gap, by the greediness of i , this can only be because of some resource $x \in [m]$, chosen before h_i^r and valued such that $v(x_j^r) > v(x) > v(h_i^r)$. Note that either $x \in P_E^r$ or $\eta(x) = r$. In any case, $x \notin P_E^{r+1}$ since for the τ of P_E^{r+1} , it holds that $v(h_i^r) > \tau$ and thus $v(x) > \tau$. We refer to the potential being *actualized*, as the envy only increases because the potential increases by less than possible in that case or even decreases. Note also that potential does not always need to become envy. In fact, if for some round r there is an $x \in P_E^r$ where $v(h_i^r) > v(x) > v(x_j^r)$, this negatively contributes to the envy. However, we do not really discuss these cases, as our goal is to find an upper bound for the envy potential and envy. We have seen that both values influence each other. Thus, we bound the sum of both. To shorten this common term, we often use the following notation.

Notation 3.26. $EP^r = v(X_j^r) - v(H_i^r) + |P_E^r|$.

Note that neither EP^r nor actually P_E^r reference i or j . This is because for the rest of the thesis we only consider those concepts for the two agents and have therefore left out these indices.

To bound the value of this expression, we first consider how it changes from round to round. Essentially, the goal is to find the smallest possible $c \in \mathbb{N}_0$ for which it holds that

$$EP^{r+1} \leq EP^r + c. \quad (3.14)$$

On a round-by-round basis, this depends on whether i or the agent currently owning X_j^r chooses the larger-valued resource. This occurs for two reasons: First, it directly influences the envy, but more importantly, it influences the potential. For instance, if $v(x_j^r) > v(h_i^r)$, then x_j^r is not part of P_E^{r+1} .

Lemma 3.27. *For any allocation $\mathbf{X}$ along with the historical choices $(h_i^r)_{(i,r) \in [n] \times [R]}$ computed by Algorithm 3.3 for a permutation instance, it holds for any $r \in [R]$ that*

1. if $v(x_j^r) < v(h_i^r)$, then $c \leq n - 3$,
2. if $v(x_j^r) > v(h_i^r)$, then $c \leq n - 1$

for the c given in Equation (3.14).

Observe that if i envies X_j^r at the start of round r , then by the envy-reducing order, only $v(h_i^r) > v(x_j^r)$ and thus the first case is possible. If i does not envy X_j^r , both cases are possible.

Proof. We begin this proof by laying out some groundwork before differentiating the two cases. First, we use Observation vi of Observation 3.22 to bound the change in envy by the resources chosen in round r :

$$EP^{r+1} = v(X_j^{r+1}) - v(H_i^{r+1}) + |P_E^{r+1}| \quad (3.15)$$

$$\leq v(X_j^r) + v(x_j^r) - v(H_i^r) - v(h_i^r) + |P_E^r| + \delta_P \quad (3.16)$$

$$= EP^r + v(x_j^r) - v(h_i^r) + \delta_P \quad (3.17)$$

where $\delta_p = |P_E^{r+1} \setminus P_E^r| - |P_E^r \setminus P_E^{r+1}|$ is the size difference of the potential. Thus, we find the smallest bound for c for which Equation (3.12) holds by finding the maximal value achieved by $v(x_j^r) - v(h_i^r) + \delta_p$. We have already discussed below Lemma 3.16 that resources chosen by i are not part of the gap. By the same argument, they are also not part of the potential. Since at most n resources are chosen in a round, $\delta_p \leq (n-1) - 0 = n-1$.

Let us now consider the first case, where $v(x_j^r) < v(h_i^r)$. To argue about the change in potential, it is helpful to refer to the agent that is in possession of the bundle X_j^r at the start of round r . So let k be the agent for which $X_j^r = H_k^r$. Here, there are two cases: either the resource that k chooses in round r is not part of the potential P_E^{r+1} , meaning the potential increases by at most $n-2$, or it is part of the potential, and we may have that the potential increases by $n-1$. However, for this to be the case, $v(x_j^r)$ is small enough, increasing the difference between the values of the two resources. Formally, if $h_k^r \notin P_E^{r+1}$, then $|P_E^{r+1} \setminus P_E^r| \leq n-2$ holds, and thus $v(x_j^r) - v(h_i^r) + \delta_p \leq -1 + (n-2) = n-3$. If $h_k^r \in P_E^{r+1}$, then $v(h_k^r) < \max_{y \in [m], v(y) > r} v(y) = \tau < v(h_i^r)$, and thus $v(x_j^r) = v(h_k^r) \leq v(h_i^r) - 2$. Hence, it does not matter that the change in bundle size might be the maximal $\delta_p \leq n-1$, as together

$$v(x_j^r) - v(h_i^r) + \delta_p \leq -2 + (n-1) = (n-2).$$

Let us now consider the other case, where $v(x_j^r) > v(h_i^r)$. Recall that i picks greedily. Our goal is to use this to argue that if the difference $v(x_j^r) - v(h_i^r)$ is greater than one, this is only because the potential is actualized, and this missing potential is just the same as the increase in envy. We start by defining this extra envy $a = v(x_j^r) - v(h_i^r) - 1 \geq 0$ and the resources of the potential that are actualized $A = \{x \in [m] \mid v(h_i^r) < v(x) < v(x_j^r)\}$. For permutation valuations, it holds that $|A| = a$. We now analyze the relationship of both values to the envy and potential. We start with the observation on the potential that $A \cap P_E^{r+1} = \emptyset$. This holds since all resources in A are valued higher than $v(h_i^r) > \tau$. We then split A into the resources $A_{=r} = \{x \in A \mid \eta(x) = r\}$ chosen in round r , which will be those we do not add to the potential, and $A_{<r} = \{x \in A \mid \eta(x) < r\}$, which will be those we remove from the potential.

Formally, we show $|P_E^{r+1} \setminus P_E^r| \leq n-2 - |A_{=r}|$. At most n resources are chosen each round, which may potentially be added to the potential. However, as seen before, h_i^r is never part of P_E^{r+1} , as all resources in the potential are valued lower than the highest-valued unallocated resource under v . Since i chooses h_i^r greedily, this is never the case. It is also not the case for x_j^r , as it is valued higher than h_i^r . For all resources in $A_{=r}$, we have seen that they also do not belong to the potential by the same argument. Furthermore, it holds that $|P_E^r \setminus P_E^{r+1}| \geq |A_{<r}|$, as all resources in $A_{<r}$ are part of P_E^r , since they are valued lower than $v(x_j^r)$ while not being part of P_E^{r+1} . We now put everything together to find

$$v(x_j^r) - v(h_i^r) + \delta_p \leq a + 1 + n - 2 - |A_{<r}| - |A_{=r}| = n - 1 + (a - |A|) = n - 1.$$

□

3.3.2.1 Bounding Envy and Envy Potential over Multiple Rounds

This section has two main results. First, with Corollary 3.29, we show that Claim i of Section 3.2.1 holds. Second, using Corollary 3.29, we provide a practical bound for the envy in allocations produced for permutation instances by Algorithm 3.3 via Theorem 3.30. The following lemma is the main work of this section, enabling both of the aforementioned results.

Lemma 3.28. *Let $\mathbf{X}$ be an allocation returned from Algorithm 3.3 given a permutation instance. Then for every $r \in [R] + 0$, it holds that*

$$EP^{r+1} \leq (n-2)r + 1$$

where EP^{r+1} refers to the sum of envy and envy potential between any two agents.

Let i be an agent envious of j in such an allocation. Using the definition $EP^{r+1} = v(X_j^{r+1}) - v(H_i^{r+1}) + |P_E^{r+1}|$ along with $|P_E^{r+1}| \geq 0$, it follows that $(n-2)r+1$ is an upper bound for the envy.

Corollary 3.29. *Consider an execution of Algorithm 3.3 on a permutation instance that returns the allocation $\mathbf{X}$ along with the historical bundles $(H_i^r)_{(i,r) \in [n] \times [R]}$. Let i be an agent envying j in $\mathbf{X}$. It holds for all $r \in [R] + 0$ that*

$$v(X_j^{r+1}) - v(H_i^{r+1}) \leq (n-2)r + 1.$$

With this, we can bound the total envy for permutation instances.

Theorem 3.30. *Consider an execution of Algorithm 3.3 on a permutation instance that returns the allocation $\mathbf{X}$. Let i be an agent envying j in $\mathbf{X}$. It holds that*

$$v(X_j) - v(X_i) \leq (n-1)r_g + 1 - R$$

for r_g as defined in Definition 3.20.

Proof. To see that this holds, combine Corollary 3.29 with Lemma 3.21 to form

$$\begin{aligned} v(X_j) - v(X_i) &\leq v(X_j^{r_g+1}) - v(H_i^{r_g+1}) - \sum_{r=r_g+1}^R (v(x_j^r) - v(h_i^r)) \\ &\leq (n-2)r_g + 1 - (R - r_g) = (n-1)r_g + 1 - R. \end{aligned}$$

□

We spend the rest of this section proving Lemma 3.28. The term $r(n-2)$ hints that this can be done by combining the two cases of Lemma 3.27. An argument which further motivates this is, that every time we use the higher bound $c \leq n-1$, the envy from agent i to the agent currently holding the restricted bundle of X_j increases, since $v(x_j^r) > v(h_i^r)$. Once this envy is positive, by the envy-reducing order, i chooses first, and thus we are in the case where $c \leq n-3$. So in the worst case, we might assume that we experience the both cases of Lemma 3.27 each in roughly $\frac{R}{2}$ rounds, which explains this bound.

This is essentially what we are going to do, though first we start out with only a proof attempt for Lemma 3.28. We then analyze where and why it fails and adapt the approach for the real proof.

Lemma 3.27 strongly favors induction proofs. So let us try to show the main claim

$$EP^{r+1} \leq (n-2)r + 1 \tag{3.18}$$

directly for all $r \in [R] + 0$ by induction over r . For this attempt, we only show steps that lead to the issues in the proof. Thus, we skip the induction start. For the induction step, we utilize Equation (3.14) to show

$$EP^{r+1} \leq EP^r + c \leq (n-2)(r-1) + 1 + c.$$

The difficult case is if $v(x_j^r) > v(h_i^r)$. Here, we only have the guarantee that $c \leq n-1$. For $c = n-1$, this only suffices to show

$$EP^{r+1} \leq (n-2)r + 2.$$

Looking back at the intuitive idea of this proof, this may make sense. In this proof, there should be some step where because of the envy, the case $v(x_j^r) > v(h_i^r)$ does not happen. Thus, the most probable first assumption is: if $EP^{r+1} = (n-2)r + 1$, then there is envy, and thus we do not have to consider $c = n-1$ for this case. Hence, we add this to the induction hypothesis, resulting in the two cases:

- (i) $EP^{r+1} \leq (n-2)r$ or
- (ii) $EP^{r+1} = (n-2)r + 1$, and at the start of round $r+1$, i envies X_j^{r+1} .

Using Induction Hypothesis ii, it is possible to show Induction Hypothesis i. However, we need to show Induction Hypothesis ii as well. Here, the challenging case is once again $v(x_j^r) > v(h_i^r)$ with $c = n-1$. Consider what happens when $EP^r = (n-2)(r-1)$ and $v(x_j^r) = v(h_i^r) + 1$. Then, the envy increases by only 1, while the potential may increase by $n-2$. Since now $EP^{r+1} = (n-2)r + 1$, we need to show that i envies X_j^{r+1} , but as the envy increased only by one. For this to be the case, it is required that $v(X_j^r) \geq v(H_i^r)$. It is probably clear by now that it might not be the most productive approach to add a third induction hypothesis claiming that if

$$EP^{r+1} = (n-2)r, \text{ then } v(X_j^r) \geq v(H_i^r).$$

If we proceed with $c = n-1$, we will run into the same issue once again. To get ahead of this infinite regress in the argumentation, we instead abstract these induction hypotheses into:

- (i) $EP^{r+1} \leq (n-2)r + 1$ and
- (ii) if $EP^{r+1} = (n-2)r - a$ for $a \in \mathbb{N}_0 + (-1)$, then $v(X_j^{r+1}) - v(H_i^{r+1}) \geq -a$.

However, using Notation 3.26, Induction Hypothesis ii simplifies to

$$|P_E^{r+1}| \leq (n-2)r,$$

which is simply false. Consider for instance $r = 1$: if all agents of $[m] - i$ contribute to the potential in this round, we have $|P_E^2| = n-1$. Though as we have seen in the proof of Lemma 3.27, for $v(h_i^r) > v(x_j^r)$ and $x_j^1 \in P_E^2$ to be the case, it is required that $v(x_j^r) \leq v(h_i^r) - 2$, somewhat diminishing the envy by one more than is required from $v(h_i^r) > v(x_j^r)$. If x_j^1 should persist in the potential for multiple rounds, this diminishes the envy even more. Thus, our initial approach was to consider these negative contributions to envy as „waste“ and thread them into the induction hypothesis somehow. Generally, this makes sense. If there is some other negative contribution to envy, then the two cases of Lemma 3.27 do not need to appear in a 50/50-split, as more rounds where $v(x_j^r) > v(h_i^r)$ may exist to balance out the „wasted envy“. We were even able to express EP^{r+1} by the waste and the number of rounds spent in either of the two cases of Lemma 3.27. However, tying this back around to show that if $EP^{r+1} = (n-2)r + 1$ holds, then i envies X_j^{r+1} , was not successful. Partially because there might be envy-cycle elimination in any round, reducing the envy. We found a better way through the perspective that the potential can only be higher than $(n-2)r$ if the envy is negative. This forms the idea behind the induction hypothesis we end up using to prove Lemma 3.28.

Proof of Lemma 3.28. We show this lemma via induction over $r \in [R] + 0$ of the following hypothesis:

- (i) $EP^{r+1} \leq (n-2)r + 1$
- (ii) if $|P_E^{r+1}| > (n-2)r$ then $v(X_j^{r+1}) - v(H_i^{r+1}) \leq (n-2)r - |P_E^{r+1}|$.

First, we show that Induction Hypothesis ii implies the first hypothesis that we added in the failed approach, that being that if $EP^{r+1} \leq (n-2)r+1$, then agent i envies X_j^{r+1} . The first step towards this is to reframe it as follows:

$$\text{if } |P_E^{r+1}| > (n-2)r \text{ then } EP^{r+1} = |P_E^{r+1}| + v(X_j^{r+1}) - v(H_i^{r+1}) \leq (n-2)r. \quad (3.19)$$

By contraposition, if $EP^{r+1} = (n-2)r+1$, then $|P_E^{r+1}| \leq (n-2)r$ and thus, by Notation 3.26, $v(X_j^{r+1}) - v(H_i^{r+1}) \geq 1$, i.e., i envies the agent in possession of bundle X_i^{r+1} .

We now perform the induction, starting with $r = 0$. Both induction hypotheses hold here since

$$EP^1 = 0 \leq (n-2)0+1, \text{ and } P_E^1 \leq 0 = (n-2)r.$$

For $r > 0$, we show both hypotheses hold for both cases of Lemma 3.27. We start with Induction Hypothesis i and the easier case, $v(h_i^r) > v(x_j^r)$. By Lemma 3.27 along with Induction Hypothesis i, it holds that

$$EP^{r+1} \leq EP^r + c \leq (n-2)(r-1) + 1 + (n-3) < (n-2)r+1.$$

For the second case, $v(x_j^r) > v(h_i^r)$, if $EP^r \leq (n-2)(r-1)$, it holds by Lemma 3.27 that

$$EP^{r+1} \leq EP^r + c \leq (n-2)(r-1) + (n-2) = (n-2)r+1.$$

To show that $EP^r \leq (n-2)(r-1)$ holds, assume it does not, i.e., $EP^r = (n-2)(r-1) + 1$. Before starting the induction proof, we showed using Induction Hypothesis ii that in this case i envies X_j^r , thus, by the envy-reducing order, i chooses before the agent owning X_j^r chooses. This is a contradiction to $v(x_j^r) > v(h_i^r)$. In conclusion, $EP^r \leq (n-2)(r-1)$ holds along with Induction Hypothesis i for this and thus all cases, provided we can show that Induction Hypothesis ii does so as well.

To show Induction Hypothesis ii, assume that $|P_E^{r+1}| > (n-2)r$. We need to show that $v(X_j^{r+1}) - v(H_i^{r+1}) \leq (n-2)r - |P_E^{r+1}|$. We approach this as before by splitting the difference of the bundles using Observation vi of Observation 3.22, enabling us to use the induction hypothesis. To do so, however, we first introduce the helper variable

$$a = EP^r - (n-2)(r-1) = v(X_j^r) - v(H_i^r) + |P_E^r| - (n-2)(r-1). \quad (3.20)$$

With this, we can now reformulate the envy as

$$v(X_j^{r+1}) - v(H_i^{r+1}) \leq v(X_j^r) - v(H_i^r) + v(x_j^r) - v(h_i^r) \quad (3.21)$$

$$= (n-2)(r-1) - |P_E^r| + a + v(x_j^r) - v(h_i^r). \quad (3.22)$$

We now differentiate between the two cases of Lemma 3.27, beginning with $v(h_i^r) > v(x_j^r)$. We extend Equation (3.22) further. To do this, we utilize the facts that

- $|P_E^{r+1}| - |P_E^r| \leq (n-1)$, or equivalently $-|P_E^r| \leq (n-2) + 1 - |P_E^{r+1}|$, and
- $v(x_j^r) - v(h_i^r) \leq -1$

to find

$$(n-2)(r-1) - |P_E^r| + a + v(x_j^r) - v(h_i^r) \leq (n-2)(r) - |P_E^{r+1}| + 1 + a - 1.$$

Thus, $v(X_j^{r+1}) - v(H_i^{r+1}) \leq (n-2)r - |P_E^{r+1}| + a$, which suffices to show Induction Hypothesis ii provided $a \leq 0$.

To complete this case, we need to consider $a > 0$. We start by showing that this is only possible for very specific values of $|P_E^r|$ and $|P_E^{r+1}|$. First, observe that since $|P_E^r| > (n-2)(r-1)$, then by Equation (3.19) that $EP^r \leq (n-2)(r-1)$. Thus,

$$a = EP^r - (n-2)(r-1) \leq 0. \quad (3.23)$$

What we are actually concerned with is the contraposition: $a > 0$ implies that $|P_E^r| \leq (n-2)(r-1)$. Recall that we assumed $|P_E^{r+1}| \geq (n-2)r + 1$. Thus, the equation

$$(n-2)r + 1 \leq |P_E^{r+1}| = |P_E^r| + \delta_p \leq (n-2)r + (n-1)$$

holds only if $|P_E^r| = (n-2)(r-1)$, $|P_E^{r+1}| = (n-2)r + 1$, and $x_j^r \in P_E^{r+1}$. This last property follows from the fact that the resources chosen by every agent in $[n] - i$ in round r contribute to the potential set, as otherwise at most $(n-2)$ resources are added.

We now have very strict values for the envy potential, which we will use to finish the proof for this case. However, from the fact that $x_j^r \in P_E^{r+1}$, we derive one last necessary equation. As we did in the same case—and subcase—of the proof of Lemma 3.27, we find that $v(x_j^r) < \tau < v(h_i^r)$, and thus

$$v(x_j^r) - v(h_i^r) \leq -2. \quad (3.24)$$

We can now string together a chain of equations, showing Induction Hypothesis ii. Below, we mention the arguments used in the relevant steps. It holds that

$$v(X_j^{r+1}) - v(H_i^{r+1}) \leq v(X_j^r) - v(H_i^r) + v(x_j^r) - v(h_i^r) \quad (3.25)$$

$$\leq EP^r - |P_E^r| - 2 \quad (3.26)$$

$$\leq (n-2)(r-1) + 1 - (n-2)(r-1) - 2 = -1 \quad (3.27)$$

$$= (n-2)r - (n-2)r - 1 \quad (3.28)$$

$$= (n-2)r - |P_E^{r+1}| \quad (3.29)$$

where for

- Equation (3.26), we used both Notation 3.26 and Equation (3.24),
- Equation (3.27), we used Induction Hypothesis i and the fact that $|P_E^r| = (n-2)(r-1)$, and
- Equation (3.29), we used the fact that $|P_E^{r+1}| = (n-2)r + 1$.

With this, the case $v(h_i^r) > v(x_j^r)$ is complete, and we turn to $v(x_j^r) > v(h_i^r)$. To show Induction Hypothesis ii for this step, we extend Equation (3.22). To do this, we bound the value of a and express the change in envy, similar to our approach in the proof of Lemma 3.27 for this case. We begin with the change in potential, which is expressed by the equation

$$|P_E^{r+1}| \leq |P_E^r| + (n-2) - (v(x_j^r) - v(h_i^r) - 1). \quad (3.30)$$

This holds because h_i^r is chosen greedily, and the only reason for it to be valued lower than $v(x_j^r) - 1$ is that envy is actualized. We used essentially the same argument to show the exact same case of Lemma 3.27. This also enables us to bound a , as the potential increases by at most $(n-2)$. Recall our assumption that $|P_E^{r+1}| > (n-2)r$, which implies that $|P_E^r| > (n-2)(r-1)$. Using Equation (3.23), we have already seen that $a \leq 0$ holds. However, we are going to require that $a < 0$ to cancel out the -1 of Equation (3.30).

We achieve this by using the fact that $v(X_j^r) - v(H_i^r) \leq 0$, as x_j^r is chosen before h_i^r . Substitution into Equation (3.20) yields

$$a = v(X_j^r) - v(H_i^r) + |P_E^r| - (n-2)(r-1) < 0 + (n-2)(r-1) - (n-2)(r-1).$$

With $a \leq -1$, together with Equation (3.30), we can complete Equation (3.22) to

$$\begin{aligned} & (n-2)(r-1) - |P_E^r| - a + v(x_j^r) - v(h_i^r) \\ & \leq (n-2)(r-1) - |P_E^{r+1}| + (n-2) - (v(x_j^r) - v(h_i^r) - 1) - a + v(x_j^r) - v(h_i^r) \\ & = (n-2)r - |P_E^{r+1}| + 1 - a \leq (n-2)r - |P_E^{r+1}|. \end{aligned}$$

Thus, we have shown that all induction hypotheses hold for both cases. $\square$

Using Induction Hypothesis i and Induction Hypothesis ii, we can make the following observation:

Observation 3.31. *If at the start of a round r of Algorithm 3.3 it holds for two agents i, j that*

$$EP^{r+1} = (n-2)r + 1,$$

then agent i envies agent j .

3.3.3 Attempting to Bound t -Tolerance

This section would have been the main part where we show that Algorithm 3.3, given a permutation instance, returns a t -tolerant EFX allocation for $t \geq \lceil \frac{R}{2} \rceil + 1$ and even $t \geq \lceil \frac{R}{2} \rceil$ for the right instances. Instead, we only show parts of it and motivate why it does not always hold.

To this end, let i once again be an agent envying another agent j . Instead of bounding the total amount of resources in X_j that do not eliminate envy, we bound κ , which we define as the amount of resources in $X_j \setminus X_j^{r_g+1}$ that do not eliminate envy. Furthermore, let us define the set of resources not eliminating envy chosen, before the round r_g

$$\mathcal{R} = \{r \in [r_g - 1] \mid v(x_j^r) < v(X_j) - v(X_i)\} \subseteq [r_g - 1],$$

which we will not be able to sufficiently upperbound. There are different bounds for κ depending on $|\mathcal{R}|$. In Section 3.3.3.1, we consider the case where $|\mathcal{R}| = 0$. In Section 3.3.3.2, we consider the case where $|\mathcal{R}| > 0$. Both cases share some common groundwork, which we provide now.

We start out with the simplest bound for κ , which is

$$\kappa \leq R - r_g. \tag{3.31}$$

This bound does not sufficiently restrict κ for small r_g . Thus, we seek to find a second bound, preferably one that bounds κ tighter for small r_g . So what happens when r_g is small? The gap round is earlier, thus the gap and in turn the envy is small. Recall Theorem 3.30 to formally see why. For smaller envy, fewer resources in X_j are valued small enough to not eliminate it when removed. This motivates that the second constraint for κ is based on the following equation, which holds for all x_j^r with $r > r_g$ that do not eliminate envy when removed:

$$v(x_j^r) < v(X_j) - v(X_i).$$

We are going to further bound this equation from both sides. On the right, if $|\mathcal{R}| = 0$, we use Theorem 3.30, while for $|\mathcal{R}| > 0$, we first need a bound for envy in that case. On the left side, we are going to construct a lower bound for $v(x_j^r)$, resulting in a chain of equations which is required to hold for any round $r \in [R]$ where x_j^r does not eliminate envy. We can then use this constraint along with the fact that $R \geq r > r_g$ and $r \in \mathbb{N}$ to upper bound the amount of rounds for which this set of constraints may hold, which forms the second upper bound for κ .

The proofs we present in both Section 3.3.3.1 and Section 3.3.3.2 use this new bound for κ . In both cases, we show that there is a constant r' such that if $r_g \geq r'$, Equation (3.31) becomes tight enough to sufficiently bound κ , and for the case where $r_g \leq r'$, the new bound bounds κ sufficiently tightly. Though we do not do this in any of the proofs, the value of r' can be found by considering for what r_g both the new equation and the simple Equation (3.31) become tight.

As mentioned, as part of the proof, we lower bound $v(x_j^r)$. Of course, initially $v(x_j^r)$ might be any value in $[m]$. However, the value $v(h_i^r)$ might not, as agent i chooses it greedily. We even already showed a lower bound for $v(h_i^r)$ with Lemma 3.17 for Algorithm 3.2. For Algorithm 3.3, the proof does not change, so we use it again in this section. Now consider that if $v(x_j^r)$ is some low value while $v(h_i^r)$ is valued higher as it is lower bounded, the total envy is lower than Theorem 3.30. This is because we worked with the upper bound $v(x_j^r) - v(h_i^r) \leq -1$ for the difference in the value of the resources. Imagine that after the execution of Algorithm 3.3, we alter the instance we used as an input, replacing the value of $v(x_j^r)$ with any other value. This never changes whether x_j^r eliminates the envy when removed or not.

Lemma 3.32. *Let $\mathbf{X}$ be an allocation produced from an additive instance using Algorithm 3.3. Let i be an agent envying j . Then, for every resource $x_j \in X_j$, it holds for any additive valuation u with*

$$v_i(x) = u(x) \text{ if } x \neq x_j$$

that x_j eliminates envy under u if and only if it does under v_i .

Proof. As always, we set $v = v_i$. Observe that

$$\begin{aligned} v(x_j) &< v(X_j) - v(X_i) \\ \Leftrightarrow 0 &< v(X_j - x_j) - v(X_i) \\ \Leftrightarrow 0 &< u(X_j - x_j) - v(X_i) \\ \Leftrightarrow u(x_j) &< u(X_j) - u(X_i). \end{aligned}$$

□

We leverage this property to show that, to bound κ , it suffices to consider a simpler valuation.

Definition 3.33 (Post-gap normalized function). *Given an allocation $\mathbf{X}$ where i envies j with the gap round r_g , and a history function η , the post-gap normalized valuation $u : [m] \rightarrow [m]$ is defined by*

$$u(x) = \begin{cases} v_i(h_i^{\eta(x)}) - 1 & \text{if } \eta(x) > r_g \wedge x \in X_j \\ v_i(x) & \text{otherwise} \end{cases}.$$

Note that a post-gap normalized valuation is not necessarily a permutation valuation. Hence, we must be careful not to accidentally use any argumentation that only works for permutation valuations. Note, however, that the most important parts of the permutation valuation are kept intact. Every value for every resource from before and inside of the gap round is kept the same. Furthermore, having the difference $v(h_i^r) - v(x_i^r) = 1$ is something that may happen with permutation valuations naturally as well. With Table 3.1, we saw an example of this in Section 3.2.1.

Also note that when we use a post-gap normalized valuation, this potentially changes more than just one value, as was the case in Lemma 3.32. Using such a valuation does not decrease the envy, allowing us to show that at least one direction of Lemma 3.32 remains, which suffices to bound κ .

Lemma 3.34. *Let $\mathbf{X}$ be an allocation computed by Algorithm 3.3 where i envies j , and r_g is the gap round. For any $r > r_g$, if $x_j^r \in [m]$ and $v(x_j^r) < v(X_j) - v(X_i)$, then $u(x_j^r) < u(X_j) - u(X_i)$.*

Proof. First, we show that u values every resource in X_j greater than or equal to v . For all resources chosen before or during round r_g , the values remain unchanged. For a round $r > r_g$, by the envy-reducing order, i chooses greedily before the agent that holds X_j^r . Thus, it holds that:

$$v(x_j^r) \leq v(h_i^r) - 1 = u(x_j^r). \quad (3.32)$$

Now assume that $v(x_j^r) < v(X_j) - v(X_i)$. We use an argument similar to Lemma 3.32. By Equation (3.32) combined with the fact that $v(X_i) = u(X_i)$, it holds that

$$0 < v(X_j - x_j^r) - v(X_i) \leq u(X_j - x_j^r) - u(X_i).$$

Applying the additivity of the valuation, this result yields $u(x_j^r) < u(X_j) - u(X_i)$. $\square$

By Equation (3.32), the envy from i might be larger under u than under v_i . Since we need to bound the envy to bound κ , this needs to be taken into consideration. However, the increase in envy under a post-gap normalized valuation still remains below the upper bound we found for permutation valuations.

Lemma 3.35. *For any allocation $\mathbf{X}$ computed by Algorithm 3.3 on a permutation instance, for an agent i envying an agent j with gap round r_g , it holds for the post-gap normalized valuation u for v_i that*

$$u(X_j) - u(X_i) \leq EP^{r_g+1} - (R - r_g),$$

where EP^{r_g+1} is the envy and envy potential under v_i .

Proof. To see that this holds, consider the resources that have a different value under u . We explain the other relevant steps below:

$$u(X_j) - u(X_i) = 0 + v(X_j) - v(X_i) + \sum_{r \geq r_g+1}^R (u(x_j^r) - v(x_j^r)) \quad (3.33)$$

$$\leq |P_E^{r_g+1}| + v(X_j^{r_g+1}) - v(H_i^{r_g+1}) + \sum_{r \geq r_g+1}^R (v(x_j^r) - v(h_i^r)) + \sum_{r \geq r_g+1}^R (u(x_j^r) - v(x_j^r)) \quad (3.34)$$

$$\leq EP^{r+1} + \sum_{r \geq r_g+1}^R (v(x_j^r) - v(h_i^r)) + \sum_{r \geq r_g+1}^R (u(x_j^r) - v(x_j^r)) \quad (3.35)$$

$$\leq EP^{r+1} + \sum_{r \geq r_g+1}^R (u(x_j^r) - v(h_i^r)) = EP^{r+1} - (R - r_g), \quad (3.36)$$

where

- Equation (3.34) follows from $0 \leq |P_E^{r_g+1}|$ and Lemma 3.21,
- Equation (3.35) follows from Notation 3.26, and
- Equation (3.36) follows from Definition 3.33.

□

With this, we have everything required to show the second bound for κ with a dependence on EP^{r_g+1} , as this changes depending on the two different cases described at the start of the section.

Lemma 3.36. *Let $\mathbf{X}$ be an allocation computed by Algorithm 3.3 for a permutation instance. For an agent i envying agent j with $EP^{r_g+1} \leq (n-1)r_g - e$ for some $e \in \mathbb{N}_0$, it holds that*

$$\kappa < R - \frac{m + R + e - n + 1}{n} + r_g.$$

Note that $e = r_g - 1$ is the case for $|\mathcal{R}| = 0$, while for $|\mathcal{R}| > 0$, we determine e in Section 3.3.3.2. Also recall that Equation (3.31) becomes less tight for smaller r_g , while the bound given by Lemma 3.36 does get tighter for smaller r_g . Thus, we can use both together to bound κ more effectively.

Proof. To bound κ , we are going to consider some round $r_g + o$ for some offset $o \in \mathbb{N}$ where $x_j^{r_g+o}$ does not eliminate envy. We know that $r_g + o$ is a natural number and that $r_g + o \leq R$. We now find some lower bound $l < r_g + o$, as then we can bound

$$\kappa \leq R - l + 1.$$

To find this bound l , we use a chain of equations. We explain why the individual steps hold below.

$$m - n(r_g + o) + 1 \leq v(h_i^{r_g+o}) - 1 \quad (3.37)$$

$$= u(x_j^{r_g+o}) \quad (3.38)$$

$$< u(X_j) - u(X_i) \quad (3.39)$$

$$\leq EP^{r_g+1} - (R - r_g) \quad (3.40)$$

$$\leq nr_g - R - e \quad (3.41)$$

where

- Equation (3.37) follows from Lemma 3.17. It is important to note that Lemma 3.17 is, strictly speaking, only shown for Algorithm 3.2 used with envy-reducing orders. However, the argument that in round $r_g + o$, by the time i gets to choose, at most $n(r_g + o) - 2$ highest valued resources may be allocated holds even if we have envy-cycle eliminations in between rounds.
- Equation (3.38) follows from Definition 3.33.
- Equation (3.39) follows from the fact that $x_j^{r_g+o}$ does not eliminate envy under v , and by Lemma 3.34 it does not under u .
- Equation (3.40) holds by Lemma 3.35.

By rearranging this, we find a lower bound for $r_g + o$:

$$\begin{aligned} m - n(r_g + o) + 1 &< nr_g - R - e \\ \Leftrightarrow m + R - nr_g + e + 1 &< n(r_g + o) \\ \Leftrightarrow r_g + o &> \frac{m + R + e + 1}{n} - r_g = l. \end{aligned}$$

With this, we can now create the bound for κ as described at the start of the proof:

$$\kappa \leq R - \frac{m + R + e + 1}{n} + r_g + 1 = R - \frac{m + R + e - n + 1}{n} + r_g.$$

□

3.3.3.1 Case: $|\mathcal{R}| = 0$

This is the case where the fact that Claim ii does not hold has no impact.

Lemma 3.37. *Let $\mathbf{X}$ be an allocation computed by Algorithm 3.3 given a permutation instance. For any agent i envying an agent j , it holds that $\kappa < \frac{R}{2} - \frac{\frac{3}{2}R}{2n-1} + 1$.*

For many instances, $\frac{\frac{3}{2}R}{2n-1} \leq 0.5$ holds. For those and for even R , this gives a bound of $\kappa \leq \lceil \frac{R}{2} \rceil$. In general, however, this only suffices for the following bound:

Corollary 3.38. *Let $\mathbf{X}$ be an allocation computed by Algorithm 3.3 given a permutation instance. For any agent i envying an agent j , it holds that*

$$\kappa \leq \lceil \frac{R}{2} \rceil + 1.$$

Proof. Recall that $EP^{r_g+1} \leq (n-2)r_g + 1$ by Lemma 3.28. Thus, we may use Lemma 3.36 with $e = r_g - 1$ to get

$$\kappa \leq R - \frac{m + R - n + r_g}{n} + r_g = R - \frac{m + R - n}{n} + \frac{n-1}{n}r_g.$$

Using this equation, consider which bound we achieve for $r_g < \frac{m+R-n}{2n-1}$:

$$\begin{aligned} \kappa &\leq R - \frac{m + R - n}{n} + \frac{n-1}{n} \frac{m + R - n}{2n-1} \\ &= R + \left(-1 + \frac{n-1}{2n-1}\right) \frac{m + R - n}{n} \\ &= R + \frac{n-1-2n+1}{2n-1} \frac{m + R - n}{n} \\ &= R + \frac{-n}{2n-1} \frac{m + R - n}{n} \\ &= R - \frac{m + R - n}{2n-1}. \end{aligned}$$

We get the exact same bound if $r_g \geq \frac{m+R-n}{2n-1}$ using Equation (3.31):

$$\kappa < R - \frac{m + R - n}{2n - 1}.$$

Thus, we can use this bound for any $r_g \in [R]$. We finalize this proof using two more inequalities. First, observe that $m \geq (R - 1)n + 1 = Rn - n + 1$ as $R = \lceil \frac{m}{n} \rceil$. Another important, more complex step in the final chain of equations is the following:

$$\frac{Rn}{2n - 1} = \frac{R}{2} \frac{n - 2^{-1} + 2^{-1}}{n - 2^{-1}} = \frac{R}{2} \left(1 + \frac{2^{-1}}{n - 2^{-1}}\right) = \frac{R}{2} + \frac{2^{-1}R}{2n - 1}.$$

Putting all of this together, we find

$$\begin{aligned} \kappa &\leq R - \frac{m + R - n}{2n - 1} \\ &\leq \frac{2Rn - R}{2n - 1} - \frac{Rn - n + 1 + R - n}{2n - 1} \\ &= \frac{2Rn - R - Rn - R + 2n - 1}{2n - 1} \\ &= \frac{Rn - 2R + 2n - 1}{2n - 1} \\ &= \frac{R}{2} - \frac{\frac{3}{2}R}{2n - 1} + 1. \end{aligned}$$

□

3.3.3.2 Case: $|\mathcal{R}| > 0$

In this section, we consider the case $|\mathcal{R}| > 0$. This is the case that breaks Claim ii and leads to the counterexample. When upper-bounding the resources that do not eliminate envy, we have to bound $|\mathcal{R}| + \kappa$. Hence Lemma 3.37 does not suffice, as it only bounds κ without dependence on $|\mathcal{R}|$. In this section, we integrate $|\mathcal{R}|$ into this bound, but find that this still does not suffice. To do this, we again use Lemma 3.36 but this time with $e \gg r_g - 1$. First, we consider a few candidates for e and how it is even possible that a resource from $X_j^{r_g}$ does not eliminate envy, especially when considering the observations we had with Lemma 3.32.

One initial observation is that by Corollary 3.23, it is necessary that $v(h_i^r) > v(x_j^{r_g}) > v(x_j^r)$. Thus, if there is one or even multiple resources like x_j^r , it is required that $v(h_i^r) - v(x_j^r)$ is a larger difference.

Notation 3.39. $d^r = v(h_i^r) - v(x_j^r)$ for all $r \in \mathcal{R}$.

Note that in the minimal case $d^r = 2$, but for x_j^r to not eliminate envy, this is usually a higher value. Lemma 3.32 motivated that x_j^r being assigned a low value might not suffice for it to not eliminate envy, as this is associated with a decrease in envy. The decrease in envy also exists in this case and is formalized by the following lemma.

Lemma 3.40. *Given any allocation $\mathbf{X}$ computed by Algorithm 3.3 for a permutation instance. For any agent i envying an agent j with the gap-round r_g :*

$$EP^{r_g+1} \leq (n - 1)r_g - \sum_{r \in \mathcal{R}} d^r. \quad (3.42)$$

Proof. Observe that for any round $r \in [r_g] \setminus \mathcal{R}$, it holds that $EP^{r+1} \leq EP^r + (n-1)$ by Lemma 3.27. For any round $r \in \mathcal{R}$, at most $n-1$ new resources are added to the potential. Meanwhile, the envy decreases by at least $v(h_i^r) - v(x_j^r) = d^r$. If there is no envy-cycle elimination involving i at the end of round r , then the envy decreases by exactly d^r . Otherwise, as $v(H_i^{r+1}) \geq v(H_i^r + h_i^r)$, the envy decreases strictly more than d^r . Thus, for such a round, it holds that $EP^{r+1} \leq EP^r + (n-1) - d^r$.

In sum, since $EP^1 = 0$, this results in

$$EP^{r_g+1} = r_g(n-1) - \sum_{r \in \mathcal{R}} d^r.$$

□

Note that this bound is constructed less carefully than the one in Lemma 3.28. For example, we do not consider the existence of rounds where $c \leq (n-3)$, with the c of Lemma 3.27 for rounds that do not appear in $\mathcal{R}$. However, the analysis of this case is complex enough as it is. Hence, we sacrifice some tightness for more clarity in the proofs.

Lemma 3.44 further motivates that only using low values for $v(x_j^r)$ may not suffice. If using low values for $v(x_j^r)$ does not suffice for x_j^r to not eliminate envy when removed, then the next logical step is to reduce the values of $v(h_i^r)$ as well. Though since h_i^r is chosen greedily under v , the only way to do this is to have other agents choose more resources valued highly under v . Or, in other words, by actualizing potential. To argue formally about this, we introduce the next variable:

Notation 3.41. Let $a^r = m - 2(r-1) - v(h_i^r)$ for all $r \in \mathcal{R}$.

The value a^r is, in a sense, a lower bound for the amount of potential actualized before $v(h_i^r)$ is chosen. To see this, consider that $v(h_i^r) = m - 2(r-1) - a^r$. Since i chooses greedily, exactly $2(r-1) + a^r$ highly valued resources are chosen before i chooses h_i^r . Of these, $r-1$ are chosen by i itself. Another $r-1$ may be chosen by j , as is the case in Table 3.1, leaving at least a^r to be chosen by other agents. Though practically it is possible that the potential is not converted into envy, increasing the envy early is an important factor for why we can bypass Lemma 3.32. By the definition of r_g , it holds that $v(X_j^{r_g}) \leq v(H_i^{r_g})$. In other words, all envy actualized via a^r has been counterbalanced by either i being chosen first for multiple rounds and decreasing envy by at least 1 each round, or by having one or multiple higher $d^{r'}$ for some rounds $r < r' < r_g$. This motivates how the definitions of d^r and a^r interact. Both individually limit EP^{r_g+1} while they essentially decrease $v(x_j^r)$ in sum. Formally, we can simply combine Notation 3.39 and Notation 3.41 to make the following observation.

Observation 3.42. For every $r \in \mathcal{R}$, it holds that $v(x_j^r) = v(h_i^r) - d^r = m - 2(r-1) - a^r - d^r$.

To see how it bounds envy and envy potential, we have to be a bit more careful than with d^r .

Lemma 3.43. Given any allocation $\mathbf{X}$ computed by Algorithm 3.3 for a permutation instance. For any agent i envying an agent j with the gap-round r_g , it holds that

$$EP^{r_g+1} \leq (n-1)r_g - r_{\max} - a^{r_{\max}}, \quad (3.43)$$

where $r_{\max} = \max \mathcal{R}$.

Proof. To start, we bound the envy and potential after round r_g using Lemma 3.27 and the fact that $v(X_j^{r_g}) \leq v(H_i^{r_g})$ by the definition of r_g :

$$EP^{r_g+1} \leq EP^{r_g} + (n-1) \quad (3.44)$$

$$= v(X_j^{r_g}) - v(H_i^{r_g}) + |P_E^{r_g}| + (n-1) \leq |P_E^{r_g}| + (n-1). \quad (3.45)$$

By finding an upper bound for $|P_E^{r_g}|$, we may bound the envy and potential. We do this by first bounding $|P_E^{r_{\max}+1}|$ and then using

$$|P_E^{r_g}| \leq |P_E^{r_{\max}+1}| + (n-1)(r_g - r_{\max} - 1), \quad (3.46)$$

which holds since in every round at most $n-1$ resources may be added to the potential, as seen, for example, in the proof of Lemma 3.27.

To bound $|P_E^{r_{\max}+1}|$, let us first consider the set of resources chosen by the end of round $r_{\max}$:

$$M^{\leq r_{\max}} = \{x \in [m] \mid \eta(x) \leq r_{\max}\}.$$

The value of $v(h_i^{r_{\max}})$ partitions this bundle into

$$M_+^{\leq r_{\max}} = \{x \in M^{\leq r_{\max}} \mid v(x) \geq v(h_i^{r_{\max}})\} \text{ and } M_-^{\leq r_{\max}} = \{x \in M^{\leq r_{\max}} \mid v(x) < v(h_i^{r_{\max}})\}.$$

Observe that $P_E^{r_{\max}+1} \subseteq M^{\leq r_{\max}}$, since for every $x \in P_E^{r_{\max}+1}$ it is true that $\eta(x) < r_{\max} + 1$. Meanwhile, it holds that $M_+^{\leq r_{\max}} \cap P_E^{r_{\max}+1} = \emptyset$ because for the τ used to define $P_E^{r_{\max}+1}$, it holds that $\tau < v(h_i^{r_{\max}})$ since $h_i^{r_{\max}}$ was chosen greedily. Thus, for every $x \in M_+^{\leq r_{\max}}$, it holds that $\tau < v(h_i^{r_{\max}}) \leq v(x)$ and thus $x \notin P_E^{r_{\max}+1}$. Putting the two observations together, we conclude that $P_E^{r_{\max}+1} \subseteq M_-^{\leq r_{\max}}$. Furthermore, $v^{-1}(\tau) \notin P_E^{r_{\max}+1}$ by definition, while $v^{-1}(\tau) \in M_-^{\leq r_{\max}}$. Thus, $P_E^{r_{\max}+1} \subsetneq M_-^{\leq r_{\max}}$. We are going to use this to bound the size of the potential via the size of $M_-^{\leq r_{\max}}$. We bound the size of $M_-^{\leq r_{\max}}$ by taking the size $|M^{\leq r_{\max}}| = nr_{\max}$ and subtracting the size of $M_+^{\leq r_{\max}}$. Here, observe that there are exactly $2(r_{\max} - 1) + a^{r_{\max}}$ resources valued higher than $h_i^{r_{\max}}$, all of which are allocated before the end of round $r_{\max}$ by greediness. Thus, since $h_i^{r_{\max}} \in M_+^{\leq r_{\max}}$ as well, it holds that $|M^{\leq r_{\max}}| = nr_{\max}$ and $|M_+^{\leq r_{\max}}| \geq 2(r_{\max} - 1) + a^{r_{\max}} + 1$. We use this to show that

$$\begin{aligned} |P_E^{r_{\max}+1}| &\leq |M_-^{\leq r_{\max}}| - 1 \\ &= |M^{\leq r_{\max}}| - |M_+^{\leq r_{\max}}| - 1 \\ &\leq nr_{\max} - 2(r_{\max} - 1) - a^{r_{\max}} - 2 = (n-2)r_{\max} - a^{r_{\max}}. \end{aligned}$$

By combining this with Equation (3.45) and Equation (3.46), we find that

$$\begin{aligned} EP^{r_g+1} &\leq |P_E^{r_g}| + (n-1) \\ &\leq (n-2)r_{\max} - a^{r_{\max}} + (n-1)(r_g - r_{\max} - 1) + (n-1) \\ &= (n-1)r_g - r_{\max} - a^{r_{\max}}. \end{aligned}$$

□

When we now place Lemma 3.40 and Lemma 3.43 against Observation 3.42, we have the formal background behind how we can overcome the ideas described in Lemma 3.32. The values $\sum_{r \in \mathcal{R}} d^r$ and $a^{r_{\max}}$ decrease EP^{r_g+1} individually while decreasing $v(x_j^r)$ in sum.

With the goal of bounding κ , we again want to utilize Lemma 3.36 with a stricter e . We may use either $\sum_{r \in \mathcal{R}} d^r$ or $a^{r_{\max}}$ for this, but we then also have to bound these values as well. There may be multiple approaches to doing this, but here we bound d^r individually using Lemma 3.43 by utilizing a similar approach to the one used in the proof of Lemma 3.36.

Lemma 3.44. *For every $r \in \mathcal{R}$, it holds that*

$$d^r > m - nr_g - 2(r - 1) - (a^r - a^{r_{\max}}) + r_{\max} + R > m - nr_g - r + R + 2.$$

Proof. For every fixed $r \in \mathcal{R}$, it holds that

$$v(x_j^r) < v(X_j) - v(X_i).$$

Observe that the value of $v(x_j^r)$ is given by Observation 3.42, while $v(X_j) - v(X_i)$ is upper bounded by Lemma 3.21 and Lemma 3.43. Thus, it holds that

$$m - 2(r - 1) - a^r - d^r = v(x_j^r) \tag{3.47}$$

$$< v(X_j) - v(X_i) \tag{3.48}$$

$$\leq (n - 1)r_g - r_{\max} - a^{r_{\max}} - (R - r_g) \tag{3.49}$$

$$= nr_g - r_{\max} - a^{r_{\max}} - R. \tag{3.50}$$

We rearrange this to find

$$\begin{aligned} m - 2(r - 1) - a^r - d^r &< nr_g - r_{\max} - a^{r_{\max}} - R \\ \Leftrightarrow d^r &> m - nr_g - 2(r - 1) - (a^r - a^{r_{\max}}) + r_{\max} + R. \end{aligned} \tag{3.51}$$

We further simplify this using $a^r < a^{r_{\max}}$, or equivalently $0 > -a^r + a^{r_{\max}}$, along with $r < r_{\max}$, to find $d^r > m - nr_g - r + 2 + R$. $\square$

Note that simplifying $0 > -a^r + a^{r_{\max}}$ may reduce the sharpness of the bound. But if there is only one round before r_g where potential is actualized, then $a^r = a^{r_{\max}}$, as will be the case in the counterexamples we construct in Section 3.3.4 and Section A. A further property to be observed is that this bound is especially weak for large r_g . For these cases it is possible to construct a tighter bound by not bounding d^r with the envy potential EP^{r_g+1} , but utilizing the simple fact that $v(h_i^r) - d^r = v(x_j^r) < v(x_j^{r_g}) < v(h_i^r) - (r_g - r)$, where the last relation holds because agent i chooses greedily at least $r_g - r - 1$ more times, bounding the maximum value of $v(x_j^{r_g})$. This results in the bound $d^r > r_g - r$, which may be better than the bound of Lemma 3.44 in extreme cases.

However, this makes finding an optimal e for Lemma 3.36 a lot harder and does not change the fact that we cannot bound $|\mathcal{R}|$ sufficiently to get an improved tolerance result.

Lemma 3.45. *Let $\mathbf{X}$ be an allocation computed by Algorithm 3.3 given a permutation instance. For any agent i envying an agent j , it holds that*

$$\kappa < \frac{R}{|\mathcal{R}| + 2} + 1 - \frac{1}{n} - \frac{R + |\mathcal{R}|2}{n(|\mathcal{R}| + 2)}.$$

The proof is essentially the same as for Lemma 3.37 with a more complex e .

Proof. We find the second bound for κ using Lemma 3.36 with $e = \sum_{r \in \mathcal{R}} d^r$ because of Lemma 3.40. We then further expand $d^r > m - nr_g - r + R + 2$ using Lemma 3.44:

$$\kappa < R - \frac{m + R + e - n + 1}{n} + r_g \tag{3.52}$$

$$\leq R - \frac{m + R + \sum_{r \in \mathcal{R}} d^r - n + 1}{n} + r_g \tag{3.53}$$

$$\leq R - \frac{m + R + \sum_{r \in \mathcal{R}} (m - nr_g - r + R + 2) - n + 1}{n} + r_g \tag{3.54}$$

$$\leq R - \frac{(|\mathcal{R}| + 1)(m + R) - \sum_{r \in \mathcal{R}} r + |\mathcal{R}|2 - n + 1}{n} + \frac{|\mathcal{R}|nr_g}{n} + r_g \tag{3.55}$$

$$\leq R - \frac{\sigma}{n} + (|\mathcal{R}| + 1)r_g, \tag{3.56}$$

where $\sigma = (|\mathcal{R}| + 1)(m + R) - \sum_{r \in \mathcal{R}} r + |\mathcal{R}|2 - n + 1$. Because of $m \geq Rn - n + 1$, we can also further bound this by

$$\sigma \geq Rn(|\mathcal{R}| + 1) + (|\mathcal{R}| + 1)(-n + 1 + R) - \sum_{r \in \mathcal{R}} r + |\mathcal{R}|2 - n + 1 \quad (3.57)$$

$$= Rn(|\mathcal{R}| + 1) + (|\mathcal{R}| + 2)(1 - n) + (|\mathcal{R}| + 1)R - \sum_{r \in \mathcal{R}} r + |\mathcal{R}|2. \quad (3.58)$$

We now show that for every $r_g \in [R]$, this gives our bound. First, consider Equation (3.56) for $r_g \leq \frac{\sigma}{n(|\mathcal{R}|+2)}$:

$$\kappa < R - \frac{\sigma}{n} + (|\mathcal{R}| + 1) \frac{\sigma}{n(|\mathcal{R}| + 2)} \quad (3.59)$$

$$= R + \left(\frac{|\mathcal{R}| + 1 \text{Gentile}}{|\mathcal{R}| + 2} - 1 \right) \frac{\sigma}{n} \quad (3.60)$$

$$= R - \frac{1}{|\mathcal{R}| + 2} \frac{\sigma}{n}, \quad (3.61)$$

which gives the same bound as Equation (3.31) for $r_g < \frac{\sigma}{n(|\mathcal{R}|+2)}$. So in any case, we may bound κ for any $r_g \in [R]$ using this equation, and by applying Equation (3.58), we conclude this proof.

$$\kappa \leq R - \frac{\sigma}{n(|\mathcal{R}| + 2)} \quad (3.62)$$

$$\leq \frac{Rn(|\mathcal{R}| + 2)}{n(|\mathcal{R}| + 2)} - \frac{Rn(|\mathcal{R}| + 1) + (|\mathcal{R}| + 2)(1 - n) + (|\mathcal{R}| + 1)R - \sum_{r \in \mathcal{R}} r + |\mathcal{R}|2}{n(|\mathcal{R}| + 2)} \quad (3.63)$$

$$= \frac{Rn}{n(|\mathcal{R}| + 2)} + 1 - \frac{1}{n} - \frac{(|\mathcal{R}| + 1)R - \sum_{r \in \mathcal{R}} r + |\mathcal{R}|2}{n(|\mathcal{R}| + 2)} \quad (3.64)$$

$$\leq \frac{R}{|\mathcal{R}| + 2} + 1 - \frac{1}{n} - \frac{R + |\mathcal{R}|2}{n(|\mathcal{R}| + 2)}, \quad (3.65)$$

where we used $r < R$ in the final step. $\square$

Let us now consider why this bound for κ does not suffice. We initially wanted to bound the amount of resources which do not eliminate envy with

$$\kappa + |\mathcal{R}| \leq \left\lceil \frac{R}{2} \right\rceil + 1.$$

Now, taking a look at Lemma 3.45, the term $-\frac{1}{n} - \frac{R+|\mathcal{R}|2}{n(|\mathcal{R}|+2)}$ may help somewhat in bounding κ , but for larger n , it becomes arbitrarily small. So, at the very least, we would need to bound $|\mathcal{R}| + \frac{R}{|\mathcal{R}|+2} \leq \frac{R}{2}$. However:

$$\begin{aligned} \frac{R}{|\mathcal{R}| + 2} + |\mathcal{R}| &> \frac{R}{2} \\ \Leftrightarrow |\mathcal{R}| &> \frac{R|\mathcal{R}| + 2R}{2|\mathcal{R}| + 4} - \frac{2R}{2|\mathcal{R}| + 4} \\ \Leftrightarrow 2|\mathcal{R}| + 4 &> R \\ \Leftrightarrow |\mathcal{R}| &> \frac{R}{2} - 2. \end{aligned}$$

So this does not hold if the amount of resources in $X_j^{r_g}$ which do not eliminate envy is almost as large as $\frac{R}{2}$. Or, in other words: the resources chosen after r_g are not the driving factor when it comes to instances requiring large t -tolerance. Of course, this still leaves open whether the way we bound d^r , κ , and EP^{r_g+1} is simply too inefficient or if it is truly possible to construct such an instance. In the next section, we do just that.

3.3.4 Constructing an Example for $t \geq \lceil \frac{m}{n} \rceil - 2$

In this section, we describe how to construct an instance, along with some constraints on the envy-decreasing orders, so that the allocation $\mathbf{X}$ produced by Algorithm 3.3 given this instance and using the constraints, is only t -tolerant EFX for $t \geq \lceil \frac{m}{n} \rceil - 2$. We explain in detail how and why this example is constructed this way, but we do not offer a complete proof considering all aspects of Algorithm 3.3, which would need to include, for example, formally verifying that fitting envy-reducing orders exist.

Furthermore, we construct this example by specifying which agent is assigned which resource in $\mathbf{X}$, as we have done before for Table 3.1. This way, we only specify the valuation $v = v_i$ of an agent i experiencing envy towards an agent j . There are many possible choices for the valuations of the other agents, for example, choices where they only highly value the resources that they actually receive in $\mathbf{X}$ in the order in which they receive them. Here, we would formally need to show that the valuations exist, produce the allocation $\mathbf{X}$, and also for convenience's sake do not create any envy-cycles during the executions, as we only use the round-restricted bundles in this section. Under the assumption that these valuations exist, such an instance can be constructed. To understand the ideas behind this, it may be helpful to follow along with the example of this that we give in Section A.

We use the observations we made in Section 3.3.3.2. We start with all the resources that do not eliminate envy being allocated before the gap round r_g . Secondly, we have seen that we have to actualize envy before the gap. Since the resource $x_j^{r_g}$ eliminates envy, by the definition of t , this envy actualization occurs in one round, for which we here use the first round. Hence, $v(x_j^1) = m$ and $v(x_i^1) = m - n + 1$, while all agents in $[n] - i - j$ choose the resources $v^{-1}(m - 1), \dots, v^{-1}(m - n + 2)$ in the first round. Hence, there is an envy of $n - 1$ after the first round.

We then balance this actualized envy with the d^r for $r \in \mathcal{R}$ as defined in Section 3.3.3.2, where $\mathcal{R}$ is the set of the next t rounds, so $\mathcal{R} = [t + 1] - 1$. For these, we simply let agent i choose the highest-valued resource uncontested, so

$$v(x_i^r) = m - n - (r - 2),$$

for all $r \in \mathcal{R}$. The resources chosen by j in such a round r are all valued lower than $v(x_i^r)$ by some value $d^r = d$, which for simplicity is constant for all these rounds. Let us now consider which bound this d has to fulfill.

First of all, we want it to balance with the actualized value, i.e., we want

$$v(X_j^R) - v(X_i^R) = 0.$$

to fulfill the definition of $r_g = R$ without carrying over any envy. For this, as $v(x_j^r) - v(x_i^r) = -d$, we have the constraint

$$v(x_j^1) - v(x_i^1) = n - 1 = td. \tag{3.66}$$

For a given t , we may easily satisfy this by choosing a sufficiently large n .

The second constraint d needs to satisfy is that it is large enough so that $v(x_j^r)$ does not eliminate envy. Formally, for every $r \in \mathcal{R}$, it holds that

$$v(x_j^r) = v(x_i^r) - d < v(X_j) - v(X_i) = 0 + v(x_j^R) - v(x_i^R). \tag{3.67}$$

To achieve this, we want to maximize the right side. For x_i^R , this is done by setting its value to one. This is achieved if all other resources were already taken and i chooses last in the round. Since i does not envy j at the start of the round, and we can distribute

the other resources in a manner such that i does not envy other agents as well, such an envy-reducing order exists.

For $v(x_j^R)$, observe that the highest-valued resource we have assigned so far is $v(x_i^{R-1}) = v(x_i^{t+1}) = m - n - t + 1$. Thus, we set $v(x_j^R) = m - n - t$. This results in envy of $v(x_j^R) - v(x_i^R) = m - n - t - 1$. When we turn back to d , we now have to achieve that $v(x_j^r) = v(x_i^r) - d = m - n - (r - 2) - d < m - n - t - 1$ for all $r \in \mathcal{R}$. Since $r \geq 2$, this holds for all $d > t + 1$ as

$$m - n - (r - 2) - d \leq m - n - d < m - n - t - 1.$$

Thus, for the example construction, we set $d = t + 2$.

Using this, we can determine the range of values in which the set $X_j^R - x_j^1$ is valued under v_i . Using the definition of x_i^r and d , this results in $v^{-1}(m - n - t - 2), \dots, v^{-1}(m - n - 2t - 1)$. With this, the only thing left to determine is the resources chosen by the agents in $[m] - i - j$ in round $[R] - 1$. These being the only resources left, they are specifically resource $v^{-1}(m - n - t - 1)$ and all the resources in the range $v^{-1}(m - n - 2t - 2), \dots, v^{-1}(2)$.

One more thought may be given to the envy-reducing orders. Consider an instance where all agents value the resources we have assigned them above in order of the rounds. It suffices to set $l \prec_1 i, l \prec_R i$ for all $l \in [n] - i$. The rest may be chosen arbitrarily.

This concludes the construction. Section A shows an example of this for $t = 4$. We specifically choose $t = 4$ because for this example we then have to use 25 agents and 250 resources allocated over 6 rounds. We choose these values to show that this example actually surpasses our initial assumed bound for the amount of resources that do not eliminate envy. Here we have to be a bit careful as we mainly assumed that only of $t \geq \lceil \frac{R}{2} \rceil + 1$ is required, which is given for this example. We strongly expect that it is possible to construct an example for a higher t which does that but as the example for $t = 4$ is already quite large, we do not do this. However below Lemma 3.37, we describe that for instances with an even amount of rounds, that the Lemma provides a bound of $\lceil \frac{R}{2} \rceil = 3$ for these instances, so for this subcase Section A is a valid counterexample.

3.4 Conclusion and Further Questions

In this chapter, we have discussed the use of round-robin for computing EF1 allocations and attempted to extend some guarantees to t -tolerant EFX allocations by allowing different orders of agents in different rounds. This approach has some merit. Especially with Corollary 3.23, we have found that the envy-reducing round-robin offers a better envy bound than the regular round-robin. We have seen, however, that extending this to better guarantees for t -tolerant EFX allocation did not succeed, not even for permutation instances. Specifically, there are instances where Algorithm 3.3 produces allocations that are t -tolerant only for $t \geq \lceil \frac{m}{n} \rceil - 2$, which is the exact same bound as for Algorithm 3.1 on Example 3.7. It is important to note here that we have not shown that round-robin returns $(\lceil \frac{m}{n} \rceil - 2)$ -EFX instance but for envy-decreasing round-robin at least this seems very likely due to the existence of round r_g and then need to acutalize potential in some round before r_g . Due to time constraints we were not able to further investigate this. However, we have demonstrated how permutation instances can be a useful restriction that allows us to make some powerful observations, such as Theorem 3.30, even if in the end this did not lead to an improved t -tolerance bound.

We also want to highlight that the example construction in Section 3.3.4 is somewhat surprising. As stated before, we knew that Claim ii did not hold, but initial counterexamples used $n = 4$ and $m = 40$. This resulted in us being able to place only one resource before the

gap that did not eliminate envy, resulting in a significant decrease in envy. This, in turn, motivated us to assume that having more resources before the gap that do not eliminate envy leads to further reductions. Only through the observations made in Section 3.3.3.2 did we find that, to break the assumed tolerance bound, we need to have most resources that do not eliminate envy occur before the gap, which then enables the gap-round to be in the last round, increasing the envy.

Note also that one of the most impactful improvements to the envy-reducing round-robin may be further constraints on the orders. Every counterexample hinges on a bad order being chosen. Sometimes bad cases still exist if both agent i and agent j share the exact same valuations; however, if they do not, there usually exists a better envy-reducing order. In Section 3.2, we considered whether it is possible to circumvent the existence of envy cycles by using different envy-reducing orders, but we have found this to be a challenging direction. Therefore, similar issues may exist for further extensions to the envy-reducing orders.

Perhaps the most interesting result of this chapter is the envy-reducing round-robin algorithm itself. Given its better envy bound, it may be worthwhile to study whether this offers better c -EFX bounds on some instances, or how it fares when it comes to PO and NW. One high-level observation it allows us to make is that Constraint i from the start of Chapter 3 is the more severe restriction on the round-robin algorithm. Even when we tried getting around larger gaps in valuation by using permutation valuations, new gaps appear in the worst cases we consider, as the other agents remove resources from the sequence of resource order decreasing under v . Hence, for an improvement, it is certainly necessary to allow for different size bundles. One approach to do this is to run Algorithm 3.3 once. If there is envy from an agent i to an agent j at the end that is not eliminated by a resource in X_j , we consider the resources $x \in X_j$ such that every other resource in $X_j - x$ eliminates envy from i to $X_j - x$. We then look at the last round $\eta(x)$ for which such an x exists, rerun Algorithm 3.3 up to round $\eta(x)$, and let the agent l that has the set $X_j^{\eta(x)}$ in round $\eta(x)$ not choose a resource in this round. This should effectively reduce the envy towards X_j . At the same time, agent l and finally agent j might become more envious of other agents, but there are $\eta(x)$ rounds left where this envy may put them earlier in the envy-reducing orders, reducing the impact of this operation. If we continue to reapply this operation on the instance created by second run of Algorithm 3.3 and so on, each time adding a round where some agent is not allowed to choose a resource, this may result in vastly different allocation. For one, we do not know if this approach terminates, but if it does, the sets may have vastly different sizes and potentially better fairness guarantees. However, this is purely speculation.

Another interesting approach may be to evaluate envy-reducing round-robin statistically or to look at probabilistic results such as EFR. The types of instances and executions that were problematic for the permutation valuations required that many agents pick from a longer consecutive sequence of resources without any gaps, which is at least statistically unlikely.

4. Conclusion

In this thesis, we have set out to investigate the space between EF1 and EFX. To this end, we have considered two algorithms computing EF1 allocations on additive instances and tried extending them. Both have their own chapter with their own conclusion and further research. Here, we simply summarize the main results again and interpret what they mean for the main question.

In Chapter 2, we have seen that using t -tolerant g -champions and Pareto improvable cycles, it is possible to compute a t -tolerant EFX allocation for $t \geq \lceil \frac{m-2}{n} \rceil$ for additive fair division instances. We have also seen that there are instances and partial allocations where $t \geq \lceil \frac{m-2}{n} \rceil$ is required so that a t -tolerant Pareto improvable cycle exists.

In Chapter 3, we have shown that round-robin yields EF1 allocations that by their bundle size are at least $(\lceil \frac{m}{n} \rceil - 1)$ -tolerant EFX allocations. We have attempted to extend this to show the existence of a $(\lceil \frac{1}{2} \lceil \frac{m}{n} \rceil \rceil + 1)$ -tolerant EFX allocation for permutation instances, but instead only provide an example construction where a t -tolerance of $t \geq \lceil \frac{m}{n} \rceil - 2$ is required. This leaves open whether for permutation instances it is always possible to find a $(\lceil \frac{m}{n} \rceil - 2)$ -tolerant EFX allocation, possibly even by using the envy-reducing round-robin. Assuming this is possible, it may also be possible to extend this proof to show EFX2 always exists for these instances, which would be a slight improvement upon EF1. And even then, this holds only for permutation instances, as this does not hold for general additive instances because of Constraint i from the start of Chapter 3.

So in summary, we have not found an approach to find an allocation strictly fairer than EF1 by utilizing some of the most common EF1 algorithms. For some restricted instances, this is certainly possible. For additive instances in general, this may also require significantly different approaches than those that worked for EF1. This becomes especially interesting considering the recent development where EFX allocations have been shown to not exist in certain more general settings [AMM⁺26],[MS26]. Hence, if the same limitations exist for additive instances as well, then it may be worth considering for what k these start to appear for EFX k .

Bibliography

- [AAB⁺23] Georgios Amanatidis, Haris Aziz, Georgios Birmpas, Aris Filos-Ratsikas, Bo Li, Hervé Moulin, Alexandros A. Voudouris, and Xiaowei Wu. Fair division of indivisible goods: Recent progress and open questions. *Artif. Intell.*, 322(C):103965, 2023.
- [AAC⁺22] Hannaneh Akrami, Noga Alon, Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, and Ruta Mehta. EFX Allocations: Simplifications and Improvements. *arXiv e-prints*, page arXiv:2205.07638, 2022.
- [ABFR⁺21] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A. Voudouris. Maximum Nash welfare and other stories about EFX. *Theoretical Computer Science*, 863:69–85, 2021.
- [ABFRV22] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. Fair Division of Indivisible Goods: A Survey. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5385–5393. International Joint Conferences on Artificial Intelligence Organization, 2022.
- [ACIW21] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair allocation of indivisible goods and chores. *Autonomous Agents and Multi-Agent Systems*, 36(1), 2021.
- [AGS25] Arash Ashuri, Vasilis Gkatzelis, and Alkmini Sgouritsa. EF2X Exists for Four Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(13):13555–13563, 2025.
- [AI82] Kenneth J. Arrow and Michael D. Intriligator. *Handbook of mathematical economics*, volume 2. North-Holland, Amsterdam, 1982.
- [AMM⁺26] Hannaneh Akrami, Alexander Mayorov, Kurt Mehlhorn, Shreyas Srinivas, and Christoph Weidenbach. A Counterexample to EFX $n \geq 3$ Agents, $m \geq n + 5$ Items, Submodular Valuations via SAT-Solving. *arXiv e-prints*, page arXiv:2604.18216, 2026.
- [AMN20] Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. Multiple birds with one stone: Beating $1/2$ for EFX and GMMS via envy cycle elimination. *Theoretical Computer Science*, 841:94–109, 2020.
- [ARS22] Hannaneh Akrami, Rojin Rezvan, and Masoud Seddighin. An EF2X Allocation Protocol for Restricted Additive Valuations. In Luc de Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 17–23. IJCAI, 2022.
- [BCFF21] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. (Almost Full) EFX Exists for Four Agents (and Beyond). *arXiv e-prints*, page arXiv:2102.10654, 2021.

- [BCFF22] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. Almost Full EFX Exists for Four Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):4826–4833, 2022.
- [Ben89] Jeremy Bentham. *An Introduction to the Principles of Morals and Legislation* (1789). T. Payne and son, London, 1789.
- [Bud11] Eric Budish. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- [CFKS23] George Christodoulou, Amos Fiat, Elias Koutsoupias, and Alkmini Sgouritsa. Fair allocation in graphs. In *Proceedings of the 24th ACM Conference on Economics and Computation*, EC ’23, page 473–488. Association for Computing Machinery, 2023.
- [CGH19] Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-Freeness Up to Any Item with High Nash Welfare: The Virtue of Donating Items. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC ’19, page 527–545. Association for Computing Machinery, 2019.
- [CGM24] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX Exists for Three Agents. *J. ACM*, 71(1), 2024.
- [CKM⁺19] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Econ. Comput.*, 7(3), 2019.
- [FHL⁺21] Alireza Farhadi, MohammadTaghi Hajiaghayi, Mohamad Latifian, Masoud Seddighin, and Hadi Yami. Almost envy-freeness, envy-rank, and nash social welfare matchings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5355–5362, 2021.
- [HGNV25] Vishwa Prakash Hv, Pratik Ghosal, Prajakta Nimbhorkar, and Nithin Varma. EFX Exists for Three Types of Agents. In *Proceedings of the 26th ACM Conference on Economics and Computation*, EC ’25, page 101–128, New York, NY, USA, 2025. Association for Computing Machinery.
- [KSV20] Maria Kyropoulou, Warut Suksompong, and Alexandros A. Voudouris. Almost envy-freeness in group resource allocation. *Theoretical Computer Science*, 841:110–123, 2020.
- [LMMS04] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC ’04, page 125–131, New York, NY, USA, 2004.
- [LR24] Claudia Lindner and Jörg Rothe. *Cake-Cutting: Fair Division of Divisible Goods*, pages 507–603. Springer Nature Switzerland, Cham, 2024.
- [Luo97] Ari Luotonen. *Web Proxy Servers*. Prentice Hall PTR, Upper Saddle River, Nj, 1997.
- [Mah23] Ryoga Mahara. Existence of EFX for two additive valuations. *Discrete Applied Mathematics*, 340:115–122, 2023.
- [Mah24] Ryoga Mahara. Extension of Additive Valuations to General Valuations on the Existence of EFX. *Mathematics of Operations Research*, 49(2):1263–1277, 2024.

- [MS26] Simon Mackenzie and Mashbat Suzuki. Counterexamples to EFX for Submodular and Subadditive Valuations. *arXiv e-prints*, page arXiv:2605.06451, 2026.
- [PR20] Benjamin Plaut and Tim Roughgarden. Almost Envy-Freeness with General Valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020.
- [Sha87] Alan C Shaw. *The logical design of operating systems*. Prentice Hall, Eaglewood Cliffs, NJ, 2 edition, 1987.
- [Var74] Hal R Varian. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63–91, 1974.

Declaration LLM Usage

The writing of certain parts of this thesis were assisted by Large Language Models (LLMs). This section transparently documents which LLMs were used and to what extent. Table 4.1 outlines their applications, ordered descendingly by their overall impact on the thesis. More detailed information regarding the specific models referenced is provided below:

- **Gemini:** Refers to Google's Gemini 3.5 Flash model, as well as Google's Gemini 3.1 Pro model for select tasks.
- **ChatGPT:** Refers to OpenAI's ChatGPT 5.5 model.
- **Claude:** Refers to Anthropic's Claude 4.6 Sonnet model, utilized with varying levels of effort and both with and without reasoning capabilities enabled.

Prior versions of these models may have also been used during the preliminary stages of this thesis. Crucially, every AI-generated output was manually reviewed, verified, and in most cases, heavily edited.

Common Prompt Template

The following system prompt template was utilized for most text-editing and proofreading tasks:

Fix grammar and typos in the text (LaTeX code) below and return it as a code block. You may furthermore fix:

- Uses of "which" and "that" according to the rules of restrictive vs. non-restrictive clauses.
- Mathematically imprecise wording. Especially instances of the word "must" where "is" is more fitting. For example: "therefore must be $t=4$ " should be "therefore $t=4$ ".
- Bad line breaks. For example, I often write "agent $\$1\$$ ". I use this so frequently that I have a custom command "\agn{1}", which internally expands to "agent~ $\$1\$$ " to prevent the text wrapping after "agent" and starting a new line with " $\$1\$$ ". Use academic writing standards to insert "~" where line wraps should be prevented.

Do not change:

- The core content. This applies strictly to mathematical equations. Do not modify or reformat them. Even if you suspect a typo, output a warning below the code block instead of modifying it yourself.
- Custom LaTeX commands, labels, refs, or citations. These may contain typos, but they are not rendered and changing them will break the document compilation.
- Set operators: "+" and "-" denote union and set minus for singleton sets. These are explicitly defined for this work-do not replace them.

Here is the text: <TEXT>

Table 4.1: Overview of LLM utilization across various thesis tasks.

Tool	Use-Case	Notes
ChatGPT, Gemini	Proofreading grammar, correcting common writing mistakes, and identifying typos in equations.	The prompt template used for the majority of these sections is provided in Common Prompt Template.
ChatGPT, Claude	Resolving LaTeX typesetting and template issues.	For example, properly integrating this specific disclosure chapter into the existing LaTeX template.
ChatGPT, Gemini	Reformulating specific paragraphs.	Only used for a select few awkwardly phrased paragraphs. In almost all instances, the suggested rephrasings were manually modified before inclusion.
ChatGPT, Claude	Literature research on highly specific topics.	Specifically utilized when exploring the application of round-robin protocols for fair division. This led to the discovery of a relevant paper [ACIW21]. It was also used to check for existing counterexamples showing that round-robin allocations are not Pareto optimal. While no example in the scientific literature was found, Claude found one in the round-robin Wikipedia entry which we have subsequently based our Example 3.5 on.
Claude	Rapid prototyping and checking of new conceptual ideas.	Primarily used during the initial phases of the thesis to attempt to find counterexamples for early conjectures. Although this approach succeeded in only about one out of every four attempts, it was highly beneficial for deepening engagement with the subject matter. None of the ideas we checked with an LLM was included in the final thesis.
Claude	Generating TikZ diagrams.	The base code for diagrams was generated by the LLM, but each figure required manual adjustments.

Appendix

A Example: 4-Tolerant EFX Allocation Computed by Envy-Reducing Round-Robin on a Permutation Instance

The ideas behind this construction are listed in Section 3.3.4. We only give the allocation $\mathbf{X}$ resulting from an execution of Algorithm 3.3 on a suitable instance. In Table A.1 we identify all resources via $v = v_i$, where i is the agent envious of another agent j such that the removal of 4 resources does not eliminate envy. A similar example was given in Table 3.1, and the ideas for why this suffices for a permutation instance are the same. The only additional part we specify are two constraints on the envy-reducing orders used during Algorithm 3.3, namely: $i \prec_1 l$ and $i \prec_R l$ for all $l \in [n] - i$, as otherwise i would choose higher-valued resources in these rounds.

By the construction given in Section 3.3.4, we require $n = t(t + 2) + 1 = 25$ agents and $m = (t + 2)n = 150$ resources. From these resources, we denote two by i and j as usual. The remaining resources are denoted by [23]. The resources whose removal does not eliminate envy are x_j^r for $r \in [5] - 1$.

The following table describes the rest of the execution of the algorithm.

$r \in [R]$	1	2	3	4	5	6
$v(x_j^r)$	150	119	118	117	116	121
$v(x_i^r)$	126	125	124	123	122	1
$v(x_1^r)$	149	120	93	70	47	24
$v(x_2^r)$	148	115	92	69	46	23
$v(x_3^r)$	147	114	91	68	45	22
$v(x_4^r)$	146	113	90	67	44	21
$v(x_5^r)$	145	112	89	66	43	20
$v(x_6^r)$	144	111	88	65	42	19
$v(x_7^r)$	143	110	87	64	41	18
$v(x_8^r)$	142	109	86	63	40	17
$v(x_9^r)$	141	108	85	62	39	16
$v(x_{10}^r)$	140	107	84	61	38	15
$v(x_{11}^r)$	139	106	83	60	37	14
$v(x_{12}^r)$	138	105	82	59	36	13
$v(x_{13}^r)$	137	104	81	58	35	12
$v(x_{14}^r)$	136	103	80	57	34	11
$v(x_{15}^r)$	135	102	79	56	33	10
$v(x_{16}^r)$	134	101	78	55	32	9
$v(x_{17}^r)$	133	100	77	54	31	8
$v(x_{18}^r)$	132	99	76	53	30	7
$v(x_{19}^r)$	131	98	75	52	29	6
$v(x_{20}^r)$	130	97	74	51	28	5
$v(x_{21}^r)$	129	96	73	50	27	4
$v(x_{22}^r)$	128	95	72	49	26	3
$v(x_{23}^r)$	127	94	71	48	25	2

Table A.1: Resources over agent and rounds valued by $v = v_i$