# Complexity of Locally Fair Allocations on Graphs

**MSc Thesis** *(Afstudeerscriptie)*

written by

**Giovanni Varricchione**
(born September 17th, 1996 in Benevento, Italy)

under the supervision of **Dr. Ronald de Haan** and **Drs. Simon Rey**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

**MSc in Logic**

at the *Universiteit van Amsterdam*.

| **Date of the public defense:** | **Members of the Thesis Committee:** |
|---|---|
| *April 29th, 2021* | Dr. Benno van den Berg (*chair*) |
| | Prof. dr. Ulle Endriss |
| | Dr. Davide Grossi |
| | Dr. Ronald de Haan |
| | Drs. Simon Rey |

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

**Abstract**

We study a new model for fair division in which agents, aside from getting allocated a bundle of items (that can include both goods and chores), have also to be assigned a position on a social graph. Hence, natural fairness criteria arise that are not global, as in the classical setting, but local. In particular, we consider three fairness criteria in their local variants: local envy-freeness, local envy-freeness up to one item and local proportionality. For these, we study how complex the task of performing a position assignment (and possibly also an item allocation) which respects a given criterion is. Specifically, we focus on how different graph topologies influence the complexity of these problems. With the exception of local envy-freeness up to one item which, thanks to known facts from the literature, proves to be easily tractable when the item allocation is not fixed, in many cases both tasks prove to be intractable for the criteria we consider. We also provide some parameterized complexity results when only the position assignment has to be done. In particular, we prove tractability, for all criteria but local proportionality, in case the social graph is either a tree or a forest and using two parameters, one of which specifically tailored for these graphs.

Finally, we comment the experimental results we have obtained from randomly generated instances, each with a fixed item allocation done using one of various criteria. The experiments' main objective is to measure how different parameters influence the likelihood of instances for which there is a position assignment that respects a given fairness criterion. In particular, local envy-freeness up to one item proves to be the criterion more easily satisfied when the item allocation is random, whereas, in most cases, local proportionality is more likely to be satisfied than local envy-freeness.

**Acknowledgments**

First of all, I would like to thank Ronald and Simon for their help in coming up with the thesis topic and, overall, for their constant support throughout our work. Ronald, ever since we had our first meeting, you have always given me very useful feedback and ideas on how to approach my studies. Your profound knowledge in the topics I have studied with you has deeply motivated me in these years. Simon, your intuitions and suggestions during our meetings always gave me new insights on the problems we were facing, and showed me new ways to tackle them. In particular, I wanted to thank you both for your constant readiness in answering my questions and for all the meetings we had (and the fun I had during them!), which have made working on the thesis much more manageable. Aside from them, I also thank Benno, Davide and Ulle for their availability in joining the Thesis Committee.

I would like to thank my parents, without whom nothing of this would have ever been possible. Your unrelenting support has and will always inspire me in my life. To you I dedicate this thesis.

I think that, out of all the people with whom I have spent these two years, Max deserves a special mention. Mainly thanks to you I made up my mind and decided to move to Amsterdam, and I will never forget all the time we have passed together before and after we have moved. You already know what you mean to me, and I hope you will forgive me if I do not repeat myself here.

Damiano, the conversations I had with you have been the most engaging I had during the MoL, and in you I have found another companion with whom to share achievements and failures without regrets. Not only that, but you were also perseverant enough to convince me to start reading *One Piece*, which you should now know grants you many points.

I also wanted to thank Adam, Koen, Matteo and Pietro for the countless *DnD* nights, and Quinzia for all the times we have watched *Propaganda Live* together. Without you quarantine would have been an awful period.

Finally, my last thanks go to Giulia, who has always been by my side since many years and has always had enough time to comfort me or simply have a lighthearted chat. I hope you never lose your cheerfulness and kindness.

To all those who I did not mention but hold a place dear in my heart: you know who you are. Thanks for being there for me throughout these years.

# Contents

# Chapter 1

# Introduction

Imagine a scenario in which a company manager has at her disposal a set of assignments for the company's employees and, by the company's rules, must assign each of them to some (unique) worker. To compensate for these chores, the manager has also a set of rewards, which need not be financial ones (Maqsood et al. [2015], Bredereck, Kaczmarczyk, and Niedermeier [2018]). It should be noted that, regardless of the items' nature, a natural assumption is that two different employees might have different opinions about each of the available items, considering each of them either a good or a chore. For example, an employee might be extremely interested in being assigned some task because of its importance, thus evaluating it positively, while another might feel overwhelmed by it, thus evaluating it negatively. Another important aspect to notice in the setting we have just given is that all these items cannot be divided, i.e. they are so-called "*indivisible items*". At this point, a sensible idea would be to define some criteria which the manager can use to define the assignment, so that in the end (hopefully) each employee does not feel cheated.

In fair division, given a set of agents and a set of items, the goal is to assign items to the agents in such a way that certain fairness criteria are met. Assume that a survey has been held at the aforementioned company: each employee (agent) has been asked what she thinks of each item in a given set. The manager now has enough information to come up with an item assignment which, hopefully, respects one (or more) given criterion (criteria). An easy example one might come up with for such a criterion is that we want all our employees not to envy anyone in the company, as this can lead to rather unpleasant situations and unmotivated workers, where with *envying* we mean that one thinks that somebody else received a *strictly* better bundle of items. On the other hand, the manager might also be satisfied with assigning the items in such a way that, if an employee envies another one, then, if we were to remove some item from either the bundle assigned to the first or to the second, this envy disappears: it might not be an optimal solution, but one which is adequate enough. As a final example, granting to each agent a bundle of items so that each one feels like they have received a "fair" share of the whole set of items, granting a sort of *egalitarian* distribution (at least from the point of view of each single agent), could also suffice. All these examples refer to criteria which have been studied in fair division, respectively "*envy-freeness*", "*envy-freeness up to one item*" and "*proportionality*". In the thesis, we will be concerned with exactly these criteria.

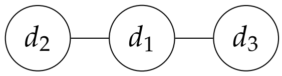
At this point, a natural question might arise: in a natural setting, does someone care about everybody else? Usually not: a tangible example is the fact that often company offices are distributed over different buildings/floors, and each worker usually has limited information about what happens in the company outside their workplace. Moreover, would a developer care about the bundle given to a marketing agent? Social scientists, starting from the seminal paper Festinger [1954], have since long time formulated a theory by which people tend to mostly compare themselves (and their possessions) to their peers. This theory has seen applications in many fields, including fair division itself (Bredereck, Kaczmarczyk, and Niedermeier [2018]).

Therefore, it makes sense that we focus more on variants of the fairness criteria previously defined which take into account this lack of knowledge (or lack of interest) from the agents' point of view. This last aspect can be easily modeled with a *graph*, a structure where agents are represented as vertices of the graph and two vertices are connected if say, for example, their two corresponding agents sit at desks close to each other, meaning that both "know" what bundle of items was assigned to the other. For the sake of the example, we will now consider exclusively local envy-freeness. Given this "local" setting, an agent can only be envious of those who she "sees" in the graph, and not of any other arbitrary agent. Notice how this can lead to situations in which we have a *local* envy-free assignment while a *global* one is impossible:

| | Milk | Wine | Beer |
|---|---|---|---|
| Armin | 1 | 0 | 0 |
| Pieck | 0 | 0 | 1 |
| Reiner | 0 | 0 | 1 |

Each row corresponds to an agent and each column to an item: a 1 signals that the agent likes the item and a 0 the opposite. As one can see, there is no item allocation which is envy-free: if we give Armin some milk, then one between Pieck and Reiner will be bound to envy the other, as only wine and beer are left to be given. On the other hand, if we were to give milk to either Pieck or Reiner, then Armin will envy the one who received it.

However, imagine that we also have to place all of them on the following graph, representing the way the office desks are placed and which desks can be seen from some particular desk:




Now, recall the first way we could have assigned the available goods: Armin gets the milk and, for the example's sake, Pieck a beer and Reiner a glass of wine. Again, from a *global* point of view we are doomed: Reiner will necessarily envy Pieck. However, assume we assign Armin to desk $d_1$, Pieck to desk $d_2$ and Reiner to desk $d_3$: then Reiner will only be able to know what his only deskmate has received, but knows nothing about who has received the beer. One could argue that, in the end, Reiner does indeed know that someone must have received a good which he prefers to his own (as by the company's rules), however there is no tangible person close to him to envy. Thus, by combining this item allocation and the desk assignment, we have obtained a *locally* envy-free assignment for the given instance.

In the thesis, we will focus on two different problems which are naturally implied by the setting. In one, the task will be to answer whether there are an item allocation and a position assignment which respect the given criterion. In the other, aside from the usual instance, also the item allocation will be fixed, and the task will be to check whether there is a position assignment which, combined with the given item allocation, satisfies the criterion. When computational complexity results will allow us, we will also briefly discuss the "search" variants of these problems, i.e. where the task is to find the position assignment or the item allocation and position assignment pair which satisfies the criterion.

A big issue which emerges when working with fair division is the computational complexity of the task. As an example, in the "classic" fair division setting, i.e. without an underlying social graph on which agents are placed, with cardinal utilities, deciding whether there is an envy-free (Lipton et al. [2004]) or proportional (Bouveret and Lemaître [2016]) item assignment are both NP-hard problems. In particular, for envy-freeness, intractability already arises when considering just two agents with additive and identical utilities, which, intuitively, should be amongst the simplest settings for the problem. On the other hand, envy-freeness up to one item represents a nice case, as an assignment which respects it can always be computed in polynomial time (Lipton et al. [2004], Aziz, Caragiannis, et al. [2019]).

Although the situation might seem grim to the reader, we do have a couple of tricks up our sleeves which can mitigate these issues. In fact, though our setting comes with an extra difficulty, i.e. the position assignment on the graph, the graph itself will also play a vital role in the complexity of the task. Thus, different graph topologies might lead to a decrease in complexity. Another direction we will explore will be that of considering different parameterizations of the problems, i.e. to observe how certain variables (which will be called "*parameters*") of the instances affect the computational complexity of the problem. If we are lucky, it could turn out that the problems' high complexity might be due to parameters which in real-life applications can be neglected.

Based on all the observations done so far, we have also decided to carry out some experiments to observe how different variables (e.g. the number of agents, the graph topology, the number of items, etc.) affect the likelihood, for each of the criteria we have defined, of positive instances, i.e. instances in which it is possible to define a position assignment which, coupled with the fixed item allocation, satisfy the given criterion.

## 1.1 Our Contribution

We explore a new setting for fair division which does not only account for the distribution of items but also for positioning agents on a social network. Though it was briefly discussed in the literature, we study it deeper and come out with some interesting findings, which also open up new directions of research. We consider two different problems in their existential and search variants, one in which only the position assignment has to be done and one in which both the position assignment and the item allocation have to be performed. We focus on three different fairness criteria (in their local variants to be precise): envy-freeness, envy-freeness up to one item and proportionality. For all problems, we study how the social graph itself affects the (parameterized) computational complexity of the problem itself.

For the first problem we have mentioned, there are both good and bad news. Regardless of the fairness criterion we are considering, it turns out that both stars and matchings prove to be graphs on which such a task is easy to perform, as it can be done in polynomial time. On the other hand, on lines (and subsequently strongly connected graphs) these problems are all NP-complete, which is intuitively due to the difficulty of finding a valid order for the agents. If we consider only local envy-freeness and local envy-freeness up to one item, then it can also be shown that, for two parameters which in some particular instances are reasonably low, these problems are fixed parameter tractable (FPT) in case the social graph is either a tree or a forest. Moreover, for graphs that have a constant treewidth (intuitively graphs that look very similar to trees or forests), these same problems are FPT as well when parameterized by the number of non-isolated vertices in the graph. Notice that all these positive results (except the one for graphs with constant treewidth) hold also in case one has to find such a position assignment. Instead, a negative result is the fact that in graphs with a non-constant treewidth these problems are again intractable, so much as they become para-NP-complete when using again the number of non-isolated vertices as the parameter. Unfortunately, none of these results applies to local proportionality.

On the other hand, if the item allocation is not fixed, the problem we introduce turns out to be intractable already in its existential variant. If one looks just at local envy-freeness, this could already be guessed from the fact that it is known in the literature that, given a fixed position assignment, finding an item allocation that is local envy-free is intractable. In fact, on all graphs that are either lines, stars, matchings or strongly connected ones, this problem is NP-complete. Moreover, these same results apply to the local proportionality as well. As a consequence of these results, it follows that the search problems are also at least NP-hard in all the cases we have just mentioned, since the search variant of any problem is at least as difficult as its existential variant. On the other hand, as it is always possible to find an item allocation which is envy-free up to one item, the existential and search problems can both be solved in polynomial time for this local criterion.

We have also performed some experiments to study how various parameters influence the likelihood of positive instances in random ones. In this case, instances come with a fixed set of agents, a set of items, a utility profile, an item allocation, a social graph, and a fairness criterion (amongst the ones we are interested in), and an instance is "*positive*" in case there is a position assignment that satisfies the given criteria with the given item allocation. The main factor which increases this likelihood is the number of items per agent, though when items are given randomly it might happen the opposite. This is most likely due to the fact that the number of item assignments is much higher, hence it is, in general, more likely that there is a way to assign the items that satisfies the global variant of the given criterion. We have also noticed how the presence of *hubs* in graphs (vertices that are connected to many other vertices) generally decreases this likelihood. As one might expect, the probability that an instance is positive if the items are given randomly is often very low. In this sense, the best results we had were when we focused on local envy-freeness up to one item, in which case the highest likelihood was approximately 70%. We have also studied how the number of envy free agent-types, a notion which we will later introduce to obtain our parameterized results, influences this likelihood, and have noted that the higher this parameter is, the lower the likelihood is. The graphs on which we have obtained the best results in these experiments were those that are *sparser* (i.e. with few edges) and that did not contain hubs.

## 1.2 Thesis overview

We will now give a quick outline of the thesis chapters.

Chapter 2 introduces the reader to needed notions in fair division and computional complexity (both in the classical and in the parameterized case). In particular, we introduce the fairness criteria which are analysed later: envy-freeness, envy-freeness up to one item and proportionality, and their local variants. We introduce basic concepts of classical and parameterized computational complexity, delving a bit deeper in the latter. For both we introduce their respective reductions and also give a list of the recurring problems which we will use to define reductions in later proofs.

Following, from Chapter 3 to Chapter 5 we study the complexity of the problems we are interested in, where each chapter focuses on one of the fairness criteria that we will introduce. Each chapter is divided in sections based on the structure of the underlying social graph. For a detailed overview of the results presented in these chapters we refer to the previous section.

Chapter 6 instead covers the experimental side of the thesis. In it, instead of focusing just on the complexity of the problems, we try also to analyse how often positive instances (i.e. instances in which there are an item allocation and a position assignment which satisfy one of the criteria we are interested in) arise in random ones. We also consider different classes of graphs, just as in the previous chapters, different kinds of utility functions and different ways of assigning items. For the latter, we consider, for example, assignments which maximize utilitarian welfare or which assign items randomly. Moreover, we will also comment on how the number of envy-free agent-types, a notion which we will introduce later on in the theoretical results, affects this ratio.

Conclusions, directions for future research and comments on what has been achieved in the thesis are finally discussed in Chapter 7.

## 1.3 Related work

Fair division has its roots in economic theory, first theorized by Steinhaus [1948] in collaboration with Knaster and Banach. The literature in fair division is florid: for the interested reader we refer to the books Brams and Taylor [1996], Moulin [2003], Lindner [2016], Lang [2016] and to Robertson and Webb [1998] for a more algorithmic approach, and to the surveys Thomson [2016], Procaccia [2013, 2016], Bouveret, Chevaleyre, and Maudet [2016], Markakis [2017], Aziz [2020] and Walsh [2020] for a more computer science-oriented point of view.

The fairness criteria we will analyse, envy-freeness, proportionality and envy-freeness up to one item have been respectively introduced in Foley [1967], Steinhaus [1948] and Budish [2011]. We refer also to Aziz and Rey [2020] for a summary on the logical relationships of these criteria, including others which we will not discuss in this thesis[1].

Starting with Lipton et al. [2004], fair division saw also developments in the field of computational complexity. Major results from Lipton et al. include the already cited NP-completeness of checking whether there is an envy-free item allocation, but also an approximation one which shows that such problem cannot be approximated by a reasonably low factor. As cited before, there have been developments in these years also about the other fairness criteria. Checking the existence of a proportional assignment has also been shown to be NP-hard, whereas envy-freeness up to one item has seen more positive results. Adding on the one we have mentioned before, Caragiannis et al. [2019] have also shown that finding an item allocation which is envy-free up to one item and Pareto efficient can be done, under the assumption that utilities are positive and additive, through a procedure that maximizes the product of utilities (also called "*Maximum Nash Welfare algorithm*"). Unfortunately, computing such an assignment has been shown to be NP-hard in Nguyen et al. [2014]. It should be noted that starting from the Caragiannis et al. paper, there has been an increasing interest in combining fairness criteria with Pareto efficiency, sprouting various papers like Plaut and Roughgarden [2020], which focuses on envy-freeness up to any item, and Barman, Krishnamurthy, and Vaish [2018], which describes a pseudo-polynomial algorithm to approximate an item allocation that maximizes the utilities product.

A recent trend in fair division, on which we also embark, is that of considering an underlying social network on which agents are or can be placed. This model was first introduced in Chevaleyre, Endriss, and Maudet [2007], where also the local variant of envy-freeness was defined. However, their model differed in the fact that it also included money and mainly revolved around allowing the agents to exchange items. Years later, Abebe, Kleinberg, and Parkes [2017] and Bei, Qiao, and Zhang [2017] almost simultaneously reintroduced this graph-variant independently. It should be noted, however, that their setting differs much from ours as they consider divisible resources, whereas we are only concerned with indivisible ones.

Along this line of research, Beynier et al. [2019] stands out as the paper which has mostly inspired us for this work. Their research was mainly concerned with the *house allocation* problem, a particular case of fair division in which each agent is to be assigned exactly a single item, but had also a social

[1] For the reader's interest, it should be noted that the preprint which can be found on arXiv.org (`1907.09279 [cs.GT]`) contains a more extensive taxonomy compared to the one in the conference article.

network, represented as an undirected graph. Like ours, their work linked the computational complexity of performing an item assignment which is locally envy-free with the topology of the underlying social network. Main results include NP-hardness of such problem in graphs in which the maximum degree (i.e. the maximum number of edges for each vertex in the graph) is equal to some constant $k \geq 1$, in graphs where the minimum degree is $n - k$ for a fixed constant $k \geq 3$ (where $n$ is the number of vertices in the graph) and tractability in the case of graphs with a minimum degree equal to $n - 2$. Unlike their work, our main focus will be on a variant of this problem in which the agents also have to be placed on the network and on a specular problem in which the item allocation is fixed and only the position assignment must be computed. It should also be noted that one section of their paper is devoted to the first problem we have just mentioned. From this point of view, our work differs simply in the fact that we are not in the special case of the house allocation problem. Thus, when compared to the work from Beynier et al., we consider ours to be a different take on this "*local*" variant of fair division, but also a continuation of it.

Bredereck, Kaczmarczyk, and Niedermeier [2018] also proposed a model akin to ours, where however the social graph is a directed one instead of an undirected one and agents have already been given a fixed position on the network. Following the trend started by Caragiannis et al. [2019], they also mix envy-freeness with efficiency criteria: notable ones include utilitarian social welfare (where the objective is to maximise the sum of all utilities) and the already mentioned Pareto efficiency. They present a plethora of complexity results, both classical and parameterized, and consider also different representations and constraints on the utility functions.

Eiben et al. [2020] is another work which has greatly inspired us, especially for our results in parameterized complexity. A major difference in their setting is the way envy-freeness is defined, as it is equivalent to the combination of a variant of (global) proportionality and our local envy-freeness, and the fact that, again, agents are already fixed on the social network. In their setting, they use (with other parameters) *cliquewidth* and *treewidth*, two values which intuitively measure how much a graph is similar to a clique and a tree respectively, to prove that finding an envy-free or locally envy-free item allocation is in XP, and strengthen the treewidth result by proving that the problem cannot become FPT even with additional parameters. Finally, using integer linear programming and a set of completely different parameters, they provide an FPT result for both problems.

We conclude this section with two other directions in this variant of fair division which share some similarities to our model, but are fundamentally different. The first one involves the possibility of allowing agents on the graph to exchange items, which are initially endowed to them. This dates back to Chevaleyre, Endriss, Estivie, et al. [2007], where the model which allowed agents to exchange items was defined, and Chevaleyre, Endriss, and Maudet [2007], where such model was extended with an underlying social graph, as mentioned before. After years this setting was partially picked up again in Gourvès, Lesca, and Wilczynski [2017], who introduced various problems in the house allocation case. One of these was concerned with checking whether it is possible for an agent to obtain a certain item through rational swaps, and it turned out that in case the graph is a star this can be done in polynomial time, while for trees this problem is NP-hard. Following, both Huang and Xiao [2019] and Bentert et al. [2019] proved the tractability of checking whether an object can "reach" an agent via rational swaps assuming the social graph is a line and obtained a positive result in the house allocation problem in case preferences are strict, and while the former also proved NP-hardness in case preferences are weak, the latter showed NP-hardness results for a different class of graphs and also under different constraints over the agents' utilities.

The second direction takes a completely different take on the graph, which is used to connect items instead of agents. This framework was first defined in Bouveret, Cechlárová, et al. [2017], later followed by Igarashi and Peters [2019], Lonc and Truszczynski [2018], Bei, Igarashi, et al. [2019] and Goldberg, Hollender, and Suksompong [2020]. Notably enough, most of this line of research focuses on answering whether there are item allocations which satisfy certain fairness and/or efficiency criteria while also allocating to each agent (usually) a connected set of items. Nevertheless, complexity-related results have also been achieved in this setting (we particularly refer to Igarashi and Peters [2019] and Goldberg, Hollender, and Suksompong [2020] amongst the cited papers for this kind of results).

# Chapter 2

# Preliminaries

In this chapter we give the required background and formalities required in the thesis. We define our model of fair division with an underlying social network, the fairness criteria and their local variants. We also introduce notions from complexity theory and parameterized complexity theory which will be used in the theoretical study of the framework.

## 2.1 Fair Division

A finite set of $n$ agents $\mathcal{N}$ and a finite one of $m$ items $\mathcal{O}$ are given. We will denote with $[k]$ the set $\{1, \ldots, k\}$. Each agent $i$ has her own utility function $u_i : \mathcal{P}(\mathcal{O}) \to \mathbb{R}$, which induce a *utility profile* $\boldsymbol{u} = (u_1, \ldots, u_n)$. For convenience and readability, when considering a single item $o_j$ we will write $u_i(o_j)$ instead of $u_i(\{o_j\})$ to denote the utility granted by such item to agent $i$. An item allocation consists of a vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ of subsets which partition $\mathcal{O}$, where agent $i$ has been assigned the bundle $\pi_i$. As any allocation $\boldsymbol{\pi}$ is a partition, for any pair of distinct agents $i, j \in \mathcal{N}$ we have that $\pi_i \cap \pi_j \neq \emptyset$ and $\bigcup_{k=1}^{n} \pi_k = \mathcal{O}$. We will work with *additive* utilities, meaning that if agent $i$ is assigned the bundle of objects $\pi_i = \{\pi_{i_1}, \ldots, \pi_{i_\ell}\}$, then $u_i(\pi_i) = \sum_{k=1}^{\ell} u_i(\pi_{i_k})$. We differentiate objects in two (*subjective*) categories, *goods* and *chores*, where the former give positive utility and the latter negative. Obviously two different agents might have different opinion on the same item $o$, even to the case where the first might consider it a good (thus giving her a positive utility) and the second a chore (giving a negative utility).

In our model, we assume also the existence of a graph $G = (V, E)$, where $V = \{1, \ldots, k\}$ is the finite set of vertices and $E \subseteq \binom{V}{2}$ is the edge relation, where with $\binom{V}{2}$ we denote the set of all unordered pairs of $V$, hence $G$ is an undirected graph. For directed graphs, the edge relation is a subset of $V \times V$, hence $E$ is a set of ordered pairs of vertices. We assume that graphs are undirected, unless we explicitly state otherwise. The following concept is one which will be mostly used in Chapter 6, in which we will discuss the experiments we have performed.

**Definition 1** (Degree of a vertex)**.** Let $G = (V, E)$ be an undirected graph and let $v \in V$ be a vertex in such graph. The degree of vertex $v$ in the graph $G$, denoted with $deg_G(v)$, is the number of vertices to which it is connected, i.e. the cardinality of the set $\{w \mid \{v, w\} \in E\}$.

In particular, we will be interested in the so-called *regular* graphs, i.e. those graphs in which all vertices have the same degree.

The objective of the central authority is to assign to each agent her place (vertex) on the network (graph). We denote with $\mu$ the assignment, referred from now on as *position assignment*, of agents to the graph's vertices, i.e. $\mu$ is a map from $\mathcal{N}$ to $V$ where no two distinct agents $i, j \in \mathcal{N}$ are such that $\mu(i) = \mu(j)$. For simplicity, we assume that $|\mathcal{N}| = |V|$, therefore any position assignment $\mu$ is a bijection between $\mathcal{N}$ and $V$. Given a vertex $v$, we denote with $N(v) = \{w \mid \{v, w\} \in E\}$ the neighborhood of $v$, i.e. the set of vertices that are connected with $v$ through $E$ in the graph $G$. We also define the *augmented* neighborhood of $v$, defined as $N^+(v) = N(v) \cup \{v\}$, i.e. the set containing the neighborhood of $v$ and $v$ itself. Given a position assignment $\mu$ in which agent $i \in \mathcal{N}$ has been assigned vertex $v \in V$, we denote with $N_\mu(i)$ the neighborhood of $i$, that is, the set of agents $j \in \mathcal{N}$ who have been assigned a vertex connected to $v$, i.e.:

$$N_\mu(i) = \{j \mid \exists w. \mu(j) = w \wedge w \in N(v)\}$$

With $N_\mu^+(i)$ we denote the *augmented* neighborhood of $i$, which, similarly to before, consists of the union between the singleton containing $i$ and her own neighborhood: $N_\mu^+(i) = N_\mu(i) \cup \{i\}$. When $\mu$ is clear from the context, we will drop it for convenience.

We will study two different cases depending on whether the item allocation $\boldsymbol{\pi}$ is fixed or not. However, we will always have that the central authority must give a position assignment $\mu$. In the first case, instances of our problems are tuples $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, while in the second $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$, where $\mathcal{N}$ is the finite set of agents, $\mathcal{O}$ the finite set of items, $\boldsymbol{u}$ the item allocation and $G$ the social graph.

Following we give a rundown of fairness notions in item allocations. Let $\mathcal{N}$ be a set of $n$ agents, $\mathcal{O}$ a set of $m$ items and $\boldsymbol{\pi}$ an item allocation (regardless of whether $\boldsymbol{\pi}$ is fixed or not).

**Definition 2** (Envy-freeness (EF))**.** An allocation $\boldsymbol{\pi}$ is *envy-free* if, for any pair of agents $i, j \in \mathcal{N}$, it holds that $u_i(\pi_i) \geq u_i(\pi_j)$.

**Definition 3** (Envy-freeness up to one item (EF1))**.** An allocation $\boldsymbol{\pi}$ is *envy-free up to one item* if, for any pair of agents $i, j \in \mathcal{N}$, either $i$ does not envy $j$ or there is an item $o \in \pi_i \cup \pi_j$ such that $u_i(\pi_i \setminus \{o\}) \geq u_i(\pi_j \setminus \{o\})$.

**Definition 4** (Proportionality (PROP))**.** An allocation $\boldsymbol{\pi}$ is *proportional* if, for any agent $i \in \mathcal{N}$, it holds that $u_i(\pi_i) \geq \frac{u_i(\mathcal{O})}{n}$.

The following implications are a known fact in the literature about fair division (Aziz, Bouveret, et al. [2018], Aziz and Rey [2020]).

**Fact 1.** *If $\boldsymbol{\pi}$ is EF, then it is also EF1 and PROP.*

Regarding envy-freeness, we also define the so-called "*envy-free graph*". This is an undirected graph in which the unordered pair $\{i, j\}$ is in the set of edges $E$ if and only if $i$ does not envy $j$ and vice versa.

**Definition 5** (Envy-free graph)**.** Given a set of agents $\mathcal{N}$, a set of items $\mathcal{O}$, a profile of utility functions $\boldsymbol{u}$ and an item allocation $\boldsymbol{\pi}$, the envy-free graph $G = (V, E)$ is defined as follows:

- $V := \mathcal{N}$, i.e. each agent is a vertex in the graph;
- $E := \{\{i, j\} \mid u_i(\pi_i) \geq u_i(\pi_j) \wedge u_j(\pi_j) \geq u_j(\pi_i)\}$, i.e. two agents are connected if and only if they both do not envy each other.

We will also use a directed version of the graph, where the ordered pair $(i, j)$ is in the edge relation $E$ if and only if $i$ does not envy $j$. However, unless stated otherwise, we will always consider the undirected version of the graph when we refer to it.

It is also possible to define an analogous graph for envy-freeness up to one item, the "*envy-freeness up to one item graph*".

**Definition 6** (Envy-free up to one item graph)**.** Given a set of agents $\mathcal{N}$, a set of items $\mathcal{O}$, a profile of utility functions $\boldsymbol{u}$ and an item allocation $\boldsymbol{\pi}$, the envy-free graph $G = (V, E)$ is defined as follows:

- $V := \mathcal{N}$, i.e. each agent is a vertex in the graph;
- $E := \{\{i, j\} \mid \exists o, o' \subseteq \pi_i \cup \pi_j.\ |o| \leq 1 \wedge |o'| \leq 1 \wedge u_i(\pi_i \setminus o) \geq u_i(\pi_j \setminus o) \wedge u_j(\pi_j \setminus o') \geq u_j(\pi_i \setminus o')\}$, i.e. two agents are connected if and only if they both do not envy each other up to one item.

Similarly to the envy-free graph, we will also use a directed version of the envy-free up to one item graph; though, unless specifically stated otherwise, we will refer to the undirected version. It is clear that both these graphs can be built in polynomial time, to be more precise in $O(|\mathcal{N}|^2)$ time for the first and $O(|\mathcal{N}|^2|\mathcal{O}|)$ for the second.

We now define *local* variants of the previous fairness notions. The intuition is that, given the position of an agent on the social graph, she will only be able to know the bundles of her neighbors, and not the ones of every other agent. Obviously, in this case we also must have that the position assignment $\mu$ has already been decided.

**Definition 7** (Local envy-freeness (LEF))**.** An item allocation $\boldsymbol{\pi}$ and a position assignment $\mu$ are *local envy-free* if, for any agent $i \in \mathcal{N}$ and any agent $j \in N_\mu(i)$, it holds that $u_i(\pi_i) \geq u_i(\pi_j)$.

**Definition 8** (Local envy-freeness up to one item (LEF1))**.** An item allocation $\boldsymbol{\pi}$ and a position assignment $\mu$ are *local envy-free up to one item* if, for any agent $i \in \mathcal{N}$ and any agent $j \in N(i)$, either $u_i(\pi_i) \geq u_i(\pi_j)$ or there is an item $o \in \pi_i \cup \pi_j$ such that $u_i(\pi_i \setminus \{o\}) \geq u_i(\pi_j \setminus \{o\})$.

**Definition 9** (Local proportionality (LPROP))**.** An item allocation $\boldsymbol{\pi}$ and a position assignment $\mu$ are *local proportional* if, for any agent $i \in \mathcal{N}$, it holds that $u_i(\pi_i) \geq \frac{\sum_{j \in N^+_\mu(i)} u_i(\pi_j)}{|N^+_\mu(i)|}$.

From which it is trivial to verify the following fact:

**Fact 2.** *If $\boldsymbol{\pi}$ is EF (EF1), then it is also LEF (LEF1) with any $\mu$. Moreover, similarly to the global case, if $\boldsymbol{\pi}$ is LEF with some $\mu$, then it is also LEF1 and LPROP with the same $\mu$.*

However, interestingly enough, PROP does not necessarily imply LPROP. Consider the following example, with the following utilities:

| | $o_1$ | $o_2$ | $o_3$ |
|---|---|---|---|
| 1 | 1 | 2 | 0 |
| 2 | 0 | 1 | 2 |
| 3 | 2 | 0 | 1 |

As for the item allocation $\boldsymbol{\pi}$, agent $i$ gets just object $o_i$ for each $i \in [3]$. Hence, it is clear that the item allocation is a proportional one, as for each agent the average utility across all items is 1, exactly the utility of their bundle. Instead, consider the graph $G = (V, E)$ that consists of a connected pair and an isolated vertex, i.e. $V = \{v_1, v_2, v_3\}$ and $E = \{\{v_1, v_2\}\}$. As one can quickly verify, it is indeed the case that there is no position assignment $\mu$ which is local proportional with $\boldsymbol{\pi}$.

Figure 2.1 summarizes the taxonomy just described.

Before moving on, we would like to discuss one of the criteria we have just defined, envy-freeness up to one item. First of all, it is a known fact that an EF1 allocation always exists, and that it can be computed in polynomial time (Lipton et al. [2004], Caragiannis et al. [2019]). However, if the set of

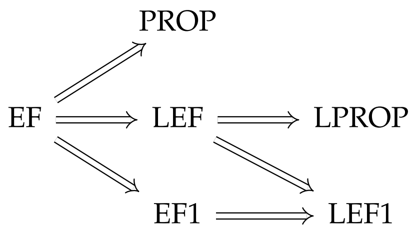



Figure 2.1: Logical relation between defined fairness criteria.

items to assign contains a mix of goods and chores, both algorithms fail to find an EF1 allocation. In a follow-up paper, Aziz, Caragiannis, et al. [2019] have proven that it is possible to find an EF1 item allocation in polynomial time even when the items are a mix of goods and chores, in case utilities are additive. This is achieved by performing a double round-robin[2] assignment, where intuitively the first round is executed to assign chores in a clockwise order and the second to assign goods in an anticlockwise order. Thus, the following proposition is a corollary of these observations and Fact 2:

**Proposition 1.** *There always exists an item allocation which is local envy-free up to one item for any position assignment when utilities are additive. Moreover, this item allocation can be computed in polynomial time.*

This, however, does not make LEF1 an uninteresting criterion: what happens if the item allocation $\boldsymbol{\pi}$ is fixed? In this case Proposition 1 tells us nothing, hence there are still open questions left to answer.

We conclude this section about fair division by introducing the problems and the recurring graph topologies which will be studied in the following chapters. We denote with $\mathcal{F}$ any fairness criteria we have just introduced. In all the following problems we assume that any item must be assigned to some agent. First, we introduce the two decision problems. The first one will be called EXISTS-$\mathcal{F}$-POSITION-ASSIGNMENT:

> **Instance**: a tuple $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, where $\mathcal{N}$ is a set of agents, $\mathcal{O}$ a set of items, $\boldsymbol{u} = (u_1, \dots, u_{|\mathcal{N}|})$ a utility profile, $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{|\mathcal{N}|})$ an item allocation and $G$ a graph.
>
> **Question**: is there a position assignment $\mu$ such that $\boldsymbol{\pi}$ and $\mu$ satisfy $\mathcal{F}$?

And the second EXISTS-$\mathcal{F}$-DISTRIBUTION:

> **Instance**: a tuple $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$, where $\mathcal{N}$ is a set of agents, $\mathcal{O}$ a

[2] In the round-robin algorithm agents are ordered in a circular order and the item allocation is performed by following such order and assigning one item to an agent each time it is her turn.

set of items, $\boldsymbol{u} = (u_1, \ldots, u_{|\mathcal{N}|})$ a utility profile and $G$ a graph.

**Question**: is there a pair $\langle \boldsymbol{\pi}, \mu \rangle$, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{|\mathcal{N}|})$ is an item allocation and $\mu$ is a position assignment, such that $\boldsymbol{\pi}$ and $\mu$ satisfy $\mathcal{F}$?

The other problems which we will analyse are the "*search*" variants of the ones we have just defined. This means that we do not want an answer to a question, but an object in output. The first problem is FIND-$\mathcal{F}$-POSITION-ASSIGNMENT:

**Instance**: a tuple $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, where $\mathcal{N}$ is a set of agents, $\mathcal{O}$ a set of items, $\boldsymbol{u} = (u_1, \ldots, u_{|\mathcal{N}|})$ a utility profile, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{|\mathcal{N}|})$ an item allocation and $G$ a graph.

**Output**: a position assignment $\mu$ such that $\boldsymbol{\pi}$ and $\mu$ satisfy $\mathcal{F}$, if there is one.

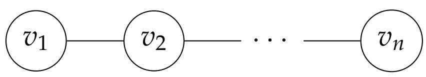
And the second FIND-$\mathcal{F}$-DISTRIBUTION:

**Instance**: a tuple $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$, where $\mathcal{N}$ is a set of agents, $\mathcal{O}$ a set of items, $\boldsymbol{u} = (u_1, \ldots, u_{|\mathcal{N}|})$ a utility profile and $G$ a graph.

**Output**: a pair $(\boldsymbol{\pi}, \mu)$, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{|\mathcal{N}|})$ is an item allocation and $\mu$ is a position assignment, such that $\boldsymbol{\pi}$ and $\mu$ satisfy $\mathcal{F}$, if there is one.

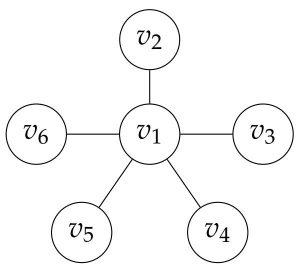
Notice that, by Proposition 1, we know that for LEF1 we will not have to consider the problems where we have to find both an item allocation and a position assignment.

We will consider three specific topologies for graphs. The first one is the line, as for example:



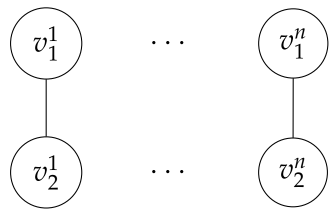

The line of $n$ vertices is defined formally as a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ and $E = \{\{v_i, v_{i+1}\} \mid 1 \leq i < n\}$.

Secondly, the star. Following is a star with 6 vertices:



The star of $n$ vertices is defined formally as a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ and $E = \{\{v_1, v_i\} \mid 1 < i \leq n\}$ ($v_1$ is the central vertex).

Finally, the so-called "*matching graph*", which we will call "*matching*" from now on:




A matching of size $2n$ is defined formally as a graph $G = (V, E)$, where $V = \{v_1^1, v_2^1 \ldots, v_1^n, v_2^n\}$ and $E = \{\{v_1^i, v_2^i\} \mid 1 \leq i \leq n\}$.

## 2.2 Computational Complexity

We now introduce notions from computational complexity which will be used in the theoretical results. We expect the reader to be familiar with basic concepts from classical complexity, such as asymptotic notation and the complexity classes P and NP. On the other hand, we will cover much more notions in parameterized complexity, such as FPT reductions and various complexity classes.

For the interested reader, we refer to textbooks such as Arora and Barak [2009] and Flum and Grohe [2006] for more complete introductions in, respectively, classical and parameterized complexity.

We will divide this section in two subsections, one for classical and the other for parameterized complexity. Aside from theoretical concepts, we will also list problems (and their complexity) that will appear in the following chapters.

### 2.2.1 Classical Complexity

We remind the reader of two concepts, as they will both play a central role in the coming chapters. In order to be as precise as possible, we introduce decision problems as subsets of a language of some alphabet. Given an alphabet $\Sigma$, i.e. a finite set of symbols, we denote with $\Sigma^\star$ the set containing all finite strings composed of symbols from $\Sigma$. Hence, a decision problem $Q$ is a subset of $\Sigma^\star$, and therefore it can also be seen as a "*language*" (i.e. a set of strings made exclusively of symbols from $\Sigma$) where its alphabet is $\Sigma$. An algorithm decides $Q$ correctly in case it answers positively if the input instance is a member of $Q$ and negatively otherwise.

**Definition 10** (Polynomial-time reduction)**.** Let $Q$ and $Q'$ be two decision problems over some alphabets $\Sigma$ and $\Sigma'$. A function $f : \Sigma^\star \mapsto (\Sigma')^\star$ is a polynomial-time reduction from $Q$ to $Q'$ if, for any input instance $x \in \Sigma^\star$:

- It can be computed in polynomial time, i.e. in time $|x|^{O(1)}$;
- $x \in Q$ if and only if $f(x) \in Q'$.

**Definition 11** (NP-completeness)**.** A problem $Q \subseteq \Sigma^\star$ is NP-complete if:

- $Q$ is in NP;
- For any problem $Q'$ in NP there is a polynomial-time reduction $f$ from $Q'$ to $Q$ ($Q$ is "NP-*hard*").

A standard technique to prove the NP-completeness of some problem $Q$ is to, once one has established its membership in NP, give a polynomial-time reduction from an NP-hard problem $Q'$ to $Q$. As this will be the main technique which we will use to prove the NP-completeness of various problems, we will now give a list of recurring (NP-complete) problems that we will use to give polynomial-time reductions from.

- HAM-PATH

  **Instance**: a graph $G = (V, E)$.

  **Question**: is there an Hamiltonian path (i.e. a path which visits each vertex exactly once) in $G$?

- 3-PARTITION

  **Instance**: a multiset of positive integers $S$ of size $3m$ such that the sum of all integers is $mT$ and each integer is in the open interval $\left(\frac{T}{4}, \frac{T}{2}\right)$.

  **Question**: is there a partition of $S$ into $m$ subsets $T_1, \ldots, T_m$ such that for each subset $T_i = \{t^i_1, \ldots, t^i_{n_i}\}$ we have that $\sum_{j=1}^{n_i} t^i_j = T$?

- PARTITION

  **Instance**: a multiset of positive integers $S$.

  **Question**: is there a partition of $S$ into two subsets $S_1, S_2$ such that the sum of their elements is equal?

- EXACT COVER BY 3-SETS

  **Instance**: a set $X$ such that $|X| = 3q$ for some integer $q$ and $S \subseteq \{T \subseteq X \mid |T| = 3\}$ a collection of triplets of elements of $X$.

**Question**: is there a subset $S' \subseteq S$ such that each element of $X$ occurs exactly in one triplet of $S'$?

It should be noted that we use a non-standard version of the 3-PARTITION problem, as the subsets $T_1, \ldots, T_n$ which partition $S$ need not be triplets, unlike the name of the problem might suggest.

We quickly justify the use of this variant by giving a reduction from an arbitrary instance $\langle S, mT \rangle$ for the standard 3-PARTITION to one for our variant. Define the new set of elements $S' = \{s + 2T \mid s \in S\}$: notice that the sum of all elements in $S'$ must be $7mT$, because in the original instance it was $mT$ and the size of $S$ (hence also of $S'$) is $3m$. The question now is whether there is a partition of $S'$ into subsets which sum up to $7T$. Clearly the reduction can be done in polynomial time. It is also easy to see that it is a correct one: if there is a partition of $S$ in triplets for the original instance, then the partition which has the corresponding triplets is a solution for the built instance, since originally each triplet summed up to $T$ and now, as we have added $2T$ to each of element of $S$, the new triplets must sum up to $7T$. For the other direction, it suffices to notice that a partition of $S'$ into subsets of elements which sum to $7T$ can only contain triplets, as now each element of $S'$ is in the open interval $(7\frac{T}{4}, 7\frac{T}{2})$ and therefore the sums of any subsets of two elements or of four elements are respectively strictly smaller and larger than $7T$. Thus, the corresponding partition in triplets of elements of $S$ is a solution to the original instance. Therefore, the variant of 3-PARTITION which we use is also NP-complete (membership in NP holds by an argument identical to the one for the standard version).

### 2.2.2 Parameterized Complexity

In the classical setting, we have already seen how a decision problem $Q$ can be defined as a language of a given alphabet $\Sigma$. A "*parameterized problem*" can be seen as language in which every string is paired up with a natural number.

**Definition 12** (Parameterized problem)**.** Given an alphabet $\Sigma$, a parameterized problem is a pair $(Q, \kappa)$, where $Q$ is a decision problem (i.e. a subset, or language, of $\Sigma^\star$) and $\kappa : \Sigma^\star \mapsto \mathbb{N}$ is a polynomial-time computable mapping from words of the alphabet $\Sigma$ to natural numbers.

The function $\kappa$ is a function which outputs the parameter of an input instance. Given an input instance $x \in \Sigma^\star$ for $Q$ and an integer $k$, we say that $(x, k) \in (Q, \kappa)$ if and only if $x \in Q$ and $\kappa(x) = k$.

As a first example that might be useful for the reader, consider the problem CLIQUE, where an input instance consists of a pair $\langle G, k \rangle$, where $G$ is a graph and $k$ an integer, and the question is whether $G$ contains a clique of size

$k^3$. A natural parameterization for CLIQUE is $\kappa(\langle G, k\rangle) := k$, i.e. using the integer $k$ as a parameter.

The reader might find this definition to be quite restraining in the sense that we allow problems to be parameterized only with a single integer parameter. We relax this constraint by allowing problems to be parameterized by any number of integers. This relaxation can be justified by considering the single parameter of the definition as a code for the tuple of parameters which we are effectively using.

We are now ready to encounter our first parameterized class.

**Definition 13** (FPT)**.** A parameterized problem $(Q, \kappa) \subseteq \Sigma^\star \times \mathbb{N}$ is in FPT if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that there is a deterministic algorithm which, given an input instance $(x, k) \in \Sigma^\star \times \mathbb{N}$, decides whether $(x, k) \in (Q, \kappa)$ in $f(k)|(x, k)|^{O(1)}$ time.

The FPT class can be seen as "parameterized P", as it identifies the problems that we consider tractable for reasonably low values of the parameters. Notice that we call these problems "tractable" mainly because the chosen parameterizations are such that for the majority of the input instance in which we are interested the parameters are relatively low, allowing us to consider negligible the $f(k)$ factor in the time it takes to compute the algorithm.

As we have defined the parameterized counterpart of the P class, we will also define now the counterpart of polynomial-time reductions.

**Definition 14** (FPT reduction)**.** Let $(Q, \kappa)$ and $(Q', \kappa')$ be two parameterized problems over some alphabets $\Sigma$ and $\Sigma'$. A function $f : \Sigma^\star \mapsto (\Sigma')^\star$ is an FPT reduction from $(Q, \kappa)$ to $(Q', \kappa')$ if:

- $x \in Q$ if and only if $f(x) \in Q'$;
- There is some computable function $g : \mathbb{N} \to \mathbb{N}$ such that there is a deterministic algorithm which runs in time $g(k)|(x, k)|^{O(1)}$ and computes $f(x)$ for any input instance $(x, k)$ for $(Q, \kappa)$;
- There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that $\kappa'(f(x)) \leq g(\kappa(x))$ for any $x \in \Sigma^\star$.

While the first two conditions might seem natural, the last one stands out as it seems to be a rather artificial condition on the size of the output instance's parameter. Nevertheless, it is crucial as otherwise the FPT class would not be closed under the resulting reductions. We will show membership in FPT by giving an FPT reduction to the Integer Linear Program (ILP) problem,

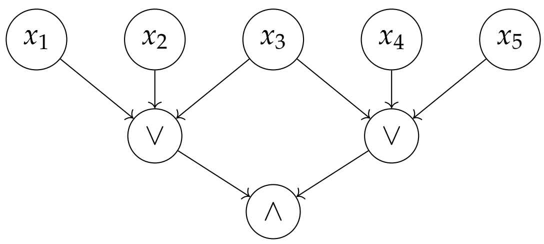
[3] A clique of size $k$ is a graph $G = (V, E)$ such that $V = [k]$ and $E = \{\{i, j\} \mid i, j \in [k]\}$, i.e. a graph of $k$ vertices all connected.

which is in FPT when parameterized by the number of variables in the set of constraints.

We now introduce parameterized classes that contain problems which are considered intractable. These classes are also closed under FPT reductions, thus a way to prove the membership of a problem in either one of them is to define an FPT reduction from some problem in such class to the one in which we are interested in.

**Definition 15** ($W[t]$)**.** A parameterized problem $(Q, \kappa)$ is in $W[t]$, for $t \geq 0$, if there is an FPT reduction which reduces an instance $(x, k)$ to a logical circuit with *weft* smaller than or equal to $t$ such that there is an assignment which assigns $k$ variables true and makes the whole circuit true if and only if $(x, k) \in (Q, \kappa)$.

A logical circuit is a DAG-like structure (Directed Acyclic Graph) in which the roots (all the vertices which no edge points towards to) are literals (i.e. occurrences of, possibly negated, propositional variables), the inner vertices are logical gates (which represent the logical operations of conjunction, disjunction and negation) and there is one final logical gate at the end of the DAG which returns the truth value of the formula under a given assignment. The "*weft*" of a logical circuit is the largest number of logical gates with more than two inputs across all paths from one of the roots to the end of the DAG[4]. Consider the following as an example:




The circuit represents the formula $(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4 \vee x_5)$. The weft of the formula is exactly 1, because on any path the first logical gate which is encountered has more than two inputs, while the last gate has exactly two inputs.

As the reader might have guessed, all these complexity classes generate a hierarchy, which is known as the "*W hierarchy*". It should be noted that it is still not known whether the hierarchy collapses somewhere (though it is trivially true that $W[i] \subseteq W[j]$ for any $i \leq j$) or whether it is equal to the FPT class, which coincides with $W[0]$. To see why this last equality holds, it suffices to observe that given an input instance $(x, k)$ for some FPT problem, we can decide it in FPT time (trivially), and then output, in polynomial time,

[4]Clearly, these logical gates with more than two inputs can only be conjunction and disjunction ones.

a logical circuit which is either always true or false, regardless of the values assigned to the variables, depending on the answer of the FPT algorithm for the original problem. To conclude the argument it suffices to notice that, by how the circuit is built, its truth value will be the same for any assignment which assigns $k$ variables true (thus the correct one based on the decision on input $(x, k)$), and that it can be built trivially with a weft of 0; therefore, it follows that such problem is indeed in $W[0]$ by the definition we have given previously for the generic class $W[t]$.

Akin to what happens in classical complexity, we will assume that the answer to whether the hierarchy collapses or whether is equal to the FPT class are both negative, hence problems which are proven to be $W[t]$-hard will be considered *effectively* intractable, for any $t \geq 1$. Though we have introduced the whole hierarchy, we will only be interested in the class $W[1]$.

The final class we will consider is the parameterized version of NP.

**Definition 16** (Para-NP)**.** A parameterized problem $(Q, \kappa) \subseteq \Sigma^\star \times \mathbb{N}$ is in para-NP if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that there is a non-deterministic algorithm which, given an input instance $(x, k) \in \Sigma^\star \times \mathbb{N}$, decides whether $(x, k) \in (Q, \kappa)$ in $f(k)|(x, k)|^{O(1)}$ time.

Notice the nice parallelism between P vs. NP and FPT vs. para-NP. Indeed, the following is a known fact about these four complexity classes.

**Fact 3** (Corollary 2.13, Flum and Grohe [2006])**.** P = NP *if and only if* FPT = *para-*NP.

This class also has its own "*hard*" problems.

**Definition 17** (Para-NP-hardness)**.** A parameterized problem $(Q, \kappa)$ is para-NP-hard if and only if there exists an FPT reduction from any parameterized problem $(Q', \kappa')$ in para-NP to it.

The following is a fact which is well known about para-NP-hard problems. In the thesis, we will use this result to show para-NP-hardness.

**Fact 4** (Corollary 2.16, Flum and Grohe [2006])**.** *A parameterized problem* $(Q, \kappa)$ *is para-*NP*-hard if it is* NP*-hard for at least one constant value of the parameterization* $\kappa$*.*

A classic example of a para-NP-hard problem is the GRAPH COLORING problem parameterized by the number of colors, as it is already NP-hard when the number of colors is fixed to 3.

It is also known that all the classes we have seen so far are closed under FPT reductions.

**Fact 5.** FPT*, the W hierarchy and para-*NP *are all closed under* FPT *reductions.*

Thus, to prove membership in any of these classes it suffices to give FPT reductions from problems which we already know are in them.

Before listing the problems that we will use to perform the FPT reductions, we have to define a graph parameter which is often used in parameterized complexity.

**Definition 18** (Tree decomposition)**.** Given a graph $G = (V, E)$, the *tree decomposition* of $G$ is a pair $\mathcal{T} = \langle T = (V', E'), \chi \rangle$ where $T$ is a tree and $\chi : V' \to \mathcal{P}(V)$ maps vertices of the tree $T$ to subsets of vertices of the graph $G$ (which are called "*bags*") so that the following two conditions hold:

1. For every edge $\{v, w\} \in E$, there is a tree vertex $t \in V'$ such that $\{v, w\} \subseteq \chi(t)$;
2. For every graph vertex $w \in V$, the set of tree vertices $t \in V'$ such that $v \in \chi(t)$ induces a non-empty subtree.

Notice that there are various ways to define the tree decomposition; the one we have chosen is the one as it was defined in Eiben et al. [2020]. It should be noted that for each graph there might be more than one tree decomposition. Hence, we can now define the treewidth of a graph, which intuitively indicates how similar is a given graph to a tree or a forest.

**Definition 19** (Treewidth)**.** Given a graph $G$, the *width* of a tree decomposition $\mathcal{T}$ of $G$ is the size of its largest bag minus one. The *treewidth* of the graph $G$, denoted with $\mathrm{tw}(G)$, is the minimal width across all tree decompositions of $G$.

To give some examples, the treewidth of a tree or a forest is 1, whereas the treewidth of a clique of size $k$ is $k - 1$. As we have already mentioned, the treewidth is a parameter which proves to be quite useful in parameterized complexity proofs and, more in general, in complexity theory as a whole. For a survey on treewidth and correlated results we refer to Bodlaender [2006].

Finally, we will now list the problems which we will employ in our proofs using parameterized complexity.

- Integer Linear Program Problem[5]

  **Instance**: a set of linear constraints.

  **Parameter**: the number of variables that appear in the constraints.

[5]Notice that we use a variant of the problem in which there is no objective function to maximize/minimize, as in our case integer linear programs with only constraints will suffice.

**Question**: is there an assignment of the variables which makes all the constraints true?

**Parameterized complexity**: FPT (Lenstra [1983]).

- SUBGRAPH ISOMORPHISM

  **Instance**: two graphs $G = (V_G, E_G)$ and $H = (V_H, G_H)$.

  **Parameter**: the number of vertices of $H$.

  **Question**: is there an isomorphism from $H$ to a (not necessarily induced) subgraph of $G$?

  **Parameterized complexity**: FPT (Alon, Yuster, and Zwick [1995]).

- CLIQUE

  **Instance**: a graph $G = (V, E)$ and an integer $k$.

  **Parameter**: the integer $k$.

  **Question**: does $G$ contain a clique of size $k$?

  **Parameterized complexity**: $W[1]$-hard (Downey and Fellows [1995]).

Chapter 3

# Complexity of Local Envy-Free Assignments

The first fairness criterion we consider is local envy-freeness, which in its definition is amongst the simplest criteria for fair division but it it also one of the most difficult to satisfy. Its global variant is amongst the most studied criteria in the literature.

This chapter will give the general guideline for the rest of the theoretical part of thesis (that includes also Chapters 4 and 5), since the results which we will prove in this one will also be adapted to the other fairness criteria, when possible.

We have decided to divide each chapter into sections based on the corresponding graph families which we will consider in them. As we move on, the families will become more and more general and include more graphs. We first start with lines and stars, both classes that belong to the one of strongly connected graphs, which follow up immediately after the two. Then, we will move to the matchings, which can be seen as the "simplest" collection of non-connected strongly connected graphs. At this point, the reader will have seen that amongst these, only the stars and the matchings prove to be graphs on which it is easy to find a way to arrange agents so that they do not envy each other, showing again the intractability of envy-freeness complexity-wise even in its local variant. Afterwards, we shift our attention to trees and forests (notice how matchings fall in the latter family) and we will work with parameterized complexity instead of classic complexity. Finally, we conclude the chapter by considering graphs that have a constant treewidth and those that do not have a constant treewidth. As any graph has a well-defined treewidth, its family will fall in either one of the two families we have just mentioned.

## 3.1 Lines and Stars

As the first families of graphs, we consider two that are relatively simple and strongly connected, lines and stars. Regardless of their similarities, as we will see shortly, the two families differ quite a lot in complexity terms with respect to our problems: the fact that in stars there is a "central" vertex makes finding a way to assign agents to their positions, so that the assignment is LEF, much easier.

**Proposition 2.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a star,* EXISTS-LEF-POSITION-ASSIGNMENT *is decidable in polynomial time.*

*Proof.* We prove the claim by giving a polynomial-time algorithm which decides the problem. The algorithm exploits a simple observation which holds in case $G$ is a star and we want to obtain an LEF position assignment given an item allocation: to check whether there is an LEF position assignment it suffices to check whether there is an agent $i$ such that $u_i(\pi_i) \geq u_i(\pi_j)$ and $u_j(\pi_j) \geq u_j(\pi_i)$ for any other agent $j \in \mathcal{N} \setminus \{i\}$. Thus, the algorithm will simply loop over the agents and check the following two conditions:

1. That the current agent $i$ is such that for any other agent $j \in \mathcal{N} \setminus \{i\}$ it holds that $u_i(\pi_i) \geq u_i(\pi_j)$;
2. That any other agent $j \in \mathcal{N} \setminus \{i\}$ is such that $u_j(\pi_j) \geq u_j(\pi_i)$.

If there is such an agent $i$ then the algorithm halts and outputs "Yes", otherwise if the loop ends and no such agent has been found it outputs "No". It is clear that the complexity of the algorithm is polynomial in the input size, quadratic in the number of agents to be precise.

We quickly check that the algorithm is correct. If there is an LEF position assignment $\mu$, then there must be some agent who is placed at the central vertex of the star and satisfies both conditions checked in the loop. Hence, the algorithm will halt and output "Yes".

On the other hand, if the algorithm outputs "Yes" then this can happen just in case there is some agent $i$ who satisfies both conditions checked in the loop. Hence, by simply taking the position assignment $\mu$ which places such agent $i$ at the central vertex and all the other agents on the outer vertices, it is clear, by the definition of LEF, that $\mu$ must be an LEF position assignment. Thus, the claim is proven. □

It is also easy to observe that, if the underlying graph $G$ is a star, the corresponding search problem, FIND-LEF-POSITION-ASSIGNMENT, can also be solved in polynomial time: instead of returning a binary answer, the algorithm will simply output a position assignment $\mu$ that places at the central vertex the agent $i$ who satisfies both conditions in the loop (if there is any) and the remaining agents will be placed in an arbitrary order on the outer vertices.

We will now show that EXISTS-LEF-POSITION-ASSIGNMENT cannot be solved in polynomial time (unless P = NP) if the social graph is a line. In order to prove the claim we will use the HAM-PATH problem.

**Proposition 3.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a line,* EXISTS-LEF-POSITION-ASSIGNMENT *is* NP*-complete even when there are as many items as agents.*

*Proof.* We first show that a solution to EXISTS-LEF-POSITION-ASSIGNMENT can be verified in polynomial time. This can be done trivially using as a certificate a position assignment $\mu$ (which is obviously polynomial in the size of the input): it suffices then to check that no agent is envious of any of her neighbors to verify whether the certificate is a correct one or not. This can be done in polynomial time, more precisely in quadratic time in the number of agents. Hence, EXISTS-LEF-POSITION-ASSIGNMENT $\in$ NP, thus is also in NP in case the underlying graph is a line.

To show NP-hardness, we provide a polynomial-time reduction from HAM-PATH to EXISTS-LEF-POSITION-ASSIGNMENT, such that the graph $G$ is a line and $|\mathcal{N}| = |\mathcal{O}|$. Let $G = (V, E)$, with $n = |V|$, be an instance for HAM-PATH. The polynomial time reduction $f$ gives in output the following instance for EXISTS-LEF-POSITION-ASSIGNMENT:

- $\mathcal{N} = [n]$, i.e. there is a corresponding agent $i$ for each vertex $v_i \in V$;
- $\mathcal{O} = \{o_1, \ldots, o_n\}$, i.e. there is a corresponding item $o_i$ for each agent $i$;
- $\boldsymbol{u} = (u_1, \ldots, u_n)$ where $u_i$ is defined as follows: let $\overline{N}(v_i) = V \setminus N^+(v_i)$ be the set of vertices which are not the vertex $v_i$, which corresponds to agent $i$, nor neighbors of it. Then, let $o_{i_1}, \ldots, o_{i_m}$ be the items that correspond to the vertices in $\overline{N}(v_i)$, let $o'_{i_1}, \ldots, o'_{i_{m'}}$ be those that correspond to the vertices in $N(v_i)$ and let $o_i$ be item that corresponds to the vertex $v_i$. The utility function $u_i$ is defined so that $u_i(o_{i_1}) > \cdots > u_i(o_{i_m}) > u_i(o_i) > u_i(o'_{i_1}) > \cdots > u_i(o'_{i_{m'}})$. For any agent $i$, define $u_i(\pi)$ for a bundle of items $\pi \subseteq \mathcal{O}$ simply as the sum of the utilities of its items;
- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ where $\pi_i = \{o_i\}$ for any $i \in \mathcal{N}$;
- $G' = (V', E')$, where $V' = \{w_1, \ldots, w_n\}$ and $E = \{\{w_i, w_{i+1}\} \mid 1 \leq i < n\}$. Clearly $G'$ is a line.

It is easy to see that the given reduction is polynomial in the input size and satisfies the two conditions we have previously mentioned. We proceed to show that $G$ has an Hamiltonian path if and only if $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' \rangle$ has a position assignment $\mu$ which is LEF with $\boldsymbol{\pi}$.

Suppose that there is an LEF position assignment $\mu$ for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' \rangle$. This means that $\mu$ assigns to each agent $i$ a vertex on the line $G'$ so that $i$ is not envious of her neighbor(s). Let $j$ be any agent who is assigned a position connected to $i$'s one. Since $\mu$ is LEF we must have that $u_i(\pi_i) \geq u_i(\pi_j)$ and $u_j(\pi_j) \geq u_j(\pi_i)$: by definition of $\boldsymbol{u}$, this can only happen just in case $v_i$ and $v_j$, the vertices in $G$ that correspond to agents $i$ and $j$, are such that $\{v_i, v_j\} \in E$. As both $i$ and $j$ were chosen arbitrarily, it follows that, for each $i, j \in \mathcal{N}$ for which $\{\mu(i), \mu(j)\} \in E'$, the pair of corresponding vertices $v_i, v_j$ in $G$ are such that $\{v_i, v_j\} \in E$ holds as well. Therefore, as each agent is assigned exactly one position on $G'$, it follows that the path which corresponds to the position assignment $\mu$ (i.e. the path $v_{i_1}, \ldots, v_{i_n}$ such that agent $i_k$ is assigned the vertex $w_k$ in $G'$ by $\mu$) is an Hamiltonian path in $G$.

Now assume that $G$ has an Hamiltonian path $P = v_{p_1}, \ldots, v_{p_n}$. Obviously, every pair of consecutive vertices $v_{p_i}, v_{p_{i+1}}$ in $P$ (for $1 \leq i < n$) must be such that $\{v_{p_i}, v_{p_{i+1}}\} \in E$, otherwise $P$ is not a valid path. Consider the position assignment $\mu$ that corresponds to $P$, i.e. such that the $i$-th agent on the line is $p_i$. We quickly show that $\mu$ is LEF: as all pairs $v_{p_i}, v_{p_{i+1}}$ in $P$ must be such that $\{v_{p_i}, v_{p_{i+1}}\} \in E$ it follows that, for their respective agents $p_i, p_{i+1}$, $u_{p_i}(\pi_{p_i}) > u_{p_i}(\pi_{p_{i+1}})$ and $u_{p_{i+1}}(\pi_{p_{i+1}}) > u_{p_{i+1}}(\pi_{p_i})$ both hold by definition of $\boldsymbol{u}$ and $\boldsymbol{\pi}$. As this holds for any pair of agents on the line, we have that no agent is envious of her neighbor(s), thus $\mu$ is indeed LEF. □

## 3.2 Strongly Connected Graphs

We now shift our attention to EXISTS-LEF-DISTRIBUTION in the context of strongly connected graphs, i.e. graphs in which there is a path (a sequence of vertices connected through the edge relation) from any vertex to any other vertex. For our purposes, it will suffice to consider agents with identical utilities. Hence, instances of the problem will be tuples $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$, where $u_i(\pi) = u_j(\pi)$ for any pair of agents $i, j \in \mathcal{N}$ and any bundle of items $\pi \subseteq \mathcal{O}$, and $G$ is strongly connected.

There are two observations to make. The first is that already Beynier et al. [2019] proved NP-completeness for such problem: we extend such result by showing that the same problem is already NP-complete even assuming that all agents have identical utilities and the underlying graph is strongly connected, thus in a setting which might be considered "simpler" (in particular for the property that agents have identical utilities). The second is that we skip EXISTS-LEF-POSITION-ASSIGNMENT in this section as its NP-completeness, in case the social graph is a strongly connected one, follows from the fact that we have shown that it is NP-complete already for lines, which are trivially strongly connected graphs (proving membership in NP can be done by the same argument).

Before proving our claim, we first show a lemma that applies to instances in which the social graph is a strongly connected one and all agents have identical utilities. This lemma will prove to be quite useful in the proof of the upcoming result.

**Lemma 1.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which agents have identical utilities and $G$ is strongly connected, $\boldsymbol{\pi}$ and a position assignment $\mu$ are LEF if and only if every agent has the same utility under $\boldsymbol{\pi}$.*

*Proof.* We prove both directions. The right-to-left direction is trivial: if every agent has the same utility, then no agent will be envious of her neighbor(s) because their utility functions are identical by assumption.

For the left-to-right, suppose by contraposition that there are two agents $i, j$ such that their utilities under $\boldsymbol{\pi}$ are different: we show that there is no position assignment $\mu$ which is LEF with $\boldsymbol{\pi}$. Without loss of generality, assume that $u(\pi_i) < u(\pi_j)$ (recall all agents have the same utility function $u$). Let $\mu$ be an arbitrary position assignment: as $G$ is strongly connected, there must be a path $P = \mu(j), \mu(k_1), \ldots, \mu(k_m), \mu(i)$ from $j$ to $i$. Either $u(\pi_{k_1}) = u(\pi_j)$ or not: in the latter, we already have that $\mu$ and $\boldsymbol{\pi}$ are not LEF. Instead, if $u(\pi_{k_1}) = u(\pi_j)$, we can repeat the same two arguments for the next agent on the path $k_2$, and so on for the following agents. Thus, either some agent $k_\ell$ on the path is such that $u(\pi_{k_\ell}) \neq u(\pi_j)$, meaning that $\mu$ and $\boldsymbol{\pi}$ are not LEF, or all agents, from $k_1$ to $k_m$, have a utility equal to $u(\pi_j)$. However, by assumption we know that $u(\pi_i) < u(\pi_j)$, therefore $u(\pi_i) < u(\pi_{k_m}) = u(\pi_j)$, proving that in any case no $\mu$ can be LEF with the given $\boldsymbol{\pi}$. □

To prove our result, we will use the unrestricted version of 3-PARTITION, as it was defined in Chapter 2.

**Theorem 1.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ in which $G$ is a strongly connected graph,* EXISTS-LEF-DISTRIBUTION *is* NP*-complete, even when agents have identical utilities.*

*Proof.* We first show that EXISTS-LEF-DISTRIBUTION is in NP. To do so, we show that, given a polynomial-size certificate, the problem can be verified in polynomial time. For any instance $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$, a certificate will be a pair $\boldsymbol{\pi}, \mu$, i.e. an item allocation and a position assignment (both polynomial in the size of the instance). We can easily check in quadratic time in the number of agents whether, under the assignment $\mu$, every pair of neighboring agents is envy-free under the allocation $\boldsymbol{\pi}$. Therefore, EXISTS-LEF-DISTRIBUTION $\in$ NP, thus it is also in NP in this particular setting.

We now show NP-hardness. Consider an arbitrary instance for 3-PARTITION made of a multiset of positive integers $S = \{x_1, \dots, x_{3m}\}$ and an integer $mT$. We provide the following polynomial-time reduction to an instance for EXISTS-LEF-DISTRIBUTION where agents have identical utilities and $G$ is strongly connected:

- $\mathcal{N} = [m]$, i.e. there is an agent for each of the subsets in a possible solution for the 3-PARTITION input instance;
- $\mathcal{O} = \{o_1, \dots, o_{3m}\}$, i.e. there is a corresponding item $o_i$ for each positive integer $x_i \in S$;
- $\boldsymbol{u} = (u, \dots, u)$ where $u(\{o_i\}) = x_i$ and, for any bundle $\pi \subseteq \mathcal{O}$, we have that $u(\pi) = \sum_{o \in \pi} u(o)$. Clearly all agents have identical utilities;
- $G$ is the line with $m$ vertices. Clearly $G$ is strongly connected.

Observe that the reduction is trivially a polynomial-time one. We now show that $S$ can be partitioned in $m$ subsets such that the sum of the elements of each subset is $T$ if and only if there is an item allocation and a position assignment that are LEF for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$.

Assume that there is a partition of $S$ in subsets $T_1, \dots, T_m$ so that for each $1 \leq i \leq m$ we have that $T_i = \{x_{i_1}, \dots, x_{i_n}\}$ is such that $\sum_{j=1}^{n} x_{i_j} = T$. Then, by simply taking the item allocation which assigns to agent $i$ the corresponding subset of items $\{o_{i_1}, \dots, o_{i_n}\}$ and the position assignment which assigns to agent $i$ the $i$-th position on the line, we have an item allocation and a position assignment that are LEF for the instance. In fact, as every agent will have a utility of $T$ by definition of $u$, no agent will be envious of her neighbor(s).

On the other hand, assume that there is an item allocation $\boldsymbol{\pi}$ and a position assignment $\mu$ that are LEF for the EXISTS-LEF-DISTRIBUTION instance. By Lemma 1 it follows that every agent must have the same utility: as $u(\mathcal{O}) = mT$, $|\mathcal{N}| = m$ and every item is assigned by definition of the problem, it follows that each bundle gives to its agent a utility of $T$. Therefore, by taking the corresponding partition of elements of $S$ (i.e. if $\pi_i = \{o_{i_1}, \dots, o_{i_n}\}$ then $T_i = \{x_{i_1}, \dots, x_{i_n}\}$) we obtain a partition $T_1, \dots, T_m$ of $S$ such that for every $T_i$ the sum of its elements is equal to $T$. □

Observe that the graph which was built in the reduction is a line, and at the same time one can also redo the same proof where the built graph is a star instead of a line. These two observations imply the following corollary.

**Corollary 1.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a star or a line,* EXISTS-LEF-DISTRIBUTION *is* NP-*complete.*

## 3.3 Matchings

We now consider the same problem but when the social graph is a matching (observe that in this case we assume that the number of agents is even). As for the stars, we can obtain a positive result both for the decision problem and the search one if the item allocation is given.

**Proposition 4.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which G is a matching,* EXISTS-LEF-POSITION-ASSIGNMENT *is decidable in polynomial time.*

*Proof.* We give a polynomial-time algorithm which can solve the problem given an arbitrary instance. The algorithm first builds the envy-free graph $G_{EF} = (V_{EF}, E_{EF})$, which can be built in polynomial time as we have already noted in Chapter 2.

Recall that a matching can be defined as a set of pairs of vertices such that any vertex appears in exactly one pair and each pair of vertices in the set is connected in the graph. Hence, if $G_{EF}$ admits a matching which contains all vertices (a "*perfect*" matching), then we know that it is possible to assign to each agent a position in the matching, by trivially pairing up agents who are paired in the perfect matching. On the other hand, if there is an LEF position assignment on the matching, then there must also be a perfect matching in $G_{EF}$, by definition of the envy-free graph itself. Now, the key observation to make is that finding the maximum size matching of any graph can be done in polynomial time (Micali and Vazirani [1980]), thus deciding whether a graph has a perfect matching can be done in polynomial time (simply check whether the size of the matching is equal to half the number of vertices). Therefore, EXISTS-LEF-POSITION-ASSIGNMENT can be decided in polynomial time. □

Like before, it is also possible to compute FIND-LEF-POSITION-ASSIGNMENT in polynomial time, by simply giving in output the perfect matching of $G_{EF}$, if there is one.

We now show that EXISTS-LEF-DISTRIBUTION is NP-complete under the assumption that agents have identical utilities and that the underlying social graph is a matching. To do this, we give a reduction from PARTITION, an NP-complete problem.

As we will see, the reduction is really a trivial one, since by Lemma 1 we know that two connected agents with identical utilities can be envy-free if and only if they have the same utility under an item allocation $\boldsymbol{\pi}$.

**Proposition 5.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ in which G is a matching,* EXISTS-LEF-DISTRIBUTION *is* NP*-complete, even when agents have identical utilities.*

*Proof.* By the same argument we have made before, we know that EXISTS-LEF-DISTRIBUTION $\in$ NP holds also in this case.

To show NP-hardness, let $S = \{x_1, \ldots, x_m\}$ be any input instance for PARTITION, and consider the following polynomial-time reduction:

- $\mathcal{N} = [2]$;
- $\mathcal{O} = \{o_1, \ldots, o_m\}$, i.e. there is a corresponding item $o_i$ for each positive integer $x_i \in S$;
- $\boldsymbol{u} = (u, u)$ where $u(\{o_i\}) = x_i$ and, for any bundle $\pi \subseteq \mathcal{O}$, we have that $u(\pi) = \sum_{o \in \pi} u(o)$. Clearly both agents have the same utility function;
- $G$ is the matching with 2 vertices.

Clearly the output instance is also an instance for our problem. Then, it is trivial to observe that there is an item allocation $\boldsymbol{\pi}$ (which allocates each item) and a (forced) position assignment $\mu$ that are LEF for the instance $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ if and only if $S$ can be partitioned into two sets such that the sums of their elements are equal. □

## 3.4 Trees

We now move to results which involve mainly parameterized complexity. For now, we will consider the family of trees for which, if we use a proper parameterization tailored for them, we can prove that FIND-LEF-POSITION-ASSIGNMENT is in FPT.

We now introduce the first of the two parameters which we will use.

**Definition 20** (Vertex-type)**.** Let $G = (V, E)$ be a tree. Given $v, w \in V$ two arbitrary vertices of $G$, we say that $v$ and $w$ have the same *vertex-type* $t_v$ if and only if the two subtrees rooted respectively in $v$ and $w$ are *identical* (up to a bijective mapping between the vertices, i.e. an *isomorphism*).

Notice that here we implicitly assume that the tree is rooted, meaning that there is some vertex in it labelled as the root. This is because otherwise the subtree rooted in some vertex cannot be defined properly.

It is easy to observe that, given two vertices, one can check whether they have the same vertex-type in linear time. Hence, vertex-types for all the graph's vertices can be computed in polynomial time.

**Fact 6.** *Given the relation*

$$T_v = \{(v, w) \mid v, w \in V, \text{ the subtrees rooted in } v \text{ and } w \text{ are isomorphic}\}$$

*each vertex-type denotes an equivalence class for $T_v$.*

It is easy to see that $T_v$ is indeed an equivalence relation, since it is reflexive, symmetric and transitive. By its own definition, it then follows that each vertex-type is indeed an equivalence class of $T_v$.

Thus, $T_v$ partitions $V$ into equivalence classes $[v]_{T_v}$, each corresponding to some vertex-type, where $v \in V$ is the "representative" of its class. We will denote with $T_V$ the set of vertex-types of $G$, with $v_{t_v}$ the representative of type $t_v \in T_V$ and with $V_{t_v}$ the set of vertices of type $t_v \in T_V$.

**Observation 1.** *There are some very important observations to make about vertex-types:*

- *If two vertices have the same vertex-type, then for each vertex-type they will have the same number of children of that type. Here, with "children" of some vertex v we intend the vertices connected to it which are also in the subtree rooted in v itself;*
- *A certain vertex-type $t_v$ can only be found at a specific depth in a single tree. This follows clearly from the fact that vertex-types are defined by the rooted subtrees of such tree;*
- *The relation $T_v$ can be computed in $O(|V|^3)$ time.*

Considering the first item of Observation 1, for an arbitrary pair of vertex-types $t_v, t'_v \in T_V$, we will denote with $e_{t_v,t'_v}$ the number of children of type $t'_v$ of any vertex of type $t_v$.

We now introduce the concept of "*agent-type*" (with respect to envy-freeness). The idea behind the agent-type is that two agents who are equivalent with respect to envy for any arbitrary agent (i.e. either they both envy/do not envy the same agents and the same agents envy/do not envy both of them) can be swapped in a position assignment $\mu$ without changing whether $\mu$ is LEF or not. Thus, let $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ be an arbitrary instance and $G_{EF} = (V_{EF}, E_{EF})$ the directed envy-free graph induced by it.

**Definition 21** (EF Agent-type)**.** Two agents $i, j \in \mathcal{N}$ have the same EF agent-type $t_a$ if and only if, for any agent $k \in \mathcal{N}$, it holds that:

- $(i,k) \in E_{EF} \iff (j,k) \in E_{EF}$: either both $i$ and $j$ do not envy $k$ or they both do;
- $(k,i) \in E_{EF} \iff (k,j) \in E_{EF}$: either $k$ does not envy both $i$ and $j$ or she does.

**Fact 7.** *Given the relation*

$$T_n = \{(i,j) \mid i,j \in \mathcal{N}, \forall k \in \mathcal{N}.[((i,k) \in E_{EF} \iff (j,k) \in E_{EF}) \\ \wedge ((k,i) \in E_{EF} \iff (k,j) \in E_{EF})]\}$$

*each agent-type denotes an equivalence class for $T_n$.*

Again, we briefly comment on why the previous claim holds. First we show that $T_n$ is reflexive, symmetric and transitive. The first two properties are trivial. It remains to prove transitivity: let $i, j, r \in \mathcal{N}$ be such that $(i, j), (j, r) \in T_n$, we show that also $(i, r) \in T_n$. As $(i, j) \in T_n$ we have that $(i, k) \in E_{EF} \iff (j, k) \in E_{EF}$ and $(k, i) \in E_{EF} \iff (k, j) \in E_{EF}$ both hold for any $k \in \mathcal{N}$, similarly as $(j, r) \in T_n$ we have that $(j, k) \in E_{EF} \iff (r, k) \in E_{EF}$ and $(k, j) \in E_{EF} \iff (k, r) \in E_{EF}$ both hold for any $k \in \mathcal{N}$. Therefore, for any $k \in \mathcal{N}$ it holds that $(i, k) \in E_{EF} \iff (j, k) \in E_{EF} \iff (r, k) \in E_{EF}$ and $(k, i) \in E_{EF} \iff (k, j) \in E_{EF} \iff (k, r) \in E_{EF}$, i.e. $(i, r) \in T_n$. Hence, we have that $T_n$ is also transitive, as desired. Thus, by definition, it follows clearly that each agent-type is indeed an equivalence class for $T_n$.

We now know that $T_n$ partitions $\mathcal{N}$, and that each agent-type corresponds to some equivalence class $[i]_{T_n}$, where $i \in \mathcal{N}$ will be regarded as the "*representative*" of such class. We will denote with $T_{\mathcal{N}}$ the set of agent-types of $\mathcal{N}$ and, with $i_{t_a}$ the representative of type $t_a \in T_{\mathcal{N}}$ and with $\mathcal{N}_{t_a}$ the set of agents of type $t_a \in T_{\mathcal{N}}$.

**Observation 2.** *Again, there are a couple of observations to make about agent-types:*

- *When doing the position assignment, placing two different agents but which have the same agent-type makes no difference if the goal is to satisfy LEF, as they both envy (do not envy) the same agents and any agent envies (does not envy) both of them;*
- *Computing the relation $T_n$ (and thus its classes) can be done in polynomial time in the size of the instance, as computing the envy-free graph can be done in polynomial time.*

Before proving that, if the social graph is a tree, FIND-LEF-POSITION-ASSIGNMENT is in FPT under the parameters we have just defined, we will first briefly comment on the two parameters themselves. Although one might argue that they are both parameters which will not be reasonably low in many cases, there are some considerations to make. For starters, the number of vertex-types is not high in $n$-ary trees and, more in general, trees which show some repeating patterns inside them. Although our running example pictured the graph as a representation of the distribution of desks inside an office, the social graph might also represent some sort of hierarchy in the company. At that point, it is not difficult to imagine that such hierarchy might be in fact a tree and, most importantly, one which does indeed show some repeating patterns. When we will move to forests, this observation will be even more impactful, as it might happen, for example, that one tree is a subtree of another one in the forest.

Though not through an immediate consideration, the (EF) agent-types can also be justified as well. In real applications, instead of the agent-types as they were was defined, it might even be more common to have agents with identical utilities being assigned bundles that grant them the same utility. A way of thinking why this situation might occur is that it might happen that the central authority does not have perfect information from every agent about the assigned bundles: at this point, it might be sensible to group agents by their job and/or known skills. Thus, agents will be divided in a coarser way with respect to their true utility functions. Moreover, in particular in the case of chores, it is not difficult to imagine that agents who have similar skills or even the same job inside the company will be assigned similar tasks, hence why they will probably have the same agent-type.

We will now move to the (positive) result. We will be able to show that, if the social graph is a tree, FIND-LEF-POSITION-ASSIGNMENT is in FPT when parameterized by the two parameters just defined, by reducing an arbitrary input instance for it to an instance for the INTEGER LINEAR PROGRAM PROBLEM, i.e. an Integer Linear Program (shortened as "ILP").

If we can define a correct FPT reduction from our problem to an ILP, this would imply the claim by simply observing how a solution to the ILP induces a position assignment $\mu$ (which will be trivially computable in polynomial time).

Before moving on to the proof, we first need to define the depth of a vertex in a tree $G = (V, E)$. In a tree, given a vertex $v$, we define its depth $d_G(v)$ in the classical way, i.e. the length of the unique path (meaning the number of traversed vertices) from the root to $v$. We denote with $D(G)$ the depth of the tree $G$ itself, i.e. the maximal depth amongst its vertices. When clear from the context, we will drop the $G$ in all these notations. Analogously, we define the depth of a vertex in a forest as the length of the unique path from the root (of the vertex's tree) to the vertex itself.

**Theorem 2.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a tree,* FIND-LEF-POSITION-ASSIGNMENT *is in* FPT *if parameterized by the number of EF agent-types $|T_{\mathcal{N}}|$ and the number of vertex-types $|T_V|$.*

*Proof.* We will prove the claim by reducing the instance $I$ to an ILP which number of variables can be bounded by a (computable) function of $|T_{\mathcal{N}}|$ and $|T_V|$. The ILP will have the following variables:

- For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$ a variable $x_{t_a, t'_a, t_v, t'_v}$ which will encode how many times an agent of type $t_a$ and one of type $t'_a$ are assigned respectively to a vertex of type $t_v$ and one of type $t'_v$ that are connected by an edge. In this case, the vertex of type $t_v$ is the parent and the one of type $t'_v$ is the child. Often in the rest of the proof we will call these respectively "*parent vertex*" and "*child vertex*";

- For $t_a \in T_\mathcal{N}$ and $t_v \in T_V$ a variable $r_{t_a,t_v}$ which value will be 1 if the tree's root is of type $t_v$ and an agent of type $t_a$ is assigned to it and 0 otherwise.

Hence, the number of variables is exactly $|T_\mathcal{N}|^2|T_V|^2 + |T_\mathcal{N}||T_V|$, which is clearly a computable function of our parameters.

We will now introduce the constraints of the ILP, with a quick description of why they are needed. Afterwards, we will prove the claim by proving that a solution to the built ILP induces the existence of an LEF position assignment (which can be also defined in polynomial time) and vice versa.

**Integrity constraints**:

– For $t_a, t'_a \in T_\mathcal{N}$ and $t_v, t'_v \in T_V$:

$$x_{t_a,t'_a,t_v,t'_v} \in \mathbb{N}_0$$

– For $t_a \in T_\mathcal{N}$ and $t_v \in T_V$:

$$r_{t_a,t_v} \geq 0$$

– For $t_a \in T_\mathcal{N}$ and $t_v \in T_V$:

$$r_{t_a,t_v} \leq 1$$

**Root constraints**:

– For $t_v \in T_V$ with $t_v$ equal to the root's vertex-type:

$$\sum_{t_a \in T_\mathcal{N}} r_{t_a,t_v} = |V_{t_v}|$$

Ensures that there is only one pair agent-type, vertex-type such that the corresponding root variable is equal to 1, and that the vertex-type is indeed the correct one;

– For $t_a \in T_\mathcal{N}$ and $t_v, t'_v \in T_V$ with $t_v$ equal to the root's vertex-type:

$$\sum_{t'_a \in T_\mathcal{N}} x_{t_a,t'_a,t_v,t'_v} - r_{t_a,t_v} e_{t_v,t'_v} = 0$$

Ensures that there is the correct number of edges from the root towards vertices of type $t'_v$, checking that $\sum_{t'_a \in T_\mathcal{N}} x_{t_a,t'_a,t_v,t'_v}$ is different from 0 just in case $r_{t_a,t_v}$ is different from 0 and that, in such case, $\sum_{t'_a \in T_\mathcal{N}} x_{t_a,t'_a,t_v,t'_v}$ is exactly equal to $e_{t_v,t'_v}$;

**Network conformity constraints**:

- For $t_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$ with $t_v$ not equal to the root's vertex-type:

$$\left(\sum_{t'_a \in T_{\mathcal{N}}} x_{t_a,t'_a,t_v,t'_v}\right) - e_{t_v,t'_v} \sum_{t'_a \in T_{\mathcal{N}}} \sum_{t''_v \in T_V} x_{t'_a,t_a,t''_v,t_v} = 0$$

Ensures that the number of edges in which a vertex of type $t_v$, to which an agent of type $t_a$ is assigned, is the parent vertex of some child vertex of type $t'_v$, to which an agent of an arbitrary agent-type has been assigned, is equal to the number of times an agent of type $t_a$ is assigned to a child vertex of type $t_v$ times the number of children of type $t'_v$ for any vertex of type $t_v$;

**Agent conformity constraints**:

- For $t_a \in T_{\mathcal{N}}$:

$$\sum_{t'_a \in T_{\mathcal{N}}} \sum_{t_v, t'_v \in T_V} x_{t'_a,t_a,t_v,t'_v} + \sum_{t_v \in T_V} r_{t_a,t_v} = |\mathcal{N}_{t_a}|$$

Ensures that the number of agents of type $t_a$ assigned over all $G$ is exactly the number of agents of such type in $\mathcal{N}$;

**LEF constraints**:

- For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$:

$$x_{t_a,t'_a,t_v,t'_v}\left(u_{i_{t_a}}(\pi_{i_{t_a}}) - u_{i_{t_a}}(\pi_{i_{t'_a}})\right) \geq 0$$

- For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$:

$$x_{t_a,t'_a,t_v,t'_v}\left(u_{i_{t'_a}}(\pi_{i_{t'_a}}) - u_{i_{t'_a}}(\pi_{i_{t_a}})\right) \geq 0$$

We now show that there is a position assignment $\mu$ which is LEF when paired up with $\boldsymbol{\pi}$ for $I$ if and only if there is a solution to the corresponding ILP we have just defined. This implies our claim because the ILP defined through the FPT reduction can be solved in FPT time (when parameterized by the number of variables), hence we can also find a position assignment $\mu$ in FPT time (with the given parameters).

We first show the left-to-right direction. Assume that there is indeed such a position assignment $\mu$. We first describe how to obtain an assignment for the ILP variables from $\mu$. This assignment will be the natural one induced by $\mu$: for the variables $r_{t_a,t_v}$, we assign 1 if the root is a vertex of type $t_v$ and an agent of type $t_a$ is assigned to it and 0 otherwise. For any other variable

$x_{t_a,t'_a,t_v,t'_v}$, we assign to it the number of times an agent of type $t_a$ is assigned to a vertex of type $t_v$, an agent of type $t'_a$ is assigned to a vertex of type $t'_v$ and there is an edge connecting a parent vertex of type $t_v$ to a child vertex of type $t'_v$. Hence, it follows clearly that the integrity constraints are satisfied. We show that the remaining constraints are satisfied as well:

**Root constraints**:

- We have two cases, either $t_v$ is the root's vertex-type or not. In the former, as the agent-type of the agent at the root is unique, observe that it will follow that $\sum_{t_a \in T_{\mathcal{N}}} r_{t_a,t_v} = 1$; similarly, in the latter, by how we have defined the assignment it will follow that $\sum_{t_a \in T_{\mathcal{N}}} r_{t_a,t_v} = 0$. In any case, the constraint is satisfied;
- For the remaining root constraints, we have three different cases. If $t_a$ and $t_v$ are such that $r_{t_a,t_v} = 0$, then it also implies that $x_{t_a,t'_a,t_v,t'_v} = 0$ for any other pair $t'_a, t'_v$ since the root is the only vertex of type $t_v$, hence the constraint is satisfied. If $t_v$ and $t'_v$ are such that $e_{t_v,t'_v} = 0$, then $x_{t_a,t'_a,t_v,t'_v} = 0$ since there are no edges connecting a parent vertex of type $t_v$ to a child of type $t'_v$, hence the constraint again is satisfied. Finally, assume that $t_a$, $t_v$ and $t'_v$ are such that $r_{t_a,t_v} = 1$ and $e_{t_v,t'_v} > 0$: then it is also clear that if we sum $x_{t_a,t'_a,t_v,t'_v}$ over all agent-types $t'_a$, we obtain exactly $e_{t_v,t'_v}$, since for each vertex of type $t'_v$ which is a child of the root there will be exactly one agent (of some type $t'_a$) assigned to it;

**Network conformity constraints**:

- As before with the second type of root constraints, also here we have three different cases. If $t_a$ and $t_v$ are such that no agent of type $t_a$ was assigned to a vertex of type $t_v$, then it must also be the case that there is no edge connecting a parent vertex (of any arbitrary type $t''_v$ to which an agent of any arbitrary type $t'_a$ was assigned) to a child vertex of such type where such an agent was assigned, hence the constraint is satisfied. On the other hand, if $e_{t_v,t'_v} = 0$, then no edge can connect a parent vertex of type $t_v$ to a child vertex of type $t'_v$, and again the constraint is satisfied. Finally, if $e_{t_v,t'_v} \neq 0$ and $\sum_{t''_v \in T_V} x_{t'_a,t_a,t''_v,t_v} \neq 0$, then observe that the number of child vertices of type $t'_v$ to which an agent of an arbitrary type has been assigned, for each vertex of type $t_v$ to which an agent of type $t_a$ has been assigned, is exactly equal to $e_{t_v,t'_v}$ simply by definition of vertex-type. Observe that the vertex-type $t_v$ is not the root's vertex-type, as in that case we would always have that $x_{t'_a,t_a,t''_v,t_v} = 0$ (for arbitrary $t'_a$, $t_a$ and $t''_v$), which would make the constraint false in case $x_{t_a,t'_a,t_v,t'_v} \neq 0$. As this holds for any vertex of type $t_v$ to which an agent of type $t_a$ has been assigned, the constraint is satisfied;

**Agent conformity constraints**:

- Let $t_a$ be an arbitrary agent-type: observe that $\mu$ must assign all agents of type $t_a$ to a (distinct) position in $G$ as it is a position assignment. Let $(t_a^1, \dots, t_a^D)$ be the number of agents which are of type $t_a$ and are assigned to a vertex at depth $i$, where $i$ ranges from 1 (the depth of the root) to $D$ (the height of the tree). Then, by the previous observation, it follows that $\sum_{i=1}^{D} t_a^i = |\mathcal{N}_{t_a}|$. To conclude that the constraint is satisfied, we have to make a couple of observations. First, $\sum_{t_v \in T_V} r_{t_a, t_v} = 1$ just in case the agent placed at the root is of type $t_a$ (in which case also $t_a^1 = 1$), otherwise it is equal to 0 (as $t_a^1$ in such case). Secondly, if $t_{v_1}^i, \dots, t_{v_m}^i$ are all the vertex-types at some depth $i$ (recall that each vertex-type can only be at a certain depth), then $\sum_{t'_a \in T_\mathcal{N}} \sum_{j=1}^{m} \sum_{t'_v \in T_V} x_{t'_a, t_a, t'_v, t_{v_j}^i}$ is indeed equal to the number of vertices at depth $i$ to which an agent of type $t_a$ has been positioned, by the way we have defined the variable assignment, thus equal to $t_a^i$. Hence, we can conclude that the constraint is indeed satisfied;

**LEF constraints**:

- First of all, observe that for any pair of agent-types $t_a, t'_a$ and for any pair of vertex-types $t_v, t'_v$, we have that $x_{t_a, t'_a, t_v, t'_v} \neq 0$ if and only if there is, somewhere in the tree, a parent vertex of type $t_v$, to which an agent of type $t_a$ has been assigned, which has a child vertex of type $t'_v$, to which an agent of type $t'_a$ has been assigned. Hence, as $\boldsymbol{\pi}$ and $\mu$ are LEF, it follows that an agent of type $t_a$ cannot envy one of type $t'_a$ as they are connected, meaning that $u_{i_{t_a}}(\pi_{i_{t_a}}) - u_{i_{t_a}}(\pi_{i_{t'_a}}) \geq 0$. Thus, it is easy to see that this last inequality implies that the constraint is satisfied;
- By a proof symmetric to the previous one, it also follows that the second kind of LEF constraints are satisfied under the variable assignment induced by $\mu$.

Hence, as all constraints are satisfied, it follows that the ILP has indeed a solution, as by our claim.

For the other direction, assume that there is a solution to the ILP, i.e. a variable assignment that satisfies all constraints. We prove that any position assignment $\mu$ which is naturally induced by a solution to the ILP is LEF when paired up with $\boldsymbol{\pi}$ for $I$. First, set $\mathcal{N}_0 := \mathcal{N}$: at each step $i$ we will assign some agent $a_i \in \mathcal{N}_i$ to some position in the tree (according to the agent-type we are considering) and we will then set $\mathcal{N}_{i+1} := \mathcal{N}_i \setminus \{a_i\}$. Observe that, as for each agent-types $t_a$ exactly $|\mathcal{N}_{t_a}|$ agents are assigned in total considering both the root and the child vertices in the tree (by the agent-conformity constraints), in the end we will have assigned exactly all the agents in $\mathcal{N}$.

We obtain such an assignment $\mu$ by starting from the root $r$ of the tree. Since the variable assignment satisfies the constraints, observe that there is only one pair of agent-type $t_a$ and vertex-type $t_v$ such that $r_{t_a,t_v} = 1$, and that $t_v$ is indeed the vertex-type of the root. Hence, to the root we assign an (arbitrary) agent $a$ of type $t_a$ and set $\mathcal{N}_1 := \mathcal{N}_0 \setminus \{a\}$.

Consider now an arbitrary child $w$, of some type $t'_v$, of the root $r$. As $r$, which is of type $t_v$, has a child vertex of type $t'_v$, observe that this implies that $e_{t_v,t'_v} > 0$; hence, as the root constraints are satisfied, this means that $\sum_{t'_a \in T_\mathcal{N}} x_{t_a,t'_a,t_v,t'_v} > 0$. Thus, take an arbitrary agent-type $t'_a$ such that $x_{t_a,t'_a,t_v,t'_v} > 0$ and we still have not assigned $x_{t_a,t'_a,t_v,t'_v}$ agents of such type to a vertex of type $t'_v$ which is a child of the root[6]: assign to $w$ an agent $a' \in \mathcal{N}_1$ of type $t'_a$ and set $\mathcal{N}_2$ accordingly. Notice that, as $x_{t_a,t'_a,t_v,t'_v} > 0$, we also get that the LEF constraints are satisfied for $t_a$ and $t'_a$, meaning that both the agent placed at the root cannot envy the one placed at the child vertex and vice versa. We can then iterate this procedure for all remaining child vertices of the root, as any of these will have some vertex-type $t''_v$, implying that $e_{t_v,t''_v} > 0$ which allows us to repeat the same argument.

Now, let $w$ be an arbitrary child vertex of type $t'_v$ of the root to which we have assigned some agent of type $t'_a$. Consider one of its children $w'$: this vertex must be of some vertex-type $t''_v$. Hence, it must be the case that $e_{t'_v,t''_v} > 0$. By construction, as an agent of type $t'_a$ was assigned to $w$, this means that $x_{t_a,t'_a,t_v,t'_v} > 0$ (where $t_a$ and $t_v$ are, as before, the agent-type of the agent assigned to the root and the vertex-type of the root). Thus, this implies that $\sum_{t''_a \in T_\mathcal{N}} x_{t'_a,t''_a,t'_v,t''_v} > 0$. In particular, such value is equal to the number of vertices of type $t''_v$ which are child of vertices of type $t'_v$ times the number of $t'_v$-vertices, i.e. each vertex of type $t''_v$ which is the child of one of type $t'_v$ will have some agent allocated to it. Thus, as we did previously, take an agent of any type $t''_a$ such that $x_{t'_a,t''_a,t'_v,t''_v} > 0$ and we still have not assigned all $t''_a$-agents to an edge of such type, and assign it to the child vertex $w'$. Again, as $x_{t'_a,t''_a,t'_v,t''_v} > 0$, by the LEF constraints it follows that the agent placed at $w$ cannot envy the one placed at $w'$ and vice versa. Finally, observe that this same argument can be repeated for all the remaining children of vertices at depth 1, and for the children of these and so on.

As observed before, since all agents for each agent-type are assigned, it follows that each agent is assigned to some position in the tree. Also, the

[6]In truth, for the first child of the root to which we have to assign an agent, it is clear that this will not be the case. On the other hand, this observation is needed in the remaining children of the root, as it could be the case that we will have already filled in all the "free" spots for agents of such type in these kind of edges.

fact that for any edge the two agents placed at its extremes do not envy each other (as ensured by the LEF constraints during the construction of $\mu$) implies that $\mu$ and $\boldsymbol{\pi}$ are LEF for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, thus proving the claim as by our previous observation. □

## 3.5 Forests

Nicely enough, it is possible to slightly alter the ILP we have previously defined so that it can be shown that the same claim holds also if the graph is a forest (i.e. a collection of trees).

**Theorem 3.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a forest,* FIND-LEF-POSITION-ASSIGNMENT *is in* FPT *if parameterized by the number of agent-types $|T_{\mathcal{N}}|$ and the number of vertex-types $|T_V|$.*

*Proof.* The approach is essentially the same, though with some minor adjustments to fit the differences concerning the social graph, which now is a forest instead of a tree. Intuitively, the problem in which one incurs when dealing with forests instead of trees is that it might happen that a tree in the forest is also a proper subtree of another tree in the forest. So, consider $t_v$ to be the type of the root of such (sub)tree: what we have to be careful about is to differentiate the two vertices, so that we are able to recognize which one is the root and which one is the inner vertex (and to do the same when considering their respective edges). Thus, we will now have the following variables in the ILP:

- For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$ a variable $x_{t_a,t'_a,t_v,t'_v}$ which has the same objective as the one in the proof of Theorem 2;
- For $t_a \in T_{\mathcal{N}}$ and $t_v \in T_V$ a variable $r_{t_a,t_v}$ which has the same objective as the one in the proof of Theorem 2;
- For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$ such that $t_v$ is a vertex-type of one of the forest's roots, a variable $x^r_{t_a,t'_a,t_v,t'_v}$ which has the same objective as its respective "non-root" variable, but has the extra condition that the parent vertex is a root of some tree in the forest.

Thus, the number of variables of the ILP can be upper bounded by $2|T_{\mathcal{N}}|^2|T_V|^2 + |T_{\mathcal{N}}||T_V|$, which is still clearly a computable function of the parameters, thus if the reduction is correct then it is an FPT one, proving the claim as in the analogous result for trees.

The constraints of the ILP also vary:

**Integrity constraints**. We remove the constraints which regarded the root variables, and introduce a new set of constraints for such variables:

– For $t_a \in T_{\mathcal{N}}$ and $t_v \in T_V$:

$$r_{t_a,t_v} \in \mathbb{N}_0$$

This ensures that the number of times an arbitrary agent-type is assigned to a vertex-type for a root is a non-negative integer;

**Root constraints**. We modify the constraints which verify that the edges starting from the roots are correct using the new variables:

– For $t_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$:

$$\sum_{t'_a \in T_{\mathcal{N}}} x^r_{t_a,t'_a,t_v,t'_v} - r_{t_a,t_v} e_{t_v,t'_v} = 0$$

**Network conformity constraints**. As we have introduced a new set of variables, we want to "link" these ones with the standard ones we also had before. Thus, we substitute the previous set of network conformity constraints with the following, which goal is to have that, for any $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$, $x_{t_a,t'_a,t_v,t'_v}$ considers both cases in which the vertex of type $t_v$ is a root or not in the total count of the outgoing edges from vertices of such type:

– For $t_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$ (now we also include the vertex-types $t_v$ that are the types of roots in the forest):

$$\sum_{t'_a \in T_{\mathcal{N}}} \left( x_{t_a,t'_a,t_v,t'_v} - x^r_{t_a,t'_a,t_v,t'_v} \right) - e_{t_v,t'_v} \sum_{t'_a \in T_{\mathcal{N}}} \sum_{t''_v \in T_V} x_{t'_a,t_a,t''_v,t_v} = 0$$

Observe that we know that $x^r_{t_a,t'_a,t_v,t'_v}$ is the correct number (i.e. the number of roots of type $t_v$ to which an agent of type $t_a$ has been assigned times $e_{t_v,t'_v}$) by the new root constraints.

As before, we show that there is an LEF position assignment $\mu$ if and only if there is a solution to the ILP defined in the reduction.

For the left-to-right direction, assume again that there is such a position assignment $\mu$. We yet consider the variable assignment which is naturally induced by $\mu$. We will omit constraints that were not changed, as we have already shown that they are satisfied by the variable assignment we are considering: while for some the proof is identical for others some slight adjustments are needed, mainly to adapt the proof to the fact that now we are considering a forest (hence there might be multiple roots) instead of a tree. Hence, it remains to check that newly introduced constraints are satisfied as well:

**Integrity constraints**: the new constraints are trivially satisfied as, for an arbitrary agent-type $t_a$ and arbitrary vertex-type $t_v$, the number of times an agent of type $t_a$ is placed at a root of type $t_v$ is obviously an integer greater than or equal to 0;

**Root constraints**: as in the proof for the analogous constraints, there are various cases. If $t_a, t_v$ and $t'_v$ are such that either $r_{t_a,t_v} = 0$ or $e_{t_v,t'_v} = 0$, then it must follow that $x^r_{t_a,t'_a,t_v,t'_v} = 0$ (for arbitrary $t'_a, t'_v$ in the first case and for arbitrary $t_a, t'_a$ in the second one). Instead, if for some $t_a, t_v$ and $t'_v$ both $r_{t_a,t_v}$ and $e_{t_v,t'_v}$ are greater than 0, then this means that there are exactly $r_{t_a,t_v}$ roots of type $t_v$ where an agent of type $t_a$ has been placed. By how the variable assignment was defined, this means that the sum across all agent-types $t'_a$ of the variables $x^r_{t_a,t'_a,t_v,t'_v}$ is exactly $e_{t_v,t'_v}$ for each such root. Hence, the constraint is satisfied;

**Network conformity constraints**: again, we have two different cases. Let $t_a$ and $t_v$ be respectively an arbitrary agent-type and an arbitrary vertex-type. If no agent of type $t_a$ has been assigned to a vertex of type $t_v$, then it follows that there is no edge connecting a root of type $t_v$ to which an agent of type $t_a$ has been assigned, and for another pair consisting of an arbitrary agent-type $t'_a$ and an arbitrary vertex-type $t'_v$, there is no edge connecting a parent vertex of type $t'_v$, to which an agent of type $t'_a$ has been assigned, to a child vertex of type $t_v$, to which an agent of type $t_a$ has been assigned; hence, in this case the constraint is satisfied. On the other hand, assume that there is at least an agent of type $t_a$ which has been assigned to a vertex of type $t_v$ and fix an arbitrary vertex-type $t'_v$. If $e_{t_v,t'_v} = 0$, then the constraint is yet satisfied. Finally, if $e_{t_v,t'_v} \neq 0$, then the number of times a parent vertex of type $t_v$, to which an agent of type $t_a$ has been assigned, is connected to a vertex of type $t'_v$, to which an agent of an arbitrary type $t'_a$ has been assigned, is indeed equal to the number of times such a parent vertex is a child of some other vertex times $e_{t_v,t'_v}$ (this holds by definition of vertex-type itself) plus the number of edges in which it is the root of some tree in the forest and it is connected to a vertex of type $t'_v$, again to which an agent of some arbitrary type $t'_a$ has been assigned. Thus, in any case the constraint is satisfied.

For the right-to-left direction, assume now that there is a solution to the ILP. As in the previous proof, we will build a position assignment $\mu$ which will also be an LEF one for the original input instance $I$. The construction is essentially the same, though in the initial case we account also for the fact that there are multiple roots. This can be easily handled using the root variables $r_{t_a,t_v}$ in the same way they were used previously, as now for a single vertex-type the root variables sum up to the number of roots of such type over all agent-types. Once agents have been assigned to the roots, the rest of the position assignment is defined in the same way as it was done before, using the edge variables $x_{t_a,t'_a,t_v,t'_v}$. Hence, the claim follows. □

We will now show a similar result, however instead of using the two parameters we have defined in the previous section, we will use only the number of vertex-types. To prove this new result, we will use the EXACT COVER BY 3-SETS problem, which is a known NP-complete problem in the literature (Garey and Johnson [1979]). Notice that we will not use the classical version of the problem, but a restricted one in which each element appears in exactly three triplets. This variant is due to Gonzalez [1985], in which it is also shown to be NP-complete.

We will show that EXISTS-LEF-POSITION-ASSIGNMENT is NP-complete even when the number of vertex-types is a constant value which does not depend on the size of the instance.

**Theorem 4.** *Let $\mathcal{G}$ be the class of graphs such that each graph $G \in \mathcal{G}$ has exactly 3 vertex-types.* EXISTS-LEF-POSITION-ASSIGNMENT, *restricted to graphs in $\mathcal{G}$, is* NP*-complete.*

*Proof.* Membership in NP follows by the fact that it is possible to check if a position assignment $\mu$ (the polynomial certificate in our case) is a solution for a given input instance $I$ by simply checking in polynomial time whether all agents connected by $\mu$ do not envy each other under the current item allocation $\boldsymbol{\pi}$. Observe that this holds for any arbitrary instance, thus also for any instance such that the social graph has only 3 different vertex-types.

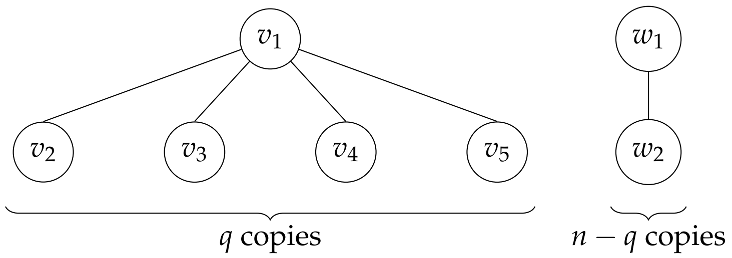
To show NP-hardness, we give a polynomial-time reduction from EXACT COVER BY 3-SETS. Let $I = \langle X, S \rangle$ be an instance for EXACT COVER BY 3-SETS, where $X = \{x_1, \ldots, x_{3q}\}$ and $S = \{S_1, \ldots, S_n\}$. The reduction is given as follows:

- $\mathcal{N} := \mathcal{N}_X \cup \mathcal{N}_S \cup \mathcal{N}_D$, where:
  - $\mathcal{N}_X := \{a_1, \ldots, a_{3q}\}$, i.e. one agent per element of $X$. We will call these "*element-agents*";
  - $\mathcal{N}_S := \{a_{S_1}, \ldots, a_{S_n}\}$, i.e. one agent per triplet of $S$. We will call these "*triplet-agents*";
  - $\mathcal{N}_D := \{a_{d_1}, \ldots, a_{d_n}\}$, i.e. one dummy agent for each triplet of $S$;
- $\mathcal{O} := \{o_1, \ldots, o_{3q}, o_{S_1}, \ldots, o_{S_n}, o_{d_1}, \ldots, o_{d_n}\}$: there is a corresponding item in $\mathcal{O}$ for each agent in $\mathcal{N}$;
- $\boldsymbol{u} = (u_1, \ldots, u_{3q}, u_{S_1}, \ldots, u_{S_n}, u_{d_1}, \ldots, u_{d_n})$, where:
  - For an arbitrary $u_i$, i.e. the utility of the element-agent $a_i$:
    * $u_i(o_i) = 1$;

  * $u_i(o_{S_j}) = 0$ for all $j$ such that $x_i \in S_j$: agent $a_i$ does not care about any item which corresponds to any of the triplets which contain her corresponding element;
  * $u_i(o) = 2$ for any other $o \in \mathcal{O}$;
- For an arbitrary $u_{S_i}$, i.e. the utility of the triplet-agent $a_{S_i}$:
  * $u_{S_i}(o_{S_i}) = 1$;
  * $u_{S_i}(o_j) = 0$ for all $j$ such that $x_j \in S_i$: agent $a_{S_i}$ does not care about the item which corresponds to any of the elements in her corresponding triplet;
  * $u_{S_i}(o_{d_i}) = 0$: agent $a_{S_i}$ does not care about the item which corresponds to the dummy agent of her triplet;
  * $u_{S_i}(o) = 2$ for any other $o \in \mathcal{O}$;
- For an arbitrary $u_{d_i}$, i.e. the utility of the dummy-agent $a_{d_i}$:
  * $u_{d_i}(o_{d_i}) = 1$;
  * $u_{d_i}(o_{S_i}) = 0$: agent $a_{d_i}$ does not care about the item which corresponds to her associated triplet $S_i$;
  * $u_{d_i}(o) = 2$ for any other $o \in \mathcal{O}$;

- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{3q}, \pi_{S_1}, \ldots, \pi_{S_n}, \pi_{d_1}, \ldots, \pi_{d_n})$ where $\pi_i = \{o_i\}$, $\pi_{S_i} = \{o_{S_i}\}$ and $\pi_{d_i} = \{o_{d_i}\}$;
- $G = (V, E)$ is a graph which consists of $q$ stars with 5 vertices and $n - q$ matchings of two vertices.

As we will shortly see, in position assignments that are LEF, central vertices of the stars will be occupied by triplet-agents, while the outer vertices by the corresponding element-agents and dummy agents. The matchings will be occupied by triplet-agents and their corresponding dummy agents. The graph $G$ built by the reduction is illustrated below.




Clearly $G$ has exactly 3 vertex-types, one for the roots of the stars with 5 vertices, one for the "roots" of the matchings and one for the leafs. Moreover, it is easy to see that the reduction is a polynomial-time one.

Hence, it remains to show that the reduction is correct. Assume that there is a subset $S' = \{S'_1, \ldots, S'_q\}$ of $S$ such that each element of $X$ appears in exactly one triplet of $S'$. Then consider the position assignment $\mu$ which is defined as follows:

- For the $q$ stars with 5 vertices, it assigns to the roots the triplet-agents $a_{S'_1}, \ldots, a_{S'_q}$ and to the leafs the dummy agents and the 3 element-agents associated to the corresponding triplet of the star. Observe that for each of these stars exactly 5 agents have been assigned, and that no agent envies each other;
- For the $n - q$ matchings, it assigns to the "roots" the triplet-agents in $S \setminus S'$ (which are exactly $n - q$) and to the leafs the dummy agents associated to the corresponding triplet of the matching. Again, observe that exactly 2 agents are assigned in each of these matchings and that no one envies each other.

Hence, there is indeed a position assignment $\mu$ which is LEF when paired up with $\boldsymbol{\pi}$ for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$.

For the other direction, assume that there is an LEF position assignment $\mu$. To prove that there is a subset of triplets that covers $X$, we will prove two different claims:

1. Agents placed at the roots of the $q$ stars can only be triplet-agents;
2. Agents placed at the $n - q$ matchings can only be pairs composed of a triplet-agent and her corresponding dummy agent.

To prove the first, assume by way of contradiction that $\mu$ places at the root of one of the $q$ stars an agent that is not a triplet-agent. Then this agent is either a dummy agent or an element-agent: in the former case, observe that each dummy agent does not envy only her corresponding triplet-agent under $\boldsymbol{\pi}$, and since she is connected to four other agents, she will necessarily envy someone. For the former we make the same argument, except the fact that we have to consider that as each element appears in exactly three triplets, the element-agent will necessarily envy the fourth agent to which she is connected (as it might happen that the other three are the corresponding triplet-agents). Thus, the claim is shown as we know by assumption that $\mu$ is LEF.

For the second claim, assume again by way of contradiction that $\mu$ places at some matching a pair which is not composed of a triplet-agent and her corresponding dummy agent. If the dummy agent is not the one corresponding to the triplet-agent, then trivially $\mu$ cannot be LEF as they will both envy each other. To conclude, there are two remaining cases: either this pair is made of an element-agent and a dummy agent or an element-agent and a triplet-agent. In the former case $\mu$ is trivially non-LEF, as the agents envy each other, in the latter case either the element-agent corresponds to an element which is in the corresponding triplet of the triplet-agent or not. If the element does not belong to the triplet, then $\mu$ is not LEF by the same argument as in the previous case, if instead the element is in the triplet, then recall that the dummy agent who is associated to the triplet-agent will necessarily envy the agent(s) to whom she is connected to, as the only one not envied by her is her corresponding triplet-agent. □

By Definition 17, this will imply that the same problem, when parameterized only by the number of vertex-types, is para-NP-hard.

**Corollary 2.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a forest,* EXISTS-LEF-POSITION-ASSIGNMENT *is para-*NP*-hard if parameterized by the number of vertex-types $|T_V|$.*

## 3.6 (Definitely-Non-)Quasi-Trees

For the final section of this chapter, we will divide graphs using their treewidth. We will consider graphs $G$ that have a constant treewidth, i.e. such that $\text{tw}(G) = O(1)$, which intuitively are all those that look very similar to trees or forests, and, on the opposite side, graphs with a non-constant treewidth, i.e. such that $\text{tw}(G) \neq O(1)$. To give an example, graphs that have a constant treewidth but which we did not mention so far are (collections of) cycles, which treewidth is equal to 2. On the other hand, graphs with non-constant treewidths are all cliques (as already mentioned in Chapter 2), as their treewidth is equal to the number of vertices in the clique minus one.

The parameter we will use is the number of non-isolated vertices in the social graph $G$, where a vertex is "isolated" in a graph if and only if it is not connected to any other vertex. In most cases, this parameter is equal to the number of vertices of the social graph and, thus, the number of agents. Notice that, if we were to use the number of agents $|\mathcal{N}|$ as the parameter, EXISTS-LEF-POSITION-ASSIGNMENT would trivially be in FPT because it suffices to simply loop over all possible position assignment and check whether there is one that is LEF. Since there are $|\mathcal{N}|!$ possible position assignments and checking whether an assignment is LEF takes $O(|\mathcal{N}|^2)$, the claim trivially follows.

However, our main interest in this parameter is not in using it for a positive result, but for a negative one. As we will see, EXISTS-LEF-POSITION-ASSIGNMENT is in FPT when using such parameter and if the input instance has a constant treewidth, however, when the treewidth is non-constant, the problem becomes $W[1]$-hard, which should give the reader a good feeling for how difficult the problem is, considering the parameter we will use.

**Proposition 6.** *Let $\mathcal{G}$ be a class of graphs such that each graph $G \in \mathcal{G}$ has a constant treewidth, i.e.* $\text{tw}(G) = O(1)$*.* EXISTS-LEF-POSITION-ASSIGNMENT*, restricted to graphs in $\mathcal{G}$, is in* FPT *if parameterized by the number of non-isolated vertices in $G$.*

*Proof.* The proof consists in reducing an instance for EXISTS-LEF-POSITION-ASSIGNMENT to one for SUBGRAPH ISOMORPHISM, such that $\text{tw}(H) = O(1)$, through an FPT reduction and by observing a property of the EXISTS-LEF-POSITION-ASSIGNMENT problem.

We first give the reduction, which is quite simple. For an arbitrary instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, the graph $H$ is a copy of $G$ but without the isolated vertices, while the graph $G'$ is the envy-free graph induced by the instance $I$. Notice that $\text{tw}(H) = O(1)$, since by assumption $\text{tw}(G) = O(1)$, and the number of vertices of $H$ is exactly the number of non-isolated vertices in $G$. Thus, as the reduction takes polynomial time in the size of the input instance, it remains to check that the reduction is a correct one for it to be an FPT one.

The correctness of the reduction follows from observing that an arbitrary instance $I$ has a solution if and only if the non-isolated subgraph of the social graph $G$ is isomorphic to some subgraph of the envy-free graph $G_{EF}$ induced by $I$ (as the agents placed at the isolated vertices trivially cannot envy and be envied by no one). If there is an LEF position assignment $\mu$, then it means that we can map vertices of the non-isolated subgraph of $G$ to a subgraph of the envy-free graph (by considering the mapping induced by the position assignment $\mu$): as $\mu$ is LEF this implies that if two agents are connected in $G$ then they will also be connected in the envy-free graph, and trivially the mapping is injective as no agent can be assigned to two different positions, meaning that there is indeed such an isomorphism.

On the other hand, if the non-isolated subgraph of $G$ and some subgraph of the envy-free graph $G_{EF}$ are isomorphic, then there must be an injective mapping from the former to the latter such that if there is an edge between two vertices in the first graph then there is one between the images of these two vertices in the second one. The position assignment $\mu$ induced by the isomorphism (where the remaining agents are assigned in an arbitrary way to the isolated vertices) must be an LEF one, as agents who are connected in $G$ are also connected in the envy-free graph $G_{EF}$.

Hence, as the FPT reduction is correct, SUBGRAPH ISOMORPHISM is in FPT when parameterized by $|V(H)|$ and $H$ has a constant treewidth (i.e.

$\text{tw}(H) = O(1)$), the claim follows. □

We now prove our last hardness result about LEF. To prove it, we will use the CLIQUE problem parameterized by the size $k$ of the clique.

**Theorem 5.** *Let $\mathcal{G}$ be a class of graphs containing all graphs $G$ consisting of a clique together with some isolated vertices—and thus $\mathcal{G}$ contains graphs of unbounded treewidth.* EXISTS-LEF-POSITION-ASSIGNMENT, *restricted to graphs in $\mathcal{G}$, is $W[1]$-hard if parameterized by the number of non-isolated vertices in $G$.*

*Proof.* We prove the claim by giving an FPT reduction from the CLIQUE problem parameterized by the integer $k$, which is a $W[1]$-hard problem, to our problem. Given an input instance $\langle G = (V, E), k\rangle$, where $V = \{v_1, \dots, v_n\}$, for the CLIQUE problem, the built instance $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' = (V', E')\rangle$ is defined as follows:

- $\mathcal{N} = \{1, \dots, n\}$, i.e. there is a corresponding agent $i$ for each vertex $v \in V$;
- $\mathcal{O} = \{o_1, \dots, o_n\}$, i.e. there is a corresponding item $o_i$ for each agent $i$;
- $\boldsymbol{u} = (u_1, \dots, u_n)$ where for any $1 \leq i \leq n$:
  - $u_i(o_i) = 0$;
  - $u_i(o_j) = 0$ for $i, j$ such that $\{v_i, v_j\} \in E$: for any pair of connected vertices of $G$ their two agents do not envy each other;
  - $u_i(o_j) = 1$ for $j \neq i$ such that $\{v_i, v_j\} \notin E$: for any pair of non-connected vertices of $G$ their two agents envy each other;
- $\boldsymbol{\pi} = (\pi_1, \dots, \pi_n)$ where $\pi_i = \{o_i\}$ for any $1 \leq i \leq n$.
- $G'$ is defined as follows:
  - $V' = \{v'_1, \dots, v'_n\}$;
  - $E = \{\{v'_i, v'_j\} \mid i, j \leq k\}$.

The construction we have just defined has two key ideas:

1. By how $\boldsymbol{u}$ and $\boldsymbol{\pi}$ are defined, it is clear that the envy-free graph induced by the built instance is a copy of $G$;
2. The social graph $G'$ is a clique of size $k$ with a set of isolated vertices so that it has the same number of vertices as $G$.

Observe that the number of non-isolated vertices in $G'$ is indeed bounded by a computable function of $k$, as it is exactly equal to it. Also, as we have mentioned in the preliminaries, the treewidth of $G'$ is $k - 1$ (as $G'$ is a clique of $k$ vertices), thus it is not constant. Since the reduction can be trivially performed in polynomial time with respect to the size of the CLIQUE instance, it remains just to check that it is a correct one.

Thus, we show that that there is a clique of size $k$ in $G$ if and only if there is a position assignment $\mu$ which is a LEF for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' \rangle$. By how $G'$ was built, there is a position assignment $\mu$ which is LEF with $\boldsymbol{\pi}$ if and only if there is a clique of size $k$ in the envy-free graph induced by the instance we have built for EXISTS-LEF-POSITION-ASSIGNMENT. In fact, if there is such a clique, then the position assignment that maps agents involved in such clique to the vertices in the clique of $G'$ (and the remaining agents to the isolated vertices) is an LEF one: the agents who are assigned to a vertex in the clique do not envy each other since they are connected in the envy-free graph, while the remaining ones have no one to envy or be envied by.

On the other hand, if there is a position assignment $\mu$ which is LEF, then observe that this implies that the envy-free graph does indeed contain a clique of size $k$, as no pair of agents assigned to connected vertices in $G'$ envies each other. This concludes the proof because the envy-free graph is a copy of $G$, thus $G$ itself must contain a clique of size $k$ as well. □

## 3.7 Summary

This concludes the first chapter with theoretical results in the thesis. As we have seen, in most of the cases deciding whether there is, and thus also finding, a position assignment $\mu$ which is LEF is an intractable problem. The only exceptions to this intractability are matchings and stars. On the other hand, we were able to obtain much more positive results in parameterized complexity. We think that, at least for this topic of research, it is definitely more profitable to pursue results in this direction.

In the following chapter we will consider local envy-freeness up to one item, the second criterion we have introduced, which is strictly tied to local envy-freeness. Compared to LEF, we will see more positive results for LEF1, in particular when the item allocation $\boldsymbol{\pi}$ is not fixed.

# CHAPTER 4

# COMPLEXITY OF LOCAL ENVY-FREE UP TO ONE ITEM ASSIGNMENTS

We now consider the second fairness criterion which we have established in the introduction, local envy-freeness up to one item. The strategy that we will use to prove the results of this chapter is essentially to adapt proofs which we have done in Chapter 3 for local envy-freeness to local envy-freeness up to one item. In most cases this will boil down to defining in a slightly more precise way the utility functions and the item allocations, so that envy-freeness up to one item is taken into account.

Before moving on, we would like to point out a mistake in which one might fall if not too careful, and which extends to local proportionality as well. Looking back at Figure 2.1 and considering the positive results which were achieved in the previous chapter, one might be tempted to declare that such results also hold for local envy-freeness up to one item. However, the pitfall here is the fact that, although position assignments which are LEF are also LEF1, it might happen that for some instances there is no LEF position assignment but there is an LEF1 one. Therefore, proofs of tractability of local envy-freeness do not necessarily follow also for local envy-freeness up to one item (and local proportionality). However, unlike what we will see in the following chapter about local proportionality, luckily in the case of local envy-freeness up to one item all positive results for local envy-freeness do indeed carry over.

Moreover, we will also be able to obtain positive results for FIND-LEF1-DISTRIBUTION, because by Proposition 1 we know that it is possible to always find an LEF1 item assignment in polynomial time, hence by giving also an arbitrary position assignment we obtain a distribution which satisfies such criterion. This also implies that the answer to EXISTS-LEF1-DISTRIBUTION is always "Yes", regardless of the input instance.

**Corollary 3.** EXISTS-LEF1-DISTRIBUTION *and* FIND-LEF1-DISTRIBUTION *can be computed respectively in constant and polynomial time for any instance.*

As a general rule of thumb, proofs of negative results for local envy-freeness will be changed to take into account the following observation:

**Observation 3.** *If $\boldsymbol{\pi}$ is such that each agent is assigned exactly one item and has a positive utility under $\boldsymbol{u}$, then it is (local when paired up with an arbitrary $\mu$) envy-free up to one item.*

To circumvent this property of local envy-freeness up to one item, we will simply assign to each agent more than one item.

## 4.1 Lines, Stars and Strongly Connected Graphs

As in Chapter 3, we first focus on lines and stars, which have the same behaviour they had in the previous chapter. We also add a small discussion about the more general family of strongly connected graphs in this section because it turns out that, thanks to Corollary 3, all problems which were mentioned in the previous chapter for these graphs are now tractable.

**Proposition 7.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a star,* EXISTS-LEF1-POSITION-ASSIGNMENT *is decidable in polynomial time.*

*Proof.* To prove the claim, we give a simple polynomial-time algorithm which answers the question. Similarly to before, it suffices to simply loop over all agents and check that the following conditions hold:

1. That the current agent $i$ is such that for any other agent $j \in \mathcal{N} \setminus \{i\}$ either $u_i(\pi_i) \geq u_i(\pi_j)$ or there is one item $o \in \pi_i \cup \pi_j$ such that $u_i(\pi_i \setminus \{o\}) \geq u_i(\pi_j \setminus \{o\})$;
2. That any other agent $j \in \mathcal{N} \setminus \{i\}$ is such that either $u_j(\pi_j) \geq u_j(\pi_i)$ or there is one item $o \in \pi_i \cup \pi_j$ such that $u_j(\pi_j \setminus \{o\}) \geq u_j(\pi_i \setminus \{o\})$.

Again, if there is such an agent $i$, the algorithm outputs immediately "Yes", otherwise, if after the loop no such agent has been found, it outputs "No". We quickly discuss the complexity of the algorithm as in this case it is not just quadratic in the number of agents, but involves the number of items as well. In fact, the algorithm runs in a time of $O(|\mathcal{N}|^2|\mathcal{O}|)$, as when we have to check envy-freeness up to one item we will loop over the set of items as well. Nevertheless, the algorithm is still clearly polynomial in the input size.

At this point, we omit the proof as it is exactly the same as the one for local envy-freeness, albeit using local envy-freeness up to one item in its place. It is in fact clear that the conditions which we check for each agent enforce the position assignment to be LEF1 given the fixed item allocation $\pi$. □

In the same manner of the previous chapter, this result implies the tractability of FIND-LEF1-POSITION-ASSIGNMENT when the underlying social graph is a star.

Regarding lines, we will use again the HAM-PATH problem to show its NP-completeness.

**Proposition 8.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G\rangle$ in which $G$ is a line,* EXISTS-LEF1-POSITION-ASSIGNMENT *is* NP*-complete.*

*Proof.* Membership in NP is proved in the usual way: we can check in polynomial time whether a position assignment $\mu$ (the polynomial certificate) is LEF1 for the given input instance because we just need to loop over all the agents and check that they do not envy their neighbors up to one item, which can trivially be done in $O(|\mathcal{N}|^2|\mathcal{O}|)$ time.

The reduction we give from an arbitrary input instance $G = (V, E)$ for HAM-PATH differs from the one for local envy-freeness in the set of items, the item allocation and the utility functions in the following way:

- $\mathcal{O} = \{o_1^1, o_1^2, \ldots, o_n^1, o_n^2\}$, i.e. there are two items $o_i^1, o_i^2$ for each vertex $v_i \in V$;
- For each agent $i$ let $o_{i_1}^1, o_{i_1}^2, \ldots, o_{i_m}^1, o_{i_m}^2$ and $o_{i'_1}^1, o_{i'_1}^2, \ldots, o_{i'_{m'}}^1, o_{i'_{m'}}^2$ be the items which correspond respectively to the $m$ agents in $\overline{N}(v_i)$ (as defined in the proof of Proposition 3) and the $m'$ agents in $N(v_i)$. The utility function $u_i$ is defined in the following way:
  - $u_i(o_i^1) = u_i(o_i^2) = u_i(o_{i'_k}^j) = 0$ for any $k \in [m']$ and $j \in [2]$;
  - $u_i(o_{i_k}^j) = 1$ for any $k \in [m]$ and $j \in [2]$;
- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ where $\pi_i = \{o_i^1, o_i^2\}$ for any $i \in \mathcal{N}$.

Trivially the reduction is still polynomial-time. We now show that it is a correct one.

Assume that there is an LEF1 position assignment $\mu$ for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{a}, \boldsymbol{\pi}, G'\rangle$. To prove that there is a Hamiltonian path, it suffices to notice that two agents who are connected by $\mu$ do not envy each other up to one item if and only if their respective vertices are connected by an edge in $G$, similarly to what happened with local envy-freeness. In fact, suppose that two agents $i, j$ such that $\{v_i, v_j\} \notin E$ are connected under $\mu$: then $u_i(\pi_j) = 2$ and, regardless of the item we remove from $\pi_j$ to obtain a new bundle $\pi'_j$, we have that $u_i(\pi'_j) = 1$, whereas $u_i(\pi_i) = 0$, i.e. $i$ does envy $j$ up to one item (and the same holds vice versa). On the other hand, if $i, j$ are such that $\{v_i, v_j\} \in E$, then they both do not envy each other up to one item because they simply do not envy each other. Hence, by the same argument we made in the proof of Proposition 3, it follows that $G$ must contain a Hamiltonian path.

For the other direction, assume that there is a Hamiltonian path $P = v_{p_1}, \ldots, v_{p_n}$ in $G$. Consider the corresponding position assignment $\mu$ such that the $i$-th agent on the line is $p_i$: it is easy to see that it is LEF1 because two agents are connected if and only if they correspond to two vertices which are consecutive in the path, meaning that the two vertices are also connected in the original graph. As the vertices are connected, it follows that the two agents do not envy each other up to one item (in fact they do not envy each other at all), thus proving the claim. □

As anticipated before, we conclude the section by briefly discussing the general class of strongly connected graphs. At this point, we would mention that because of the result we have just proven, it follows that EXISTS-LEF1-POSITION-ASSIGNMENT is NP-complete when the underlying social graph is strongly connected. Naturally, we would then move on to see what happens with EXISTS-LEF1-DISTRIBUTION, as we have not treated it when considering the previous families of graphs. This discussion however already ends when we recall Corollary 3, which grants us the following result:

**Theorem 6.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ where $G$ is a strongly connected graph,* EXISTS-LEF1-DISTRIBUTION *is computable in constant time and* FIND-LEF1-DISTRIBUTION *in polynomial time.*

At the end of this section we are finally able to observe for the first time the extreme difference between local envy-freeness up to one item and the other two criteria, local envy-freeness and local proportionality (which we will see in the next chapter). Counter-intuitively, because here the item allocation is not fixed, a problem which should be more difficult than its counterpart, where the item allocation is instead fixed, becomes tractable.

## 4.2 Matchings

To prove the tractability of EXISTS-LEF1-POSITION-ASSIGNMENT (and subsequently FIND-LEF1-POSITION-ASSIGNMENT) when the underlying graph is a matching, it suffices to make the following observation.

**Observation 4.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ where $G$ is a matching, there is a position assignment $\mu$ which is LEF1 for $I$ if and only if the envy-free up to one item graph $G_{EF1} = (V_{EF1}, E_{EF1})$ admits a perfect matching.*

The proof of this is identical to the proof of the fact that an instance $I$ where the social graph is a matching admits an LEF position assignment if and only if there is a perfect matching in the envy-free graph $G_{EF}$ induced by it, which was showed in the proof of Proposition 4.

Then, by simply noticing that the envy-free up to one item graph can be computed as well in polynomial time in the size of $I$ as the envy-free graph from Proposition 4, the claim follows by using again the same algorithm from Micali and Vazirani [1980] to decide whether there is a perfect matching in polynomial time. Hence, we get the following result.

**Proposition 9.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a matching,* EXISTS-LEF1-POSITION-ASSIGNMENT *is decidable in polynomial time.*

Which implies tractability for FIND-LEF1-POSITION-ASSIGNMENT as well, since the algorithm from Micali and Vazirani [1980] outputs a perfect matching if there is any.

To conclude, from Corollary 3 we can obtain the tractability of EXISTS-LEF1-DISTRIBUTION and FIND-LEF1-DISTRIBUTION.

**Theorem 7.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ where $G$ is a matching,* EXISTS-LEF1-DISTRIBUTION *is decidable in constant time and* FIND-LEF1-DISTRIBUTION *is computable in polynomial time.*

## 4.3 Trees

We will now move on to results which are concerned with parameterized complexity, as we did in Chapter 3.

The first concept we define is one which the reader will find familiar, as it is the counterpart for local envy-freeness up to one item of one we have already introduced for local envy-freeness in Chapter 3. Consider an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ and the directed envy-free up to one item graph $G_{EF1} = (V_{EF1}, E_{EF1})$ induced by it.

**Definition 22** (EF1 Agent-type)**.** Two agents $i, j \in \mathcal{N}$ have the same EF1 agent-type $t_a$ if and only if, for any agent $k \in \mathcal{N}$, it holds that:

- $(i,k) \in E_{EF1} \iff (j,k) \in E_{EF1}$: either both $i$ and $j$ do not envy $k$ up to one item or they both do;
- $(k,i) \in E_{EF1} \iff (k,j) \in E_{EF1}$: either $k$ does not envy both $i$ and $j$ up to one item or she does.

As before, the following fact holds by an argument similar to the one we provided for the EF agent-types.

**Fact 8.** *Given the relation*

$$T_n = \{(i,j) \mid i,j \in \mathcal{N}, \forall k \in \mathcal{N}.[((i,k) \in E_{EF1} \iff (j,k) \in E_{EF1}) \\ \wedge ((k,i) \in E_{EF1} \iff (k,j) \in E_{EF1})]\}$$

*each agent-type denotes an equivalence class for $T_n$.*

Before moving to the first result of this section, we will introduce another key object which will prove to be quite useful in the following proof. Again, consider an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ and the directed envy-free up to one item graph $G_{EF1} = (V_{EF1}, E_{EF1})$ induced by it. To define the ILP as we did in the proof of the analogous results for local envy-freeness, we will use a (slightly modified) *characteristic function* of the set of edges of $G_{EF1}$, which we will denote with $\mathbb{1}_{E_{EF1}}$ and is defined as follows:

$$\mathbb{1}_{E_{EF1}}((i,j)) = \begin{cases} 1 & \text{if } (i,j) \in E_{EF1} \\ -1 & \text{otherwise} \end{cases}$$

Clearly given in input the instance $I$, this function can be computed in polynomial time since the directed envy-free up to one item graph can be computed in polynomial time as well.

**Fact 9.** *Given an arbitrary instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, the characteristic function $\mathbb{1}_{E_{EF1}}$ can be computed in polynomial time.*

Moreover, by taking a representative $i_{t_a}$ for each agent-type $t_a \in T_{\mathcal{N}}$, we can also define an analogous function for EF1 agent-types, which will be denoted with $\mathbb{1}_T$, such that $\mathbb{1}_T(t_a, t'_a) := \mathbb{1}_{E_{EF1}}((i_{t_a}, i_{t'_a}))$. It is clear that this function can be computed in polynomial time given the characteristic function $\mathbb{1}_{E_{EF1}}$.

**Lemma 2.** *Let $i, j \in \mathcal{N}$ be agents of, respectively, EF1 agent-types $t_a, t'_a$, then $i$ does not envy $j$ up to one item if and only if $\mathbb{1}_T(t_a, t'_a) = 1$.*

The lemma follows from the simple observation that, if we consider the representatives $i_{t_a}, i_{t'_a}$ of EF1 agent-types $t_a, t'_a$, then we know, by definition of EF1 agent-type itself, that $(i,j) \in E_{EF1} \iff (i_{t_a}, i_{t'_a}) \in E_{EF1}$ and, by how $\mathbb{1}_T$ was built, that $(i_{t_a}, i_{t'_a}) \in E_{EF1} \iff \mathbb{1}_T(t_a, t'_a) = 1$. Thus, $(i,j) \in E_{EF1} \iff \mathbb{1}_T(t_a, t'_a) = 1$, as stated by the claim.

We are now ready to prove the result analogous to Theorem 2 but for envy-freeness up to one item.

**Theorem 8.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a tree,* FIND-LEF1-POSITION-ASSIGNMENT *is in* FPT *if parameterized by the number of EF1 agent-types $|T_{\mathcal{N}}|$ and the number of vertex-types $|T_V|$.*

*Proof.* The proof consists of another reduction to the INTEGER LINEAR PROGRAM PROBLEM, so that the number of variables is polynomial in the number of agent-types and vertex-types.

In fact, the built ILP will have the same set of variables, and will only change in what we previously called "*LEF constraints*" which will be replaced by the, rather unimaginatively named, "*LEF1 constraints*". In order to define such constraints, we first must compute the function $\mathbb{1}_T$. Due to Fact 9 we know that such construction can be done in polynomial time in the size of $I$, hence the reduction is still FPT.

The new constraints are defined in the following way:

**LEF1 constraints:**

– For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$:

$$x_{t_a,t'_a,t_v,t'_v} \mathbb{1}_T(t_a, t'_a) \geq 0$$

– For $t_a, t'_a \in T_{\mathcal{N}}$ and $t_v, t'_v \in T_V$:

$$x_{t_a,t'_a,t_v,t'_v} \mathbb{1}_T(t'_a, t_a) \geq 0$$

We now prove that the reduction is again a correct one, i.e. that there is an LEF1 position assignment $\mu$ for the instance $I$ if and only if the built ILP has a solution.

For the left-to-right direction, assume that there is some LEF1 position assignment $\mu$. The assignment of variables which we will consider is the natural one induced by $\mu$, as it was defined in the proof of Theorem 2. Since all the constraints except for the LEF1 ones are the same, we omit the proof that the induced variable assignment satisfies them, as it is identical to the proof of the just mentioned theorem. Thus, it remains to show that the LEF1 constraints are satisfied. Recall that for arbitrary agent-types $t_a, t'_a$ and arbitrary vertex-types $t_v, t'_v$ we have that $x_{t_a,t'_a,t_v,t'_v} > 0$ if and only if there is at least an agent of type $t_a$ placed at a vertex of type $t_v$ which is the parent of a vertex of type $t'_v$ where an agent of type $t'_a$ has been placed. Hence, as $\mu$ is LEF1, it must be the case that both the agent of type $t_a$ does not envy the agent of type $t'_a$ up to one item and vice versa: then, by Lemma 2, it follows that $\mathbb{1}_T(t_a, t'_a) = \mathbb{1}_T(t'_a, t_a) = 1$, therefore both constraints for these types are satisfied since in both cases the product is larger than 0.

For the right-to-left direction the proof is identical to the one of the same direction in the proof of Theorem 2. The position assignment $\mu$ is built in the same way, hence it only remains to prove that it is LEF1 for the input instance. Observe that because of Lemma 2 we know that each time $x_{t_a,t'_a,t_v,t'_v} > 0$ it must be the case that both $\mathbb{1}_T(t_a, t'_a)$ and $\mathbb{1}_T(t'_a, t_a)$ are both equal to 1 as the variable assignment satisfies the constraints of the ILP. Hence, by the lemma it follows that each pair of agents of such types do not envy each other up to one item, meaning that the resulting position assignment must be LEF1 since, if two agents of types $t_a, t'_a$ are connected, then the vertex types $t_v, t'_v$ of the vertices on which they are placed are such that $x_{t_a,t'_a,t_v,t'_v} > 0$.

Thus, as we know that the INTEGER LINEAR PROGRAM PROBLEM is in FPT when parameterized by the number of variables in the ILP, the result follows in the like it did in the proof of Theorem 2. □

## 4.4 Forests

In the same fashion of Chapter 3, we can prove that, by slightly modifying the built ILP, FIND-LEF1-POSITION-ASSIGNMENT is in FPT for the same parameterization even if the social graph is a forest.

**Theorem 9.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a forest,* FIND-LEF1-POSITION-ASSIGNMENT *is in* FPT *if parameterized by the number of agent-types $|T_{\mathcal{N}}|$ and the number of vertex-types $|T_V|$.*

We omit the proof because it is identical to the one of the analogous theorem for local envy-freeness. The set of constraints is changed in the same way, and the only difference is that now we have LEF1 constraints instead of LEF constraints. However, due to the proof of Theorem 8, we already know that an LEF1 position assignment induces an assignment of variables which satisfies these constraints, and that an assignment of variables which satisfies these constraints induces as well an LEF1 position assignment, thus the result follows.

Next, we prove the intractability of EXISTS-LEF1-POSITION-ASSIGNMENT even when the number of vertex-types is constant. This, similarly to what has happened before, leads to the para-NP-hardness of the same problem when parameterized solely by the number of vertex-types.

**Theorem 10.** *Let $\mathcal{G}$ be the class of graphs such that each graph $G \in \mathcal{G}$ has exactly 3 vertex-types.* EXISTS-LEF1-POSITION-ASSIGNMENT*, restricted to graphs in $\mathcal{G}$, is* NP*-complete.*

*Proof.* We provide yet again a reduction from EXACT COVER BY 3-SETS, in the variant where each instance $I = \langle X, S \rangle$ is such that each element $x \in X$ appears in exactly three triplets of $S$.

The polynomial-time reduction we provide is the same, albeit for some small details which we must take into account due to Observation 3. Following are the changes we make to the reduction:

- $\mathcal{O} := \{o_1^1, o_1^2, \ldots, o_{3q}^1, o_{3q}^2, o_{S_1}^1, o_{S_1}^2, \ldots, o_{S_n}^1, o_{S_n}^2, o_{d_1}^1, o_{d_1}^2, \ldots, o_{d_n}^1, o_{d_n}^2\}$: there are two corresponding items in $\mathcal{O}$ for each agent in $\mathcal{N}$;
- For each utility function $u$, we define $u(o) = 3$ for all the items which are not related to the agent whose utility is $u$, while for all the new "duplicated" items the utility which they grant to the agent is the same

as the utility which their analogous item granted in the proof for local envy-freeness;

- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{3q}, \pi_{S_1}, \ldots, \pi_{S_n}, \pi_{d_1}, \ldots, \pi_{d_n})$ where $\pi_i = \{o_i^1, o_i^2\}$, $\pi_{S_i} = \{o_{S_i}^1, o_{S_i}^2\}$ and $\pi_{d_i} = \{o_{d_i}^1, o_{d_i}^2\}$.

It is clear that the new reduction is still a polynomial-time one and, since we did not change the output graph, it still has exactly three vertex-types. Therefore it remains just to prove its correctness, i.e. that there is an exact cover of $X$ from the triplets of $S$ if and only if there is an LEF1 position assignment $\mu$ for the built instance.

For the first direction, assume that there is a subset $S' = \{S'_1, \ldots, S'_q\}$ of triplets of $S$ which covers exactly $X$. Notice that by taking the same position assignment $\mu$ as the one we provided in the proof of Theorem 4, we obtain yet again a position assignment which is LEF. Since a position assignment that is LEF is also LEF1, the claim follows.

For the other direction, assume that there is an LEF1 position assignment $\mu$ for the built instance. We prove the claim for local envy-freeness up to one item corresponding to the same claim we have proven in the proof of the corresponding theorem for local envy-freeness, i.e. that the central vertices of the stars can only be occupied by triplet-agents and that the matchings can only be occupied by a triplet-agent and her corresponding dummy agent.

Assume by way of contradiction that for some star the central vertex is not occupied by a triplet-agent, then it must be occupied either by an element-agent or by a dummy agent. In the former case $\mu$ cannot be LEF1, because there are exactly three agents which she does not envy up to one item, the triplet-agents who correspond to the triplets which contain her corresponding element, therefore the remaining vertex of the star will be occupied by an agent which she does envy up to one item (notice how, by removing either one of the two items, the remaining utility of that agent's bundle will be 3 for the element-agent). For the latter case, this same argument holds as well because each dummy agent does not envy up to one item only one other agent, her corresponding triplet-agent, hence in the remaining vertices of the star all the agents will be envied up to one item by the dummy agent.

On the other hand, assume that there is a matching which is not occupied by a pair consisting of a triplet-agent and a dummy agent: clearly it cannot be the case that the assigned pair consists of an element-agent and a dummy agent, or an element-agent and a triplet-agent such that the corresponding element is not in the corresponding triplet or a triplet-agent and a dummy agent who is not the one of the former agent, since in any case the two agents will necessarily envy each other up to one item. Therefore, the only possible case is that the assigned pair consists of an element-agent and a triplet-agent such that the corresponding element is in the corresponding triplet: but then the triplet-agent's dummy agent will necessarily be connected to an agent that she envies up to one item, hence this also cannot happen. Thus, by the same argument which we have used previously it follows that there must be a subset of $S$ which covers exactly $X$, proving the claim. □

Hence, by Definition 17, we also get the following result.

**Corollary 4.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a forest,* EXISTS-LEF1-POSITION-ASSIGNMENT *is para-*NP*-hard if parameterized by the number of vertex-types $|T_V|$.*

## 4.5 (Definitely-Non-)Quasi-Trees

We conclude this chapter in the same way we ended Chapter 3, i.e. by treating graphs based on their treewidth.

**Proposition 10.** *Let $\mathcal{G}$ be a class of graphs such that each graph $G \in \mathcal{G}$ has a constant treewidth, i.e.* $\text{tw}(G) = O(1)$. EXISTS-LEF1-POSITION-ASSIGNMENT, *restricted to graphs in $\mathcal{G}$, is in* FPT *if parameterized by the number of non-isolated vertices in $G$.*

Like before, we also omit the proof of this proposition because it is identical to the proof of Proposition 6. The crucial observation to make is that, similarly to what happens for the envy-free graph, there is a position assignment $\mu$ which is LEF1 for the instance if and only if there is a subgraph isomorphism from the non-isolated subgraph of $G$ to a subgraph of the envy-free up to one item graph $G_{EF1}$ induced by the input instance itself. Knowing this, the proof follows exactly as it did before given the fact that SUBGRAPH ISOMORPHISM is in FPT when parameterized by $|V(H)|$, if $H$ has a constant treewidth.

We will now prove the final result of the chapter.

**Theorem 11.** *Let $\mathcal{G}$ be a class of graphs containing all graphs $G$ consisting of a clique together with some isolated vertices—and thus $\mathcal{G}$ contains graphs of unbounded treewidth.* EXISTS-LEF1-POSITION-ASSIGNMENT, *restricted to graphs in $\mathcal{G}$, is $W[1]$-hard if parameterized by the number of non-isolated vertices in $G$.*

*Proof.* As in the analogous proof in the Chapter 3, to prove the claim we give an FPT reduction from the CLIQUE problem, which implies the claim as we know that the latter problem is $W[1]$-hard when parameterized by the size of the clique. Given an input instance $\langle G = (V, E), k \rangle$, where $V = \{v_1, \ldots, v_n\}$, for the CLIQUE problem, the built instance $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' = (V', E') \rangle$ is defined as follows:

- $\mathcal{N} = \{1, \ldots, n\}$, i.e. there is a corresponding agent $i$ for each vertex $v_i \in V$;
- $\mathcal{O} = \{o_1^1, o_1^2, \ldots, o_n^1, o_n^2\}$, i.e. there are two corresponding items $o_i^1, o_i^2$ for each agent $i$;
- $\boldsymbol{u} = (u_1, \ldots, u_n)$ where for any $k \in [2]$ and $1 \leq i \leq n$:
  - $u_i(o_i^k) = 0$;
  - $u_i(o_j^k) = 0$ for $i, j$ such that $\{v_i, v_j\} \in E$;
  - $u_i(o_j^k) = 1$ for $j \neq i$ such that $\{v_i, v_j\} \notin E$;
- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ where $\pi_i = \{o_i^1, o_i^2\}$ for any $1 \leq i \leq n$.
- $G'$ is as it was defined in the proof of Theorem 5.

We now observe that two agents do not envy each other up to one item if and only if their corresponding vertices are connected in the original graph $G$. If the vertices are connected, then the claim holds simply because the two agents do not envy each other, hence they also do not envy each other up to one item. On the other hand, assume that agents $i$ and $j$ are such that $\{v_i, v_j\} \notin E$: then $u_i(\pi_i) = 0$ and $u_i(\pi_j) = 2$. Moreover, it is easy to verify that there is no item which can be removed from $\pi_i \cup \pi_j$ so that, given the new bundles without the removed item, $i$'s new bundle gives to her a utility larger than or equal to the one which $j$'s new bundle grants to $i$. Therefore, $i$ does indeed envy $j$ up to one item.

Similarly to what happened before, there is an LEF1 position assignment for the built instance if and only if there is a clique of size $k$ in the original input graph $G$ by the same argument we have made in the analogous proof of Theorem 5. □

## 4.6 Summary

As anticipated in Section 3.7, in this section there have been more positive results, all due to the fact that it is always possible to find an item allocation that is EF1 (which always exists) in polynomial time. This obviously means that, in case the central authority has the task to perform the item allocation as well, the problems become trivial for LEF1. Aside from this, there were no differences between LEF and LEF1 complexity-wise.

In the next chapter we will examine the last criterion we defined, local proportionality. Unlike these first two criteria we have seen, we will not be able to obtain some parameterized complexity results for local proportionality, but all the other ones will follow via adaptations of proofs we have seen in Chapter 3, similarly to what has happened in this chapter.

CHAPTER 5

# COMPLEXITY OF LOCAL PROPORTIONAL ASSIGNMENTS

We conclude the theoretical section of the thesis by considering the last fairness criterion we defined in the introduction, local proportionality. Amongst the three criteria we are interested in this surely stands out as the most different one, as local envy-freeness and local envy-freeness up to one item are clearly connected by their definitions. However, as the reader will also see, this uniqueness will not be visible up until we reach the realm of parameterized complexity.

As hinted by the previous paragraph, results in classical complexity will be the same as in Chapter 3, whereas in parameterized complexity most of the results will not carry over. Similarly to what happened with local envy-freeness up to one item, the proofs in this chapter will mainly be adaptations of the respective proofs from Chapter 3.

## 5.1 Lines and Stars

As usual, we will first consider again graphs that are lines or stars. Like before, stars prove to be amongst the simplest graphs to treat with respect to local proportionality, whereas lines again are intractable.

**Proposition 11.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a star,* EXISTS-LPROP-POSITION-ASSIGNMENT *is decidable in polynomial time.*

*Proof.* To prove the claim, we give a simple polynomial-time algorithm that answers the question. Similarly to before, it suffices to simply loop over all agents and check two conditions:

1. That the current agent $i$ is such that $u_i(\pi_i) \geq \frac{u_i(\mathcal{O})}{|\mathcal{N}|}$;
2. That any other agent $j \in \mathcal{N} \setminus \{i\}$ is such that $u_j(\pi_j) \geq u_j(\pi_i)$.

If there is such an agent $i$, then the algorithm immediately outputs "Yes" and halts. Instead, if after the loop no agent satisfies the two conditions, the algorithm outputs "No". It is clear that the algorithm is a polynomial one, quadratic in the number of agents to be precise. We now show that there is an LPROP position assignment if and only if the algorithm outputs "Yes".

If there is such a position assignment $\mu$, then obviously there must be some agent $i$ who is assigned to the central vertex. Notice that for $\mu$ to be LPROP, $i$ must satisfy the two conditions of the algorithm, hence the algorithm will output "Yes" once the loop reaches $i$.

On the other hand, assume that the algorithm answers positively. This can happen only if, when looping over the agents, there is some agent $i$ that satisfies the two conditions. Consider a position assignment $\mu$ which places $i$ at the central vertex and all the remaining agents randomly at the remaining vertices. We show that $\mu$ must be LPROP:

1. $\mu$ is LPROP for $i$: since $N^+(i) = \mathcal{N}$ it is clear that $\sum_{j \in N^+(i)} \pi_j = \mathcal{O}$, and by the way the algorithm is defined it must be the case that $u_i(\pi_i) \geq \frac{u_i(\mathcal{O})}{|\mathcal{N}|}$, thus the claim holds;
2. $\mu$ is LPROP for any $j \in \mathcal{N} \setminus \{i\}$: to prove this claim it suffices to notice that when an agent is connected to only one other agent, local proportionality is equivalent to local envy-freeness. Thus, by simply observing that the only neighbor of $j$ under $\mu$ is $i$, and that by the algorithm it is ensured that $j$ does not envy $i$, the claim follows.

Hence, a positive answer from the algorithm implies the existence of a position assignment which is LPROP, thus proving the proposition. □

Notice that by using this algorithm it is also possible to define another polynomial-time one for FIND-LPROP-POSITION-ASSIGNMENT, where the output position assignment $\mu$ is defined as in the proof of the right-to-left direction.

Let us move now to the lines. Again, we will use the HAM-PATH problem to prove the NP-completeness of EXISTS-LPROP-POSITION-ASSIGNMENT.

**Proposition 12.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a line,* EXISTS-LPROP-POSITION-ASSIGNMENT *is* NP*-complete.*

*Proof.* Membership in NP can be proven exactly in the same way as it was proven for local envy-freeness: a (polynomial) certificate is a position assignment $\mu$ and the task of the verifier is to check that each agent has a bundle which grants her at least the average utility in her augmented neighborhood with respect to $\mu$. Trivially this can be done in quadratic time.

To prove NP-hardness, we give a reduction from HAM-PATH. The reduction is almost the same as the one for local envy-freeness, as the only difference is in the utility functions. Let $i$ be any agent and let $v_i$ be her corresponding vertex in $G$. As before, we denote with $o'_{i_1}, \ldots, o'_{i_{m'}}$ the objects which correspond to agents whose corresponding vertices are all and only those in $N(v_i)$: we define $u_i(\{o'_{i_j}\}) = u_i(\{o_i\})$ for any such $i_j$.

Thus, we proceed to show that there is an LPROP position assignment for the derived instance $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' \rangle$ if and only if there is a Hamiltonian path in the input graph $G$ for HAM-PATH.

Assume that there is an LPROP position assignment $\mu$ for $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G' \rangle$. This means that $\mu$ assigns to each agent $i$ a vertex on the line $G'$ so that $i$ has a utility proportional to that of her neighbor(s). Observe that, for any agent $i$, local proportionality in this case is satisfied just in case all of her neighbors $j$ are such that $u_i(\pi_i) = u_i(\pi_j)$. In fact, suppose by way of contradiction that there is some agent $i$ for which one of her neighbors $j$ is such that $u_i(\pi_i) \neq u_i(\pi_j)$. Then, by construction of the reduction, it follows that $u_i(\pi_j) > u_i(\pi_i)$. Thus, observe that the average utility from $i$'s point of view, in her augmented neighborhood $N^+(i)$, is strictly larger than $u_i(\pi_i)$, because there are at most two agents (including $i$ herself) such that the bundles assigned to them have a utility equal to $u_i(\pi_i)$ (from $i$'s perspective) and at least one with a utility strictly larger than $u_i(\pi_i)$. Hence, $\boldsymbol{\pi}$ and $\mu$ do not satisfy local proportionality for $i$, meaning that $\mu$ is not LPROP. Therefore, for any agent $i$, we have that her two neighbors $j_1, j_2$ are such that $u_i(\pi_i) = u_i(\pi_{j_1}) = u_i(\pi_{j_2})$, which can happen, by construction, if and only if the vertex to which they correspond, $v_i, v_{j_1}, v_{j_2}$, are such that $\{v_{j_1}, v_{j_2}\} \subseteq N(v_i)$ (and the same holds for those agents who have only one other neighbor). Thus, it follows that the path $P$ which "follows" the position assignment $\mu$ is indeed a Hamiltonian path of $G$, as all vertices are visited exactly once since each agent can only be placed at one vertex in $G'$.

Now suppose that $G$ has an Hamiltonian path $P = v_{p_1}, \ldots, v_{p_n}$. Obviously, every pair of consecutive vertices $v_{p_i}, v_{p_{i+1}}$ in $P$ must be such that $\{v_{p_i}, v_{p_{i+1}}\} \in E$, otherwise $P$ is not a valid path. Consider the position assignment which corresponds to $P$, i.e. such that the $\mu(p_i) = w_i$. We quickly show that $\mu$ is LPROP: let $1 < i < n$ be arbitrary (for $i = 1$ or $i = n$ the argument is the same but there are less agents to consider, thus we omit the two cases). Since $P$ is a path it must be the case that $\{v_{p_i}, v_{p_{i-1}}\}$ and $\{v_{p_i}, v_{p_{i+1}}\}$ are both in $E$, meaning that, for the respective agents $p_{i-1}, p_i, p_{i+1}$, it holds that $u_{p_i}(\pi_{p_{i-1}}) = u_{p_i}(\pi_{p_i}) = u_{p_i}(\pi_{p_{i+1}})$. As $p_{i-1}$ and $p_{i+1}$ are the only neighbors of $p_i$, by definition it follows that $\mu$ is LPROP for $p_i$. Therefore, $\mu$ must be LPROP for the built instance as $p_i$ was chosen arbitrarily. □

## 5.2 Strongly Connected Graphs

We generalise lines and stars and consider now strongly connected graphs. In the same way as in Chapter 3, due to Proposition 12, we know that EXISTS-LPROP-POSITION-ASSIGNMENT is NP-complete for this class.

Therefore, we shift our focus again to EXISTS-LPROP-DISTRIBUTION. The proof of the following theorem is a direct consequence of the corresponding Theorem 1 and of the following lemma.

**Lemma 3.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which agents have identical utilities and $G$ is strongly connected, $\boldsymbol{\pi}$ and a position assignment $\mu$ are LPROP if and only if every agent has the same utility under $\boldsymbol{\pi}$.*

*Proof.* For the right-to-left direction, we trivially have that if all connected agents have the same utility, then the item allocation $\boldsymbol{\pi}$ and the position assignment $\mu$ are LPROP by definition.

For the other direction, assume by contraposition that there are two connected agents $i, j$ who do not have the same utility under $\boldsymbol{\pi}$. Without loss of generality, assume that $u(\pi_i) < u(\pi_j)$. Also, assume that all other agents connected to $i$ have at least the same utility as $i$: if there is some other agent who has a lower utility than $i$, then consider her in place of $i$ and take $i$ in place of $j$ and, if the same happens for this new agent, repeat the process. Notice that, as the set of agents is finite and all agents have the same utility function $u$, there must be some agent $i'$ with minimal utility and which can be reached from $i$ (but not necessarily connected to $i$ herself). Thus, this process of choosing "new" agents $i$ and $j$ will, at some point, stop. Then, $u(\pi_i) < \frac{\sum_{k \in N^+(i)} u(\pi_k)}{|N^+(i)|}$, because all agents surrounding $i$, except for $j$, have at least the same utility as $i$, but $j$ has a strictly larger utility than $i$. Hence, $\boldsymbol{\pi}$ and $\mu$ cannot be LPROP, which proves the claim. □

From this lemma, it follows also that all proofs which exploited identical utilities for local envy-freeness hold as well for local proportionality. Hence, we get the following result.

**Theorem 12.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ in which $G$ is strongly connected,* EXISTS-LPROP-DISTRIBUTION *is* NP*-complete, even when agents have identical utilities.*

From which we can derive the corresponding corollary by making the same observation, as we have made in Chapter 3, about how we can replace the line used in the construction with a star.

**Corollary 5.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ in which $G$ is a star or a line,* EXISTS-LPROP-DISTRIBUTION *is* NP*-complete.*

## 5.3 Matchings

Graphs which are matchings are a very particular case with respect to local proportionality. It is rather easy to verify the following fact:

**Lemma 4.** *For all instances $I$ with a fixed item allocation and such that $G$ is a matching, a position assignment $\mu$ is LPROP if and only if it is LEF.*

This holds due to the simple observation that in matchings each agent will be paired with another agent, hence the utility of a given agent will be higher than or equal to the average utility (from her point of view) of her augmented neighborhood if and only if her utility is higher than or equal to the utility which the bundle assigned to her only neighbor would have granted her. Hence, all results we were able to prove with respect to local envy-freeness also hold for local proportionality.

**Proposition 13.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G$ is a matching,* EXISTS-LPROP-POSITION-ASSIGNMENT *is decidable in polynomial time.*

**Proposition 14.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, G \rangle$ in which $G$ is a matching,* EXISTS-LPROP-DISTRIBUTION *is* NP*-complete, even when agents have identical utilities.*

## 5.4 Trees, Forests and (Definitely-Non-)Quasi-Trees

We conclude this chapter by merging all these graph families into a single section. As the reader might recall, in the previous chapters these sections were mainly devoted to parameterized complexity results.

In those proofs, we exploited two different but very useful concepts: the "*vertex-type*" and the "*envy-free graph*" or "*envy-free up to one item graph*". The first allowed us to avoid using as a parameter the total number of vertices, which is a trivial parameterization of the problem as it is exactly equal to the number of agents: at that point, a simple brute force search would have done the job. The second were instead crucial in proving the correctness of reductions we have used.

As a first bad news, a "*proportional graph*" cannot be defined in the same way as the envy-free (up to one item) graph. Proportionality, and its local variant as well, is a criterion which cannot be defined on a single pair of agents, but must take into account the whole "neighborhood" of a given agent to determine whether the instance is fair or not. Thus, a simple graph where each vertex corresponds to an agent will not be of much use in this context, as it will never be able to consider the multiple ways in which the neighborhood of an agent can be formed. We propose two different ways to tackle this issue, both which try to address this lack of expressive power.

The first one is to use, instead of a graph, a *hypergraph*. Hypergraphs are graphs in which single edges can connect more than two vertices: it is then clear how this feature can be used to define the proportional *hypergraph*. The second way we propose is to define a graph in which vertices are divided into two subsets, one in which each vertex represents an agent and the other in which every vertex represents a subset of agents. Hence, each edge connects an agent to a subset of agents such that the item allocation is proportional from the point of view of the first agent. Essentially this is a way to disguise the hypergraph we have previously mentioned into a normal graph. One advantage this has is the simple fact that we are still using graphs instead of a different structure, hence all interesting properties and complexity results still apply.

We believe that finding a sensible (better if also efficient) way to define such a concept with respect to proportionality will prove to be extremely useful, especially after seeing how it can be employed when we are considering the other two fairness criteria we are interested in instead of proportionality. An easy example to observe the potential this has is to consider the fact that the CLIQUE problem can be easily used to perform (parameterized) reductions if a hypothetical proportionality graph is defined, as we have already seen with the envy-free graph and the envy-free up to one item graph.

On the other hand, as one might guess, the main issue with vertex-types when one wants to determine whether an item allocation and a position assignment satisfy local proportionality for some agent, is the fact that it is not possible to determine who are the neighboring agents of the given one using only vertex-types, as the vertices connected to the one at which the agent is placed will be confused with all the other vertices of the same type, hence it will not be possible to understand which agents have been assigned to those vertices. Note also how this issue is particularly severe when one tries to devise an ILP to try to solve the EXISTS-LPROP-POSITION-ASSIGNMENT and FIND-LPROP-POSITION-ASSIGNMENT, as we did for example in Theorems 2 and 8. In those proofs, we were able to overlook the ambiguous nature of vertex-types, and in fact we exploited it, to ensure that the assignments were LEF or LEF1, because of the fact that these criteria, unlike local proportionality, are only interested in pairs of adjacent agents. The main problem in taking this same approach with local proportionality is the inherent difficulty in reconstructing the unique neighborhood of a fixed agent assigned to some vertex, because given the vertex-type of the latter, we effectively are not able to determine the correct agents who are in the neighborhood of the fixed agent. One way in which this might be fixed is by changing the way variables are defined in the ILP: an idea which naturally comes to mind is to define variables in such a way that they talk about neighborhoods of a fixed vertex and not of a single edge. Nevertheless, one should always remember that the number of variables should be polynomial in the parameters of the problem for the algorithm to have a chance to be an FPT one, hence we tend to believe that in order to fix this issue the parameters will also have to be changed in some way.

The reader might also recall that the only negative result in parameterized complexity about forests was about vertex-types, where we used a variant of EXACT COVER BY 3-SETS to prove the NP-completeness of EXISTS-LEF-POSITION-ASSIGNMENT and EXISTS-LEF1-POSITION-ASSIGNMENT even when the number of vertex-types of the social graph is equal to a fixed constant, thus implying the para-NP-hardness of the same problem when parameterized by the number of vertex-types. (Un)Fortunately, that same result holds also when we consider local proportionality, as we will see now.

**Theorem 13.** *Let $\mathcal{G}$ be the class of graphs such that each graph $G \in \mathcal{G}$ has exactly 3 vertex-types.* EXISTS-LPROP-POSITION-ASSIGNMENT, *restricted to graphs in $\mathcal{G}$, is* NP*-complete.*

*Proof.* Akin to previous proofs in this chapter, the approach we will take on this one is that of adapting the proof of the corresponding theorem from Chapter 3 so that it takes into account local proportionality instead of local envy-freeness.

First of all, the problem is in NP by the usual argument, as one can simply check in polynomial time whether a position assignment $\mu$ (i.e. the polynomial certificate) is LPROP for a given instance by checking that for each agent the average utility of her augmented from her point of view is smaller than or equal to her own utility under $\boldsymbol{\pi}$.

To prove NP-hardness, we reduce again from the variant of EXACT COVER BY 3-SETS. Hence, let $I = \langle X, S \rangle$ be an arbitrary instance of EXACT COVER BY 3-SETS such that each element of $X$ appears in exactly three triplets of $S$.

The reduction is identical to the one in the proof of Theorem 4, except for the utilities of each agent which are now defined so that, for an arbitrary utility function $u$, $u(o) = 5$ for any item $o \in \mathcal{O}$ that was such that previously $u(o) = 2$, i.e. all those items that have no correlation to the agent. Trivially, the built instance $I' = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ is indeed an instance of EXISTS-LPROP-POSITION-ASSIGNMENT where the social graph $G$ has exactly three vertex-types and the reduction takes polynomial time to be performed.

We now show that the reduction is correct. For the first direction, assume that there is a subset $S' = \{S'_1, \ldots, S'_q\}$ of $S$ such that each item of $X$ appears in exactly one triplet of $S'$. Consider the same position assignment $\mu$ which was defined in the analogous proof for local envy-freeness, we now show that it is also LPROP for $I'$:

- For the $q$ stars with 5 vertices, the utilities of the central agents are indeed greater than or equal to the average utilities of their respective augmented neighborhoods, since for each of them it is exactly equal to $\frac{1}{5}$, which is clearly smaller than their own utilities that are exactly equal to 1. Moreover, $\mu$ is also LPROP for the agents on the outer vertices of these stars, since in that case local proportionality corresponds to local envy-freeness as they have only one neighbor, the central vertex in the star, and in the previous proof we had already seen how the assignment was LEF;

- For the remaining $n - q$ matchings, $\mu$ is LPROP by the same argument we have just made for the agents on the outer vertices of the stars: as these agents are paired with only one other agent, the assignment is LPROP because we already know that it is LEF.

Thus, we have proven that the existence of a solution to the instance $I$ implies the existence of an LPROP position assignment for $I'$.

For the other direction, assume that there is a position assignment $\mu$ which is LPROP for $I'$. We prove, yet again, the same claim we have proven in the proofs of the two corresponding theorems in the previous two chapters, i.e. that the central vertices of the stars can only be occupied by triplet-agents, and the same applies to the matchings (for these, pick arbitrarily one of the two vertices as the "central" vertex). For suppose that in the stars the central vertex is either a dummy agent or an element-agent. In the former case $\mu$ is trivially not LPROP for the dummy agent, as amongst the four agents to which she is connected, there are at least three whose utility of their bundles for her is equal to 5, meaning that the average utility in the augmented neighborhood is at least $\frac{16}{5}$, which is clearly larger than or equal to 1. In the latter case, the same must apply because by the definition of the variant of EXACT COVER BY 3-SETS we know that there are only three other agents for the element-agent whose bundle give a utility of 0 to her, thus the fourth agent in the star will have a bundle which gives a utility of 5 to her, hence making $\mu$ not LPROP. For the matchings, we can also prove that the agents placed there can only be a triplet-agent and her corresponding dummy agent. If the dummy agent is not the corresponding one, then she will envy the triplet-agent (and so will the triplet-agent envy her), thus making the position assignment not LPROP. Instead, if there is an element-agent, then in any case the position assignment will not be LPROP for the dummy agent corresponding to the triplet-agent, as she will be connected to at least one other agent, which necessarily will not be the correct triplet-agent, meaning that the utility of her bundle for the dummy agent will be equal to 5. Hence, by the same argument we have made in the analogous proofs for local envy-freeness and local envy-freeness up to one item, it follows that this implies the existence of a subset $S'$ which exactly covers $X$, thus proving the theorem. □

Thus, as in Chapters 3 and 4, by Definition 17 this implies that the same problem, when parameterized only by the number of vertex-types, is para-NP-hard if the social graph is a forest.

**Corollary 6.** *Given an instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ in which $G = (V, E)$ is a forest,* EXISTS-LPROP-POSITION-ASSIGNMENT *is para-*NP*-hard if parameterized by the number of vertex-types $|T_V|$.*

## 5.5 Summary

This chapter marks the end of the theoretical part of the thesis. In it we have seen mostly negative results, apart from some classes of graphs, the results for LEF1 concerning the problems in which the central authority has to compute the item allocation as well and some parameterized results. For what concerns local proportionality, we have seen that most of the parameterized results we were able to obtain for the other two criteria were not reproducible for it. We think that finding parameters which fit this criterion is surely an interesting direction of research, because of how different it is compared to local envy-freeness and local envy-freeness up to one item.

Following is the final chapter of the thesis, in which we will discuss the experiments we have performed. These will highlight how various parameters affect the likelihood that a given random instance $I = \langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, paired with some fairness criterion $\mathcal{F}$ amongst the three that we have studied, is a positive one, i.e. one for which there is a position assignment $\mu$ that satisfies $\mathcal{F}$.

# Chapter 6

# Experiments

In the final chapter of the thesis we will present some experimental results we have obtained during our work[7]. Section 6 from Beynier et al. [2019] has been our main inspiration for these experiments.

Similarly to their work, we will also mainly focus on the occurrence of positive instances (for any of the three fairness criteria we have defined) amongst randomly generated ones. In this case, instances will be of the form $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$ (it should be noted that a parameter of our experiments will be the way the item allocation $\boldsymbol{\pi}$ is performed) paired with some fairness condition $\mathcal{F}$ amongst the three we have discussed in the thesis, i.e. LEF, LEF1 and LPROP, and it will be considered "positive" if there is some position assignment $\mu$ that satisfies $\mathcal{F}$ with $\boldsymbol{\pi}$ for the given instance.

The main question we want to answer is "*how much do some parameters influence the likelihood of positive instances?*" We think this question is particularly interesting because answering it allows us to know which kind of instances (based on the parameters we will use) are the most promising ones when the central authority has the only task of performing the position assignment.

We briefly outline the chapter. In Section 6.1 we will present to the reader the setting of our experiments. We will specify how we have generated the random instances, the set of parameters and which libraries we have used. Following, in Section 6.2 we will discuss the experiments without considering the agent-types as parameters, on which we will instead focus in Section 6.3.

## 6.1 Setting of the experiments

First, we explain how the random instance is built. Recall that an instance in these experiments will be a tuple $\langle \mathcal{N}, \mathcal{O}, \boldsymbol{u}, \boldsymbol{\pi}, G \rangle$, where $\mathcal{N}$ is the finite set

[7] All code is available at: `https://github.com/giovannivarr/Complexity-of-Locally-Fair-Allocations-on-Graphs`.

of $n$ agents, $\mathcal{O}$ the finite set of $m$ items, $\boldsymbol{u} = (u_1, \ldots, u_n)$ the utility profile, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ the item allocation and $G = (V, E)$ the social graph.

**Utility profile** The first random component of the instances we discuss is how we generate the agents' utilities. For each agent, the utility that an arbitrary item generates for her is drawn either from a normal or a (continuous) uniform distribution. For the first, the mean and the standard deviations are respectively equal to 0 and 1, while for the latter the minimum and maximum possible utilities are respectively $-1$ and 1. Due to the fact that the final results were very similar regardless of the distribution from which the agents' utilities were drawn, we will only present those which we have obtained with the uniform distribution. We have also performed experiments in which the items are either all goods or all chores for every agent: in these cases utilities are drawn from a uniform distribution where the minimum and maximum possible values are respectively 0 and 1 for the only goods experiments and $-1$ and 0 for the only chores experiments.

**Item allocation** Instances can also be (possibly) randomized with respect to the item allocation, assigning the items randomly to the agents but in such a way that every agent gets at least one item. The other ways in which the item allocation can be performed are instead deterministic and they revolve around either maximizing the utilitarian welfare (i.e. the sum of the utilities of each agent under the given assignment) or minimizing enviness or uproportionality. To perform these three last item allocations, we have decided to use the Python-MIP library, through which we were able to define and solve ILPs. The objective functions to minimize enviness (leftmost) and unproportionality (rightmost) are the following.

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} E(i,j)(u_i(\pi_j) - u_i(\pi_i)) \qquad \min \sum_{i \in \mathcal{N}} U(i) \left( \frac{u_i(\mathcal{O})}{|\mathcal{N}|} - u_i(\pi_i) \right)$$

Where in the objective functions for the item allocations which minimize enviness or unproportionality, $E$ and $U$ are the following functions[8].

$$E(i,j) = \begin{cases} 1 & \text{if } u_i(\pi_j) - u_i(\pi_i) > 0 \\ 0 & \text{otherwise} \end{cases} \qquad U(i) = \begin{cases} 1 & \text{if } \frac{u_i(\mathcal{O})}{|\mathcal{N}|} - u_i(\pi_i) > 0 \\ 0 & \text{otherwise} \end{cases}$$

We specify that for experiments involving envy-freeness up to one item, we have decided to consider only random item allocations. This is because

[8]It should be noted that the two functions we have defined would make the objective functions for the ILPs non-linear, hence not usable in an ILP. Thus, the ILPs which we have defined use dummy variables to circumvent this issue.

in this way we can simulate the fact that the central authority is given a fixed item allocation, as otherwise it would simply have to find an EF1 item allocation which, as we already know, always exists and can be found in polynomial time.

**Social graph** Finally, we (randomly) generate graphs using the NetworkX library (Hagberg, Schult, and Swart [2008]). Graphs can be of different type: aside from lines, stars, matchings and trees which we have already seen, we also include binomial, regular and Barabási-Albert graphs.

Binomial graphs fall into the broader class of Erdős–Rényi graphs, which were first introduced in Erdős and Rényi [1959]. To be more precise, the ones generated by NetworkX follow the model which was defined in Gilbert [1959] where, given in input an integer $n$ and a probability $p$, the output graph $G$ has $n$ vertices and there is an edge between any two (distinct) vertices with probability $p$, independently of all the other edges. In our experiments, we have fixed $p$ to always be $\frac{1}{2}$.

A regular graph is defined as a graph where each vertex has the same constant degree $k$. For regular graphs we have fixed the number of vertices to be 8 and, since regular graphs with $k = 1$ are matchings, the degree will vary between 2 and 7.

Finally, the Barabási-Albert graphs, first introduced in Barabási and Albert [1999], that belong to the so-called class of *scale-free graphs*. These are graphs in which the degree distribution follows asymptotically a power law, i.e. the fraction of vertices in the graph with a degree of $k$ is proportional to the value $k^{-\gamma}$ for some constant $\gamma$. Barabási-Albert graphs fall in this category because their degree distribution is proportional to $k^{-3}$. We quickly explain how these graphs are generated: at first the network consists of $m_0$ isolated vertices[9]. It should be noted that in our experiments we have chosen to fix $m_0$ to be equal to 3 (so that graphs can be generated for any number of agents). At step $i$, given the current graph $G_i = (V_i, E_i)$, a new vertex is added to $V_i$ and it is connected to $m$ vertices, where the probability that it is connected to some older vertex $v \in V_i$ is equal to $\frac{deg_{G_i}(v)}{\sum_{v' \in V_i} deg_{G_i}(v')}$, where $deg_{G_i}(v)$ is the degree of $v$ in the current graph $G_i$ (see Definition 1). Notice that this implies that the Barabási-Albert graphs which are generated are always strongly connected graphs: each new vertex is connected to $m_0$ vertices amongst the ones already in the graph, which are all connected, except in the beginning where, however, the first new vertex that is added will be connected to all the $m_0$ isolated vertices. As the reader can notice, this way of generating the graph will lead to a graph in which, when a new

[9] In NetworkX $m_0$ is equal to another variable which we will shortly see, $m$, as by the source code of the method which we have used.

vertex is added, it will probably be connected to those vertices that have a large number of connections and thus, in the final graph, there will be a set of vertices which have high degree (usually the first $m_0$ vertices). This characteristic of the Barabási-Albert graphs mimics one of natural social networks, in which there is generally a group of individuals who are very popular in the community (Barabási and Albert [1999]). While the typical example of the desks-in-the-office does not apply in this case, we deem these networks interesting for their inherent property of being "natural" ones.

**Experiments using EF agent-types** In the experiments in which we use also the number of EF agent-types as a parameter, we have fixed the number of agents to 8 and varied the number of agent-types from 2 to 8. For each instance, we randomize the number of agents of a given agent-type and the relations between them, i.e. which agent-types are envied or not by another.

We now give a list of the different parameters we have used and their possible values:

- The number of agents, which varies from 4 to 8;
- The number of items per agent, which varies from 1 to 4;
- The way the social graph is generated;
- The way the agents' utilities are generated;
- The way the item allocation is performed;
- The degree of the regular graphs, which varies from 2 to 7;
- For what concerns the experiments using agent-types, the number of agent-types, which varies from 2 to 8;
- The fairness criterion to be satisfied, which can either be local envy-freeness (LEF), local envy-freeness up to one item (LEF1) or local proportionality (LPROP).

For each combination of the parameters we have generated 1000 random instances and computed the likelihood of positive ones. To do this, we divide the number of instances for which there is a position assignment that satisfies the fairness criterion with the item allocation by 1000, the total number of instances we have generated for that particular parameter combination. We represent the results as a heatmap, which we draw using the seaborn library (Waskom and team [2020]).

**Heatmap description** We conclude the section by briefly describing how the heatmaps are composed. The values on the $x$-axis are either the degree of the graph (for regular graphs) or the number of agents (for all other graph classes) and those on the $y$-axis the number of items per agent. For each of the figures in the next section, in their caption we will specify all the other parameters, the way the utilities are generated (e.g. "*uniform utilities*" in case each agent's utility was generated by drawing values from the uniform distribution), the way the items are assigned, the class of the social graph, the fairness criterion to satisfy and, eventually, other notes if necessary. When we present two heatmaps paired (e.g. Figure 6.2), we will specify the parameters which differentiate the two in a smaller caption beneath each heatmap. The parameters' values will be presented in the captions separated by a dash "-". The value in each cell is the likelihood of positive instances for the specific combination of parameters given by those in the caption and those of the cell's coordinates. As an example, take Figure 6.2 and consider the value in the leftmost cell on the first row of the first heatmap from the left, which is 0.76. This means that, if the agents are 4, there is only one item per agent (hence 4 items in total), utilities are drawn from the uniform distribution, items are assigned to maximize utilitarian welfare, the underlying social graph is a tree and the fairness criterion to satisfy is local envy-freeness, then the probability that a random instance is a positive one (i.e. there is a position assignment that satisfies LEF with the item allocation) is 0.76. In other words, during the experiments, out of all the 1000 randomly generated instances, approximately 760 were positive.

The heatmaps in Section 6.3 differ a bit as there are only two, one for regular graphs and another for all the remaining graph classes. On both heatmaps, the values on the $y$-axis are the number of EF agent-types. As before, for the regular graphs' heatmap, the values on the $x$-axis are the graph degree, instead, for the other heatmap, the values on the $x$-axis are the graph classes. Hence, a cell at coordinates $(\mathcal{G}, j)$ will express the likelihood of positive instances in case the social graph belongs to the graph class $\mathcal{G}$ and there are $j$ different EF agent-types.

## 6.2 Likelihood of Positive Instances

In this section we will present the results we have obtained for our first kind of experiments. In these, we have randomly generated 1000 instances for each possible combination of the parameters and evaluated how frequent were positive instances, i.e. instances for which there is a position assignment that satisfies the fairness criterion when paired up with the item allocation of that same instance.

As we have already mentioned in Section 6.1, we will only present plots that have been generated from random instances in which the utilities were drawn from the uniform distribution.

Figure 6.1 is a line plot which shows the trend, for each of the graph classes

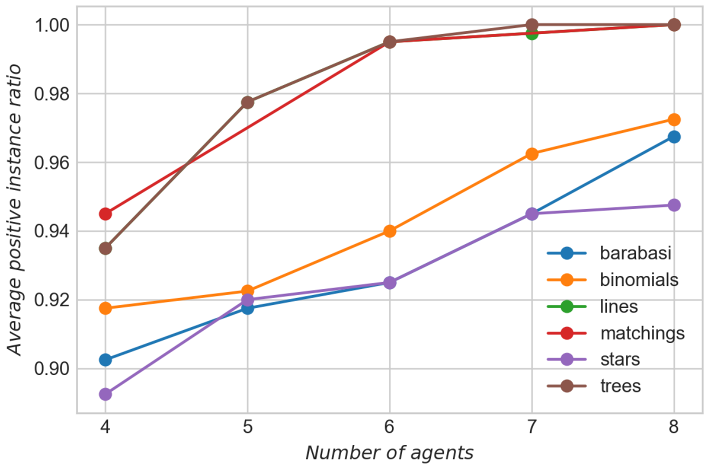



Figure 6.1: Each line represents the average likelihood of positive instances (averaged over the number of items per agent) for that particular graph class as the number of agents increases. Utilities are drawn from uniform distributions and the item allocation maximizes utilitarian welfare.

we have considered (except regular graphs), of the average ratio of positive instances (averaged over the number of items per agent) as the number of agents increases. The fairness criterion is local envy-freeness and the items are assigned to maximize utilitarian welfare.

From the plot, graph classes can be divided in approximately three groups. In the upper one there are lines, matchings and trees, in the middle one only binomial graphs, and in the lower one Barabási-Albert graphs and stars. We will choose a representative from each of these classes in order to present the experiments we have performed: for the upper one trees, for the middle one binomial graphs (obviously) and for the lower one Barabási-Albert graphs.

Before moving one, we will just quickly comment about the similarity between Barabási-Albert graphs and stars. Both classes show lower probabilities of positive instances compared to all other graphs. We think it is the case that this occurs because of the fact that in both graphs there tend to be vertices that are *hubs*, i.e. vertices that are connected to many other vertices. This is clear in the stars, where there is always exactly one vertex that is connected to all the other ones. For Barabási-Albert graphs, we have indirectly discussed the presence of these so-called hubs back when we defined them in the previous section. We would also like to point out that in Barabási-Albert graphs the likelihood of positive instances is slightly higher compared to stars: we believe that this is the case mainly because in Barabási-

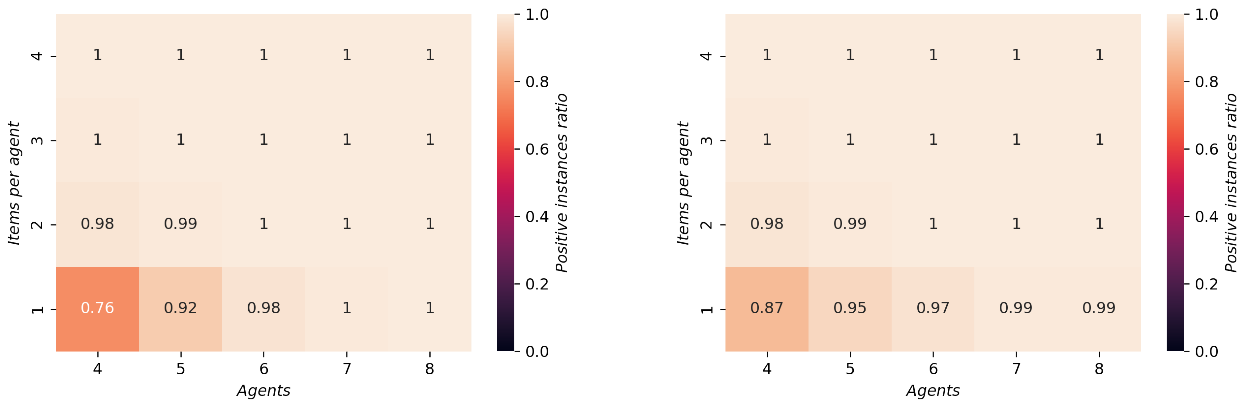
Albert graphs there can be multiple hubs, not necessarily connected to all the other vertices of the graph, unlike what happens in stars.

### 6.2.1 LEF and LPROP

The first outcomes on which we want to comment is the difference in terms of likelihood between LEF and LPROP. In Figures 6.2, 6.3 and 6.4, on the left there are plots for local envy-freeness and on the right for local proportionality for the corresponding graphs (we will keep this layout also in the next figures). The item allocation is the one that maximizes utilitarian welfare.

**Observation 5.** *When item allocations maximize welfare or are generated randomly, LPROP can be satisfied more easily than LEF.*

As one can observe, results for local proportionality are, in general, better than those for local envy-freeness. This can be particularly observed in binomial graphs and Barabási-Albert graphs. This is most likely due to the fact that position assignments that are LEF are also LPROP. Moreover, it might also happen that a position assignment which is not LEF is instead LPROP, because, though an agent might envy one of her neighbors under the given assignment, the other agents who are connected to her can "balance" the enviness so that the average utility of the agents in the augmented neighborhood for the envious agent is small enough.

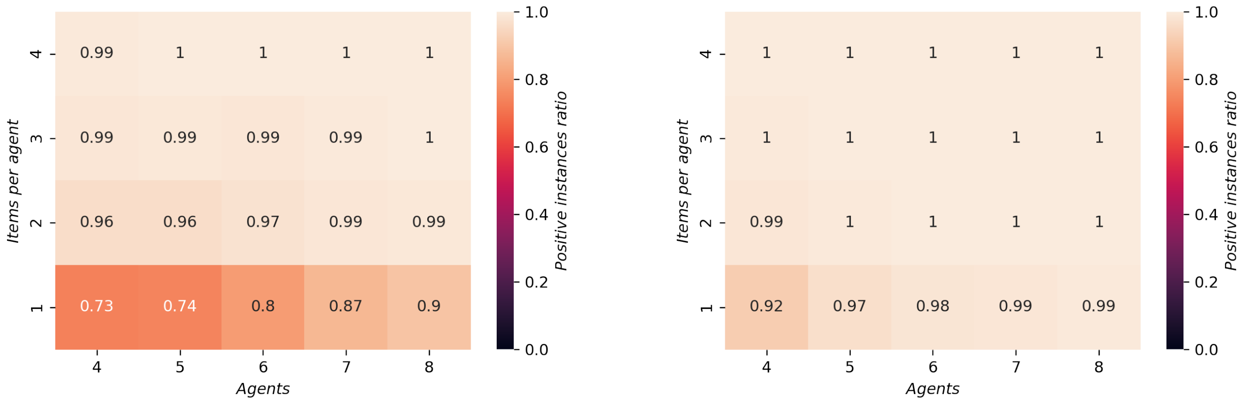



(a) LEF (b) LPROP

Figure 6.2: Uniform utilities - max utilitarian welfare - trees

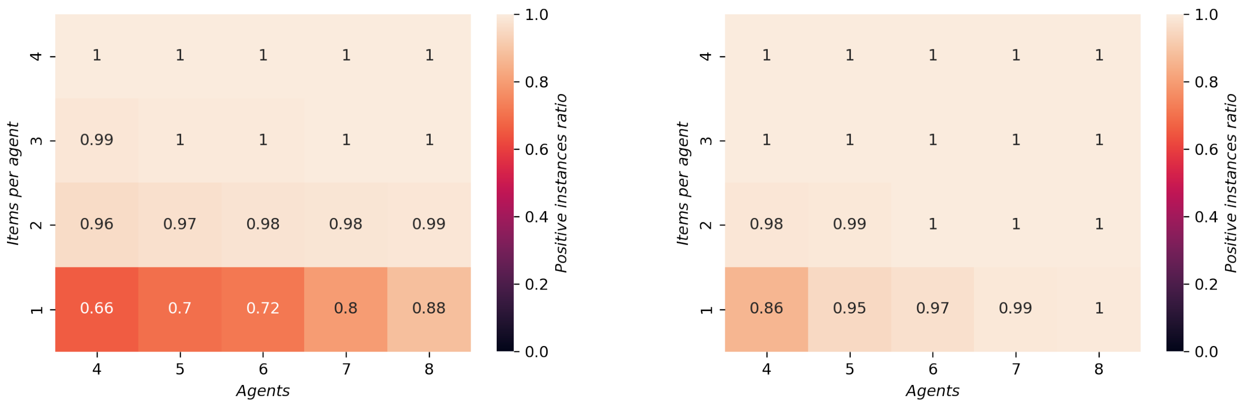
### 6.2.2 Random item allocations

The same pattern between LEF and LPROP can be noticed also when items are given to the agents randomly. In Figure 6.5 we show the results we have




(a) LEF (b) LPROP

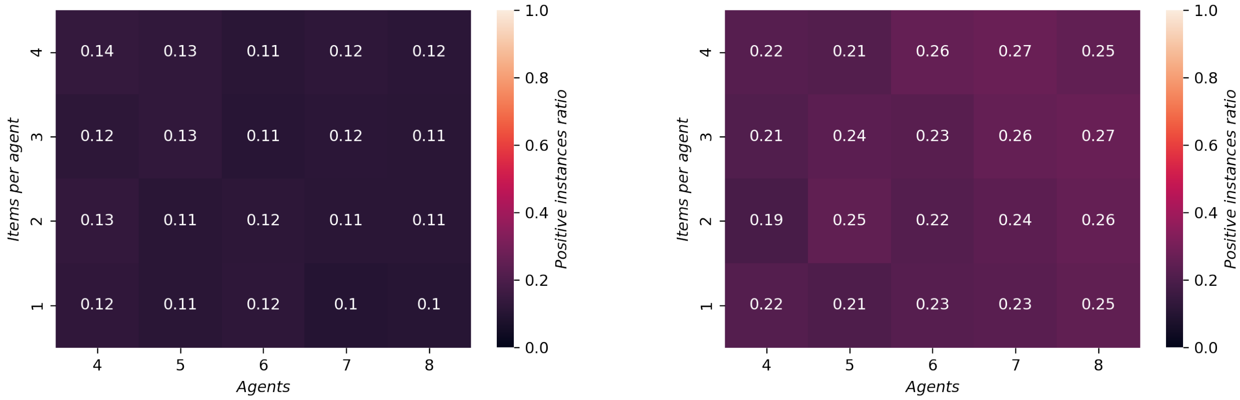
Figure 6.3: Uniform utilities - max utilitarian welfare - binomial graphs




(a) LEF (b) LPROP

Figure 6.4: Uniform utilities - max utilitarian welfare - Barabási-Albert graphs

obtained for random item allocations on trees, both for local envy-freeness and local proportionality.




(a) LEF (b) LPROP

Figure 6.5: Uniform utilities - random - trees

With respect to random allocations, local envy-freeness up to one item has proven to be the criterion satisfied most often. It should be noted that for the LEF1 experiments the number of items per agent varied from 2 to 4.

**Observation 6.** *Under random item allocations, LEF1 is the criterion that can be satisfied more easily.*

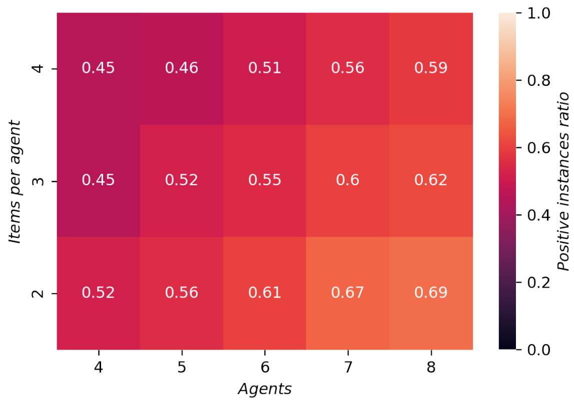



Figure 6.6: Uniform utilities - random - trees - LEF1

Aside from the same previous remark we have made on how the set of

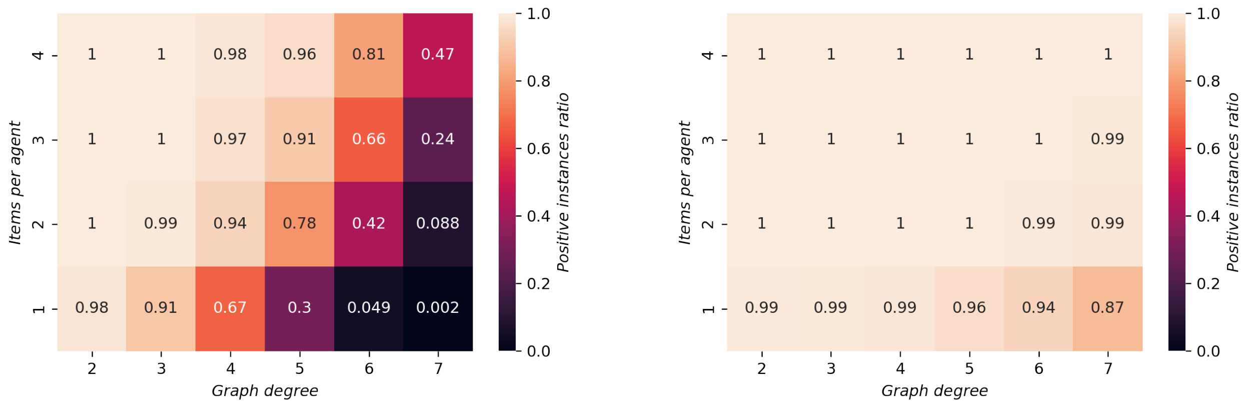
LEF allocations is a subset of the set of LEF1 ones, another factor which might influence this result is the fact that, whenever an agent (with a non-negative utility) is connected to one with a single item, the assignment is automatically LEF1. Though this situation does not always occur, it contributes positively to the occurrence of instances with an LEF1 position assignment. Figure 6.6 shows the results for random allocations when the fairness criterion to satisfy is LEF1. As it can be seen, the peak is reached with 8 agents and 2 items per agent, with a likelihood of 0.69. This is a much higher probability when compared to the maximum likelihoods for LEF and LPROP in the same setting, which are, respectively, 0.14 and 0.27.

### 6.2.3 Regular graphs

Akin to what has happened for the graph classes we have seen so far with respect to the comparison between LEF and LPROP, we have obtained similar results also for regular graphs. Moreover, like what was reported in Beynier et al. [2019], the higher the degree is and the lower the chance that the random instance is positive. The same also holds for local proportionality, however in that case the probabilities are much higher, hence the issue is not as severe.

**Observation 7.** *As the degree of a regular graph increases, the probability of the instances being positive decreases.*

This can clearly be seen in Figure 6.7, where in both heatmaps the likelihood of positive instances decreases when moving from left to right on the $x$-axis: on the first row of the heatmap for LEF, the likelihood if there is a single item per agent is 0.98 if the graph degree is 2 (leftmost cell) and 0.002 if the graph degree is 7 (rightmost cell). This decrease is in turn mitigated by the number of items per agent, as by increasing it the likelihood increases too.

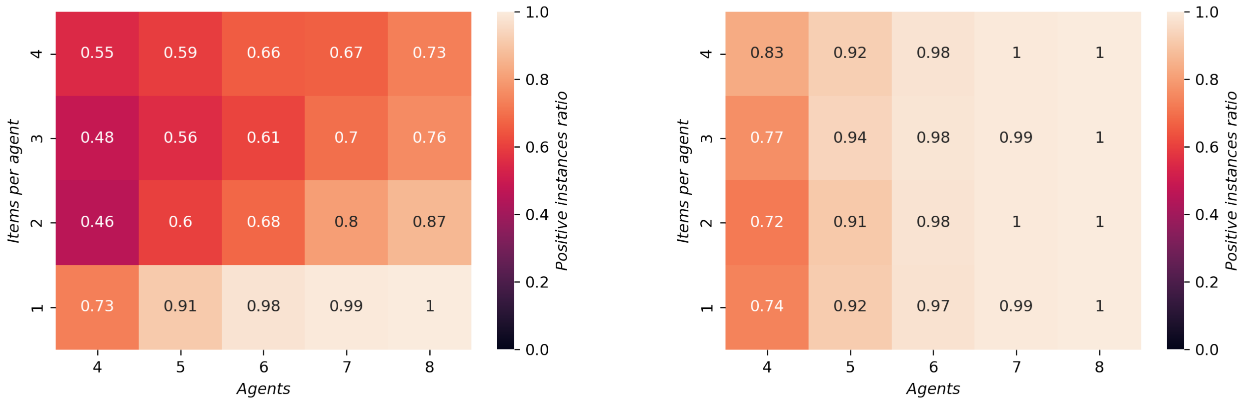



(a) LEF (b) LPROP

Figure 6.7: Uniform utilities - max utilitarian welfare - regular graphs

### 6.2.4 Curious phenomena

We conclude the section with some rather peculiar results. We have performed some experiments also in the case in which the available items are either exclusively goods or exclusively chores. Though it makes sense to expect slightly worse results when compared to having both goods and chores (intuitively because by having both of them the item allocation can be balanced better with respect to the utilities), one would expect also to have approximately the same results for these two kinds of experiments. Indeed, if one considers experiments in which the item allocation either minimizes enviness or is generated randomly, this is true. On the other hand, for those in which the allocation maximizes utilitarian welfare, this intuitive fact is false when there are two or more items per agent, as the results for the only goods case are far worse than those for the only chores case. Figure 6.8 presents the results for the case in which the underlying graph is a tree.




(a) Only goods (b) Only chores

Figure 6.8: Uniform utilities - max utilitarian welfare - trees - LEF

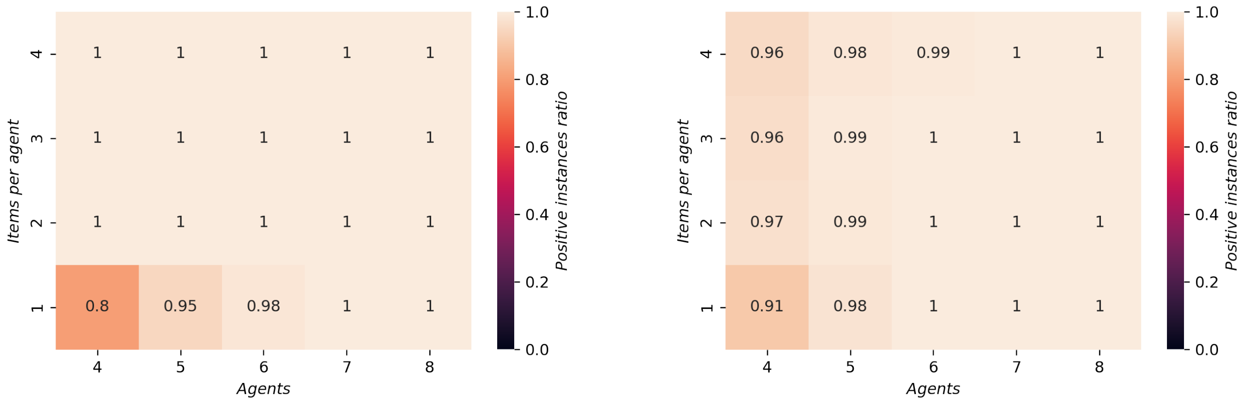
To get an intuition on why this happens, we first have to observe that, given an instance with only goods, for what concerns our case, by subtracting 1 to the utilities of each item for each agent we obtain an "equivalent", with respect to the objective of maximizing utilitarian welfare, instance with only chores. Here, with "equivalent", we intend that the item allocation which maximizes utilitarian welfare is the same. This is because the utility of each item is shifted by $-1$ for each agent, hence the item allocation which maximizes utilitarian welfare in the only goods case will also maximize the utilitarian welfare in the only chores case.

Let us focus for now on the only goods case: what happens when there are two or more items per agent is that when allocating items to maximize utilitarian welfare, once each agent has at least an item, the remaining ones will be assigned to the agent to which the items grant the largest utilities. Because of this, agents who have received a single item might envy those who have received more than one, hence a position assignment which connects these two agents cannot be LEF. Assume that $i$ is an agent who envies some other agent $j$, and assume that $i$ has received a bundle with a single item while $j$ one with $k$ items. Now, if we consider the equivalent instance with only chores, observe that the enviness of $i$ towards $j$ has decreased by $k - 1$: this follows from the fact that the utility granted by each item to $i$ has decreased by 1, and as she had only received one item her utility has decreased only by 1, whereas the utility for $i$ of the bundle assigned to $j$ has decreased by $k$. Thus, if the enviness of $i$ towards $j$ was lower than or equal $k$, in the equivalent instance with only chores $i$ will not envy $j$. On the contrary, if the difference between the utility of $j$'s bundle and that of the single item granted to $i$, from the point of view of $j$, was smaller than $k$, in the new instance $j$ will now envy $i$. However, this will happen less often than the previous kind of change in enviness, because of the fact that the item allocation was performed in order to maximize utilitarian welfare: intuitively, the bundle granted to $j$ should give her a rather large utility.

Instead, consider the case in which we have an instance with only chores, let $i$ and $j$ be agents to which a single item and a bundle of $k$ items were respectively assigned and assume that $i$ does not envy $j$. Perform the inverse transformation on the instance now, i.e. the utility granted by each item to each agent increases by 1, so that we obtain an equivalent (for the goal of maximizing utilitarian welfare) instance with only goods. Now, observe that analogously to what has happened before, this transformation has increased the enviness of $i$ towards $j$ by $k - 1$: this might, in turn, lead $i$ to envy $j$.

We give two simple examples to show how the change in enviness occurs. For the only goods to only chores example, assume that $i$ is such that $|\pi_i| = 1$ and $u_i(\pi_i) = 0.5$ and that $j$ is such that $|\pi_j| = 3$ and $u_i(o) = 0.5$ for each item in $o \in \pi_j$. In this case, $i$ envies $j$ because $u_i(\pi_i) = 0.5 < 1.5 = u_i(\pi_j)$. However, if we move to the corresponding only chores instance, $i$ will no longer envy $j$ because now $u_i(\pi_i) = -0.5 \geq -1.5 = u_i(\pi_j)$. This is because the difference between $u_i(\pi_j)$ and $u_i(\pi_i)$ in the only goods case is smaller than the difference between the sizes of $\pi_j$ and $\pi_i$, hence subtracting 1 to the utilities granted by the items in $\pi_j$ to $i$ counterbalances the enviness. On the other hand, if we consider the same example but reversed, i.e. from the only chores to the only goods instance, we can observe how the enviness can be created. As before, this increase in enviness is due to the difference in sizes of the two bundles.

Thus, our intuition is that an instance with only chores will, in general,




(a) LEF - min enviness item allocation

(b) LPROP - min unproportional item allocation

Figure 6.9: Uniform utilities - trees

have more agents that do not envy other agents under an allocation which maximizes utilitarian welfare, therefore the difference in the likelihood of positive instances.

Another interesting phenomenon is the fact that assigning items in order to minimize the unproportionality, i.e. the sum, for each agent, of the difference between the utility granted by her allocated bundle and the average utility of the whole set of items (averaged over the number of agents) for that same agent, leads in general to worse results for local proportionality if compared to those we obtained when minimizing enviness for local envy-freeness. This is a direct consequence of the fact that if an item allocation is (global) proportional, then it might happen that there is no position assignment which is local proportional (recall Figure 2.1). Figure 6.9 shows this difference between LEF and LPROP, in particular in the first two columns (from the left), where for LPROP no combination of values reaches a likelihood of 1, unlike what happens in LEF where in the first two columns as soon as there two or more items per agent the likelihood is 1.

### 6.2.5 Last remarks

Excluding random allocations, throughout most of the experiments we have always seen a sharp increase when the number of items per agent was larger than or equal to 2. In many cases the likelihood even increased up to 1. An easy explanation for this phenomenon is the fact that as there are more items, there are many more different ways of allocating them. Let $n$ be the number of agents: if there is only one item per agent then the number of possible allocations is $n!$, while if there are two then they are

$n^{2n} + \sum_{i=1}^{n-1}(-1)^i\binom{n}{i}(n-i)^{2n}$. To give the reader a feeling for how much these two values differ, if $n = 4$ then the number of possible assignments if there is one item per agent is 24, whereas if there are two it is 40824. More in general, if we have $n$ agents and $k$ items to assign (with $k \geq n$ obviously), then the number of ways to assign the items (so that each agent gets assigned at least one item) is:

$$n^k + \sum_{i=1}^{n-1}(-1)^i\binom{n}{i}(n-i)^k$$

Another parameter which also seems to be linked to an increase in the likelihood of positive instances is the number of agents itself. The intuition is essentially the same behind the items' one. Moreover, a larger number of agents allows for more diversified utilities.

## 6.3 Effects of EF Agent-Types

In this final section, we will explore how the number of EF agent-types influences the likelihood of positive instances, amongst random ones, for LEF. Clearly, if there is only one agent-type, every random instance will also be a positive one, because every pair of agents will not envy each other as they all have the same agent-type.

Another simple observation one can make is that, if there are only two agent-types and the graph is strongly connected, then no (random) instance can be a positive one, simply because somewhere two agents of different type must be connected and at least one of the two types must envy the other. This effect can clearly be seen in Figure 6.10.

As one can also observe the graphs with the highest likelihood are the matchings. This can be explained by the fact that, as each vertex is connected to only another one, the probability that a position assignment which is LEF exists are higher because in these graphs each each agent will be connected to only another one. Because of this, it is easier (compared to other families of graphs) to arrange agents in such a way that they do not envy each other, even if they are of not of the same type. Moreover, if each agent-type is such that there is an even number of agents of that type, it follows that there is a position assignment which is LEF, namely any of the ones such that agents of the same type are connected.

Trees and lines follow immediately after in the classes with highest likelihood, probably because of the fact that also in their case the number each vertex is not connected to many other vertices (though not as few as in a matching).

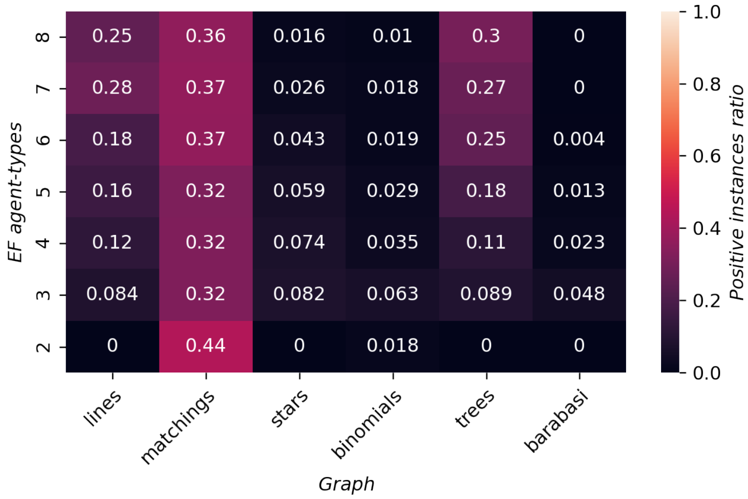



Figure 6.10: EF agent-types experiments for all graph classes except regular graphs.

On the other side of the spectrum, we have the Barabási-Albert graphs, in which each vertex is connected to three other vertices (by how we have generated the graphs). An interesting result one might notice is the fact that binomials graph have a higher probability of generating a positive instance compared to Barabási-Albert graphs, although in any graph we generate there is an edge between two vertices with a probability of $\frac{1}{2}$, hence one can expect each vertex to be connected to three or four other vertices on average. We believe that in this case the presence of hubs (which we discussed in the previous section) in Barabási-Albert graphs is amongst the main culprits of this difference, together with the fact that it is also possible that binomial graphs which are generated can be *sparse*, i.e. with few edges.

On the other hand, as one might expect, results for regular graphs are extremely negative.

Like before, as soon as we consider only strongly connected graphs (degrees larger than or equal to 4), if there are only two agent-types then no random instance can be a positive one. One can notice that regular graphs which degree is equal to 2 are those with the better results: this can be justified, yet again, by the fact that they are those with fewer edges.

## 6.4 Summary

We have presented the results of our experiments, where we were able to observe which parameters mostly influenced the likelihood of positive

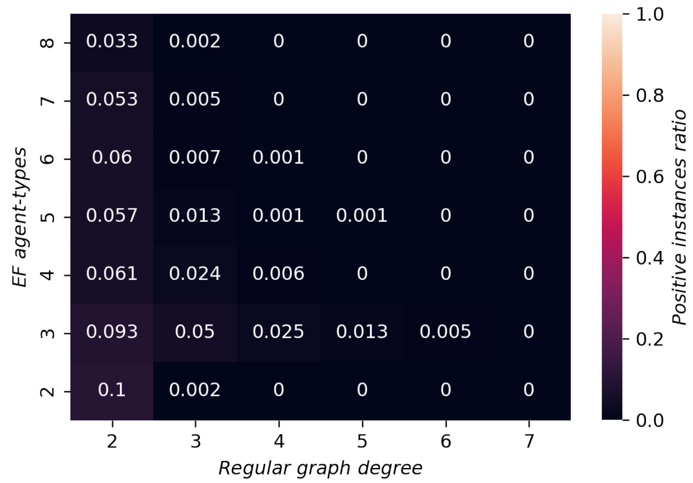



Figure 6.11: EF agent-type results for regular graphs.

instances amongst random ones. The number of items per agent is probably the one that mostly impacts the probability for each fairness criterion. If we compare local envy-freeness and local proportionality, the latter is the most probable one. On the other hand, local envy-freeness up to one item has proved to be the criterion that is most probable to be satisfied when the allocation is performed randomly.

Trees, lines and matchings are the graphs that have shown a higher likelihood, whereas Barabási-Albert graphs and stars lie on the opposite side of the spectrum. Regular graphs deserve a special mention as they were also studied in Beynier et al. [2019]: we can confirm that for local envy-freeness the higher the degree is and the lower the likelihood is. This also happens for local proportionality, however the probability of positive instances is much higher in this case and is always almost 1. Finally, for any of the fairness criteria and graph class, the likelihood is not very high in case the item allocation is done randomly.

We have also studied how the number of EF agent-types in the instance affects this probability (for local envy-freeness). In general, an increase in such number decreases the likelihood in stars, binomial graphs and Barabási-Albert graphs, whereas in trees, lines and matchings the trend is not so clear and the likelihood might increase or decrease as the number of agent-types grows. Finally, regular graphs generally have a very low likelihood. This is mainly due to the fact that all vertices in the graph will be connected to many other ones, hence having more than one agent-type will, in most of the cases, lead to worse results due to agent-types envying other types.

# Chapter 7

# Conclusion

We have presented a new direction in fair division in which the main task to perform is to place agents on a social network in such a way that certain fairness criteria are satisfied. Aside from placing the agents, we have also considered the case in which the item allocation has to be done as well. Our main goal was to study the (parameterized) computational complexity of these problems and to try to understand how it changed based on two main factors: the fairness criterion which was to be satisfied and the structure of the underlying social graph.

In Chapters 3 through 5 we have given various results in complexity theory, divided on the fairness criterion to satisfy. With respect to classical complexity, local envy-freeness and local proportionality share the exact same results in terms of tractability and intractability for each combination of the problems and graph classes we have considered. On the other hand, local envy-freeness up to one item differs, in classical complexity, from the other two criteria just in case the item allocation has to be performed. This is because of Proposition 1, thanks to which it is always possible to find an LEF1 item allocation in polynomial time. In terms of parameterized complexity, local envy-freeness and local envy-freeness up to one item behave in the same way, while for local proportionality we were not able to prove all of the results. In particular, we have been able to find some parameterizations which grant positive results (e.g. Theorems 2 and 8) in case the underlying social graph is a tree or a forest.

Tables 7.1, 7.2 and 7.3 present all the complexity results we have obtained for, respectively, local envy-freeness, local envy-freeness up to one item and local proportionality.

We have also performed some experiments which main objective was to study how often positive instances arise in random ones. Here with "positive" instances we refer to those instances for which there is a position assignment that satisfies the criterion we are interested in with the given item allocation. For these experiments, we have used a variety of parameters: the structure of the graph, the number of agents, the number of items per agent, the fairness criterion to satisfy, the distribution from which the agents' utilities are drawn and the way the item allocation is performed. It turns out that the parameter which contributes mostly to the increase in such likelihood is the number of items per agent, except in case the item allocation is performed randomly. We have also studied how EF agent-types (see Definition 21) affect this probability, and observed that in most of the cases having more than one agent-type leads to a sharp decrease of the likelihood that the instance is positive for local envy-freeness. Notice that, by how we have performed this last kind of experiments, we can substitute local envy-freeness with local envy-freeness up to one item, as the EF agent-types can be substituted by the EF1 agent-types without changing nothing in the experiments. This is because we generate randomly the relations between agent-types in these experiments, and not the utilities.

## 7.1 Future research

We conclude the thesis by giving some directions for future research which we think are promising or noteworthy to be mentioned.

We believe that parameterized complexity offers many directions to explore in this field. For starters, we think that finding new (non-trivial) parameterizations which treat also graphs that are not trees nor forests is definitely a natural direction in which one can move. We feel optimistic with respect to parameterized complexity because of the positive results which we were able to achieve in our work and also of those that can be found in other works (e.g. Eiben et al. [2020]).

Still with respect to parameterized complexity, a possible direction could be to study how local proportionality affects the complexity of the problems we have studied. As the reader might recall, we were not able to replicate most of the parameterized results we achieved with respect to local envy-freeness and local envy-freeness up to one item for local proportionality. In Section 5.4 we have proposed some ways to slightly modify notions we have used in order to try to tackle this problem; another different way to address it might be to define some different parameters specifically designed to be used with local proportionality.

Another direction which we think would be interesting to explore is to consider social graphs that are directed instead of undirected. If we see the social graph as some kind of hierarchy in the company, using a directed graph would reflect the fact that perhaps some employee high up in the hierarchy has knowledge of employees below her, but those underneath her do not have information about her. In this case it might also make sense to start again from classical complexity and see if there are any differences compared to undirected graphs. As an example of this kind of setting we refer, as we did in Section 1.3, to Bredereck, Kaczmarczyk, and Niedermeier [2018]. Out of the three criteria we have studied, in their work they have only considered local envy-freeness (which they call "*weak graph-envy-freeness*"). Thus, both local envy-freeness up to one item and local proportionality have yet to be studied in this case.

Another natural continuation would be that of considering other fairness criteria, like envy-freeness up to any item, a variant of envy-freeness up to one item in which an agent does not envy up to any item another one if and only if, by removing any item in the union of their bundles, she does not envy her given the new bundles. Inspired by works like Aziz, Caragiannis, et al. [2019] and Plaut and Roughgarden [2020], it could also be interesting to study combinations of fairness and efficiency criteria, where an example of an efficiency criteria is Pareto efficiency. Finally, one could also consider group criteria, where the intuition behind this is that this can model the interaction between groups inside a company.

| | E-LEF-P | F-LEF-P | E-LEF-D | F-LEF-D | Parameters |
|---|---|---|---|---|---|
| Stars | P<br>Prop. 2 | P<br>Prop. 2 | NP-c<br>Cor. 1 | $\geq$ NP-h<br>Cor. 1 | - |
| Lines | NP-c<br>Prop. 3 | $\geq$ NP-h<br>Prop. 3 | NP-c<br>Cor. 1 | $\geq$ NP-h<br>Cor. 1 | - |
| Strongly conn. | NP-c<br>Prop. 3 | $\geq$ NP-h<br>Prop. 3 | NP-c<br>Thm. 1 | $\geq$ NP-h<br>Thm. 1 | - |
| Matchings | P<br>Prop. 4 | P<br>Prop. 4 | NP-c<br>Prop. 5 | $\geq$ NP-h<br>Prop. 5 | - |
| Trees | FPT<br>Thm. 2 | FPT<br>Thm. 2 | - | - | Number of vertex-types and EF agent-types |
| Forests | FPT<br>Thm. 3 | FPT<br>Thm. 3 | - | - | Number of vertex-types and EF agent-types |
| | para-NP-h<br>Cor. 2 | $\geq$ para-NP-h<br>Cor. 2 | - | - | Number of vertex-types |
| tw $= O(1)$ | FPT<br>Prop. 6 | $\geq$ FPT<br>Prop. 6 | - | - | Number of non-isolated vertices |
| tw $\neq O(1)$ | $W[1]$-h<br>Thm. 5 | $\geq W[1]$-h<br>Thm. 5 | - | - | Number of non-isolated vertices |

Table 7.1: Summary of results for local envy-freeness. "E", "F", "P" and "D" respectively stand for "EXISTS", "FIND", "POSITION-ASSIGNMENT" and "DISTRIBUTION", while "h" and "c" respectively abbreviate "hard" and "complete". With tw $= O(1)$ ($\neq O(1)$) we denote the class of graphs with a constant (non-constant) treewidth (see Definition 19). A $\geq$ indicates that there is no exact result, but that the problem's complexity is *at least* the one after the $\geq$.

| | E-LEF1-P | F-LEF1-P | E-LEF1-D | F-LEF1-D | Parameters |
|---|---|---|---|---|---|
| Stars | P<br>Prop. 7 | P<br>Prop. 7 | P<br>Cor. 3 | P<br>Cor. 3 | - |
| Lines | NP-c<br>Prop. 8 | $\geq$ NP-h<br>Prop. 8 | P<br>Cor. 3 | P<br>Cor. 3 | - |
| Strongly conn. | NP-c<br>Prop. 8 | $\geq$ NP-h<br>Prop. 8 | P<br>Cor. 3 | P<br>Cor. 3 | - |
| Matchings | P<br>Prop. 9 | P<br>Prop. 9 | P<br>Cor. 3 | P<br>Cor. 3 | - |
| Trees | FPT<br>Thm. 8 | FPT<br>Thm. 8 | P<br>Cor. 3 | P<br>Cor. 3 | Number of vertex-types and EF1 agent-types |
| Forests | FPT<br>Thm. 9 | FPT<br>Thm. 9 | P<br>Cor. 3 | P<br>Cor. 3 | Number of vertex-types and EF1 agent-types |
| | para-NP-h<br>Cor. 4 | $\geq$ para-NP-h<br>Cor. 4 | - | - | Number of vertex-types |
| tw $= O(1)$ | FPT<br>Prop. 10 | $\geq$ FPT<br>Prop. 10 | P<br>Cor. 3 | P<br>Cor. 3 | Number of non-isolated vertices |
| tw $\neq O(1)$ | $W[1]$-h<br>Thm. 11 | $\geq W[1]$-h<br>Thm. 11 | P<br>Cor. 3 | P<br>Cor. 3 | Number of non-isolated vertices |

Table 7.2: Summary of results for local envy-freeness up to one item. We use the same notation as in Table 7.1.

| | E-LPROP-P | F-LPROP-P | E-LPROP-D | F-LPROP-D | Parameters |
|---|---|---|---|---|---|
| Stars | P<br>Prop. 11 | P<br>Prop. 11 | NP-c<br>Cor. 5 | $\geq$ NP-h<br>Cor. 5 | - |
| Lines | NP-c<br>Prop. 12 | $\geq$ NP-h<br>Prop. 12 | NP-c<br>Cor. 5 | $\geq$ NP-h<br>Cor. 5 | - |
| Strongly conn. | NP-c<br>Prop. 12 | $\geq$ NP-h<br>Prop. 12 | NP-c<br>Thm. 12 | $\geq$ NP-h<br>Thm. 12 | - |
| Matchings | P<br>Prop. 13 | P<br>Prop. 13 | NP-c<br>Prop. 14 | $\geq$ NP-h<br>Prop. 14 | - |
| Trees | - | - | - | - | - |
| Forests | para-NP-h<br>Cor. 6 | $\geq$ para-NP-h<br>Cor. 6 | - | - | Number of vertex-types |
| $\text{tw} = O(1)$ | - | - | - | - | - |
| $\text{tw} \neq O(1)$ | - | - | - | - | - |

Table 7.3: Summary of results for local proportionality. We use the same notation as in Table 7.1.

# References

[1] Rediet Abebe, Jon Kleinberg, and David C. Parkes. "Fair Division via Social Comparison". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2017)*. 2017, pp. 281–289.

[2] Noga Alon, Raphael Yuster, and Uri Zwick. "Color-Coding". In: *Journal of the ACM* 42.4 (1995), pp. 844–856.

[3] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[4] Haris Aziz. "Developments in Multi-Agent Fair Allocation". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. 2020, pp. 13563–13568.

[5] Haris Aziz, Sylvain Bouveret, Ioannis Caragiannis, Ira Giagkousi, and Jérôme Lang. "Knowledge, Fairness, and Social Constraints". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*. 2018, pp. 4638–4645.

[6] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. "Fair Allocation of Indivisible Goods and Chores". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019)*. 2019, pp. 53–59.

[7] Haris Aziz and Simon Rey. "Almost Group Envy-free Allocation of Indivisible Goods and Chores". In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*. 2020, pp. 39–45.

[8] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512.

[9] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. "Finding Fair and Efficient Allocations". In: *Proceedings of the 2018 ACM Conference on Economics and Computation (EC 2018)*. 2018, pp. 557–574.

[10] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. *The Price of Connectivity in Fair Division*. 2019. arXiv: `1908.05433 [cs.GT]`.

[11] Xiaohui Bei, Youming Qiao, and Shengyu Zhang. "Networked Fairness in Cake Cutting". In: *Proceedings of the 26th International Joint

*Conference on Artificial Intelligence (IJCAI 2017)*. 2017, pp. 3632–3638.

[12] Matthias Bentert, Jiehua Chen, Vincent Froese, and Gerhard J. Woeginger. *Good Things Come to Those Who Swap Objects on Paths*. 2019. arXiv: `1905.04219 [cs.DS]`.

[13] Aurélie Beynier, Yann Chevaleyre, Laurent Gourvès, Ararat Harutyunyan, Julien Lesca, Nicolas Maudet, and Anaëlle Wilczynski. “Local Envy-Freeness in House Allocation Problems”. In: *Autonomous Agents and Multi-Agent Systems* 33.5 (2019), pp. 591–627.

[14] Hans L. Bodlaender. “Treewidth: Characterizations, Applications, and Computations”. In: *Graph-Theoretic Concepts in Computer Science*. Ed. by Fedor V. Fomin. Springer-Verlag Berlin Heidelberg, 2006, pp. 1–14.

[15] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. “Fair Division of a Graph”. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*. 2017, pp. 135–141.

[16] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. “Fair Allocation of Indivisible Goods”. In: *Handbook of Computational Social Choice*. Ed. by Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. Cambridge University Press, 2016. Chap. 12, pp. 284–310.

[17] Sylvain Bouveret and Michel Lemaître. “Characterizing conflicts in fair division of indivisible goods using a scale of criteria”. In: *Autonomous Agents and Multi-Agent Systems* 30.2 (2016), pp. 259–290.

[18] Steven J. Brams and Alan D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[19] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. “Envy-Free Allocations Respecting Social Networks”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*. 2018, pp. 283–291.

[20] Eric Budish. “The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes”. In: *Journal of Political Economy* 119.6 (2011), pp. 1061–1103.

[21] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. “The Unreasonable Fairness of Maximum Nash Welfare”. In: *ACM Transactions on Economics and Computations* 7.3 (2019).

[22] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. “Reaching Envy-Free States in Distributed Negotiation Settings”. In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*. 2007, pp. 1239–1244.

[23] Yann Chevaleyre, Ulle Endriss, and Nicolas Maudet. “Allocating Goods on a Graph to Eliminate Envy”. In: *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007)*. 2007, pp. 700–

705.

[24] Rod G. Downey and Michael R. Fellows. "Fixed-parameter tractability and completeness II: On completeness for $W[1]$". In: *Theoretical Computer Science* 141.1-2 (1995), pp. 109–131.

[25] Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. "Parameterized Complexity of Envy-Free Resource Allocation in Social Networks". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. 2020, pp. 7135–7142.

[26] Paul Erdős and Alfred Rényi. "On random graphs I.". In: *Publicationes Mathematicae* 6 (1959), pp. 290–297.

[27] Leon Festinger. "A Theory of Social Comparison Processes". In: *Human Relations* 7.2 (1954), pp. 117–140.

[28] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer-Verlag Berlin Heidelberg, 2006.

[29] Duncan K. Foley. "Resource allocation and the public sector". In: *Yale economic essays* 7.1 (1967), pp. 45–98.

[30] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP*-Completeness*. W. H. Freeman & Co., 1979.

[31] Edgar N. Gilbert. "Random Graphs". In: *The Annals of Mathematical Statistics* 30.4 (1959), pp. 1141–1144.

[32] Paul W. Goldberg, Alexandros Hollender, and Warut Suksompong. "Contiguous Cake Cutting: Hardness Results and Approximation Algorithms". In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. 2020, pp. 1990–1997.

[33] Teofilo F. Gonzalez. "Clustering to minimize the maximum intercluster distance". In: *Theoretical Computer Science* (1985), pp. 293–306.

[34] Laurent Gourvès, Julien Lesca, and Anaëlle Wilczynski. "Object Allocation via Swaps along a Social Network". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*. 2017, pp. 213–219.

[35] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". in: *Proceedings of the 7th Python in Science Conference (SciPy 2008)*. 2008, pp. 11–15.

[36] Sen Huang and Mingyu Xiao. "Object Reachability via Swaps along a Line". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*. 2019, pp. 2037–2044.

[37] Ayumi Igarashi and Dominik Peters. "Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*. 2019, pp. 2045–2052.

[38] Jérôme Lang. "Fair Division of Indivisible Goods". In: *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by Jörg Rothe. Springer-Verlag

Berlin Heidelberg, 2016. Chap. 8, pp. 493–550.
[39] Hendrik W. Lenstra. "Integer Programming with a Fixed Number of Variables". In: *Mathematics of Operations Research* 8.4 (1983), pp. 538–548.
[40] Claudia Lindner. "Cake-Cutting: Fair Division of Divisible Goods". In: *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by Jörg Rothe. Springer-Verlag Berlin Heidelberg, 2016. Chap. 7, pp. 395–491.
[41] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. "On Approximately Fair Allocations of Indivisible Goods". In: *Proceedings of the 5th ACM Conference on Electronic Commerce (EC 2004)*. 2004, pp. 125–131.
[42] Zbigniew Lonc and Miroslaw Truszczynski. "Maximin Share Allocations on Cycles". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. 2018, pp. 410–416.
[43] Haider Maqsood, Aamir Alamzeb, Abdul H. Abu-Bakr, and Hashim Muhammad. "A literature Analysis on the Importance of Non-Financial Rewards for Employees' Job Satisfaction". In: *Abasyn University Journal of Social Sciences* 8 (2015), pp. 341–354.
[44] Evangelos Markakis. "Approximation algorithms and hardness results for fair division with indivisible goods." In: *Trends in Computational Social Choice*. Ed. by Ulle Endriss. AI Access, 2017. Chap. 12, pp. 231–247.
[45] Silvio Micali and Vijay V. Vazirani. "An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs". In: *Proceedings of the 21st Annual Symposium on Foundations of Computer Science (SFCS 1980)*. 1980, pp. 17–27.
[46] Hervé Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.
[47] Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe. "Computational complexity and approximability of social welfare optimization in multiagent resource allocation". In: *Autonomous Agents and Multi-Agent Systems* 28.2 (2014), pp. 256–289.
[48] Benjamin Plaut and Tim Roughgarden. "Almost Envy-Freeness with General Valuations". In: *SIAM Journal on Discrete Mathematics* 34.2 (2020), pp. 1039–1068.
[49] Ariel D. Procaccia. "Cake Cutting: Not Just Child's Play". In: *Communications of the ACM* 56.7 (2013), pp. 78–87.
[50] Ariel D. Procaccia. "Cake Cutting Algorithms". In: *Handbook of Computational Social Choice*. Ed. by Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. Cambridge University Press, 2016. Chap. 13, pp. 311–330.
[51] Jack M. Robertson and William Webb. *Cake-cutting algorithms - be fair if you can*. Taylor & Francis, 1998.
[52] Hugo Steinhaus. "The problem of fair division". In: *Econometrica* 16.1

(1948), pp. 101–104.
[53] William Thomson. "Introduction to the Theory of Fair Allocation". In: *Handbook of Computational Social Choice*. Ed. by Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. Cambridge University Press, 2016. Chap. 11, pp. 261–283.
[54] Toby Walsh. "Fair Division: The Computer Scientist's Perspective". In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*. 2020, pp. 4966–4972.
[55] Michael Waskom and the seaborn development team. *mwaskom/seaborn*. Sept. 2020. URL: `https://doi.org/10.5281/zenodo.592845`.