**Technische Universität Berlin**
Electrical Engineering and Computer Science
Institute of Software Engineering and Theoretical Computer Science
Algorithmics and Computational Complexity (AKT)

# On Fair and Envy-Free Allocations Respecting Acyclic Social Networks

**Bachelorarbeit**

**von Hanno Arnoldi (365785)**

zur Erlangung des Grades „Bachelor of Science“ (B. Sc.)
im Studiengang Computer Science (Informatik)

| | |
|---|---|
| Erstgutachter: | Prof. Dr. Rolf Niedermeier |
| Zweitgutachter: | Prof. Dr. Markus Brill |
| Betreuer: | Dr. Robert Bredereck, |
| | Andrzej Kaczmarczyk, |
| | Prof. Dr. Rolf Niedermeier |

2. Mai, 2019

## Eidesstattliche Erklärung / Statutory Declaration

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

I hereby declare that the thesis submitted is my own, unaided work, complete without any unpermitted external help. Only the sources and resources listed were used.

| | |
|---|---|
| Arnoldi, Hanno | Place, Date |

## Zusammenfassung

Diese Arbeit konzentriert sich auf die Entwicklung von Algorithmen für die Suche von fairen Allokationen. Eine Grundvoraussetzung einer fairen Allokation, in dieser Arbeit, ist das Konzept der Neidfreiheit. Für eine neidfreie Allokation muss der Agent seine erhaltenen Ressourcen höher wertschätzen, als die Ressourcen die einem beliebigen anderen Agenten zugeteilt wurden. Häufig existieren jedoch keine nicht-trivialen neidfreien Allokationen, sodass es häufig NP-schwer ist zu entscheiden, ob eine neidfreien Allokation existiert oder nicht. Bei Einbeziehung der sozialen Beziehungen der Agenten erhalten wir ein lokaleres Konzept der Neidfreiheit, die sogennante Graph-Neidfreiheit. In Graph-neidfreien Allokationen vergleicht ein Agent seine erhaltenen Ressourcen nur mit den Ressourcen seiner Nachbarn. Aus diesem Grund muss für eine Graph-neidfreie Allokation nur gelten, dass jeder Agent seine erhaltenen Ressourcen höher wertschätzt als die Ressourcen seiner Nachbarn. Folgerichtig wird das Konzept der Graph-Neidfreiheit anstelle der allgemeinen Neidfreiheit genutzt.
Es werden gerichtete azyklische Graphen genutzt, um die sozialen Beziehungen der Agenten darzustellen. Der Grund ist, das hierarchische Strukturen häufig durch azyklische Graphen dargestellt werden. Azyklische Graphen haben zusätzlich den weiteren Effekt, dass das Problem einfacher wird. Die Nutzung von zusätzlichen Fairnesskriterien soll außerdem triviale Allokationen verhindern. Als zusätzliche Fairnesskriterien wurden in dieser Arbeit *egalitarian rank fairness* und *half-feedback envy-freeness* eingeführt. Diese werden in der Arbeit später genauer erläutert. Diese Arbeit analysiert die Berechnungskomplexität von Allokation die Graph-neidfrei sind und zusätzlich eine der beiden eben eingeführten Fairnesskriterien erfüllen.

## Abstract

In this thesis we focus on the development of Algorithms for the search of fair allocations. One natural criterion for fairness, in this thesis, is the concept of envy-freeness. For an allocation to be envy-free it is required that every agent values its resources at least as much as the resources of all other agents. Non-trivial envy-free allocations often do not exist, so that deciding whether an envy-free allocations can often be NP-hard. Taking into account the social networks of the agents, it is possible to obtain a more local concept of envy-freeness, the so called graph-envy-freeness. In a graph-envy-free allocation an agent compares the received resources only with its peers. Thus, for an allocation to be graph-envy-free every agent has to value the received resources at least as much as the resources received by its peers. Hence, we use the concept of graph-envy-freeness instead of general envy-freeness.
Directed acyclic graphs are used to represent the social networks of the agents. Acyclic graphs are often used to represent hierarchical structures and have the additional effect, that the problem is simplified. We also introduce additional fairness criteria to prevent trivial allocations. In this thesis we work with the additional fairness criteria *egalitarian*

*rank fairness* and *half-feedback envy-freeness.* These will be described later in this thesis. We analyze the computational complexity for allocation that are graph-envy-free and either egalitarian rank fair or half-feedback envy-free.

# Contents

# 1 Introduction and Motivation

The question how to allocate limited resources to a group of agents occurs frequently. It appears in economics and computer science from different perspectives. The problem of resource allocation, for example, is often discussed in a subcategory of economics, namely social choice theory [1]. In computer science, problems of resource allocation appear often in context of artificial intelligence [2], operations research [3], multiagent systems [4] and electronics commerce [5]. Some examples for multiagent resource allocations are airport traffic management [6], public transport [7], network routing [8], industrial procurement via agents [9], allocation of food [10] and fair and efficient exploitation of Earth Observation Satellites [11], [12].

Depending on the different scenarios, there can be a huge variety of allocation problems [13]. Below we discuss some aspects:

**What is the nature of given resources? Are they divisible or indivisible?**

Historically, the first academic article concerning resource allocation was published by Hugo Steinhaus [14]. In the article, Steinhaus analyzes the distribution of divisible resources. To help to visualize it he is using a cake that can be cut arbitrarily. Thus, these kind of problems are often called "cake cutting."
However, not every resource can be divided. For example, this is the case when kids get different presents and these have to be divided. Presents such as clothes, for example a jacket, cannot be divided. Motivated by this example, in this thesis we focus on indivisible resources.

**What is the nature of the mechanism that leads to the allocation?**

We distinguish between two forms of mechanisms: a centralized and a decentralized form. In the centralized mechanism one authority receives all preferences of the agents and computes the allocation.
On the contrary, in the decentralized mechanism, the agents compute the allocation themselves, as a result of a negotiation. We apply centralized mechanisms to solve our problems.

A visual example for a centralized mechanism would be when a teacher asks the children what toys they want and distributes the toys to the children. The decentralized way is when the teacher gives the group toys and they have to divide them themselves.

## How do the agents express their preferences?

If people have to evaluate something, then too many details are often seen as a hindrance. Thus, the way to evaluate something is often restricted. The same holds for preferences as preferences are also a kind of evaluation where people express how much they like something.
Preferences are expressed additively, monotonically, and numerically. Thus, the value of a single resource is always a nonnegative integer. The value of a set of resources equals the sum of all the values of the resources in the set. We call a set of resources a bundle. In this thesis we work with two special cases of preferences, which are identical and 0/1 preferences. Preferences are called identical if every agent shares the same values for the resources and 0/1 if the agents can only express whether a resource is liked or not. These two properties build the four variants of preference restrictions, that is, additive, identical, 0/1, and identical 0/1 preferences.

For example if you have to feed animals of the same species in a zoo, then these animals usually have the same preferences concerning the food they like more or less. This is an example for identical preferences. But if they were animals of different species, then they would have different preferences for the food and thus have additive preferences. The preference is additive, since the animals can choose not to eat the food and therefore additional food never has a negative impact.
When people work in a shift system, then a workday is usually divided into morning, day and night shifts. The 0/1 preference can be used to express which shifts they want and which they reject. Since every worker lives his own live, the individual availability can be different. Thus, their preferences of different shifts can be different as well. Therefore, the employer can try to divide the shifts as fairly as possible.

## What is the nature of the graphs used in this thesis?

In this thesis we use graphs to represent the relations between agents. Relations between agents can often be of a hierarchical nature. Since hierarchical structures are often represented in an acyclic manner, we use acyclic graphs in this thesis. Also another technical reason is that deciding the existence of a graph-envy-free allocation in a cyclic graph is in general, an NP-hard task. For acyclic graphs, strong graph-envy-freeness allocations might even not exist [15]. An explanation of graph-envy-freeness is given in the next paragraph.

## How to measure the quality of an allocation?

In this thesis we focus on fair allocations. One natural criterion for fairness is the concept of envy-freeness. If we give presents to children, then we automatically try to be fair, so they do not fight over them. Children usually do not fight if they like their present more than the presents of the other children. Hence they do not envy each other. Therefore, an allocation is called *envy-free* if every agent values its bundle the same or higher than the bundles of all other agents. A bundle of an agent is the set of resources the agent receives with an allocation. Since allocations with the restriction of general

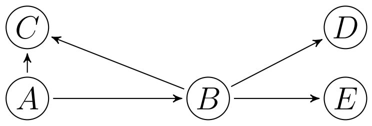


Figure 1.1: The social network for the scenario in Example 1.1.

envy-freeness frequently do not exist, we focus on graph-envy-freeness in this thesis. As mentioned in the paragraph above, we focus on graph-envy-freeness in acyclic graphs. Further, we want the allocations to be complete, since an allocation which allocates nothing to every agent is always envy-free. This holds because the value for every agent's bundle is the same as the bundle of every other agent.

Unfortunately, the restriction of graph-envy-freeness and completeness is not sufficient to ensure a fair allocation, see Example 1.1 which comes in the next paragraph. In order to refine our criteria, we add two further concepts to graph-envy-freeness and completeness. Our objective is to construct a new perception of a fair allocation. The justification of the concepts is depicted in the following examples.

### Scenario

In our scenario we talk about the distribution of resources, in form of supplies, to schools. In the following example we consider three stages of education: elementary, middle, and high school. Students from an elementary school will go to a middle school later in their lives. Then, the elementary school students will have the same resources as the middle school students have today. Thus, while they are elementary students they do not envy middle school students. The same relationship holds for the elementary school students and the high school students as well as for the middle school students and the high school students. However, students of a higher stage of education can be envious of students from a lower stage of education. This results from the fact that once a student has completed an educational stage he cannot go back. We also, fix that students can only be envious when their schools are located geographically close. This is because students that live far away from each other do not meet each other and thus do not envy one another.

**Example 1.1.** Our scenario includes one high school $A$, one middle school $B$, and three elementary schools $C, D$, and $E$. The high school $A$ is located near the middle school $B$, and the elementary school $C$. The middle school $B$ is located near the elementary schools $C, D$, and $E$. Based on this information, the social network in Figure 1.1 is generated.

A council of a city, where the schools $A, B, C, D$, and $E$ are located, receives a donation for the schools in the form of supplies for the computer lab. The donation consists of one electronic whiteboard, one computer, and five calculators. Now the council has to decide how it wants to allocate the donations fairly between the schools. Since all the schools are in the same need of the supplies, they evaluate the supplies in the same way.

| Item | Electronic whiteboard | Computer | Calculator |
|---|---|---|---|
| Value of a supply | 6 | 3 | 1 |

Table 1.1: The values of the supplies used in Example 1.1.

$$\begin{tikzcd}
C & & D \\
A \arrow[u] \arrow[r] & B \arrow[ul] \arrow[ur] \arrow[r] & E
\end{tikzcd}$$

| Agent | Resources | Value |
|---|---|---|
| $A$ | Electronic whiteboard, Computer, Calculator $\times$ 5 | 14 |
| $B, C, D, E$ | | 0 |

Figure 1.2: A graph-envy-free allocation.

The utility values of these supplies are shown in Table 1.1.

The council decides that a fair allocation of the supplies means that the basic requirement for the allocation is to be graph-envy-free. The table in Figure 1.2 and Table 1.2a show possible graph-envy-free allocations. The reason why the sole criterion of graph-envy-freeness is not sufficient is because both allocations are graph-envy-free but most people would say that Table 1.2a shows a fairer allocation than the allocation depicted in Figure 1.2. This is because the the allocation in Figure 1.2 allocates all resources to a single agent and Table 1.2a spreads the resources among the agents.

Thus, the council applies two concepts *egalitarian rank fairness* and *half-feedback envy-freeness*, that represents its perception of fairness. The egalitarian welfare of an allocation is the value of the smallest valued of a bundle in the allocation. An allocation is *egalitarian rank fair* if the egalitarian welfare is maximized and then the number, of bundles which are valued the same as the value of the egalitarian welfare, is minimal. If we only maximize the egalitarian welfare without minimizing the number of bundles evaluated the same as the value of the egalitarian welfare, then we might get an allocation depicted in Table 1.2a instead of the allocation depicted in Table 1.2b. The allocation depicted in Table 1.2a is not considered fair, since the agent $A$ receives all the remaining resources, if the egalitarian welfare cannot be raised. The allocation depicted in Table 1.2b the agents $C$ and $D$ receive the remaining resources. Thus we prevent the cases where one agent receives all the remaining resources. The Tables 1.2 and 1.3 and the table in Figure 1.2 depict the bundle and its value for each school.

A peer of an agent $a$ is every agent $b$ he compares himself with, if an arc $(a, b)$ exists. An allocation is *half-feedback envy-free* if every peer of an agent has at least a bundle of half the value (rounded down) of the value of the agent's bundle. The allocation depicted in Table 1.2b does not fulfill the criteria for half-feedback envy-freeness. This is due to the bundle of $A$. Its value (of six) is more than double the value of the bundle of $C$ (two).

With an allocation depicted in Table 1.3a the criteria for half-feedback envy-freeness are satisfied. Comparing the allocation resulting from the criteria of egalitarian rank

| Agent | Resources | Value |
|---|---|---|
| $A$ | Electronic whiteboard, Calculator $\times$ 2 | 8 |
| $B$ | Computer | 3 |
| $C$ | Calculator | 1 |
| $D$ | Calculator | 1 |
| $E$ | Calculator | 1 |

(a) An allocation with solely maximized egalitarian welfare.

| Agent | Resources | Value |
|---|---|---|
| $A$ | Electronic whiteboard | 6 |
| $B$ | Computer | 3 |
| $C$ | Calculator $\times$ 2 | 2 |
| $D$ | Calculator $\times$ 2 | 2 |
| $E$ | Calculator | 1 |

(b) A possible allocation of resources with the given resources.

| Agent | Resources | Value |
|---|---|---|
| $A$ | Electronic whiteboard | 6 |
| $B$ | Computer | 3 |
| $C$ | Calculator $\times$ 2 | 2 |
| $D$ | Calculator | 1 |
| $E$ | Calculator | 1 |

(c) A possible allocation with four calculators instead of five calculators.

Table 1.2: Allocations with respect to egalitarian rank fairness.

fairness Table 1.3a and half-feedback envy-freeness Table 1.2b the value of the bundles of $C$ and of $D$ have changed., While optimizing for the criteria of egalitarian rank fairness both bundles have the same utility value two. But to fulfill the criteria of half-feedback envy-freeness the bundle of $C$ has to be valued three and the bundle of $D$ has to be evaluated only one Table 1.3a. It is so because $A$ with a bundle valued six is located close to $C$.

To show that the existence of a fair allocation optimizing the criteria of egalitarian rank fairness does not always imply the existence of a fair allocation fulfilling the criteria for half-feedback envy-freeness we use another example.

If we had four instead of the five calculators mentioned above, then an allocation optimizing the criteria of egalitarian rank fairness is still possible, as demonstrated in Table 1.2c. This is because there always exists an allocation optimizing the criteria of egalitarian rank fairness if there exists a graph-envy-free allocation. We notice that egalitarian rank fairness optimizes the criteria of egalitarian welfare and rank. On the contrary half-feedback envy-freeness only decides whether an allocation fulfills the criteria or not.

Indeed half-feedback envy-freeness is impossible, because the supply electronic whiteboard with a value of six has to be allocated to $A$ and due to that $B$ and $C$ have to

| Agent | Resources | Value |
|---|---|---|
| $A$ | Electronic whiteboard | 6 |
| $B$ | Computer | 3 |
| $C$ | Calculator $\times$ 3 | 3 |
| $D$ | Calculator | 1 |
| $E$ | Calculator | 1 |

(a) A possible allocation of resources with the given resources.

| Agent | Resources | Value |
|---|---|---|
| $A$ | Electronic whiteboard | 6 |
| $B$ | Computer | 3 |
| $C$ | Calculator $\times$ 3 | 3 |
| $D$ | Calculator | 1 |
| $E$ | Calculator | 0 |

(b) A unfair allocation with four calculators instead of five calculator.

Table 1.3: Allocations with respect to half-feedback envy-freeness.

get a bundle valued at least three. Since $B$ has a bundle valued three $D$ and $E$ both have to get a bundle valued at least one. This is not possible since the only remaining unassigned supply is a single calculator valued one thus either the bundle of $D$ or of $E$ fails to fulfill the criteria for half-feedback envy-freeness as it is shown in Table 1.3b for $E$.

## Results

Finding a (weakly/strongly) graph-envy-free allocation can lead to some trivial cases [15]. Thus, we use additional concepts to evaluate the fairness of an allocation. We use acyclic graphs since in cyclic ones there might not exist a strongly graph-envy-free allocation and for weakly graph-envy-freeness with cyclic graphs the problem is often NP-hard. One insight of this thesis is that a problem for a strong graph-envy-free allocation is at least as complex as for a weak graph-envy-free allocation.
Another finding is that if resources can be given arbitrary positive integers as values, then the problem is NP-hard even for identical preferences. Therefore, the restriction on preferences is important for the complexity of the problem.
All results are visually collected in Tables 5.1 and 5.2 in Chapter 5.

## Related Work

As the basic concepts of this thesis is not completely new there already exist researches in a similar direction. In the papers [15]–[21] a social network over agents is assumed and envy-freeness for each agent is defined in relation to their neighborhood. This is because envy-freeness without constraints in itself is often already too strong. Chen and Shah [22] also assume that agents do not know what the other agents received in the allocation. There the goal is goal a bit different. The main goal is to study what hiding information implies for the agents. This thesis is strongly related to “Envy-Free Allocations Respecting Social Networks” [15]. This is due to the fact that this thesis is a continuation of the paper.

# 2 Basics and Models

In this chapter we provide the basic concepts, definitions and models we use in this thesis. We use graphs to represent the relations between agents, use bundles to show which resources are allocated to which agent, and use different concepts to define what we understand by a fair allocation. All necessary notions are formally defined in this chapter. As we mentioned before some of the following concepts are not new and were already dealt with similarly, for example in "Handbook of computational social choice" [23] and the papers [15]–[21] In the section below we introduce our graph notation.

## 2.1 Graphs

A directed graph $G = (V, E)$ consists of a set $V$ of vertices and a set $E$ of arcs. By $N(v)$ we denote the (out-)neighborhood of a vertex $v \in V$, that is, the set $W \subset V$ of vertices, such that for each vertex $w \in W$ an *arc* $a = (v, w) \in E$ from vertex $v$ to vertex $w$ exists. An undirected graph $G' = (V', E')$ consists of a set $V'$ of vertices and a set $E'$ of edges. Edges are undirected and thus an edge $e \in E'$ between two vertices $v, w \in V'$ can be written as a set $\{v, w\}$ and not as a tuple like an arc.

An arc $a = (v, w)$ is an outgoing arc for vertex $v$ and an ingoing arc for vertex $w$. A vertex that has only ingoing arcs is called a *sink* and a vertex that has only outgoing arcs is called a *source*. A isolated vertex $u$ is a sink as well as a source. If an arc $a = (v, w)$ exists, vertex $v$ is a *parent* of vertex $w$. In addition vertex $w$ is a *child* of vertex $v$.

In this thesis we focus on graphs without cycles. A directed graph $G = (V, E)$ has a *cycle* if a subgraph $H = (W, F)$ exists where set $W \subseteq V$ and set $F \subseteq E$, such that $W = \{v_1, \ldots, v_n\}$ and $F = \{(v_n, v_1), (v_i, v_{i+1}) \mid 1 \leq i \leq n-1\}$. Subgraph $H$ is a cycle of length $n$.

Since we will talk about graph-envy-free allocations with respect to $G$, we define directed graph classes we deal with.
One directed graph class we focus on in this thesis is the class of *DAGs* (directed acyclic graph). DAGs are the least restricted graphs we focus on. A DAG is, as the name implies, a directed graph without a cycle.

The next graph class is a bit more restricted and consists of *trees*. A tree is a directed graph without a cycle where every vertex has an indegree of one or zero.

The last graph class we work with is a *directed path*. A directed acyclic graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ is a path if $E = \{(v_i, v_{i+1}) \mid 1 \leq i \leq n-1\}$. In a path vertex $v_1$ is a source and vertex $v_n$ is a sink. A path is the most restricted class used in our thesis.

In the following sections we are defining some concepts for allocation problems we need.

## 2.2 Allocation & Bundle

We consider the setting with a finite set of agents $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ and a finite set of resources $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ that are indivisible.

**Definition 2.1.** An *allocation* of a set of resources $\mathcal{R}$ to a set of agents $\mathcal{A}$ is a mapping $\pi\colon \mathcal{A} \to 2^{\mathcal{R}}$. For every $a, a' \in \mathcal{A}$ we call $\pi(a)$ the *bundle* of $a$ under $\pi$. The bundles $\pi(a)$ and $\pi(a')$ are disjoint for all distinct $a$ and $a'$. We require that for every resource $r \in \mathcal{R}$, $r$ is allocated to a bundle such that the allocation is *complete*.

Our definition ensures that every resource can only be assigned to one agent.

There are different methods to model preferences of agents over resources. We focus on the preferences expressed numerically.

**Definition 2.2.** A preference relation $\preceq$ is called *additive* over all subsets of $\mathcal{R}$ if there is a utility function $u\colon \mathcal{R} \to \{0\} \cup \mathbb{N}$ such that for every $X, Y \subseteq \mathcal{R}$ it holds that $X \preceq Y$ if and only if $u(X) \leq u(Y)$, where $u(X)$, for every $X \subseteq \mathcal{R}$, is defined as $\sum_{r \in X} u(r)$.

An additive preference relation is called *monotonic*, if and only if every value of the utility function is non-negative. We restrict the preferences in this work to be additive and monotonic. Moreover, we call them 0/1 if the utility function maps to zero or one for every agent. An agent $a_i \in \mathcal{A}$ has a utility function $u_{a_i}$. We call the preferences *identical* if every agent has the same utility function. Thus for identical preferences we get a family of utility functions $U(u, |\mathcal{A}|)$ with $|\mathcal{A}|$ utility functions that are identical to $u$. We call resources *trivial* that are evaluated zero by every agent. The agents preferences are unary encoded by the utility function $u_{a_i}\colon \mathcal{R} \to \mathbb{N}, a_i \in \mathcal{A}$.

## 2.3 Graph-Envy-Freeness

As we focus on graph-envy-free allocations in this thesis, we first want to formally define what graph-envy-freeness is.

**Definition 2.3.** Let $\mathcal{A}$ be a set of agents, $\mathcal{R}$ be a set of resources and $G = (\mathcal{A}, E)$ be a directed graph. We call an allocation $\pi$ *weak graph-envy-free (GEF)* with respect to $G$ if for each pair of distinct agents $a, a' \in \mathcal{A}$ it is true that when $a' \in N(a)$, then $u_a(\pi(a)) \geq u_a(\pi(a'))$. By the replacement of the $\geq$ with $>$, we obtain the definition of a *strong graph-envy-free (sGEF)* allocation.

When we talk about strong or weak graph-envy-free allocations in this thesis, we omit "with respect to the graph $G$" when $G$ is clear from the context.

One weakness of the concept of weak graph-envy-free allocations is that an allocation $\pi$ which allocates nothing to each agent is always weak graph-envy-free. Thus, we have the restriction that an allocation has to be complete, as mentioned above in Definition 2.1. Even with this restriction, there still exist trivial cases. One of these trivial cases occurs, for example, for agents embedded in a tree. An allocation $\pi$ is weak graph-envy-free even for a single non-empty bundle consisting of all resources that are given to some source-agent of the tree [15].

Therefore, we need an additional concept of fairness to ensure a fair allocation of resources. Our proposals are discussed in the next section.

## 2.4 Fairness Concepts Beyond Envy-Freeness

One aspect we need for our definition of fairness is the egalitarian welfare.

**Definition 2.4.** The *egalitarian welfare* $E(\pi)$ of an allocation $\pi$ is defined as the smallest value of a bundle in $\pi$, that is $E(\pi) = \min_{a \in \mathcal{A}}(u_a(\pi(a)))$.

In our work, the happiness of an agent $a \in \mathcal{A}$ is depicted by the utility value of its bundle $u_a(\pi(a))$. Agents with a value of $E(\pi)$ of their bundle are called unhappy agents. Another aspect of our fairness criteria is rank.

**Definition 2.5.** The *rank* $R(\pi)$ of an allocation $\pi$ is defined as the number of agents with a bundle of valued $E(\pi)$, that is, $R(\pi) = |\{a \in \mathcal{A} \mid u_a(\pi(a)) = E(\pi)\}|$.

Intuitively the rank of an allocation $\pi$ is the number of unhappy agents.

Since the rank of an allocation $\pi$ is the number of minimal valued bundles in $\pi$, there has to exist at least one agent who has a bundle of value $E(\pi)$. Hence, we have the following observation.

**Observation 2.6.** *The rank of an allocation is always greater than zero.*

The definition of our first new fairness concept, that we call egalitarian rank fairness(ER-F), is formally presented below.

**Definition 2.7.** A complete allocation $\pi$ is fair with respect to egalitarian rank fairness, if and only if there exists no other allocation $\pi'$ where $E(\pi) < E(\pi')$ or if $E(\pi) = E(\pi')$ then $R(\pi) > R(\pi')$.

The reasons why the definition of egalitarian rank fairness is as it is presented in Definition 2.7 are described in the following part Example 2.8. If we switched the priority of the two conditions, then there could be an allocation, where one agent gets an empty bundle even if there were enough resources for an allocation where every agent gets at least one resource. This would be the case because the condition to minimize the agents with the smallest utility value would be fulfilled.

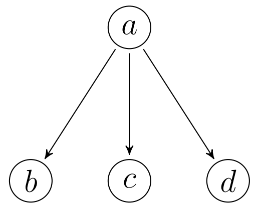


Figure 2.1: A graph for Example 2.8.

If we had only rank as condition, then the same allocation could happen.

If we had only the egalitarian welfare as condition, then there could be an allocation where, as soon as the resources cannot be allocated to every bundle perfectly, there could be a single bundle, where the remaining resources are allocated to. This would be a bundle of a source-agent. Hereafter, we give an example for a weak graph-envy-free allocation with identical 0/1 preferences with respect to ER-F presenting the issues mentioned above.

**Example 2.8.** Let us focus on six resources $\mathcal{R} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ where $u(r) = 1$ for all $r \in \mathcal{R}$, a directed acyclic graph (Figure 2.1 ) $G = (\mathcal{A}, E)$ with $\mathcal{A} = \{a, b, c, d\}$ and $E = \{(a, b), (a, c), (a, d)\}$.

Suppose we changed the priority order. This results in maximizing the egalitarian welfare after minimizing the rank. Then, there exists an allocation $\pi$ that fulfills the criteria with $\pi(a) = \{r_1, r_2, r_3, r_4\}, \pi(b) = \{r_5\}, \pi(c) = \{r_6\}, \pi(d) = \emptyset$ as there is only the agent $d$ with an empty bundle. The goal of the minimization of the rank is satisfied. Thus, the allocation would be called fair, if the minimization of the rank is the primary condition. The same holds if minimizing the rank is the only condition.

If there is only the maximization of the egalitarian welfare, $\max_{\pi \in P}(E(\pi))$ as a condition, then there exists an allocation $\pi$ that fulfills the criteria with $\pi(a) = \{r_1, r_2, r_3\}$, $\pi(b) = \{r_4\}, \pi(c) = \{r_5\}, \pi(d) = \{r_6\}$ as the maximum egalitarian welfare is $E(\pi) = 1$. The goal of the maximization of the egalitarian welfare is satisfied. Hence, the allocation would be called fair, if the maximization of the egalitarian welfare is the only condition or primary condition.

Observe that if we use our definition of fairness (Definition 2.7), then we get an allocation $\pi$ with $\pi(a) = \{r_1, r_2\}, \pi(b) = \{r_3, r_4\}, \pi(c) = \{r_5\}, \pi(d) = \{r_6\}$, which has a broader allocation of the resources. Thus the allocation can be perceived as fair.

Therefore, we need the maximization of the egalitarian welfare as well as minimization of the rank in the given priority order.

The definition of our second fairness concept, that we call half-feedback envy-freeness (HF-EF), is formally presented below.

**Definition 2.9.** A complete allocation $\pi$ is fair with respect to half-feedback envy-freeness, if and only if it holds for every arc $a = (v, w) \in E$, that $u_w(\pi(w)) \geq \left\lfloor \frac{u_w(\pi(v))}{2} \right\rfloor$.

Thus, an allocation is considered fair with respect to half-feedback envy-freeness, if and only if it holds for all child-agents, that the child-agent has a bundle of at least half

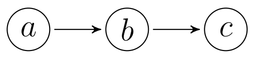


Figure 2.2: A graph for Example 2.10.

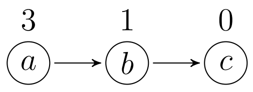


Figure 2.3: A graph for an allocation with four resources.

the value of the bundle of his parent-agent.
Hereafter we have an example of a strongly graph-envy-free allocation with identical 0/1 preference relation.

**Example 2.10.** Let $\mathcal{A}$ be a set of agents, $G = (\mathcal{A}, E)$ be a graph depicted in Figure 2.2 and $\mathcal{R}$ be a set of resources with four resources that are liked by all agents. One possible allocation is shown in Figure 2.3. The number next to a vertex represents the value of the bundle the agent receives, who is represented by the vertex.

If we are given five resources, there exists no possible fair allocation with respect to half-feedback envy-freeness, since the first four resources are allocated as demonstrated in Figure 2.3. Then the last resource cannot be allocated without violating HF-EF. Also, not allocating the resource is not an option, because our allocations have to be complete.

## 2.5 Fairness Problems Handled in this Thesis

In section we define the two problems we work on in this thesis.

Egalitarian Rank Fairness (ER-F) (s)GEF-Allocation

**Input:** A set of agents $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, a set of resources $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$, a family $U = \{u_{a_1}, u_{a_2}, \ldots, u_{a_n}\}$ of agents utility functions for the resources $\mathcal{R}$, a directed graph $G = (\mathcal{A}, E)$ and two positive integers $k, l$.

**Question:** Is there a complete and (strong) graph-envy-free allocation $\pi$ where $E(\pi) \geq l$ and $R(\pi) \leq k$?

For the problems which are polynomial-time solvable we use an algorithm maximizing $l$ and then a minimizing $k$. If an optimization problem is polynomial-time solvable, then the decision problem is also polynomial-time solvable, since with the maximized and minimized upper and lower bound we can directly see, if there is an allocation for a fixed $k$ and $l$.

Half-Feedback Envy-Freeness (HF-EF) (s)GEF-Allocation

**Input:** A set of agents $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$, a set of resources $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$, a family $U = \{u_{a_1}, u_{a_2}, \dots, u_{a_n}\}$ of agents utility functions for the resources $\mathcal{R}$ and a directed graph $G = (\mathcal{A}, E)$.

**Question:** Is there a complete and (strong) graph-envy-free allocation $\pi$ such that the allocation is also half-feedback envy-free?

## 2.6 Common Computational Problems

As we are going to demonstrate computational hardness in this thesis, one possible way is to show it through a polynomial-time many-one reduction. The Clique is a well known NP-complete problem [24].

Clique

**Input:** An undirected graph $G = (V, E)$ and a positive integer $k$.

**Question:** Is there a clique of size $k$, that is, a size-$k$ subset of the vertices such that they are pairwise adjacent?

Another problem we use for a polynomial-time many-one reduction is C-GEF-Allocation. Here we are assuming a variant of C-GEF-Allocation where we are assigning resources to agents which are embedded on a cycle $G = (\mathcal{A}, E)$. The agents have identical monotonic preferences for the resources. The problem is NP-hard [15].

C-GEF-Allocation (Cycle)

**Input:** A set $\mathcal{A}$ of $n$ agents, a set $\mathcal{R}$ of $m$ indivisible resources, a utility function $u$ for all agents and a cycle $G = (\mathcal{A}, E)$.

**Question:** Is there a complete, graph-envy-free allocation $\pi$ of $\mathcal{R}$ to $\mathcal{A}$?

# 3 Paths

In the following chapter we focus on directed paths. We chose directed paths as the first graph class, since it is the most restricted class.

Although it looks so constraint, a natural scenario to apply an allocation on a path is for the allocation of prices after a race. The first place of the race is represented by the source and the last place is represented by the sink as it is shown below.

$$\begin{tikzcd}\text{1 st} \arrow[r] & \text{2 nd} \arrow[r] & \cdots \arrow[r] & n\text{ th}\end{tikzcd}$$

## 3.1 Allocations With Respect to Egalitarian Rank Fairness

In all the following subsections we discuss the computational complexity of Egalitarian Rank Fairness (s)GEF-Allocation in the case that the input graph is a directed path. In each subsection we focus on different preference relations.

### 3.1.1 Identical 0/1 Preferences on a Path

We start with the most restricted preferences, which are the identical 0/1 preferences. With identical 0/1 preferences every agent $a \in \mathcal{A}$ has the same utility function $u\colon \mathcal{R} \to \{0,1\}$. Thus, the only relevant resources from $\mathcal{R}$ are resources which have a utility value $u(r) = 1 \mid r \in \mathcal{R}$. Resources $r' \in \mathcal{R}$ with a utility value of $u(r') = 0$ can be allocated arbitrarily to any agent as no agent envies any other agent these resources.

**Observation 3.1.** *The only relevant information regarding resources in the identical 0/1 preference model for us is the cardinality of $\mathcal{R}' \colon = \{r \in \mathcal{R} \mid u(r) = 1\}$.*

Thus, because of Observation 3.1, whenever we have identical 0/1 preferences we may assume without loss of generality that every resource $r \in \mathcal{R}$ has a utility value of one, so that $u(\mathcal{R}) = |\mathcal{R}|$.

#### Weak Graph-Envy-Freeness

The goal of this subsection is to show that ER-F GEF-Allocation with the input graph being a path and agents having identical 0/1 preferences is polynomial-time solvable. In order to find an egalitarian rank fair and weak graph-envy-free allocation we provide Algorithm 0.

The idea of this algorithm is firstly to allocate the resources evenly so that every agent receives $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ resources. The remaining resources are allocated such that the first $|\mathcal{R}| - (|\mathcal{A}| \left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor)$ agents receive a single resource.

**Algorithm 0:**

```
Input:
$\mathcal{R}$ — a set of non-trivial resources
$\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ — a set of agents
$u(r)$ — a utility function for resources
$G = (\mathcal{A}, E)$ — a directed path
1 mr ← $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$;
2 rer ← $(|\mathcal{R}| - \mathrm{mr}\,|\mathcal{A}|)$;          ▷ variable rer for unassigned resources
3 Assign every agent $a_i \in \mathcal{A}$ a label $l(a_i)$;
4 for $i \leftarrow 1$ to $|\mathcal{A}|$ by 1 do
5     if $i \leq$ rer then
6         $l(a_i) = \mathrm{mr} + 1$;
7     else
8         $l(a_i) = \mathrm{mr}$;
9 Allocate $l(a_i)$ arbitrary resources from $\mathcal{R}$ to every agent $a_i \in \mathcal{A}$;
```

**Theorem 3.2.** *Algorithm 0 solves* ER-F GEF-ALLOCATION *with the input graph being a path and agents having identical 0/1 preferences in $O(|\mathcal{A}|)$ time, where $\mathcal{A}$ denotes the set of agents.*

*Proof.* It can be inferred from Observation 3.1, that the utility value of all the resources from $\mathcal{R}$ sum up to $|\mathcal{R}|$. Assuming identical preferences, the maximum possible egalitarian welfare $E(\pi)$ of an allocation $\pi$ is achieved when every agent receives a bundle which is proportional. The bundle of an agent $a_i$ of an allocation $\pi$ is proportional if $u(\pi(a_i)) = \frac{u(\mathcal{R})}{|\mathcal{A}|}$. Thus we get an upper bound of $\frac{|\mathcal{R}|}{|\mathcal{A}|}$ for the egalitarian welfare. Since we work with indivisible resources the maximum egalitarian welfare has an upper bound of $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$. This derives from the fact that $|\mathcal{R}| \mod |\mathcal{A}|$ does not always equal zero.

Lines 1 and 8 ensure that every agent receives at least $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ resources as follows in line 9. In line 2, we set the variable rer for the number of the remaining resources $c = |\mathcal{R}| - (|\mathcal{A}| \left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor)$, in order not to allocated more, than the available resources $\mathcal{R}$. In lines 6 and 9, we allocate the remaining resources $c$ to every agent $a_i$ where $1 \leq i \leq c$, but only one resource to each agent. Hence by doing this, we reduce the rank by $c$ and therefore the rank is minimized in order for a maximum egalitarian welfare of $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$. The allocation is weakly graph-envy-free since it holds for every agent $a_i$ that $u(\pi(a_i)) = \left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor + 1 \mid 1 \leq i \leq c$ and $u(\pi(a_i)) = \left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor \mid c < i \leq |\mathcal{A}|$.
The reason for the runtime being $O(|\mathcal{A}|)$ is that the only operations depending on the

input are the labeling (line 3) and update of the labels (line 4) that depend on the distance of the agent to the source. For both operations we iterate through the whole set of agents once. Since both operations are not nested and every other operation executes in the same time regardless of the size of the input we obtain our runtime of $O(|\mathcal{A}|)$. □

### Strong Graph-Envy-Freeness

In this subsection we show that ER-F sGEF-Allocation with the input graph being a path and agents having identical 0/1 preferences is polynomial-time solvable. In order to find an egalitarian rank fair and strong graph-envy-free allocation we provide Algorithm 1.
The idea of this algorithm is firstly to first ensure an initial strong graph-envy-free allocation. Then the remaining resources are divided evenly so that every agent receives additional $\left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor$ resources where $b = |\mathcal{R}| - \sum_{i=1}^{|\mathcal{A}|} |\mathcal{A}| - i$. The remaining resources are allocated so that the first $|\mathcal{R}| - (|\mathcal{A}| \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor)$ agents receive a single resource.

**Algorithm 1:**

```
   Input:
   $\mathcal{R}$ — a set of non-trivial resources
   $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ — a set of agents
   $u(r)$ — a utility function for resources
   $G = (\mathcal{A}, E)$ — a directed path
1  if $\sum_{i=1}^{n} n - i > |\mathcal{R}|$ then
2      return;
3  Assign every agent $a_i \in \mathcal{A}$ a label $l(a_i)$ where $l(a_i) = n - i$;
4  rer ← $|\mathcal{R}| - \sum_{i=1}^{n} n - i$;        ▷ variable rer for unassigned resources
5  mr ← $\left\lfloor \frac{\text{rer}}{|\mathcal{A}|} \right\rfloor$;                ▷ variable mr for additional resources
6  for $i \leftarrow 1$ to $|\mathcal{A}|$ by 1 do
7      $l(a_i) \leftarrow l(a_i) + \text{mr}$;                  ▷ update label $l(a_i)$
8  rer ← $(\text{rer} - \text{mr}\,|\mathcal{A}|)$;          ▷ update unassigned resources rer
9  for $i \leftarrow 1$ to $|\mathcal{A}|$ by 1 do
10     if $i \leq \text{rer}$ then
11         $l(a_i) = l(a_i) + 1$;
12     else
13         $l(a_i) = l(a_i)$;
14 Allocate $l(a_i)$ arbitrary resources from $\mathcal{R}$ to every agent $a_i \in \mathcal{A}$;
```

**Theorem 3.3.** *Algorithm 1 solves* ER-F sGEF-Allocation *with the input graph being a path and agents having identical 0/1 preferences in $O(|\mathcal{A}|)$ time, where $\mathcal{A}$ denotes the set of agents.*

*Proof.* With line 1, we check whether there are enough resources for a strongly graph-envy-free allocation. With the initial labeling in line 3 we ensure a strongly graph-envy-free allocation, since it holds for every agent $a_i$ that $l(a_{i-1}) > l(a_i) > l(a_{i+1})$ where $2 \leq i \leq n-1$. Since we require the allocation to be strongly graph-envy-free the maximal egalitarian welfare of an allocation can only be $\left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor$ where $b = |\mathcal{R}| - \sum_{i=1}^{|\mathcal{A}|} |\mathcal{A}| - i$ is the number of unassigned resources after the initial labeling. The reason for this is that we need $\sum_{i=1}^{|\mathcal{A}|} |\mathcal{A}| - i$ resources to ensure a strongly graph-envy-free allocation. We ensure with lines 5 and 7 that every agent receives at least $\left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor$ resources in line 14. In line 8 we update the variable rer for the number of remaining resources $c = b - (|\mathcal{A}| \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor)$ trying not to allocated more than the available resources $\mathcal{R}$. In lines 10 and 11 we allocate the remaining $c$ resources to the agents $a_i$ where $1 \leq i \leq c$, but only one resource to each agent. Since $l(a_{n-1}) > l(a_n)$ is always true, an allocation $\pi$, resulting from line 14, has the minimal rank $R(\pi) = 1$, because it always holds that $u(\pi(a_{n-1})) > u(\pi(a_n))$. The egalitarian welfare is also maximized since $u(\pi(a_n)) = \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor$. The allocation is strong graph-envy-free since it holds for every agent $a_i$ that $u(\pi(a_i)) = |\mathcal{A}| - i + \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor + 1 \mid 1 \leq i \leq c$ and $u(\pi(a_i)) = |\mathcal{A}| - i + \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor \mid c < i \leq |\mathcal{A}|$. Therefore the condition for a strong graph-envy-free allocation of $u(\pi(a_i)) > u(\pi(a_{i+1})) \mid 1 \leq i \leq |\mathcal{A}|$ is always fulfilled.
The reason for the runtime being $O(|\mathcal{A}|)$ is that the only operations depending on the input are the labeling (line 3), the first update of the label (line 6) and the second update of the labels (line 9) that depend on the distance of the agent to the source. For all three operations we iterate through the whole set of agents once. Since all three operations are not nested and every other operation executes in the same time regardless of the size of the input we obtain our runtime of $O(|\mathcal{A}|)$. □

### 3.1.2 Identical Preferences on a Path

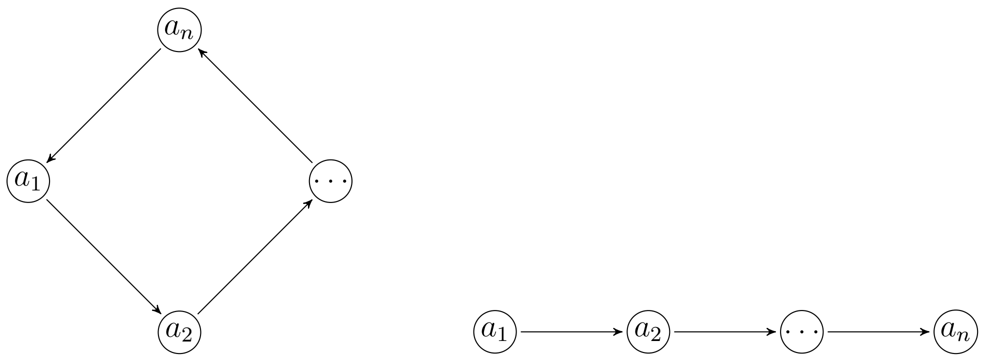
In the following we consider identical preferences. Thus, every agent $a \in \mathcal{A}$ has the same utility function $u \colon \mathcal{R} \to \{0\} \cup \mathbb{N}$.

#### Weak Graph-Envy-Freeness

In this subsection we examine the computational complexity of ER-F GEF-Allocation for agents having identical preferences with the input graph being a path.

**Theorem 3.4.** ER-F GEF-Allocation *is NP-hard even if the input graph is a path and agents having identical preferences.*

*Proof.* In the following we show NP-hardness for ER-F GEF-Allocation with the input graph being a path and agents having identical preferences. Therefore we use a polynomial-time many-one reduction from C-GEF-Allocation (Cycle) (Section 2.6).



(a) cycle (b) Path

Figure 3.2: A visualization of the first step in the Reduction 1, 2 and 3.

**Reduction 1.** Let $I = (\mathcal{A}, \mathcal{R}, u, G)$ be an instance of C-GEF-Allocation (Cycle) where $\mathcal{A}$ is a set of agents $\{a_1, a_2, \dots, a_n\}$, $\mathcal{R}$ is a set of resources $\{r_1, r_2, \dots, r_m\}$, $u\colon \mathcal{R} \to \mathbb{N} \cup \{0\}$ is the agents' utility function and $G = (\mathcal{A}, E)$ is a cycle.

First, we create graph $G'$ as follows: We start with $G' = G$. The graph $G'$ is then opened by deleting the arc $(a_n, a_1)$, as it is depicted in Figure 3.2. Thus, we get a new set of arcs $E' = E \backslash \{(a_n, a_1)\}$ for the path $G' = (\mathcal{A}, E')$.

We set the lower bound $l$ on the egalitarian welfare to $\frac{u(\mathcal{R})}{|\mathcal{A}|}$ and the upper bound $k$ on the rank to $n$. Since we have identical preferences the family of utility functions $U$ consists of $|\mathcal{A}|$ identical utility functions $u\colon \mathcal{R} \to \mathbb{N} \cup \{0\}$.

After these steps we obtain an instance $I' = (\mathcal{A}, \mathcal{R}, U, G', k, l)$ of ER-F GEF-Allocation.

We show that every solution to $I$ is also a solution to $I'$:
Let us assume that there exists an allocation $\pi$ for $I$ that is a solution to C-GEF-Allocation (Cycle). Every agent $a \in \mathcal{A}$ has to have the same value for its bundle $\pi(a)$ because for a graph-envy-free allocation in a cycle it has to hold that $u(\pi(a_i)) \geq u(\pi(a_{i+1})) \mid 1 \leq i \leq n-1$ and also $u(\pi(a_n)) \geq u(\pi(a_1))$. Hence, the rank $R(\pi)$ of the allocation $\pi$ is $n$ because every agent has the same utility value for their bundle in a cycle. Since for a graph-envy-free allocation in a path we only require that $u(\pi(a_i)) \geq u(\pi(a_{i+1})) \mid 1 \leq i \leq n-1$, every solution to C-GEF-Allocation (Cycle) is also a solution to ER-F GEF-Allocation with the input graph being a path and agents having identical preferences. Because the sink agent and the source agent have the same value for their bundle, the egalitarian welfare is maximized in a scenario of ER-F GEF-Allocation with the input graph being a path and agents having identical preferences.

We show that every solution to $I'$ is also a solution to $I$:
Let us assume that there exists an allocation $\pi$ for $I'$ that is a solution to ER-F GEF-Allocation. Since $R(\pi) = n$, every agent has to value its bundle equally to value

$E(\pi)$. Therefore it is true that $u(\pi(a_1)) = u(\pi(a_2)) = \ldots = u(\pi(a_n))$. Thus, the condition for graph-envy-freeness in a cycle which is $u(\pi(a_i)) \geq u(\pi(a_{i+1})) \mid 1 \leq i \leq n-1$ and $u(\pi(a_n)) \geq u(\pi(a_1))$ is fulfilled. Thus the allocation $\pi$ is also a solution for C-GEF-Allocation (Cycle), because every agent has the same value for their bundle and, therefore, no agent envies the bundle of his outneighbor.

The reduction can be executed in polynomial-time and thus ER-F GEF-Allocation is NP-hard.

□

### Strong Graph-Envy-Freeness

In this subsection we examine the computational complexity of ER-F sGEF-Allocation for agents having identical preferences with the input graph being a path.

**Theorem 3.5.** ER-F sGEF-Allocation *is NP-hard even if the input graph is a path and agents having identical preferences.*

*Proof.* In the following we show NP-hardness for ER-F sGEF-Allocation with the input graph being a path and agents having identical preferences. We use again a polynomial-time many-one reduction from C-GEF-Allocation (Cycle) (Section 2.6).

**Reduction 2.** Let $I = (\mathcal{A}, \mathcal{R}, u, G)$ be an instance of C-GEF-Allocation (Cycle) where $\mathcal{A}$ is a set of agents $\{a_1, a_2, \ldots, a_n\}$, $\mathcal{R}$ is a set of resources $\{r_1, r_2, \ldots, r_m\}$, $u\colon \mathcal{R} \to \mathbb{N}$ is the agents' utility function for the resources in $\mathcal{R}$ and $G = (\mathcal{A}, E)$ is a cycle.
This reduction works in two steps.
First we create graph $G'$ as follows: We start with $G' = G$. The graph $G'$ is opened by deleting the arc $(a_n, a_1)$, as it is depicted in Figure 3.2. Thus we get a new set of arcs $E' = E \backslash \{(a_n, a_1)\}$ for the path $G' = (\mathcal{A}, E')$. Since we have identical preferences the family of utility functions $U$ consists of $|\mathcal{A}|$ identical utility functions $u\colon \mathcal{R} \to \mathbb{N} \cup \{0\}$.

In the second step we create a new set of resources $\mathcal{R}'$ as follows: We add a resource $r_{m+i}$ to the set of resources $\mathcal{R}$ for every agent $a_i$. Let the utility value of the resource $r_{m+i}$ be $N - |\mathcal{A}| - i$ where $N > u(\mathcal{R})$. Finally, we obtain the set of resources $\mathcal{R}' = \mathcal{R} \cup \{r_{m+1}, r_{m+2}, \ldots, r_{m+n}\}$. We set the lower bound $l$ for the egalitarian welfare to $\frac{u(\mathcal{R})}{|\mathcal{A}|}$ and the upper bound $k$ for the rank to one.

After these steps we obtain an instance $I' = (\mathcal{A}, \mathcal{R}', U, G', k, l)$ of ER-F sGEF-Allocation.

We show that every solution to $I$ is also a solution to $I'$:
Let us assume that there exists an allocation $\pi$ for $I$ that is a solution to C-GEF-Allocation (Cycle). Every agent $a \in \mathcal{A}$ has to have the same value for their for bundle $\pi(a)$, because for a graph-envy-free allocation in a cycle it holds, that $u(\pi(a_i)) \geq u(\pi(a_{i+1})) \mid 1 \leq i \leq n-1$ and $u(\pi(a_n)) \geq u(\pi(a_1))$. With the added resources described in the second step of the reduction, there exists a fair and strongly graph-envy-free allocation $\pi'$ for $I'$. Here the difference between two values of bundles of adjacent

agents $a_i, a_{i+1}$ in a path is one. Thus, for $1 \leq i \leq n$ each agent $a_i$ receives a bundle $\pi'(a_i)$ that is $\pi(a_i) + r_{m+i}$. This is due to the fact that the bundles without $\{r_{m+1}, \ldots, r_{m+n}\}$ are equal as mentioned above. For the resources $\{r_{m+i}, r_{m+i+1} \mid 1 \leq i \leq n-1\}$ it holds that they only have a value difference of one. Since for a strongly graph-envy-free allocation in a path it only has to hold, that if $u(a_i) > u(a_{i+1}) \mid 1 \leq i \leq n-1$ every fair allocation $\pi$ for a cycle can be reduced to a fair and strongly graph-envy-free allocation $\pi'$ for a path with the additional resources $\{r_{m+1}, r_{m+2}, \ldots r_{m+n}\}$. The egalitarian welfare is maximized, because the difference between two values of bundles of two adjacent agents in a path with identical preferences is one. This is due to the fact that every agent values its bundle without the resources $\{r_{m+1}, r_{m+2}, \ldots r_{m+n}\}$ the same. Thus the value difference of the bundles $\{\pi'(a_i), \pi'(a_{i+1}) \mid 1 \leq i \leq n-1\}$ is one. A rank of one is also minimal.
We show that every solution to $I'$ is also a solution to $I$:
Let us assume that there exists an allocation $\pi'$ for $I'$ that is a solution to ER-F sGEF-Allocation. First we swap the resources with equivalently valued resources, so that for all the resources $\{r_{m+1}, r_{m+2}, \ldots r_{m+n}\}$ it holds that $r_{m+i} \in \pi(a_i)$. Then after removing the resources $\{r_{m+1}, r_{m+2}, \ldots r_{m+n}\}$ from the respective bundles $\pi(a_1), \pi(a_2), \ldots, \pi(a_n)$ we obtain an allocation $\pi$ for $I$ where every agent has a value of $\frac{u(\mathcal{R})}{|\mathcal{A}|}$ for their bundle. Therefore, it is true that $u(a_1) = u(a_2) = \ldots = u(a_n) = E(\pi)$ and thus, the conditions $u(a_i) \geq u(a_{i+1}) \mid 1 \leq i \leq n-1$ and $u(a_n) \geq u(a_1)$ for the cycle are fulfilled.
The reduction can be executed in polynomial-time and thus ER-F sGEF-Allocation is NP-hard.

□

## 3.2 Allocations With Respect to Half-Feedback Envy-Freeness

In all the following subsections we discuss the computational complexity of Half-Feedback Envy-Freeness (s)GEF-Allocation in the case where the input graph is a directed path. In each subsection we focus on different preference relations.

### 3.2.1 Identical 0/1 Preferences on a Path

As in Section 3.1 we start again with identical 0/1 preferences, since it is the most restricted preference relation we use in this thesis. Every agent $a \in \mathcal{A}$ has the same binary utility function $u\colon \mathcal{R} \to \{0, 1\}$ as in Section 3.1.1.

#### Weak Graph-Envy-Freeness

The same Algorithm 0 as in Section 3.1.1 can be used to solve Half-Feedback Envy-Freeness GEF-Allocation. This is due to the fact, that the maximum value difference resulting from the Algorithm 0 is one. One divided by two and rounded down is zero. Thus the allocation resulting from Algorithm 0 fulfills the criteria for half-feedback envy-freeness as well.

### Strong Graph-Envy-Freeness

The goal of this subsection is to show that HF-EF sGEF-Allocation with the input graph being a path and agents having identical 0/1 preferences is polynomial-time solvable. In order to find a half-feedback envy-free and strong graph-envy-free allocation we provide Algorithm 2.

The idea of this algorithm is that we first check if there are enough resources available to ensure a strongly graph-envy-free allocation. Then we search for special problematic cases where no half-feedback envy-free allocation exists, even if there are more resources then needed to ensure a strong graph-envy-free allocation. Afterwards we ensure an initial strong graph-envy-free allocation. Then the remaining resources are divided evenly so that every agent receives additional $\left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor$ resources where $b = |\mathcal{R}| - \sum_{i=1}^{|\mathcal{A}|} |\mathcal{A}| - i$. The remaining resources are allocated such that the first $c = |\mathcal{R}| - (|\mathcal{A}| \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor)$ agents receive a single resource, with the exception of $c = |\mathcal{A}| - 1$. Then the source-agent receives two resources, whereas the others only receive one. Otherwise it will result in one of the problematic cases.

**Algorithm 2:**

```
Input:
$\mathcal{R}$ — a set of non-trivial resources
$\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ — a set of agents
$u(r)$ — a utility function for resources
$G = (\mathcal{A}, E)$ — a directed path
1  if $(n-1)\frac{(n-1)+1}{2} > |\mathcal{R}|$ then
2      return;
3  else if $|\mathcal{A}| = 2$ && $|\mathcal{R}| = 2$ then
4      return;
5  else if $|\mathcal{A}| = 3$ && $|\mathcal{R}| = 5$ then
6      return;
7  Assign every agent $a_i \in \mathcal{A}$ a label $l(a_i)$ where $l(a_i) = n - i$;
8  rer ← $|\mathcal{R}| - n\frac{n+1}{2}$;          ▷ variable rer for unassigned resources
9  mr ← $\lfloor \frac{\text{rer}}{|\mathcal{A}|} \rfloor$;          ▷ variable mr for additional resources
10 for $i \leftarrow 1$ to $|\mathcal{A}|$ by 1 do
11     $l(a_i) \leftarrow l(a_i) + \text{mr}$;          ▷ update label $l(a_i)$
12 rer ← (rer −(mr $|\mathcal{A}|$));          ▷ update unassigned resources $rer$
```

**Algorithm 2:** Continuation

```
13 if $|\mathcal{R}| = (n-1)\frac{(n-1)+1}{2} + n - 1$ then
14     $l(a_1) = l(a_1) + 1$;
15     for $i \leftarrow 1$ to rer $-1$ by $1$ do
16         if $i \leq$ rer $-1$ then
17             $l(a_i) = l(a_i) + 1$;
18         else
19             $l(a_i) = l(a_i)$;
20 else
21     for $i \leftarrow 1$ to rer by $1$ do
22         if $i \leq$ rer *where* $a_i \in \mathcal{A}$ then
23             $l(a_i) = l(a_i) + 1$;
24         else
25             $l(a_i) = l(a_i)$;
26 Allocate $l(a_i)$ arbitrary resources from $\mathcal{R}$ to every agent $a_i \in \mathcal{A}$;
```

**Theorem 3.6.** *Algorithm 2 solves* HF-EF sGEF-Allocation *with the input graph being a path and agents having identical 0/1 preferences in $O(|\mathcal{A}|)$ time, where $\mathcal{A}$ denotes the set of agents.*

*Proof.* The minimal number of resources needed for strongly graph-envy-free allocation with $n$ agents are $\sum_{i=1}^{n} n - i = (n-1)\frac{(n-1)+1}{2}$ non-trivial resources. This is needed since every agent $a_i$ needs at least a bundle valued $n - i$. This is because the agent $a_n$ gets nothing and each agents parent receives one more resource ending with agent $a_1$ receiving a bundle valued $n - 1$. There are two exceptions where we have enough resources for a strongly graph-envy-free allocation but where no half-feedback envy-free allocation exists.

The first exception (mentioned in line 3) occurs when there are two resources and two agents. For the allocation to be complete and strongly graph-envy-free both resources must be given to agent $a_1$. Thus we get $\left\lfloor \frac{2}{2} \right\rfloor > 0$ and the allocation is not half-feedback envy-free. The other exception (mentioned in line 5) occurs when there are five resources and three agents. The only allocations that are complete and strongly graph-envy-free are the allocation $\pi$ and $\pi'$. In the allocation $\pi$ the agent $a_1$ receives four resources, the agent $a_2$ receives one resource, and the agent $a_3$ receives nothing. In the allocation $\pi'$ the agent $a_1$ receives three resources, the agent $a_2$ receives two resource, and the agent $a_3$ receives nothing. This is easy to verify by checking every possible allocation via a brute-force method since the numbers are still small. Both allocations are not half-feedback envy-free, because for $\pi$ between the agents $a_1$ and $a_2$ it is true that $\left\lfloor \frac{4}{2} \right\rfloor > 1$ and for $\pi'$ between the agents $a_2$ and $a_3$ it is true that $\left\lfloor \frac{2}{2} \right\rfloor > 0$. The reason why only these two counterexamples exist is that those are the only ones where problematic cases cannot

be avoided. This is explained in detail below.
Since we allocate the number of resources indicated by the label to the corresponding agents, we ensure the existence of a strongly graph-envy-free allocation with the initial labeling in 7. The allocation resulting from the algorithm is also strongly graph-envy-free since an agent $a_i$ gets either one more or the same number of additional resources as an agent $a_j$ when there is an arc $(a_i, a_j) \in E$. Thus with the original value difference the allocation is strongly graph-envy-free (lines 11, 17 and 23).
The only situation, where a value difference of two is problematic for the criteria of half-feedback envy-freeness is when we have two adjacent agents $a_i$ and $a_{i+1}$ where $a_i$ has a bundle valued two and $a_{i+1}$ has a bundle valued zero. This is because in this case we have $\left\lfloor \frac{n+2}{2} \right\rfloor > n \mid n = 0$. For every $n > 0$ we get $\left\lfloor \frac{n+2}{2} \right\rfloor \leq n$ thus fulfilling the criteria for half-feedback envy-freeness. Problematic situations only appear when $|\mathcal{R}| = (n-1)\frac{n-1+1}{2} + n - 1$. This is because after the allocation of $(n-1)\frac{n-1+1}{2}$ when the remaining $n-1$ are allocated starting by $a_1$ the last agent that receives a resource is $a_{n-1}$. This results in the problematic situation where $a_{n-1}$ has a bundle valued two and $a_n$ a bundle valued zero. In this situation the agent $a_1$ is given an additional resource. This can result in a value difference of three between the agents $a_1$ and $a_2$. This is only the case when $|\mathcal{A}| = 3$ since in every other case $a_2$ gets an additional resource resulting again in a value difference of two. These two special cases have been already checked in the beginning.
Therefore Algorithm 2 finds a solution for HF-EF sGEF-Allocation with the input graph being a path and agents having identical 0/1 preferences, if there exists an allocation fulfilling the criteria.
The reason for the runtime being $O(|\mathcal{A}|)$ is that the only operations depending on the input are the labeling (line 7), the first update of the label (line 10) and the second update of the labels (line 15 or 21) that depend on the distance of the agent to the source. For all three operations we iterate through the whole set of agents once. Since all three operations are not nested and every other operation executes in the same time regardless of the size of the input we obtain our runtime of $O(|\mathcal{A}|)$. □

### 3.2.2 Identical Preferences on a Path

As in Section 3.1 we focus on agents having identical preferences in the next subsection. Thus, every agent $a \in \mathcal{A}$ has the same utility function $u \colon \mathcal{R} \to \{0\} \cup \mathbb{N}$. Hereafter we analyze our problem of finding a half-feedback envy-free allocation for the variants of weak graph-envy-freeness and strong graph-envy-freeness.

#### Weak Graph-Envy-Freeness

In this subsection we examine the computational complexity of HF-EF GEF-Allocation for agents having identical preferences with the input graph being a path.

**Theorem 3.7.** HF-EF GEF-Allocation *is NP-hard even if the input graph is a path and agents having identical preferences.*

*Proof.* To show NP-hardness for HF-EF GEF-Allocation with the input graph being a path and agents having identical preferences we use a polynomial-time many-one reduction from C-GEF-Allocation (Cycle).

**Reduction 3.** Let $I = (\mathcal{A}, \mathcal{R}, u, G = (\mathcal{A}, E))$ be an instance of C-GEF-Allocation (Cycle) where $\mathcal{A}$ is a set of agents $\{a_1, a_2, \dots, a_n\}$, $\mathcal{R}$ is a set of resources $\{r_1, r_2, \dots, r_m\}$, $u\colon \mathcal{R} \to \mathbb{N}$ is the agents utility function for the resources in $\mathcal{R}$ and $G = (\mathcal{A}, E)$ is a cycle.

This reduction works in two steps. First we create graph $G'$ as follows: We start with $G' = G$. The graph $G'$ is opened by deleting the arc between two agents $(a_n, a_1)$, as it is depicted in Figure 3.2. We create a new set of agents $\mathcal{A}' = \mathcal{A} \cup \{a_0\}$. Thus we get a new set of arcs $E'$ where $E' = (E \setminus \{(a_n, a_1)\}) \cup \{(a_0, a_1)\}$ for the path $G' = (\mathcal{A}', E')$. Since we have identical preferences the family of utility functions $U$ consists of $|\mathcal{A}|$ identical utility functions $u\colon \mathcal{R} \to \mathbb{N} \cup \{0\}$.

In the second step we create a new set of resources $\mathcal{R}' = \mathcal{R} \cup \{r_{m+1}, r_{m+2}, \dots, r_{m+n+1}\}$. The newly added resources have a utility value of $u(r_{m+1}) = \frac{1}{1}N$ and $u(r_{m+i}) = \frac{1}{2^{i-1}}N - C \mid 2 \leq i \leq n+1$, where $C = \frac{u(\mathcal{R})}{|\mathcal{A}|}$. We require that $\frac{1}{2^n}N - C > \sum_{i=1}^{n} u(r_i)$.

After these steps we obtain an instance $I' = (\mathcal{A}', \mathcal{R}', u, G')$ of HF-EF GEF-Allocation.

We show that every solution to $I$ is also a solution to $I'$:
Let us assume that there exists an allocation $\pi$ for $I$ that is a solution to C-GEF-Allocation (Cycle). Every agent $a_i \in \mathcal{A} \mid i \in \{1, 2, \dots, n\}$ has a bundle with the value $C$. Every agent $a_i \in \mathcal{A}' \mid i \in \{0, 1, \dots, n\}$ gets the resource $r_{m+i+1}$ added to its bundle and we obtain an allocation $\pi'$ for $I'$. Hence, each agent $a_i \in \mathcal{A}' \mid 1 \leq i \leq n$ has a bundle with the value $\frac{1}{2^i}N - C + C$. Since the agent $a_0$ already has bundle with the value $\frac{1}{1}N$ every agent $a_i \in \mathcal{A}'$ has a bundle with the value $\frac{1}{2^i}N$. Thus it is always true that $u(a_i) = 2u(a_{i+1}) \mid 0 \leq i \leq n-1$ and the conditions for a complete, weakly graph-envy-free and half-feedback envy-free allocation $\pi'$ are fulfilled.

We show that every solution to $I'$ is also a solution to $I$:
Let us assume that there exists an allocation $\pi'$ for $I'$ that is a solution to HF-EF GEF-Allocation. Since $\frac{1}{2^n}N - C > \sum_{i=1}^{n} u(r_i)$ it is required that the resources $\{r_{m+1}, r_{m+2}, \dots, r_{m+n+1}\}$ are allocated to so that $r_{m+i} \in \pi'(a_{i-1}) \mid 1 \leq i \leq n+1$. Since agent $a_0$ received the resource $r_{m+1}$ it has a bundle valued $N$. For the allocation to be strong graph-envy-free each agent $a_i \mid 1 \leq i \leq n$ that has a bundle valued $\frac{1}{2^i}N - C$ by receiving the resource $r_{m+i+1}$. Thus, the remaining resources from $\mathcal{R}'$ are allocated, so that every agent $a_i$ receives a set of additional resources valued $C$. Then every agent $a_i \mid 0 \leq i \leq n$ has a bundle valued $\frac{1}{2^i}N$ and the allocation is half-feedback envy-free. After removing the resources $\{r_{m+1}, r_{m+2}, \dots, r_{m+n+1}\}$ from the respective bundles $\pi'(a_0), \pi'(a_1), \dots, \pi'(a_n)$ and the agent $a_0$ from $\mathcal{A}'$ we obtain the allocation $\pi$ and the set of agents $\mathcal{A}$ for $I$. Every agent $a_i \in \mathcal{A}$ now has a bundle valued $C$. Therefore, it is true that $u(\pi(a_1)) = u(\pi(a_2)) = \ldots = u(\pi(a_n)) = E(\pi)$ and the conditions $u(\pi(a_i)) \geq u(\pi(a_{i+1})) \mid 1 \leq i \leq n-1$ and $u(\pi(a_n)) \geq u(a_1)$ for the cycle are fulfilled.

The reduction can be executed in polynomial-time and thus HF-EF GEF-Allocation is NP-hard. □

#### Strong Graph-Envy-Freeness

In this subsection we examine the computational complexity of HF-EF sGEF-Allocation for agents having identical preferences with the input graph being a path.

**Theorem 3.8.** *Even* HF-EF sGEF-Allocation *for identical preferences in a path is NP-hard.*

*Proof.* The same proof as in HF-EF GEF-Allocation with the input graph being a path and agents having identical preferences can be applied here as well. This is because in the proof above every agent $a_i$ receives a bundle valued $\frac{1}{2^i}N$. Thus, for every arc $(a_i, a_{i+1}) \in E \wedge a, b \in \mathcal{A}$ it holds that $u(\pi(a_i)) > u(\pi(a_{i+1}))$. Therefore the proof is also applicable for the variant of strongly graph-envy-freeness. □

### 3.2.3 0/1 Preferences on a Path

In the following we have 0/1 preferences. Every agent $a_i \in \mathcal{A}$ has its own utility function $u_{a_i} \colon \mathcal{R} \to \{0, 1\}$. If a resource $r \in \mathcal{R}$ is valuated zero by every agent $a \in \mathcal{A}$ the resource can be allocated arbitrarily. Thus we only focus on resources, which are valuated as one by at least one agent.

**Observation 3.9.** *There is no resource $r \in \mathcal{R}$ that is valuated zero by every agent $a \in \mathcal{A}$.*

#### Weak Graph-Envy-Freeness

The goal of this subsection is to show that HF-EF GEF-Allocation with the input graph being a path and agents having 0/1 preferences is polynomial-time solvable. In order to find a half-feedback envy-free and weak graph-envy-free allocation we provide Algorithm 3.
The idea of this algorithm is that the agents get assigned one resource after another. The order depends on the distance to the source. The procedure starts with the source-agent and continues on to the sink-agent. Then the procedure starts again with the source-agent. This continues until all resources are assigned.

**Algorithm 3:**

```
   Input:
   $\mathcal{R}$ — a set of non-trivial resources
   $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ — a set of agents
   $U = \{u_1, u_2, \ldots, u_n\}$ — a family of agents' utility functions for resources
   $G = (\mathcal{A}, E)$ — a directed path
1  while there exists unassigned resources do
2      for $i \leftarrow 1$ to $|\mathcal{A}|$ by $1$;            ▷ Loop through the agents
3      do
4          Assign a liked and unassigned resource $r_j$ to agent $a_i$;
```

**Theorem 3.10.** *Algorithm 3 solves* HF-EF GEF-ALLOCATION *with the input graph being a path and agents having 0/1 preferences in $O(|\mathcal{A}||\mathcal{R}|)$ time, where $\mathcal{A}$ denotes the set of agents and $\mathcal{R}$ denotes the set of resources.*

*Proof.* Theorem 3.10 is proven by induction.
The base case we choose is for $\mathcal{R} = \emptyset$. With $\mathcal{R} = \emptyset$ the allocation that is complete, weakly graph-envy-free and also half-feedback envy-free is the allocation that allocates nothing to every agent. Since the set of resources is empty and there are no resources left to allocate the allocation is complete. No agent envies another agent since every agent receives no resource. For the same reason the allocation is also half-feedback envy-free. The algorithm with $\mathcal{R} = \emptyset$ does not execute the while loop in line 1. Thus, the empty allocation is obtained by our algorithm. Hence, the proof is correctly applicable for the base case.
Our inductive assumption is that Algorithm 3 finds a complete, weakly graph-envy-free and half-feedback envy-free allocation with $k$ iterations. In each iteration a single resource is assigned to an agent. The allocation that is found also has a maximum value difference of one between bundles of $a_i$ and of $a_{i+1}$ from the viewpoint of $a_{i+1}$ where $1 \leq i \leq n - 1$.
For the inductive step we show that there is also a complete, weakly graph-envy-free and half-feedback envy-free allocation for $k + 1$ iterations. In each iteration a single resource is assigned to an agent. The allocation is complete since the allocation obtained with $k$ iterations is complete and then in the $k + 1$th iteration the $k + 1$th resource is allocated. The reason is that in each iteration a single resource is allocated. The allocation with $k$ iterations is weakly graph-envy-free. We assume that after the additional iteration the allocation is not weakly graph-envy-free. For this to happen, a pair of agents $(a, b) \in E \wedge a, b \in \mathcal{A}$ needs to have the same value for their bundle. Then $b$ must receive the next resource in the $k + 1$th iteration. This is impossible, since the agent $a$ receives a liked resource in the $k$th iteration such that $a$ has a bundle valued one more than $b$. If the agent $b$ receives a resource in the $k + 1$th iteration they have the same value for their bundle.
Lastly the allocation is also half-feedback envy-free since the allocation with $k$ iterations is half-feedback envy-free and the value difference is maximal one, from the viewpoint

of $b$. After the $k+1$th iteration the value difference is at most one. The value difference between the bundles of $b$ and of $c$ where $(b, c) \in E \wedge b, c \in \mathcal{A}$ is maximal one from the viewpoint of $c$. This is because the value difference between $b$ and $c$ in the $k$th iteration was zero and now after the $(k+1)$th iteration is at most one. One divided by two and then rounded down is zero, hence the allocation stays half-feedback envy-free. Therefore the allocation obtained by Algorithm 3 is a solution to HF-EF GEF-Allocation with the input graph being a path and agents having 0/1 preferences.
The reason for the runtime being $O(|\mathcal{A}||\mathcal{R}|)$ is that in the worst case only the last agent $a_n$ likes the resources $r \in \mathcal{R}$. Every other agent values every resource as zero. Thus for every single resource we have to iterate through the whole set of agents resulting in $O(|\mathcal{A}||\mathcal{R}|)$. □

# 4 Trees & DAGs

In this chapter we focus on the remaining two graph classes we study, namely trees and DAGs. The reason for combining these two graph classes in one chapter is due to the fact that some cases have the same solution.

## 4.1 Allocations With Respect to Egalitarian Rank Fairness

In all the following subsections we discuss the computational complexity of Egalitarian Rank Fairness (s)GEF-Allocation. As we are analyzing less restricted graph classes, compared to paths, we get new insights into their computational complexity. For some problems with the same preference relations the runtime increases by changing the input graph. In the case of the ER-F sGEF-Allocation with agents having identical 0/1 preferences, the computational complexity changes accordingly to the input graph. If the input graph is a DAG it becomes NP-hard, while the problem with the input graph being a path is polynomial-time solvable.

### 4.1.1 Identical 0/1 Preferences on a Tree & DAG

As in Section 3.1 we start again with agents having identical 0/1 preferences, since it is the most restricted preference relation we use in this thesis. Every agent $a \in \mathcal{A}$ has the same binary utility function $u\colon \mathcal{R} \to \{0,1\}$ as in Section 3.1.1.

#### Weak Graph-Envy-Freeness on a Tree & DAG

In this subsection, we show that ER-F GEF-Allocation with the input graph being a DAG and agents having identical 0/1 preferences is polynomial-time solvable. In order to find an egalitarian rank fair and weak graph-envy-free allocation we provide Algorithm 4.
The idea of this algorithm is firstly to allocate the resources evenly such that every agent receives $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ resources. The remaining resources are allocated such that $|\mathcal{R}| - (|\mathcal{A}| \left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor)$ agents with the smallest distance to a source receive a single resource.

**Algorithm 4:**

```
   Input:
   $\mathcal{R}$ — a set of non-trivial resources
   $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ — a set of agents
   $u(r)$ — a utility function for resources
   $G = (\mathcal{A}, E)$ — a DAG
1  mr ← $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$;
2  Assign every agent $a_i \in \mathcal{A}$ a label $l(a_i)$ where $l(a_i) = \text{mr}$;
3  rer ← $|\mathcal{R}| - (\text{mr}\,|\mathcal{A}|)$;   ▷ rer is the number of unassigned resources
4  for each $a_i \in \mathcal{A}$ do
5      aDist$_i$ ← source_dist$(G, a_i)$;  ▷ source_dist is a function to determine
         the longest distance to the source
6  $A_j$ ← $\{a_i \in \mathcal{A} \mid \text{aDist}_i = j\}$;   ▷ group the agents by their distance to
     the source agent
7  while rer $> 0$ do
8      for $j \leftarrow 1$ to max(aDist$_i$) by 1 do
9          if $|A_j| >$ rer then
10             update $|A_j|$ arbitrary labels $l(a_i) \leftarrow l(a_i) + 1$ where $a_i \in A_j$;
11         else
12             update every label $l(a_i) \leftarrow l(a_i) + 1$ where $a_i \in A_j$;
13         rer $-|A_j|$;
14 Allocate $l(a_i)$ arbitrary resources from $\mathcal{R}$ to every agent $a_i \in \mathcal{A}$;
```

**Theorem 4.1.** *Algorithm 4 solves* ER-F GEF-ALLOCATION *with the input graph being even a DAG and agents having identical 0/1 preferences in $O(|\mathcal{A}|)$ time, where $\mathcal{A}$ denotes the set of agents.*

*Proof.* As we mentioned in the proof for Theorem 3.2, the maximum egalitarian welfare $E(\pi)$ of an allocation $\pi$ has an upper bound of $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$. The minimal rank is $|\mathcal{A}| - c$ where $c = (|\mathcal{A}| \left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor)$ since , except for $c$ resources, the other resources are needed to ensure the egalitarian welfare of $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$. Thus, the optimal allocation has an egalitarian welfare of $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ and a rank of $|\mathcal{A}| - c$.

Lines 1 and 2 ensure that every agent receives a bundle valued at least $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ in line 14. In line 12 every agent $a_i$ where $a_i \in A_0 \cup \ldots \cup A_j \mid \sum_{k=0}^{j} |A_k| \leq c$ with $j$ being maximal, receives an additional resource. Then the number of remaining resources is $d = c - \sum_{k=0}^{j} |A_k|$. Then $d$ arbitrary agents $a \in A_{j+1}$ receive an additional resource. Thus, by doing all these steps, we reduce the rank by $c$. Hence Algorithm 4 generates an allocation with a maximal egalitarian welfare of $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ and then a minimal rank of

$|\mathcal{A}| - c$.

To show that the allocation is weakly graph-envy-free, we consider two cases. In the first case we look at two agents with a different distance to the source. In the second case we look at the agents with the same distance. In the first case the allocation is weakly graph-envy-free, since the value of a bundle from an agent $a \in A_i$ from a set with a smaller distance to the source always has a bundle valued at least as much as a bundle from an agent $b \in A_j \mid j > i$ from a set with a bigger distance to the source. This is because the agents with a smaller distance to the source receive a resource before the agents with a bigger distance to the source (line 8). In the second case, for the agents $a \in A_i$ and $b \in A_i$ with the same distance to the source, the allocation is always graph-envy-free, since they are not adjacent. This is due to the fact that the longest distance for one agent $a$ would be higher than for the other agent $b$, if an arc $(b, a) \in E$ exists. This reasoning is symmetric and also holds the other way around for $(a, b) \in E$. Therefore, they cannot be in the same set $A_i$ if they are adjacent.
The algorithm also works for DAGs with multiple connected components. This is due to the reason that every agent receives a bundle valued at least $\left\lfloor \frac{|\mathcal{R}|}{|\mathcal{A}|} \right\rfloor$ such that the egalitarian welfare is maximized. Thus it is not relevant, whether the DAG consists of multiple connected components. Since the rank is reduced by allocating a single additional resource to the agents depending on their distance to a source the algorithm also works for DAGs consisting of multiple connected components. The reason for the runtime being $O(|\mathcal{A}|)$ is, that the only operations depending on the input are the labeling (line 2), grouping the agents according to the distance to (line 4) and update of the labels (line 8) that depend on the distance of the agent to the source. For all three operations we go through the whole set of agents once. Since all three operations are not nested and every other operation executes in the same time regardless of the size of the input we obtain our runtime of $O(|\mathcal{A}|)$. $\square$

Since Algorithm 4 solves ER-F GEF-Allocation with the input graph being a DAG and agents having identical 0/1 preferences in $O(|\mathcal{A}|)$ time it also solves the problem for a tree since a tree is also a DAG.

### Strong Graph-Envy-Freeness on a Tree

In this subsection we show that ER-F sGEF-Allocation with the input graph being a tree and agents having identical 0/1 preferences is polynomial-time solvable. In order to find an egalitarian rank fair and strong graph-envy-free allocation we provide Algorithm 5.
The idea of this algorithm is firstly to ensure an initial strong graph-envy-free allocation. Then the remaining resources are divided evenly such that every agent receives additional $\left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor$ resources where $b = |\mathcal{R}| - \sum_{i=1}^{n} h_{a_i}$ with $h_{a_i}$ being the length of the longest possible path to a sink. Afterwards, we reduce the rank as much as possible with the remaining resources where $c = |\mathcal{R}| - (|\mathcal{A}| \left\lfloor \frac{b}{|\mathcal{A}|} \right\rfloor)$ is the number of remaining resources. For this purpose we use the algorithm depicted in “A Depth-First Dynamic

Programming Algorithm for the Tree Knapsack Problem" [25] as a blackbox.
The input for the blackbox is an integer (budget), a tree and for every vertex of the tree two integers (cost and value). The blackbox then finds, for the given budget, the vertices, that give the highest possible attainable value. A vertex can only be chosen if its parent was chosen as well.
In our case every the budget is the number of remaining resources $c$ and the tree is our input graph $G$. Since we want to reduce the rank which depends on the sinks, every sink has a value of one and a cost of one. Every other vertex has a value of zero and a cost of one. This is due to the fact that they do not reduce the rank if they receive a resource but have to receive a resource such that the allocation stays strongly graph envy-free. The output of the blackbox is the set of vertices needed to reach the most sinks in the tree with the given budget. Thus the rank is reduced as much as possible.
The function rank_reduction uses the blackbox in our algorithm.

**Algorithm 5:**

```
   Input:
   $\mathcal{R}$ — a set of non-trivial resources
   $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ — a set of agents
   $u(r)$ — a utility function for resources
   $G = (\mathcal{A}, E)$ — a directed tree
 1 $h_{a_i}$ is the height of the agents $a_i \in \mathcal{A}$ in the tree;  ▷ the height of an agent
     is the length of the longest possible path to a sink
 2 if $\sum_{i=1}^{n} h_{a_i} > |\mathcal{R}|$ then
 3     return;
 4 Assign every agent $a_i \in \mathcal{A}$ a label $l(a_i)$ where $l(a_i) = h_{a_i}$;
 5 rer ← $|\mathcal{R}| - \sum_{i=1}^{n} h_{a_i}$;               ▷ rer for unassigned resources
 6 mr ← $\left\lfloor \frac{\text{rer}}{|\mathcal{A}|} \right\rfloor$;               ▷ mr for additional resources
 7 $l(a) \leftarrow l(a) + \text{mr}$ where $a \in \mathcal{A}$;                  ▷ update label $l(a)$
 8 rer ← $(\text{rer} - (\text{mr}\,|\mathcal{A}|))$;             ▷ update remaining resources rer
 9 Use *rank_reduction* (rer, $G$) and receive an output set $B$;
10 for each $b \in B$ do
11     increase $l(b)$ by one;
12 Allocate $l(a)$ arbitrary resources from $\mathcal{R}$ to every agent $a \in \mathcal{A}$;
13 Function rank_reduction(Budget, Tree (set of values for each agent, set of
     costs for each agent ))
14     return $B$ after the use of the blackbox;
```

**Theorem 4.2.** *Algorithm 5 solves* ER-F sGEF-Allocation *with the input graph being a tree and agents having identical 0/1 preferences in $O(|\mathcal{A}|^2)$ time, where $\mathcal{A}$ denotes the set of agents.*

*Proof.* With line 2, we check if there are enough resources for a strongly graph-envy-free allocation. With the initial labeling in line 4, we ensure a strongly graph-envy-free allocation. The number of the resources needed is $b = \sum_{i=1}^{n} h_{a_i}$ where $h_{a_i}$ is the height of a vertex in a tree. This is the length of the longest path to all sinks that are reachable from it. Thus, if every agent gets a bundle valued its height, then the allocation is strongly graph envy-free, since there is always a value difference of at least one. Since we require the allocation to be strongly graph-envy-free the upper bound of an egalitarian welfare of an allocation is $\frac{c}{|\mathcal{A}|}$, while $c = |\mathcal{R}| - b$ is the number of remaining resources after ensuring that the allocation is strong graph-envy-free. Since we have indivisible resources we have a upper bound of $\left\lfloor \frac{c}{|\mathcal{A}|} \right\rfloor$ for the egalitarian welfare. Lines 6 and 7 ensure that every agent receives a bundle valued at least $\left\lfloor \frac{c}{|\mathcal{A}|} \right\rfloor$ in the end (line 12). Hence, the egalitarian welfare of the allocation is maximized. The then remaining resources are $d = c - (|\mathcal{A}| \left\lfloor \frac{c}{|\mathcal{A}|} \right\rfloor)$. As for an optimal minimization of the rank for the allocation we use the algorithm depicted in "A Depth-First Dynamic Programming Algorithm for the Tree Knapsack Problem" [25] as a blackbox. As the sinks of the graph have the least value for their bundle every sink has a value of one and every other vertex has a value of zero in the blackbox. The cost for each vertex is also one since every vertex in the path to the sinks has to get a resource for the allocation to stay strongly graph-envy-free. So the blackbox gives us a set of vertices, building a subgraph, of the size $d$ with the greatest number of sinks from the original input graph $G$. Since for every chosen vertex the parent has to be chosen as well, the source is always in the set of vertices. Thus, after assigning the remaining resources $d$ to the agents representing the subgraph we reduce the rank by the number of sinks. The reason for the runtime being $O(|\mathcal{A}|^2)$ is that the only operations depending on the input are the labeling (line 4), update of the labels (line 11) that depend on the distance of the agent to the source and the usage of the function rank_reduction (line 9). For the operations in lines 4 and 11 we only go through the whole set of agents once. The runtime for the function (line 9) is $O(|\mathcal{A}|^2)$. This is due to the reason, that the function has a runtime of $O(nH)$ [25] where $n$ is the number of agents $|\mathcal{A}|$ and $H$ the budget. In our case the budget can only have a maximum value of $|\mathcal{A}| - 1$ since otherwise every other agent would receive an additional resource. Since all three operations are not nested and every other operation executes in the same time regardless of the size of the input we obtain our runtime of $O(|\mathcal{A}|^2)$. □

### Strong Graph-Envy-Freeness on a DAG

In this subsection we look into the computational complexity of ER-F sGEF-Allocation with the input graph being a DAG and agents having identical 0/1 preferences.

**Theorem 4.3.** ER-F sGEF-Allocation *is NP-hard with the input graph being a DAG and agents having identical 0/1 preferences.*

*Proof.* In this subsection we show NP-hardness for ER-F sGEF-Allocation with the input graph being a DAG and agents having identical 0/1 preferences. Therefore we use a polynomial-time many-one reduction from Clique.

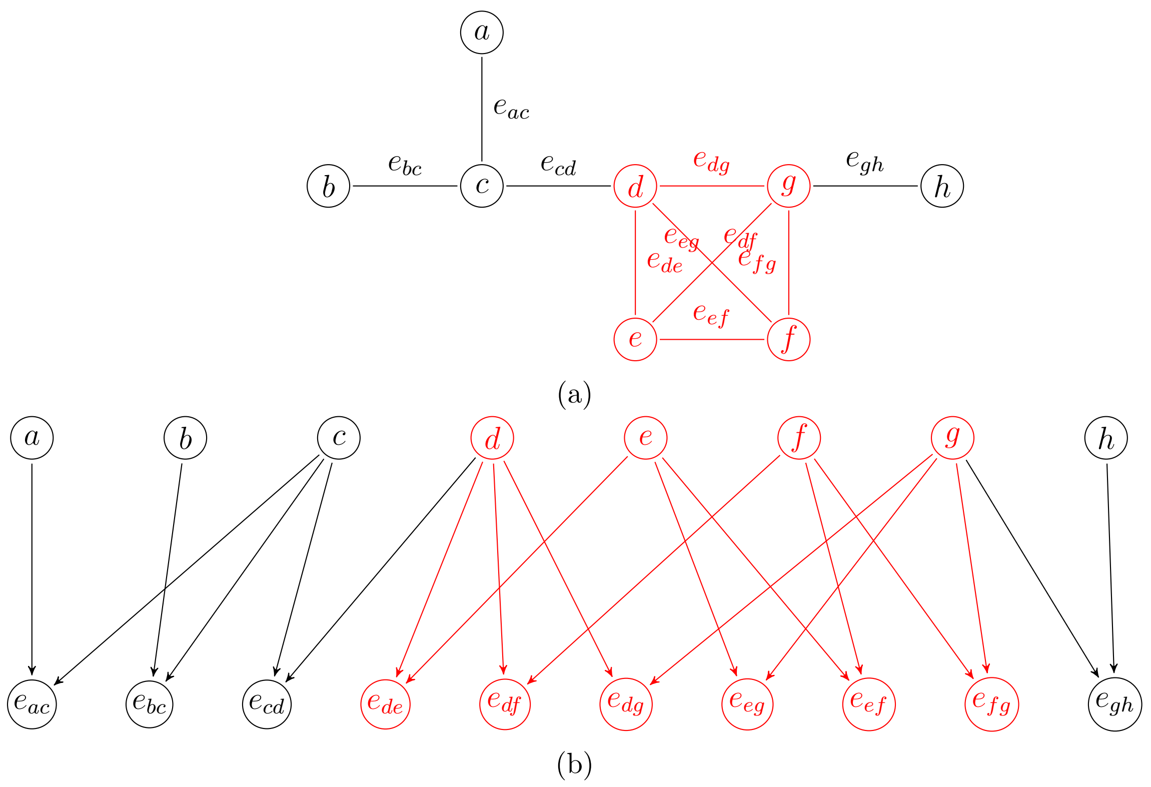


Figure 4.1: (a) An undirected graph with a clique in the vertex set $\{d, e, f, g\}$ (b) A translated directed acyclic graph according to the reduction (1. step).

**Reduction 4.** Let $I = (G, k)$ be an instance of CLIQUE where $G = (V, E)$ is an undirected graph and $k$ a positive integer.

We give the following reduction consisting of three steps.
In the first step, we create a graph $G' = (\mathcal{A}, E')$ as follows. Set $\mathcal{A}$ of agents consists of vertex agents represented by $A_v$ and edge agents represented by $A_e$ in our graph $G'$.
Then, the vertices of graph $G$ are vertices of the set $A_v$ of $G'$. Also the edges $e \in E$ of graph $G$ are vertices of the set $A_e$. For the set of arcs $E'$ it holds that for every edge $(a, b) \in E$ we have two arcs $(a, c), (b, c) \in E'$ where $a, b \in A_v \wedge c \in A_e$. An example is shown in Figure 4.1.
In the second step, we introduce the number of resources $|\mathcal{R}|$ that are value one to be $A_v| + k + \frac{1}{2}k(k-1)$.
In the third step, we set the lower bound $l$ for the value of the egalitarian welfare to zero and the upper bound $p$ for the rank to $p = |A_e| - \frac{1}{2}k(k-1)$.
After three steps we obtain an instance $I' = (\mathcal{A}, R, G', p, l)$ of ER-F sGEF-ALLOCATION.

We show that every solution to $I$ is also a solution to $I'$:
Let $S \subseteq V$ be a clique of size $k$ in $G$. We define an allocation $\pi$ where we give every vertex agent corresponding to $S$ a bundle valued two and every other vertex agent a bundle valued one. We also give every edge agent, where both connecting vertex agents have two resources in their bundle, a bundle valued one. Thus, the allocation is strongly

graph-envy-free since every vertex agent has at least one more resource than the edge agent the vertex agent is connect to. Firstly, no vertex agent envies edge agents who have an empty bundle, because every vertex agent has at least one resource in their bundle. For every edge agent who has a resource, their corresponding vertex agent $A_c \subseteq A_v$ does not envy it, because every vertex agent in $A_c$ has two resources. The edge agents with a resource are the edges of the clique. The reason is that the vertex agents in $A_c$ are the $k$ vertex agents who build the clique in the graph $G$.
The egalitarian welfare of this allocation is zero and the the rank is $|A_e| - \frac{1}{2}k(k-1)$. The egalitarian welfare cannot be higher, since at least $2|A_v| + |A_e|$ resources are needed for a egalitarian welfare of one. As a clique of size $k$ is a graph with the most edges which are $\frac{1}{2}k(k-1)$. Therefore, the rank is reduced as much as possible. Thus, the rank is optimal.

We show that every solution to $I'$ is also a solution to $I$:
Suppose there is an allocation $\pi$ that solves ER-F sGEF-Allocation with the input graph being a DAG and agents having identical 0/1 preferences. The number of given resources is $|A_v| + k + \frac{1}{2}k(k-1)$ where $A_v$ is the set of vertex agents and $k$ the size of the clique. The lower bound $l$ for the egalitarian welfare is zero, since at least $2|A_v| + |A_e|$ resources are needed for an egalitarian welfare of one. The upper bound $p$ for the rank is $|E| - \frac{1}{2}k(k-1)$. One resource must be assigned to each agent $a \in A_v$. For the remaining $k + \frac{1}{2}k(k-1)$ resources we must assign $k$ resources to agents in $A_v$ and $\frac{1}{2}k(k-1)$ to agents in $A_e$. Thus, the rank is reduced by $\frac{1}{2}k(k-1)$. The reason for that being the correct solution is that $A_e$ represents the edges in $G$ and $A_v$ the vertices in $G$. A graph of size $k$ can have at most $\frac{1}{2}k(k-1)$ edges. The number $\frac{1}{2}k(k-1)$ is due to the fact that in a clique each pair of vertices shares an edge, so there is an edge for every two vertices of $k$. So the number of edges needed in a clique of size $k$ is $\binom{k}{2}$ which equals $\frac{1}{2}k(k-1)$. This is only the case for a clique of size $k$. Thus, to reduce the rank by $\frac{1}{2}k(k-1)$ a clique of size $k$ has to exist in $G$.
Thus, every vertex agent $a \in A_v$ with two resources, which are $k$, represents a vertex of a clique in $G$ and every edge agent $a \in A_e$ with one resource, which are $\frac{1}{2}k(k-1)$, represents an edge of the clique in $G$. Thus the $\frac{1}{2}k(k-1)$ edge agents and the $k$ node-agents build the clique with the size $k$.
The reduction is executed in polynomial time and hence our problem ER-F sGEF-Allocation with the input graph being a DAG and agents having identical 0/1 preferences is NP-hard.

□

## 4.2 Allocations With Respect to Half-Feedback Envy-Freeness

In all the following subsections we discuss the computational complexity of Half-Feedback Envy-Freeness (s)GEF-Allocation. This is the only section, where the case of finding a strongly graph-envy-free allocation with the addition to our fairness concept for agents having identical 0/1 preferences is left open. The reasons for that are explained in Chapter 5.

### 4.2.1 Identical 0/1 Preferences on a Tree & DAG

As in Section 3.1 we start again with agents having identical 0/1 preferences, since it is the most restricted preference relation we use in this thesis. Every agent $a \in \mathcal{A}$ has the same binary utility function $u\colon \mathcal{R} \to \{0,1\}$ as in Section 3.1.1.

#### Weak Graph-Envy-Freeness on a Tree & DAG

The same Algorithm 4 as in Section 4.1.1 can be used to solve Half-Feedback Envy-Freeness GEF-Allocation. This is due to the fact, that the maximum value difference resulting from the Algorithm 4 is one. One divided by two and rounded down is zero. Thus the allocation resulting from Algorithm 4 fulfills the criteria for half-feedback envy-freeness.

### 4.2.2 0/1 Preferences in a Tree & DAG

In the following we study our problem HF-EF GEF-Allocation for agents having 0/1 preferences. Every agent $a \in \mathcal{A}$ has its own utility function $u_a\colon \mathcal{R} \to \{0,1\}$. If a resource $r \in \mathcal{R}$ is evaluated zero by every agent $a \in \mathcal{A}$ the resource can be allocated arbitrarily. Thus, we only focus on non-trivial resources.

#### Weak Graph-Envy-Freeness on a Tree & DAG

The goal of this subsection is to show that HF-EF GEF-Allocation with the input graph being a DAG and agents having 0/1 preferences is polynomial-time solvable. In order to find a half-feedback envy-free and weak graph-envy-free allocation we provide Algorithm 6.
The idea of this algorithm is that the agents are assigned one resource after another. The order depends on the distance to the source. The procedure starts with the agents with shortest distance, in our case the sources, and continues on to the agents with the longest distance. Then the procedure starts again with the agent with shortest distance. This continues until all resources are assigned.

**Algorithm 6:**

```
Input:
$\mathcal{R}$ — a set of non-trivial resources
$\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ — a set of agents
$U = \{u_1, u_2, \ldots, u_n\}$ — a family of agents' utility functions for resources
$G = (\mathcal{A}, E)$ — a directed tree
1 for $i \leftarrow 1$ to $|\mathcal{A}|$ by 1 do
2     aDist$_i$ ←source_dist$(G, a_i)$; ▷ source_dist is a function to determine
        the longest distance to the source
3 For each $A_j = \{a_i \mid \text{aDist}_i = j\}$; ▷ group the agents by their distance to
    the source agent
4 while there exist unassigned resources do
5     for $i \leftarrow 1$ to max(aDist$_i$) by 1;      ▷ Loop through the sets of agents
6      do
7         for $j \leftarrow 1$ to $A_i$ by 1;       ▷ Loop through the agents in the set
8          do
9             Assign a liked and unassigned resource $r_j$ to ever agent $a \in$ aDist$_i$;
```

**Theorem 4.4.** *Algorithm 6 solves* HF-EF GEF-Allocation *even with the input graph being a DAG and agents having 0/1 preferences in $O(|\mathcal{A}||\mathcal{R}|)$ time, where $\mathcal{A}$ denotes the set of agents and $\mathcal{R}$ denotes the set of resources.*

*Proof.* We proof Theorem 3.10 by induction.
The base case we choose is for $\mathcal{R} = \emptyset$. With $\mathcal{R} = \emptyset$ the allocation that is complete, weakly graph-envy-free and also half-feedback envy-free is the allocation that allocates nothing to every agent. Since the set of resources is empty and no resources are left to allocate the allocation is complete. No agent envies another agent since every agent has no resource. For the same reason the allocation is also half-feedback envy-free. The algorithm with $\mathcal{R} = \emptyset$ does not go in the while loop in line 4. Thus, the empty allocation is obtained by our algorithm. Hence, it works correctly for the base case.
Our inductive assumption is that Algorithm 6 finds a complete, weakly graph-envy-free and half-feedback envy-free allocation with $k$ iterations. In each iteration a single resource is assigned to an agent. The allocation that is found also has a maximum value difference of one between bundles of $a_i$ and of $a_{i+1}$ from the viewpoint of $a_{i+1}$ where $1 \leq i \leq n-1$.
The allocation is complete since the allocation obtained with $k$ iterations is complete and then in the $(k+1)$th iteration the $(k+1)$th resource is allocated. The reason is that in each iteration a single resource is assigned to an agent. The allocation with $k$ iterations is weakly graph-envy-free. We assume that after the additional iteration the allocation is not weakly graph-envy-free. For this to happen, two adjacent agents $a \in A_n$ and $b \in A_m$ where $(a, b) \in E \land n < m$ need to have the same value for their bundle and $b$ must receive the next resource in the $(k+1)$th iteration. This can never happen, since

agent $a$ receives a liked resource in the $k$th iteration such that $a$ has a bundle valued one more than $b$. If the agent $b$ receives a resource in the $(k+1)$th iteration they have the same value for their bundle.
Lastly the allocation is also half-feedback envy-free since the allocation with $k$ iterations is half-feedback envy-free and the value difference is at most one, from the viewpoint of $b$. After the $(k+1)$th iteration the value difference is zero and the value difference between $b$ and its peers is at most one, and one divided by two and then rounded down is zero. Hence, the allocation stays half-feedback envy-free.
Thus, the allocation obtained by Algorithm 6 is a solution to HF-EF GEF-Allocation with the input graph being a DAG and agents having 0/1 preferences.
The algorithm also works for DAGs with multiple connected components. This is due to the fact that the order of assigning the resources is only depending on the longest possible distance to a source. Thus, there is no problem, if the input graph is a DAG consisting of multiple connected components. The reason for the runtime being $O(|\mathcal{A}||\mathcal{R}|)$ is that in the worst case only the last agent in the order likes the resources $r \in \mathcal{R}$. Every other agent values every resource as zero. Thus for every single resource we have to go through the whole set of agents resulting in $O(|\mathcal{A}||\mathcal{R}|)$. $\square$

Since Algorithm 6 solves HF-EF GEF-Allocation with the input graph being a DAG and agents having 0/1 preferences in $O(|\mathcal{A}||\mathcal{R}|)$ time it also solves the problem for a tree since a tree is also a DAG.

# 5 Conclusion

This thesis takes up on a call for further fairness criteria posed by "Envy-Free Allocations Respecting Social Networks" [15]. In the paper [15] it is mentioned that including further fairness concepts to envy-freeness appears promising. The fairness concepts we introduced and studied are egalitarian rank fairness and half-feedback envy-freeness. This takes me to the overview of the results of this thesis.
One observation we made is that for both of our criteria egalitarian rank fairness and half-feedback envy-freeness every time agents have additive preferences the problem of finding a fair allocation, for both variants of graph-envy-freeness, is NP-hard.
Another result is that finding an egalitarian rank fair and strong graph-envy-free allocation is always at least as hard as finding an egalitarian rank fair and weak graph-envy-free allocation. The same holds for half-feedback envy-freeness.
From the paper "Envy-Free Allocations Respecting Social Networks" [15] we know that finding a strong graph-envy-free allocation is NP-hard. Thus it can be inferred that for our case of finding a half-feedback envy-free and strong graph-envy-free allocation or an egalitarian rank fair and strong graph-envy-free allocation is also at least NP-hard.
One more insight is that when agents having identical 0/1 preferences the same algorithm can be used to find an allocation that is weak graph-envy-free and either egalitarian rank fair or half-feedback envy-free.
Another finding is that egalitarian rank fairness has a strict bound and half-feedback envy-freeness only requires that for every agent $a$ every outneighbor has a bundle valued at most the same and at least the half the value of $a$ his bundle. Thus, proving NP-hardness for egalitarian rank fairness can be easier than for half-feedback envy-freeness, this is because for half-feedback envy-freeness it is harder to force a specific allocation. This is the case when agents have identical 0/1 preferences, the input graph is a DAG and the allocations has to be strong graph-envy-free. There, we have a polynomial-time many-one reduction from Clique for egalitarian rank fairness but we only conjecture that the problem is NP-hard for half-feedback envy-freeness.
Finally the problem of finding a half-feedback envy-free and weak graph-envy-free allocation is easier to solve when agents have 0/1 preferences than for egalitarian rank fairness. However finding a half-feedback envy-free and strong graph-envy-free allocation with agents having identical 0/1 preferences has more possible problematic situation to check than for egalitarian rank fairness. For half half-feedback envy-freeness we have two situation, where we have enough resources for a strong graph-envy-free allocation, but there exists no allocation that is strong graph-envy-free and half-feedback envy-free. The results of this thesis are depicted in Tables 5.1 and 5.2.
In the next paragraph we explain the cases which remained open and the reasons for that.

| **Weak Graph-Envy-Freeness** | | | |
|---|---|---|---|
| Graph Class / Preferences | Paths | Trees | DAGs |
| identical 0/1 | $O(\lvert\mathcal{A}\rvert)$(Th.3.2) | $O(\lvert\mathcal{A}\rvert)$(Thm.4.1) | $O(\lvert\mathcal{A}\rvert)$(Thm.4.1) |
| identical | NP-h(Thm.3.4) | NP-h(Thm.3.4) | NP-h(Thm.3.4) |
| 0/1 | | | |
| additive | NP-h(Thm.3.4) | NP-h(Thm.3.4) | NP-h(Thm.3.4) |
| **Strong Graph-Envy-Freeness** | | | |
| Graph Class / Preferences | Paths | Trees | DAGs |
| identical 0/1 | $O(\lvert\mathcal{A}\rvert)$(Th.3.3) | $O(\lvert\mathcal{A}^2\rvert)$(Thm.4.2) | NP-h(Thm.4.3) |
| identical | NP-h(Thm.3.5) | NP-h(Thm.3.5) | NP-h([15]) |
| 0/1 | | | NP-h([15]) |
| additive | NP-h(Thm.3.5) | NP-h(Thm.3.5) | NP-h([15]) |

Table 5.1: Results for egalitarian rank fairness, where $\mathcal{A}$ denotes the set of agents.

| **Weak Graph-Envy-Freeness** | | | |
|---|---|---|---|
| Graph Class / Preferences | Paths | Trees | DAGs |
| identical 0/1 | $O(\lvert\mathcal{A}\rvert)$(Thm.3.2) | $O(\lvert\mathcal{A}\rvert)$(Thm.4.1) | $O(\lvert\mathcal{A}\rvert)$(Thm.4.1) |
| identical | NP-h(Thm.3.7) | NP-h(Thm.3.7) | NP-h(Thm.3.7) |
| 0/1 | $O(\lvert\mathcal{A}\rvert\lvert\mathcal{R}\rvert)$(Thm.3.10) | $O(\lvert\mathcal{A}\rvert\lvert\mathcal{R}\rvert)$(Thm.4.4) | $O(\lvert\mathcal{A}\rvert\lvert\mathcal{R}\rvert)$(Thm.4.4) |
| additive | NP-h(Thm.3.7) | NP-h(Thm.3.7) | NP-h(Thm.3.7) |
| **Strong Graph-Envy-Freeness** | | | |
| Graph Class / Preferences | Paths | Trees | DAGs |
| identical 0/1 | $O(\lvert\mathcal{A}\rvert)$(Thm.3.6) | | |
| identical | NP-h(Thm.3.8) | NP-h(Thm.3.8) | NP-h([15]) |
| 0/1 | | | NP-h([15]) |
| additive | NP-h(Thm.3.8) | NP-h(Thm.3.8) | NP-h([15]) |

Table 5.2: Results for half-feedback envy-freeness, where $\mathcal{A}$ denotes the set of agents and $\mathcal{R}$ denotes the set of resources.

Agents have: additive preferences when ever agent $a$ has a utilitiy function $u_a\colon \mathcal{R} \to \mathbb{N}$,
identical preferences when ever agent $a$ has the same utilitiy function $u\colon \mathcal{R} \to \mathbb{N}$,
0/1 preferences when ever agent $a$ has a utilitiy function $u_a\colon \mathcal{R} \to \{0, 1\}$,
identical 0/1 preferences when ever agent $a$ has the same utilitiy function $u\colon \mathcal{R} \to \{0, 1\}$.

## Open Cases

We start with explaining why some of the problems when agents have 0/1 preferences remained open. The main obstacles to find the allocation already start with the input graph being a path. The justification for leaving the problem of finding an egalitarian rank fair and weak graph-envy-free allocation with agents having 0/1 preferences open is analyzed below. The problem is how to maximize the egalitarian welfare. One idea was to allocate one liked resource at a time starting by the source-agent. However this can lead to a situation where one of the later agents receives nothing in his iteration, because every resource he liked was already allocated. This is the case even if this could have be prevented by allocating a different resource to an earlier agent. The idea then was start allocating the resources starting with the agents that only like a few resources, in that case it is hard to ensure graph-envy-freeness.

For finding a half-feedback envy-free and strong graph-envy-free allocation, with agents having 0/1 preferences, the problem is to find the minimal strong graph-envy-free allocation. This is necessary, to decide whether a strong graph-envy-free allocation even exists for a specific set of resources. There can be a minimal strong graph-envy-free allocation for $n$ agents with only $n-1$ resources but only if all adjacent agents do not like the same resource. However even for larger sets of resource a strong graph-envy-free allocation does not always exist, for example if one agent does not like any resource and has an outneighbor, then there does not exist a strong graph-envy-free allocation. Therefore it is already hard to decide whether a strong graph-envy-free allocation exists. For finding an egalitarian rank fair and strong graph-envy-free allocation with agents having 0/1 preferences we have to consider both problems mentioned above.

The next open case is for finding a half-feedback envy-free and strong graph-envy-free allocation with the input graph being a tree and agents having identical 0/1 preferences. The problems occur, when there are more resources than needed for the minimal allocation. Then there can be pairs of adjacent agents $(a, b)$ where agent $a$ has a bundle valued exactly double the bundles value of its outneighbor $b$ after the minimal allocation. Then the problem is, since the allocation has to be complete, how to allocated the resources, so that the allocation stays half-feedback envy-free and strong graph-envy-free.

The last open case is for half-feedback envy-freeness with agents having identical 0/1 preferences and the input graph being a DAG regarding strong graph-envy-freeness. We conjecture that the problem is NP-hard. The problem for the reductions we tried were that half-feedback envy-freeness does not require strict boundaries.

## Outlook

First, the open cases can be further analyzed.
Second, the studied preference relations can be to extend. One possible extension is for the agents to have a utility function that maps to zero, one or two. The preference relations can be extended, until we find a rule that describes for which preference relations the problem is polynomial-time solvable and when it is NP-hard.
Finally, further graph classes can be studied, for example directed funnels and directed planar graphs. These can be interesting in the case where agents have identical 0/1 preferences and the allocation has to be strong graph-envy-free. This is the case, since for egalitarian rank fairness the problem is NP-hard with a Dag as an input graph and still polynomial-time solvable with the input graph being a tree.

# Literature

[1] K. J. Arrow, *Social choice and individual values.* Yale university press, 2012, vol. 12 (cit. on p. 7).

[2] M. P. Wellman, “The economic approach to artificial intelligence”, *ACM Computing Surveys*, vol. 27, no. 3, pp. 360–362, 1995 (cit. on p. 7).

[3] W. L. Winston and J. B. Goldberg, *Operations research: applications and algorithms.* Duxbury Press, Belmont, California, 2004, vol. 3 (cit. on p. 7).

[4] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaitre, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar, and P. Sousa, “Issues in multiagent resource allocation”, *Informatica*, vol. 30, no. 1, 2006 (cit. on p. 7).

[5] J. Y. Bakos, “A strategic analysis of electronic marketplaces.”, *MIS quarterly*, vol. 15, no. 3, 1991 (cit. on p. 7).

[6] G. Jonker, J.-J. Meyer, and F. Dignum, “Towards a market mechanism for airport traffic control”, in *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA’05)*, Springer, 2005, pp. 500–511 (cit. on p. 7).

[7] E. Cantillon and M. Pesendorfer, “Auctioning bus routes: The London experience”, 2006 (cit. on p. 7).

[8] R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode, “Selfish routing in non-cooperative networks: A survey”, in *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS’03)*, Springer, 2003, pp. 21–45 (cit. on p. 7).

[9] A. Giovannucci, J. A. Rodriguez-Aguilar, A Reyes, F. X. Noria, and J. Cerquides, “Towards automated procurement via agent-aware negotiation support”, in *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS’04)*, IEEE Computer Society, 2004, pp. 244–251 (cit. on p. 7).

[10] M. Aleksandrov, H. Aziz, S. Gaspers, and T. Walsh, “Online Fair Division: Analysing a Food Bank Problem”, in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI’15)*, 2015, pp. 2540–2546 (cit. on p. 7).

[11] M. Lemaître, G. Verfaillie, and N. Bataille, “Exploiting a common property resource under a fairness constraint: A case study”, in *Proceedings of the 16th International Joint Conference on Artifical Intelligence (IJCAI’99)*, Morgan Kaufmann Publishers Inc., 1999, pp. 206–211 (cit. on p. 7).

[12] M. Lemaître, G. Verfaillie, H. Fargier, J. Lang, N. Bataille, and J.-M. Lachiver, "Equitable allocation of earth observing satellites resources", in *Proceedings of the 5th ONERA-DLR Aerospace Symposium (ODAS'03)*, 2003 (cit. on p. 7).

[13] Y. Chevaleyre, U. Endriss, N. Maudet, *et al.*, *Allocating goods on a graph to eliminate envy.* Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 2007 (cit. on p. 7).

[14] H. Steinhaus, "The problem of fair division", *Econometrica*, vol. 16, no. 1, pp. 101–104, 1948 (cit. on p. 7).

[15] R. Bredereck, A. Kaczmarczyk, and R. Niedermeier, "Envy-Free Allocations Respecting Social Networks", in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'18)*, International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 283–291 (cit. on pp. 8, 12, 13, 15, 17, 20, 45, 46).

[16] S. Bouveret and J. Lang, "Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity", *Journal of Artificial Intelligence Research*, vol. 32, pp. 525–564, 2008 (cit. on pp. 12, 15).

[17] S. Bouveret, K. Cechlárová, E. Elkind, A. Igarashi, and D. Peters, "Fair division of a graph", in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, 2017, pp. 135–290 (cit. on pp. 12, 15).

[18] R. Abebe, J. Kleinberg, and D. C. Parkes, "Fair division via social comparison", in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS'17)*, International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 281–289 (cit. on pp. 12, 15).

[19] X. Bei, Y. Qiao, and S. Zhang, "Networked fairness in cake cutting", in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, 2017, pp. 3632–3638 (cit. on pp. 12, 15).

[20] H. Aziz, S. Bouveret, I. Caragiannis, I. Giagkousi, and J. Lang, "Knowledge, fairness, and social constraints", in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, 2018 (cit. on pp. 12, 15).

[21] Y. Chevaleyre, U. Endriss, and N. Maudet, "Distributed fair allocation of indivisible goods", *Artificial Intelligence*, vol. 242, pp. 1–22, 2017 (cit. on pp. 12, 15).

[22] Y. Chen and N. Shah, "Ignorance is often bliss: Envy with incomplete information", Working paper, Harvard University, Tech. Rep., 2017 (cit. on p. 12).

[23] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, *Handbook of computational social choice.* Cambridge University Press, 2016 (cit. on p. 15).

[24] R. M. Karp, "Reducibility among combinatorial problems", *Complexity of Computer Computations*, pp. 85–103, 1972 (cit. on p. 20).

[25] G. Cho and D. X. Shaw, "A depth-first dynamic programming algorithm for the tree knapsack problem", *Journal on Computing*, vol. 9, no. 4, pp. 431–438, 1997 (cit. on pp. 38, 39).