Envy-Free Allocations Respecting Social Networks

Robert Bredereck
TU Berlin
Berlin, Germany
robert.bredereck@tu-berlin.de

Andrzej Kaczmarczyk
TU Berlin
Berlin, Germany
a.kaczmarczyk@tu-berlin.de

Rolf Niedermeier
TU Berlin
Berlin, Germany
rolf.niedermeier@tu-berlin.de

ABSTRACT

Finding an envy-free allocation of indivisible resources to agents is a central task in many multiagent systems. Often, non-trivial envy-free allocations do not exist, and finding them can be a computationally hard task. Classic envy-freeness requires that every agent likes the resources allocated to it at least as much as the resources allocated to *any* other agent. In many situations this assumption can be relaxed since agents often do not even know each other. We enrich the envy-freeness concept by taking into account (directed) social networks of the agents. Thus, we require that every agent likes its own allocation at least as much as those of all its (out)neighbors. This leads to a "more local" concept of envy-freeness. We also consider a strong variant where every agent must like its own allocation more than those of all its (out)neighbors.

We analyze the classic and the parameterized complexity of finding allocations that are envy-free with respect to one of the variants of our new concept, and that either are complete, are Pareto-efficient, or optimize the utilitarian social welfare. To this end, we study different restrictions of the agents' preferences and of the social network structure. We identify cases that become easier (from Σ_2^P-hard or NP-hard to P) and cases that become harder (from P to NP-hard) when comparing classic envy-freeness with our graph-based envy-freeness. Furthermore, we spot cases where graph envy-freeness is easier to decide than strong graph envy-freeness, and vice versa.

ACM Reference Format:
Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. 2018. Envy-Free Allocations Respecting Social Networks. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Modern management strategies emphasize the role of teams and team-work. To have an effective team one has to motivate the team members in a proper way. One method of motivating team members is to reward them for achieving a milestone. On the one hand, it is crucial that every member of a team feels rewarded fairly. On the other hand, in every team there are hierarchical or personal relations, which one should attend to in the rewarding process. Since, according to the recent labor statistics in the US [19], the average cost of employee benefits (excluding legally required ones) is around 25% of the whole cost of labor, it is important to effectively use rewarding instruments. It is tempting to follow a simplistic belief that tangible incentives motivate best and thus reward employees with cash bonuses and pay raises. However, it has been shown that to keep the employee satisfaction high, an employer should also honor the employees with non-financial rewards [12].

We propose a model for the fair distibution of indivisible goods which can be used to find an allocation of non-financial rewards[1] such that each team member is satisfied with its rewards and, at the same time, is not worse off compared to any other peer whom she is in relation with. Besides the rewarding scenario our model has numerous further potential applications, just to mention targeting marketing strategies (giving non-monetary bonuses to loyal customers), allocating physical resources to virtual resources in virtualization technologies (both network and machine virtualization), and sharing charitable donations between cities or communities which may envy each other.

Returning to our initial example of reward management, it is a well-established fact that team members evaluate the fairness of rewarding based on comparisons with their peers. This phenomenon, first described seventy years ago by the social psychologist Leon Festinger [10], is probably one of the reasons of the popularity of fair allocation (division) problems in computer science. Naturally, when evaluating the subjective fairness of rewards, every team member tends to compare itself to similar peers, neglecting those who differ substantially in position, abilities, or other aspects. This has already been reflected by one of Festinger's hypotheses; however, so far, most reserach in computer science has focused on fairness notions based on "global" comparisons, that is, pairwise comparisons between all members of society.

In this work we aim at incorporating "local" comparisons into the fair allocation scenario. Having a collection of indivisible resources we look for a way to distribute them fairly among a group of agents which, prior to the distribution, shared their opinions on how they appreciate the resources. For example, imagine that a company is to reward a team of three employees responsible for a successful project. The team consists of a key account manager (KAM) being the chief of the group, an internet sales manager (ISM), and a business-to-business (B2BSM) sales manager. The company intends some non-financial rewards to recognize the employees' performances. The rewards are 'participating in a language course', 'being the company's representative for an episode of a documentary program', 'moving to a new high-end office', and 'receiving an employee-of-the-month award'. The employees (agents) were surveyed for their favorite rewards, yielding the results given in Table 1.

Each agent considers a rewarding unfair if after exchanging all its rewards with all rewards of some peer, the agent would get more approved rewards. According to the company's rewarding policy, all rewards must be handed out. Considering the standard model

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

[1] Financial rewards can be interpreted as divisible resources while we focus on indivisible resources.

	KAM	B2BSM	ISM
language course	☐	☑	☑
TV episode	☐	☑	☑
high-end office	☑	☑	☑
employee-of-the-month award	☑	☑	☑

Table 1: The results of a survey concerning employees' (0/1) preferences over the possible rewards. Checked boxes indicate the approved rewards of a particular person.

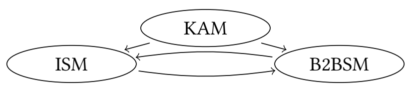

Figure 1: An illustration of who compares to whom for the introductory example. Every node represents an employee and arcs represent directions of comparisons, for instance, if an arc points from the key account manager to the internet sales manager, then the former compares herself to the latter.

of resource allocation, where each agent can compare itself to each other agent, the company cannot find a fair reward allocation. At least one agent has to get two rewards. As a consequence, two employees have at most one reward. However, a rewarding policy in the company assumes that a team's chief is always a basis of team success and thus deserves a better reward. Hence, both sales managers do not compare their rewards to the ones of their boss. Naturally, the key account manager's reward should be at least as good as the ones of the others. To illustrate these relations, we use the directed graph depicted in Figure 1. In this case, the company can reward the key account manager with the office and the employee-of-the-month award, and distribute the two remaining rewards equally to the internet and business-to-business managers. Doing so, the company achieves a fair rewarding. The key account manager has two favorite rewards and there is no incentive to exchange them. The remaining team members do not compare themselves to their boss, so they do not envy him or her. Finally, both the business-to-business and internet managers have one favorite reward, so there is no envy. Thus, by introducing the graph of relations between the employees, we were able to represent social comparisons.

Related Work. In 1948, Steinhaus [23] asked how to fairly distribute a continuous resource, a "cake", among a set of agents with (possibly different) heterogeneous valuations of the resource. From this first mathematical model of fair allocation two main research directions evolved. The difference lies in the nature of the resources—divisible or indivisible. The former type yields the so-called cake cutting problem. We refer to the books [7, 18, 22] and recent surveys [5, 17, 20, 21] on fair division problems, and next discuss literature related to our setting.

Abebe et al. [1] and Bei et al. [2] introduced social networks of agents into the fair division problem. They defined (local) fairness concepts based on social networks and then compared them to the classic fairness notions and designed new protocols to find envy-free allocations. Although their models defined local envy-freeness, it differs from our concept since they considered divisible resources.

An allocation, instead of being executed by a central mechanism, might emerge from a sequence of trades between the agents initially endowed with random resources; this setting gives birth to the problem of *distributed* allocation of indivisible resources. Gourvés et al. [11] studied this problem of embedding agents in a social network describing the possible agent interactions and restricting allocations to give a single resource to every agent. They addressed the computational hardness of several questions such as existence of a Pareto-efficient allocation, reachability of a particular allocation, or reachability of a resource for a candidate. For all these questions, they proved that finding an answer is in general NP-hard but it is polynomial-time solvable for some constrained cases. Additionally, Chevaleyre et al. [8] enriched the distributed allocation problem with monetary payments for the trades. They defined a version of graph envy-freeness which takes into account both allocations of resources and the payments of agents. They showed several results describing convergence of trades converging to a fair allocation. Additionally, they proved that the problem of finding a deal reducing unfairness among the agents is NP-hard.

The somewhat orthogonal model where relations of resources, instead of agents, are described by a graph was recently studied by Bouveret et al. [4]. They proved that to decide whether there is a fair allocation such that every assigned bundle contains only resources forming a connected component is, in general, NP-hard. Furthermore, Suksompong [24] studied the existence and properties of approximate versions of various fairness concepts in the special case of resources lying on a path.

Our Contributions. Our work follows the recent trend of combining fair allocation with social networks. We introduce social relations into the area of fair allocation of *indivisible* resources *without monetary payments*. Making use of a greater model flexibility resulting from embedding agents into a social network, we define two new versions of the classic envy-freeness property; namely, (weak) graph-envy-freeness and strong graph-envy-freeness. Even though Chevaleyre et al. [8] also introduced a property called graph-envy-freeness, their version differs from ours significantly because, instead of being a property of an allocation, it describes a particular state of the negotiations between the agents, including monetary payments (which has the flavor of divisible resources) paid to the agents so far.

We study problems of finding (weakly/strongly) graph-envy-free and efficient allocations employing separately completeness, Pareto-efficiency, and maximization of utilitarian social welfare as efficiency criteria. We assume that the agents' preferences over the resources are cardinal, additive, and monotonic. We go beyond the general case (with no further constraints on agents' preferences and an arbitrary social network), and we analyze our problems with respect to social networks being directed acyclic graphs or strongly connected components, and with respect to identical or 0/1 preferences over the resources. As a result, we explore a broad and diverse landscape of the classic computational complexity of the introduced problems. Our results reveal that in comparison to classic envy-freeness, our model sometimes simplifies the task of finding a proper allocation and sometimes makes it harder. Similarly, we identify cases where finding a (weakly) graph-envy-free allocation is easier than finding a strongly graph-envy-free allocation but also cases where the opposite is true. Additionally, our work assesses the parameterized computational complexity of several cases with respect to a few natural parameters such as the number of agents, the number of resources, and the maximum number of neighbors of an agent.

In the following sections, firstly, we present basic concepts and introduce our new model and computational problems (Section 2). Then, we analyze the problem of finding complete graph envy-free allocations (Section 3). We show that the majority of results for the case of complete graph envy-free allocations can be transferred to the other efficiency concepts; we also provide results where this transfer is impossible (Section 4). We end with conclusions and suggestions for future work (Section 5). Due to the lack of space, we defer several proof details to the full version of the paper.

2 MODEL AND BASIC DEFINITIONS

We start with basic concepts for describing graphs which we use to model relations between agents. For a directed graph $G = (V, E)$, consisting of a set V of vertices and a set E of arcs, by $N(v)$ we denote the outneighborhood of vertex $v \in V$, i.e., the set $W \subset V$ of vertices such that for each vertex $w \in W$ there exists an arc $e = (v, w) \in E$, i.e., arc e is directed from v to w. Where needed, we complement our notation by using a subscript indicating the graph we consider.

We continue with defining some standard concepts for allocation problems needed to formally introduce our problems.

Definition 2.1. An *allocation* of a set of resources $\mathcal{R}$ to a set of agents $\mathcal{A}$ is a mapping $\pi\colon \mathcal{A} \to 2^{\mathcal{R}}$ such that $\pi(a)$ and $\pi(a')$ are disjoint whenever $a \neq a'$. For any agent $a \in \mathcal{A}$, we call $\pi(a)$ the *bundle* of a under π.

There are different ways to model preferences of agents over resources; we focus on preferences expressed numerically.

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

For additive preferences, $\preceq$ is caled *monotonic* if and only if the values of the utility function are non-negative. In our work, we restrict preferences to be additive and monotonic. We call them 0/1 if the utility function maps to $\{0, 1\}$ for every agent, and *identical* if every agent has the same utility function.

Next, we formally define our graph fairness concepts based on comparisons between neighbors in a social network.

Definition 2.3. Fix a group $\mathcal{A}$ of agents, a set $\mathcal{R}$ of resources, and a directed graph $G = (\mathcal{A}, E)$ (i.e., the agents are the vertices of G). We call allocation π *(weakly) graph-envy-free* if for each pair of (distinct) agents $a_1, a_2 \in \mathcal{A}$ such that $a_2 \in N(a_1)$ it holds that $u_1(\pi(a_1)) \geq u_1(\pi(a_2))$. By replacing the weak inequality in our criterion with a strict inequality we obtain the definition of a *strongly graph-envy-free* allocation.

An allocation which gives nothing to every agent is always (weakly) graph-envy-free; to overcome this trivial case we combine our fairness concepts with different measures of allocation efficiency.

Definition 2.4. Consider an allocation π of a set $\mathcal{R}$ of resources to a set $\mathcal{A}$ of agents and a family $U = \{u_1, u_2, \ldots, u_{|\mathcal{A}|}\}$ of utility functions where some function u_i represents preferences of agent a_i. We call π *complete* if $\bigcup_{a \in \mathcal{A}} \pi(a) = \mathcal{R}$. We call π *Pareto-efficient* if there exists no allocation π' that *dominates* π, where dominating means that for all $a_i \in \mathcal{A}$ it holds that $u_i(\pi(a_i)) \leq u_i(\pi'(a_i))$ and for some $a_j \in \mathcal{A}$ it holds that $u_j(\pi(a_j)) < u_j(\pi'(a_j))$. We call $\sum_{a_i \in \mathcal{A}} u_i(\pi(a_i))$ the *utilitarian social welfare* $\mathcal{W}_\pi$ of allocation π.

A problem parameterized by ρ is *fixed-parameter tractable* if it is solvable in $f(\rho) \cdot |I|^{O(1)}$ time for some computable function f and the input size $|I|$ according to the problem's encoding; $W[t]$-hard, $t \geq 1$, problems are presumably not fixed-parameter tractable. We call a problem para-NP-hard if it is NP-hard even for a constant value of the parameter.

Throughout the paper we make heavy use of the graph problem Clique to show our results regarding computational hardness.

Definition 2.5. In the Clique problem, given an undirected graph and an integer k, we ask whether there is a clique of size k, i.e., a size-k subset of the vertices such that they are pairwise adjacent.

Clique is a well-known NP-complete [14] problem which is W[1]-complete [9] when parameterized by the size of the clique.

Problem Description. For fair allocation applications, it is important not only to know that there exists an allocation with particular features, but also to know how it looks like. This is why we define our problems in the form of search problems, instead of decision problems. Obviously, our problems also have natural decision variants.

Subsequently, X-(s)GEF-Allocation stands for X-(strongly) graph-envy-free allocation where $X \in \{\text{C}, \text{E}, \text{W}\}$, 'C' referring to complete, 'E' referring to Pareto-efficient, and 'W' referring to utilitarian social welfare. We start with defining our problems with respect to completeness and Pareto-efficiency.

C-GEF-Allocation (resp. C-sGEF-Allocation)
Input: A set $\mathcal{A}$ of n agents, a set $\mathcal{R}$ of m indivisible resources, a family $U = \{u_1, u_2, \ldots, u_n\}$ of agents' utility functions, and a directed graph $G = (\mathcal{A}, E)$.
Task: Find a complete, graph-envy-free (resp. strongly graph-envy-free) allocation of $\mathcal{R}$ to $\mathcal{A}$.

Analogously, we define the E-GEF-Allocation and E-sGEF-Allocation problems where we search for a Pareto-efficient and (weakly/strongly) graph-envy-free allocation. In the case of utilitarian social welfare we slightly change the task when defining the respective W-GEF-Allocation, W-sGEF-Allocation problems: We search for a (weakly/strongly) graph-envy-free allocation which maximizes the utilitarian social welfare.

Basic Observations. We start with two observations. Our first observation basically says that graph-envy-freeness can be checked in polynomial time. It is enough that for each agent one compares its own bundle value to values it assigns to the neighbors' bundles.

Observation 1 (graph-envy-freeness Test). *Given a set $\mathcal{R}$ of resources, a set $\mathcal{A}$ of agents with their preferences over bundles of $\mathcal{R}$, and some allocation $\pi\colon \mathcal{A} \to 2^{\mathcal{R}}$, one can decide in polynomial time whether π is (weakly/strongly) graph-envy-free.*

C-GEF-Allocation			
	DAG	SCC	General
identical 0/1	P (Obs. 3)	P (Cor. 1)	NP-h (Thm. 1)
identical	P (Obs. 3)	NP-h (Pr. 1, ♣)	NP-h (Pr. 1, ♣)
0/1	P (Obs. 3)	NP-h (Th. 2)	NP-h (Th. 2)
additive	P (Obs. 3)	NP-h (Pr. 1, ⋄)	NP-h (Pr. 1, ⋄)

C-sGEF-Allocation			
	DAG	SCC	General
identical 0/1	P (Pr. 3)	$O(1)$ (Obs. 5)	P (Pr. 3)
identical	NP-h (Pr. 2)	$O(1)$ (Obs. 5)	NP-h (Pr. 2)
0/1	NP-h (Th. 3)	NP-h (Th. 3)	NP-h (Th. 3)
additive	NP-h (Th. 3)	NP-h (Th. 3)	NP-h (Th. 3)

Table 2: Computational complexity results for C-(s)GEF-Allocation for different graph and preference restrictions. DAG and SCC stand for directed acyclic graphs and strongly connected graphs respectively. Entry "NP-h" refers to NP-hard cases, "P" refers to polynomial-time solvable cases, and "$O(1)$" refers to trivial (constant-time solvable) cases. Results marked with ♣ can, using Observation 6, also be derived from Bouveret and Lang [6]. Results marked with ⋄ also derive from Lipton et al. [16].

C-GEF-Allocation			
parameter	preferences	restrictions	complexity
outdegree	identical 0/1	outdegree= 2	p-NP-h (Th. 1)
#agents	identical	outdegree= 1	W[1]-h (Pr. 1)
#resources	0/1	str. connected	W[1]-h (Th. 2)

C-sGEF-Allocation			
parameter	preferences	restrictions	complexity
outdegree	additive	outdeg.= 1	p-NP-h (Th. 4)
#agents	additive	outdeg.= 1	W[1]-h (Th. 4)

Table 3: Parameterized complexity results for C-(s)GEF-Allocation. Entry "p-NP-h" denotes para-NP-hard cases, that is, cases that remain hard even for constant parameter values. Entry "W[1]-h" denotes W[1]-hard cases, that is, cases which are presumably not fixed-parameter tractable.

Due to Observation 1 every NP-hardness proof in Section 3 implies NP-completeness of the corresponding decision problem discussed in the proof.

Intuitively, considering (weakly/strongly) graph-envy-free allocations, we can rule out all resources which have no value for any agent. We state this claim as the following observation.

Observation 2. *Without loss of generality, there are only resources to which at least one agent assigns positive utility.*

Observation 2, albeit simple, results in a useful consequence for the case of identical 0/1 preferences: All variants of X-(s)GEF-Allocation boil down to distributing a certain number of indistinguishable resources.

3 FINDING COMPLETE ALLOCATIONS

We analyze the classic complexity (Table 2) and the parameterized complexity (Table 3) for finding allocations that are complete and (weakly/strongly) graph-envy-free. In Section 3.1, we discuss our results for the weak version of envy-freeness and in Section 3.2 we discuss our results for the strong version of envy-freeness. We identify cases where using our graph-based envy-freeness concept leads to decreased complexity (from NP-hard to P) and cases where it leads to increased complexity (from P to NP-hard), each time comparing to classic envy-freeness.

3.1 Weakly Graph-envy-free Allocations

As a warm-up, we consider the case where the graph encoding the envy-relation is acyclic and where the preferences are additive monotonic (being the least-restrictive preference type considered in this paper). This case is well-motivated because it describes hierarchical situations where only higher ranked agents may envy lower ranked agents.

For weak graph-envy-freeness there is a trivial solution that allocates all resources to a single source agent. Indeed, nobody can envy a source agent because it has no incoming arcs, the remaining agents do not envy each other because none has a resource, and the allocation is complete since all items are allocated.

Observation 3. *C-GEF-Allocation for monotonic additive preferences and an acyclic input graph is solvable in linear time.*

Next, we consider the most restrictive preference type, identical 0/1 preferences, together with the fairly large class of strongly connected graphs. Here, because of transitivity of the "greater or equal" relation, we obtain a very simple tractable case where all agents must obtain the same number of resources. To show this, we start with the following observation, directly yielding a simple algorithm.

Observation 4. *Let $\pi : \mathcal{A} \rightarrow 2^{\mathcal{R}}$ be a graph-envy-free allocation. Then, for every pair $\{a, a'\}$ of agents that belong to the same strongly connected component, it holds that (1) $u(\pi(a)) = u(\pi(a'))$ for identical preferences, and (2) $|\pi(a)| = |\pi(a')|$ for identical 0/1 preferences.*

Corollary 1. *C-GEF-Allocation for identical 0/1 preferences and an input graph being strongly connected is solvable in linear time.*

Proof. Using Observation 4, our algorithm checks whether the number of resources is divisible by the number of agents and returns true if and only if this is the case. □

Observation 4 (2) allows us to view agents from the same strongly connected component as "uniform block of agents". This view will be very helpful to obtain the following theorem which basically states that, even with identical 0/1 preferences, GEF-Allocation becomes intractable as soon as the graph is not strongly connected.

Theorem 1. *C-GEF-Allocation for identical* 0/1 *preferences is* NP*-hard even if each vertex has outdegree at most two.*

Proof. For the sake of readability we extend the concept of envy from agents to sets of agents. We say that a strongly connected component A' envies another strongly connected component A'' if there exists an agent from A' that envies an agent from A''. For identical 0/1 preferences, a solution to C-GEF-Allocation has to allocate exactly the same number of resources to every agent being a part of the same strongly connected component (Observation 4 (2)). Thus, we say that we are allocating some number of resources to a strongly connected component (instead of an agent) when we uniformly distribute these resources to the agents that belong to the component. As a consequence, we can allocate only multiples of t resources to a strongly connected component consisting of t agents.

To prove Theorem 1, consider a Clique instance formed by an undirected graph $\bar{G} = (\bar{V}, \bar{E})$ with a set $\bar{V} = \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{\bar{n}}\}$ of vertices and a set $\bar{E} = \{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_{\bar{m}}\}$ of edges, and a clique size k. Without loss of generality, assume that $1 < k < \bar{n}$ and $\bar{m} > \binom{k}{2}$.

We present a polynomial-time many-one reduction from Clique to C-GEF-Allocation. We introduce $\bar{n}^2\bar{m}(\bar{n}^2 + 1) + \bar{m}$ agents and $\bar{n}^4\bar{m} + k\bar{n}\bar{m} + \binom{k}{2}$ resources which are assigned utility one by each agent. We specify an input graph G over the agents in two steps. First, we define strongly connected components of G separately and then add arcs connecting them. By connecting two strongly connected components we mean adding an arc between two arbitrarily chosen vertices, one from each connected component. In a first step, we build the following strongly connected components:

(1) For each vertex $\bar{v} \in \bar{V}$, we introduce a *vertex component* $G_{\bar{v}}$ which consists of $\bar{n} \cdot \bar{m}$ vertices;
(2) For each edge $\bar{e} \in \bar{E}$, we introduce an *edge component* $G_{\bar{e}}$ which consists of one vertex;
(3) We introduce a *root component* G^* which consists of $\bar{n}^4 \cdot \bar{m}$ vertices.

Then, we connect the strongly connected components to form an input graph G of the C-GEF-Allocation instance. Figure 2 depicts graph G resulting from the following steps:

(1) For each edge $\bar{e} = \{\bar{v}', \bar{v}''\} \in \bar{E}$, we connect $G_{\bar{v}'}$ and $G_{\bar{v}''}$ to edge component $G_{\bar{e}}$ (with an arc pointing to $G_{\bar{e}}$);
(2) we connect the root component with every vertex component (with an arc starting at the root component).

To prove the correctness of the reduction, we have to show that there is a k-clique in $\bar{G}$ if and only if there is a complete and graph-envy-free allocation for the constructed C-GEF-Allocation instance. Assume that there is a k-clique $C = (V_C, E_C)$ in graph $\bar{G}$. We create a complete and graph-envy-free allocation as follows:

- We give $\bar{n}^4 \cdot \bar{m}$ resources to G^*;
- we give $\bar{n} \cdot \bar{m}$ resources to every vertex component associated with a vertex from V_C; and
- we give one resource to every edge component associated with an edge from E_C.

The allocation is complete because we assign

$$\bar{n}^4\bar{m} + \bar{n}\bar{m}|V_C| + |E_C| = \bar{n}^4\bar{m} + k\bar{n}\bar{m} + \binom{k}{2}$$

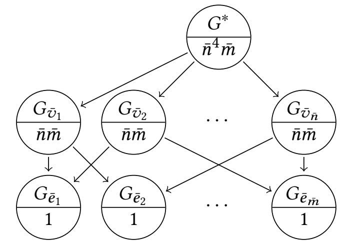

Figure 2: The input graph of a C-GEF-Allocation instance constructed in the proof of Theorem 1. The circles represent strongly connected components. Labels indicate a name (upper part) and the number of agents in the component (lower part). The connections represent arcs between two arbitrarily chosen agents from different components.

resources. Every edge component is a single vertex without outgoing arcs which means that, by definition, no edge component envies. Every vertex component $G_{\bar{v}}$, $\bar{v} \in \bar{G}$, might envy only edge vertices it is connected to. If $\bar{v} \in V_C$, then no vertex in $G_{\bar{v}}$ envies anybody, because every vertex in $G_{\bar{v}}$ has one resource and every vertex of every edge component has at most one resource. If $\bar{v} \notin V_C$, then $\bar{v}$ cannot envy because all edge components representing $\bar{v}$'s incident edges, which are not a part of clique C, have no resource allocated. Finally, the root component does not envy because each of its agents get one resource and no other agent gets more.

Conversely, assume that there exists a complete and graph-envy-free allocation for the constructed instance of C-GEF-Allocation. On the one hand, the root component has to get at least $\bar{n}^4\bar{m}$ resources because it consists of $\bar{n}^4\bar{m}$ agents. On the other hand, because of a lack of resources, the root component cannot get $2\bar{n}^4\bar{m}$ resources. This derives from the following inequality holding for $\bar{n} > 1$:

$$k\bar{n}\bar{m} + \binom{k}{2} \leq \bar{n}^2\bar{m} + \bar{n}^2 \leq 2\bar{n}^2\bar{m} < \bar{n}^4\bar{m}.$$

Thus, every agent in the root component gets one resource. Since every agent in the root component might envy all other agents (even all agents in the edge components due to transitivity of the "greater than or equal to" relation), every other agent can get at most one resource. Besides the root component's resources, there are still $k\bar{n}\bar{m} + \binom{k}{2}$ resources left. For every feasible solution there exist exactly k vertex components whose agents have a one-resource bundle. Because

$$(k + 1)\bar{n}\bar{m} > k\bar{n}\bar{m} + \binom{k}{2},$$

we cannot allocate resources to more than k vertex components. Contrarily, if one allocates $\bar{n}\bar{m}$ resources to $k-1$ vertex components, then there are still $\bar{n}\bar{m}+\binom{k}{2}$ resources left. However, we have only $\bar{m}$ edge components, each one capable of having at most one resource. Thus, a feasible allocation chooses exactly k vertex components and $\binom{k}{2}$ edge components. Moreover, every vertex component has to be connected to chosen edge components. This exactly corresponds to choosing k distinct vertices and $\binom{k}{2}$ distinct edges such that every edge is incident to two of the chosen vertices.

To see that the described construction can be realized with maximum outdegree at most two, observe that strongly connected components can be obtained through simple directed cycles (where each vertex has one outgoing arc). Each component has fewer "outgoing arcs" than vertices so that we can use an individual agent (creating a second outgoing arc) in each case.

The reduction is clearly executable in polynomial time. ◻

Another way to try to extend the tractability from Corollary 1 to a broader setting is to keep the graph being strongly connected but to consider identical monotonic additive preferences (so, allowing more than just values 1 or 0). However, Observation 4 (1) allows for a (quite straight-forward) reduction from the NP-hard and W[1]-hard EEF Existence [6]. As a result, C-GEF-Allocation for identical monotonic additive prefences inherits intractability even for the case with few agents and the graph G being a directed cycle (implying that the outdegree of each vertex is at most one, meaning that each agent envies at most one other agent).

Proposition 1. *C-GEF-Allocation for identical monotonic additive preferences is* NP*-hard and* W[1]*-hard when parameterized by the number of agents even if the input graph is a cycle.*

We finally consider C-GEF-Allocation for the case of few resources. With the classic envy-freeness notion (or G being complete for C-GEF-Allocation), the problem of finding a complete, envy-free allocation can easily be seen to be fixed-parameter tractable (using an analogous technique as used by Bliem et al. [3, Proposition 1]). For graph-envy-freeness, however, it turns out that the problem becomes W[1]-hard even for 0/1 preferences and G being strongly connected. This provides an example where the complexity for C-GEF-Allocation differs between the cases of complete directed graphs and general strongly connected graphs.

Theorem 2. *C-GEF-Allocation for* 0/1 *preferences is* NP*-hard and* W[1]*-hard when parameterized by the number of resources even if the input graph is strongly connected.*

Proof. Consider an instance of Clique with graph $\bar{G} = (\bar{V}, \bar{E})$ and clique size k. Let $m := k + \binom{k}{2} + 1$ be the number of resources in the new instance of C-GEF-Allocation. Specifically, we have a special resource r_*, k vertex resources $\mathcal{R}_\text{v} = \{r_1, r_2, \dots, r_k\}$, and $\binom{k}{2}$ edge resources $\mathcal{R}_\text{e} = \{r'_1, r'_2, \dots, r'_{\binom{k}{2}}\}$. We have a special agent a_*, dummy agents $\mathcal{A}_\text{c} = \{a_1, a_2, \dots, a_{m+1}\}$, a starting agent s, and an ending agent t. Additionally, we add vertex agents $\mathcal{A}_\text{v} = \{v_1, v_2, \dots, v_{\bar{n}}\}$ and edge agents $\mathcal{A}_\text{e} = \{e_1, e_2, \dots, e_{\bar{m}}\}$ that correspond to vertices and edges of graph $\bar{G}$. The construction of the input graph for the new instance of C-GEF-Allocation, illustrated in Figure 3, is as follows:

(1) Create a cycle over all agents from $\mathcal{A}_\text{c} \cup \{s, t\} \cup \{a_*\}$ such that every two adjacent agents are connected with bidirectional arcs. The order of the agents is arbitrary except that a_* is adjacent to agents s and t.
(2) For every vertex or edge agent a create two arcs (a, a_*) and (a_*, a).
(3) Connect agent s with every vertex agent by an arc pointing to a vertex agent.
(4) Connect agent t with every edge agent by an arc pointing to an edge agent.

res. \ agents	s	$\mathcal{A}_\text{c}$	t	a_*	$\mathcal{A}_\text{v}$	$\mathcal{A}_\text{e}$
r_*	0	1	0	1	0	0
$\mathcal{R}_\text{v}$	0	1	1	1	1	0
$\mathcal{R}_\text{e}$	1	1	0	1	1	1

Table 4: Utilities of the resources reported by the agents in the reduction in the proof of Theorem 2.

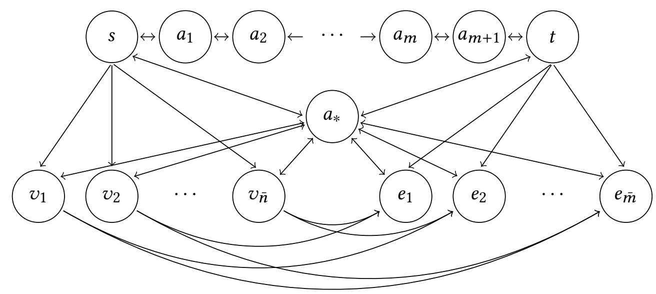

Figure 3: The general graph constructed in the reduction in the proof of Theorem 2. Every node is labeled with its name.

(5) Encode the structure of the input graph $\bar{G}$ by connecting a vertex agent to an edge agent by an arc starting at the vertex agent whenever a corresponding vertex (in $\bar{G}$) is incident to the corresponding edge (in $\bar{G}$).

Finally, Table 4 depicts the utility values given by the agents to the resources.

If there exists a solution to the Clique instance, then we can obtain a complete, graph-envy-free allocation for the C-GEF-Allocation instance by assigning the vertex and edge resources to agents representing, respectively, the vertices and the edges of the clique. We give the special resource to the special agent. Indeed, one can check using Table 4 that such an allocation is always graph-envy-free. The special resource, r_*, given to agent a_* makes neither s nor t envious. Even though s can envy vertex agents, they get only vertex resources to which s assigns the value of zero. As a result, s is unenvious. By symmetry, the same holds for t. Since both s and t get no resource, none of the dummy agents envies. Because every clique's edge connects only the clique's vertices, every edge agent that gets a resource may be envied only by the vertex agent that also got a resource. Consequently, there is no envying vertex agent. With observing that the edge agents do not envy special agent a_* because they give zero utility to the special item, we conclude that our allocation is complete and graph-envy-free.

Proving that a solution to C-GEF-Allocation yields a solution to Clique is more involved. Observe that every dummy agent reports utility of one for every resource and all edges between agents $\mathcal{A}_\text{c} \cup \{s, t\}$ are bidirectional. Thus, if at least one agent from $\mathcal{A}_\text{c} \cup \{s, t\}$ was assigned a resource, then all dummy agents would have to get one, too. This is impossible because there are $m + 1$ dummy agents and only m resources. Hence, no complete and graph-envy-free allocation assigns a resource to any agent from the set $\mathcal{A}_\text{c} \cup \{s, t\}$.

Now, consider resource r_*. Assigning r_* to one of the either vertex or edge agents makes a_* envious. Consequently, a_* has to be assigned one of the remaining resources. However, every possible choice from the remaining resources makes either s or t envious. Since we have proven that we are not allowed to give a resource to any of s and t, one has no choice but to assign r_* to a_*. Indeed, since no agent is envious after such an allocation, if there exists a complete and graph-envy-free allocation, then r_* is assigned to a_*.

Next, we show that every vertex resource can be given only to a vertex candidate and that every vertex agent gets at most one vertex resource. To justify the first part of the claim, let us assume that some vertex resource v is given to either a_* or some edge agent. This immediately implies that t has to get one of the resources which, according to our very first observation, make finding a solution impossible if given to t. Conversely, one can safely assign v to one of the vertex agents. Towards showing that every vertex agent gets at most one vertex resource, let us assume that some vertex candidate is assigned two vertex resources. By Table 4, we see that a_* is envious now (even when a_* has been assigned r_*). However, giving a_* any resource except for r_* ends up in the situation where either s or t has to be assigned a resource which is forbidden. By symmetry arguments, we can use a similar deduction to observe that every edge resource can be given only to an edge agent and that every edge agent gets at most one edge resource.

Altogether, the observations stated above show that a solution for the C-GEF-Allocation instance needs to allocate exactly k vertex resources to exactly k vertex agents and exactly $\binom{k}{2}$ edge resources to exactly $\binom{k}{2}$ edge agents. By our construction, every time an edge resource is assigned to some edge agent e, a vertex resource has to be assigned to every vertex agent connected with e. Since a vertex resource is connected to an edge resource if and only if the vertex is incident to the edge, the vertex and edge agents with allocated resources represent a k-clique.

The reduction works in polynomial time which implies that C-GEF-Allocation is NP-hard. Additionally, the reduction uses a number of resources upper-bounded by a (polynomial) function of k, which implies W[1]-hardness. ◻

3.2 Strongly Graph-envy-free Allocations

We move on to the strong variant of our envy-freeness concept and analyze how this stronger notion effects computational complexity. Again, we start with directed acyclic graphs to model hierarchical structures. Here, strong graph-envy-freeness seems to be a very reasonable assumption. In contrast to C-GEF-Allocation, which is trivial to solve in this setting, it turns out that C-sGEF-Allocation is intractable even for identical preferences. Reducing from the NP-hard Unary Bin Packing [13], we mainly use the fact that in a (directed) length-k path of agents, the first agent has to get a bundle with utility at least $k-1$.

Proposition 2. *C-sGEF-Allocation with identical monotonic additive preferences is* NP*-hard even if the input graph is acyclic.*

Using a reduction from Clique, we can show that for acyclic graphs C-sGEF-Allocation remains hard in case of 0/1 preferences. The proof is based on the observation that if we have a group B of agents connected to some agent $a \notin B$ which has to get some resource r, then, depending on whether the agents in B like resource r or not, we can distinguish two cases. The first case is that every agent in B gets at least two resources. The second case is that for every agent in B it is enough to get one resource.

Theorem 3. *C-sGEF-Allocation with* 0/1 *preferences is* NP*-hard for the input graph being either a directed acyclic graph or a strongly connected component.*

By Proposition 2 and Theorem 3, identical 0/1 preferences are the last hope to identify a tractable case for acyclic graphs. Indeed, for this preference type we develop a polynomial-time algorithm which even works for all directed graphs. As a first step, we observe that, in contrast to C-GEF-Allocation, which was NP-hard, C-sGEF-Allocation is trivial for monotonic additive preferences and strongly connected graphs (including cliques and, hence, the "standard but strong" envy-freeness concept). The intuitive idea is that any directed cycle implies a cycle for the transitive "greater than" relation when comparing the utility values of the agents' resources along this cycle.

Observation 5. *Let G be a graph that contains a strongly connected component with more than one vertex. Then, there is no strongly graph-envy-free allocation if the agents have identical preferences.*

Next, we present Algorithm 1 which, applying Observation 5, finds a complete, graph-envy-free allocation for the case of identical 0/1 preferences and arbitrary input graphs.

Algorithm 1: Let $\mathcal{R}$ be a set of resources, let $\mathcal{A}$ be a set of agents such that every agent assigns the preference value of one to every resource, and let $G = (\mathcal{A}, E)$ be a directed graph.

if $|\mathcal{A}| = 1$ **then**
  Allocate all resources to the single vertex; **return**;
if *There exists a cycle in G* **then**
  No allocation is possible; **return**;
Build a graph $G' = (\mathcal{A} \cup \{v_s\}, E')$ where
$E' = \{(u,v) \colon (v,u) \in E\} \cup \{(v_s, u) \colon u \in \mathcal{A} \wedge |N_G(u)| = 0\}$;
Assign every vertex $w \in V$ a label $\ell(w)$ being the length of the longest path from v_s decreased by one;
if $|\mathcal{R}| \geq \sum_{w \in W} \ell(w)$ **then**
  Assign $\ell(w)$ arbitrary resources from $\mathcal{R}$ to every agent $w \in V$;
  Assign the remaining resources to arbitrary agents with zero in-degree in graph G; **return**;
No allocation is possible; **return**;

Proposition 3. *C-sGEF-Allocation for identical* 0/1 *preferences can be solved in linear time.*

Proof. Algorithm 1 solves the problem. Because of space constraints we omit the proof of correctness and give the running time analysis only. Using breadth-first search, we can assign the labels to the agents and check in linear time whether a graph has a cycle. Since the same holds for our procedure of building auxiliary graph G', Algorithm 1 works in linear time. ◻

Proposition 3 complements our analysis of the classic computational complexity landscape of sGEF-Allocation for the considered restrictions on graphs and preferences. However, we strengthen our intractability result for the case of general monotonic additive preferences stated in Theorem 3. In Theorem 4 we show that C-sGEF-Allocation remains intractable even in case of few agents, the input graph being acyclic, and every agent having outdegree at most one.

Theorem 4. *C-sGEF-Allocation for monotonic additive preferences is* NP*-hard and* W[1]*-hard when parameterized by the number of agents even if the input graph G is a directed path.*

4 EFFICIENCY AND SOCIAL WELFARE

In this section, we briefly discuss to which extent our results from Section 3 transfer to settings where one searches for graph-envy-free allocations that are not necessarily complete, but that are Pareto-efficient or that optimize the utilitarian social welfare. The following observation, the proof of which is based on proving relations between its parts sequentially, shows that many hardness results provided in Section 3 directly transfer.

Observation 6. *Let $\mathcal{R}$ be a set of resources and $\mathcal{A}$ be a set of agents with identical additive monotonic preferences. Then, the following three statements are equivalent:*

(1) there is a complete and (weakly/strongly) graph-envy-free allocation,
(2) there is a Pareto-efficient and (weakly/strongly) graph-envy-free allocation, and
(3) there is a (weakly/strongly) graph-envy-free allocation $\pi : \mathcal{A} \to 2^{\mathcal{R}}$ with $\mathcal{W}_\pi = w$,

where $w = \sum_{r \in \mathcal{R}} \max_{a \in \mathcal{A}} u_a(r)$.
For $0/1$ preferences, it holds that (3) $\leftrightarrow$ (2) $\to$ (1).

Theorem 2, Theorem 3, and Theorem 4 are not fully covered by the above observation. However, since in the respective reductions every resource must be allocated to one of the agents that "values it the most" in every graph-envy-free and complete allocation, the proofs indeed can be extended to also work for Pareto-efficiency and utilitarian social welfare.

As for tractability, to adapt our results from Section 3 is a bit more complicated than adapting them for NP-hardness. Of course, for Pareto-efficiency and identical additive monotonic preferences one can use the algorithm from Observation 3 as direct consequence of Observation 6 ((1) $\leftrightarrow$ (2)).

Surprisingly, it turns our that, while for E-GEF-Allocation all polynomial-time cases still hold but require a slightly more involved algorithm, W-GEF-Allocation becomes intractable for directed acyclic graphs and additive monotonic preferences. To prove this, we reduce from Clique and only need the utility values 0, 1, and 2.

Proposition 4. *W-GEF-Allocation is* NP*-hard for the input graph being a directed acyclic graph even for three-valued utility functions.*

Finally, we describe an algorithm that shows that polynomial-time solvability of the remaining cases of C-GEF-Allocation holds also for the same special cases of the W-GEF-Allocation and E-GEF-Allocation problems.

Algorithm 2: Let $\mathcal{R}$ be a set of resources, let $\mathcal{A}$ be a set of agents with preferences encoded by the utility functions $u_a : \mathcal{R} \to \mathbb{N}, a \in \mathcal{A}$, and let G be a DAG. The sets $\mathcal{R}$ and $\mathcal{A}$ are ordered (arbitrarily).

```
while R ≠ ∅ do
    Remove all agents a with u_a(r) = 0, ∀r ∈ R;
    Allocate the first resource r* to the first agent a* with zero
      in-degree which values r* the most among the agents
      with zero in-degree;
    Remove r* from R;
```

Proposition 5. *Algorithm 2 runs in polynomial time and solves*

- *E-GEF-Allocation for acyclic input graphs and monotonic additive preferences, and*
- *W-GEF-Allocation for acyclic input graphs and $0/1$ preferences.*

Note that de Keijzer et al. [15] have shown that finding a Pareto-efficient and envy-free allocation is not only NP-hard but Σ_2^{P}-hard even for monotonic additive preferences. So, Proposition 5 decreases the complexity of E-GEF-Allocation from Σ_2^{P} for general directed graphs to polynomial-time solvability for DAGs.

5 CONCLUSION

Combining social networks with fairness in the context of resource allocations is a promising line of (future) research. Our work significantly differs from the one of Chevaleyre et al. [8] which, among many other things, has a more distributed and (because of considering monetary payments) more divisible-resources flavor. The majority of our results are computational hardness results. In a sense, they lay the foundations for a more refined search for islands of tractability concerning practically motivated use cases of our basic models. To this end, there are plenty of opportunities. First, one may study further natural parameters, including the number of resources, maximum utility values, or structural graph parameters such as treewidth. Note, however, that these parameters may need to be combined in order to achieve fixed-parameter tractability results (e.g., a small maximum utility value does not guarantee fixed-parameter tractability). Second, it appears natural to deepen our studies by considering various special graph classes for the underlying social network. In addition, one may move from directed to undirected graphs or one may consider graphs that only consist of small connected components. Again note, however, that the class of bounded-degree graphs (reflected by a parameterization using maximum degree as the parameter) alone, as shown in this work, may not be enough to achieve (fixed-parameter) tractability. Finally, including further fairness concepts beyond the ones we studied appears to be promising as well.

ACKNOWLEDGMENTS

We are grateful to anonymous reviewers for their feedback helping us to improve our presentation. RB was from September 2016 to September 2017 on postdoctoral leave at the University of Oxford (GB), supported by the DFG fellowship BR 5207/2. AK was supported by the DFG project AFFA (BR 5207/1 and NI 369/15).

REFERENCES

[1] Rediet Abebe, Jon Kleinberg, and David C. Parkes. 2017. Fair Division via Social Comparison. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS'17)*. 281–289.

[2] Xiaohui Bei, Youming Qiao, and Shengyu Zhang. 2017. Networked Fairness in Cake Cutting. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 3632–3638.

[3] Bernhard Bliem, Robert Bredereck, and Rolf Niedermeier. 2016. Complexity of Efficient and Envy-Free Resource Allocation: Few Agents, Resources, or Utility Levels. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. 102–108.

[4] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. 2017. Fair Division of a Graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 135–141.

[5] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. 2016. Fair Allocation of Indivisible Goods. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 13, 311–329.

[6] Sylvain Bouveret and Jérôme Lang. 2008. Efficiency and Envy-freeness in Fair Division of Indivisible Goods: Logical Representation and Complexity. *Journal of Artificial Intelligence Research* 32 (2008), 525–564.

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

[8] Yann Chevaleyre, Ulle Endriss, and Nicolas Maudet. 2017. Distributed fair allocation of indivisible goods. *Artificial Intelligence* 242 (2017), 1–22.

[9] Rod G. Downey and Michael R. Fellows. 1995. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science* 141, 1 (1995), 109–131.

[10] Leon Festinger. 1954. A Theory of Social Comparison Processes. *Human Relations* 7, 2 (1954), 117–140.

[11] Laurent Gourvés, Julien Lesca, and Anaëlle Wilczynski. 2017. Object Allocation via Swaps along a Social Network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 213–219.

[12] Maqsood Haider, Alamzeb Aamir, AA Hamid, and Muhammad Hashim. 2015. A literature analysis on the importance of non-financial rewards for employees' job satisfaction. *Abasyn University Journal of Social Sciences* 8 (2015), 341–354.

[13] Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. 2013. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences* 79, 1 (2013), 39–49.

[14] Richard M. Karp. 1972. Reducibility among Combinatorial Problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*. Springer US, 85–103.

[15] Bart de Keijzer, Sylvain Bouveret, Tomas Klos, and Yingqian Zhang. 2009. On the Complexity of Efficiency and Envy-Freeness in Fair Division of Indivisible Goods with Additive Preferences. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT'09)*. 98–110.

[16] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC'04)*. 125–131.

[17] Evangelos Markakis. 2017. Approximation Algorithms and Hardness Results for Fair Division with Indivisible Goods. In *Trends in Computational Social Choice*, Ulle Endriss (Ed.). AI Access, Chapter 12, 231–247.

[18] Hervé Moulin. 2003. *Fair Division and Collective Welfare.* MIT Press.

[19] Bureau of Labor Statistics, U.S. Department of Labor. 2017. Employer Costs for Employee Compensation—June 2017. (2017). https://www.bls.gov/news.release/pdf/ecec.pdf [Online; accessed 11-November-2017].

[20] Ariel D. Procaccia. 2013. Cake Cutting: Not Just Child's Play. *Communications of the ACM* 56, 7 (2013), 78–87.

[21] Ariel D. Procaccia. 2016. Cake Cutting Algorithms. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 13, 311–329.

[22] Jack M. Robertson and William A. Webb. 1998. *Cake-Cutting Algorithms - Be Fair if You Can.* A K Peters.

[23] Hugo Steinhaus. 1948. The Problem of Fair Division. *Econometrica* 16 (1948), 101–104.

[24] Warut Suksompong. 2017. Fairly Allocating Contiguous Blocks of Indivisible Items. In *Proceedings of the 10th International Symposium on Algorithmic Game Theory (SAGT'17)*. 333–344.