

On Improving Resource Allocations by Sharing

Robert Bredereck,¹ Andrzej Kaczmarczyk,^{2,3} Junjie Luo,⁴ Rolf Niedermeier,² Florian Sachse²

¹ Humboldt-Universität zu Berlin, Berlin, Germany

² Algorithmics and Computational Complexity, TU Berlin, Berlin, Germany

³ AGH University, Kraków, Poland

⁴ Nanyang Technological University, Singapore

robert.bredereck@hu-berlin.de, andrzej.kaczmarczyk@agh.edu.pl, junjie.luo@ntu.edu.sg, rolf.niedermeier@tu-berlin.de, sachse.florian@gmail.com

Abstract

Given an initial resource allocation, where some agents may envy others or where a different distribution of resources might lead to higher social welfare, our goal is to improve the allocation *without* reassigning resources. We consider a sharing concept allowing resources being shared with social network neighbors of the resource owners. To this end, we introduce a formal model that allows a central authority to compute an optimal sharing between neighbors based on an initial allocation. Advocating this point of view, we focus on the most basic scenario where a resource may be shared by two neighbors in a social network and each agent can participate in a bounded number of sharings. We present algorithms for optimizing utilitarian and egalitarian social welfare of allocations and for reducing the number of envious agents. In particular, we examine the computational complexity with respect to several natural parameters. Furthermore, we study cases with restricted social network structures and, among others, devise polynomial-time algorithms in path- and tree-like (hierarchical) social networks.

1 Introduction

The fair allocation of resources undoubtedly is a key challenge for modern societies and economies. Applications can be found in such diverse fields as cloud computing, food banks, or managing carbon loads in the context of global warming. Naturally, this topic received high attention in the scientific literature. This also holds true for the special case of indivisible resources (Bouveret, Chevaleyre, and Maudet 2016), which we concentrate on here. Moreover, we take into account the role of social networks built by agents, a growing line of research (Abebe, Kleinberg, and Parkes 2017; Bei, Qiao, and Zhang 2017; Bouveret et al. 2017; Bredereck, Kaczmarczyk, and Niedermeier 2018; Chevaleyre, Endriss, and Maudet 2017; Beynier et al. 2019; Lange and Rothe 2019; Huang and Xiao 2019). We bring one further new aspect into this scenario, reflecting the increasing relevance of “sharing economies” (Belk, Eckhardt, and Bardhi 2019; Schor and Cansoy 2019), where agents share resources in a peer-to-peer fashion. Resources to share may be almost everything, for instance, knowledge, machines, time, or natural resources. More specifically, sharing in our

scenario, which takes into account the relationships between agents expressed by social networks, means that *two* adjacent agents in the social network may share the very same resource, thus increasing the utility of the resource allocation for at least one of them (assuming positive utility for each resource). We assume this to be organized and decided by a central authority like, for example, the boss of a company. To get started with this new setting, we focus on a very basic scenario. That is, in our model only two neighbors may share *and*, reflecting the (very human) principle of protection of acquired possession, no agent shall loose its already allocated resources. This conservative principle naturally makes sharing easier to implement, keeping “restructuring costs” lower, and it may even help to “keep peace” among agents. Moreover, it sometimes comes very naturally as depicted in the subsequent knowledge sharing example. Besides improving egalitarian or utilitarian welfare, we focus on the perhaps most basic fairness criterion, envy-freeness. Since it is not always possible that complete envy-freeness is achieved (consider one indivisible resource and two agents desiring it), we aim at, given an initial resource allocation, improving it by decreasing the number of envious agents through resource sharing. Moreover, we allow for modeling relationship aspects of sharing based on the social network formed by the agents.

Before becoming more specific about our model, let us first introduce the following example related to knowledge sharing. Assume that agents are employees of a company, each having a bundle of qualifications. An agent may “envy” another agent because the other agent has some special qualification. The central authority wants to improve the situation by building teams of two agents where, due to a daily extensive cooperation, one teaches the other the missing qualification (for instance, a realization of this is the concept of *pair programming* that also has other benefits besides knowledge sharing (Williams et al. 2000)).

Model of sharing allocation. Roughly speaking, our model is as follows (see Section 2 for formal definitions). The input is a set of agents and a set of indivisible resources initially assigned to the agents. Typically, every agent may be assigned several resources. Each agent has an individual utility value for each resource. The general goals are to decrease the overall degree of envy, to increase the sum of

“utility scores” of all agents, or to increase the minimal “utility score” among all agents. Importantly, the only way an agent can improve its individual “utility score” is by participating in a sharing with other agents.

We assume that if an owner shares, then this does not decrease its own overall utility value. This approach is justified when the burden of sharing is neutralized by its advantages. Indeed, in our knowledge sharing example a hassle of co-operation is often compensated by a better working experience or higher quality outcomes (as shown by Williams et al. (2000)). Note that such complicated mutual dependencies that would be extremely hard to describe formally form a natural field for our approach. Further application examples include irregularly used resources (like printers or compute servers). Here, the coordination with another person is uncritical and splitting the maintenance costs neutralizes the inconvenience of cooperation.

We enrich our model by using two graphs, an undirected sharing graph and a directed attention graph, to model social relations between agents and to govern the following two constraints of our model. The sharing graph models the possibility for two agents to share resources because, e.g., they are close to each other or there is no conflict between the time they use resources. We focus on the case when only neighbors in the sharing graph can share a resource (a missing qualification in our knowledge sharing example). With respect to lowering the degree of envy, we assume that agents may only envy their outneighbors in the directed attention graph. This graph-based envy concept has recently been studied by many works in fair allocation (Bredereck, Kaczmarczyk, and Niedermeier 2018; Aziz et al. 2018; Beynier et al. 2019).

Agents may naturally be conservative in the sense of keeping control and not sharing too much. Furthermore, as in our initial example, it might simply be too ineffective to share a qualification among more than two employees simultaneously (due to, e.g., increased communication overhead or additional resources needed). We address this in the most basic way and assume that each resource can be shared to *at most one neighbor* of its owner and an agent can participate in a bounded number of sharings. This strong restriction already leads to tricky algorithmic challenges and fundamental insights. In particular, the model also naturally extends on well-known matching scenarios in a non-trivial way.

There are numerous options to further extend and generalize our basic model, as discussed in Section 5 and in the concluding Section 6. However, keeping our primary model simple, we aim at spotting its fundamental properties influencing the complexity of related computational problems.

Related work. To the best of our knowledge, so far the model we consider has not been studied. Since obtaining envy-free allocations is not always possible, there has been work on relaxing the concept of envy. In particular, in the literature *bounded-maximum envy* (Lipton et al. 2004), *envy-freeness up to one good* (Budish 2011), *envy-freeness up to the least-valued good* (Caragiannis et al. 2019), *epistemic envy-freeness* (Aziz et al. 2018), and *maximin share guarantee* (Budish 2011) have been studied. However, these

concepts combat nonexistence of allocations that are envy-free by considering approximate versions of it; they basically do not try to tackle the question of how to achieve less “envy” in an allocation. By way of contrast, our approach tries to find a way to lessen envy not by relaxing the concept of envy, but rather by enabling a small deviation in the model of indivisible, non-shareable resources. To this end, we make resources shareable (in our basic model by two agents). This approach is in line with a series of recent works which try to reduce envy (i) by introducing small amounts of money (Brustle et al. 2020; Halpern and Shah 2019; Caragiannis and Ioannidis 2020), (ii) by donating a small set of resources to charity (Caragiannis, Gravin, and Huang 2019; Chaudhury et al. 2021), or (iii) by allowing dividing a small number of indivisible resources (Sandomirski and Segal-Halevi 2019; Segal-Halevi 2019). In particular, the papers mentioned in point (iii) consider a model of indivisible resources that could be shared by an arbitrary group of agents and where, unlike in our study, each agent only gets a portion of the utility of the shared resources. Contrary to our setting, this model assumes no initial allocation. As a result, an envy-free allocation always exists and the goal is to seek one with a minimum number of shared resources. In contrast, our goal is to improve an *initial* allocation through sharing resources between pairs of agents. Another line of research considers the improvement of allocations by exchanging items (Chevaleyre, Endriss, and Maudet 2007; Gourvès, Lesca, and Wilczynski 2017; Huang and Xiao 2019). There has been quite some work on bringing together resource allocation and social networks (Abebe, Kleinberg, and Parkes 2017; Bei, Qiao, and Zhang 2017; Bouveret et al. 2017; Bredereck, Kaczmarczyk, and Niedermeier 2018; Chevaleyre, Endriss, and Maudet 2017; Beynier et al. 2019; Huang and Xiao 2019). In particular, the concept of only local envy relations to neighbors in a graph gained quite some attention (Aziz et al. 2018; Beynier et al. 2019; Bredereck, Kaczmarczyk, and Niedermeier 2018; Eiben et al. 2020). Modifying existing allocations to maintain fairness over time has also been studied in online settings with changing agents (Friedman, Psomas, and Vardi 2015, 2017) or resources arriving over time (He et al. 2019).

Our contributions. Introducing a new model for (indivisible) resource allocation with agents linked by social networks, we provide a view on improving existing allocations for several measures without, conceivably impossible, reallocations.

We analyze the (parameterized) computational complexity of applying our model to improve utilitarian social welfare or egalitarian social welfare (Definition 4), and to decrease the number of envious agents (Definition 5). We show that a central authority can (mostly) find a sharing that improves social welfare (measured in both the egalitarian and utilitarian ways) in polynomial time, while decreasing the number of envious agents is NP-hard even if the sharing graph is a clique and the attention graph is a bidirectional clique. To overcome NP-hardness, we also study the influence of different natural parameters (such as agent utility

ENVY-REDUCING SHARING ALLOCATION (ERSA)							
$\mathcal{G}_s = \mathcal{G}_t$		same utility		few agents		few resources	
clique	tree- or pathwidth	$\mathcal{G}_t = \text{clique}$	$\mathcal{G}_s = \text{clique}$	n	$k = 0, \Delta k = 1$	m	#shared resources
NP-h Thm. 3	XP, W[1]-h Thm. 6	P Thm. 7	NP-h Thm. 8	FPT Thm. 4	p-NP-h Thm. 5	XP, W[1]-h Obs. 3, Thm. 8	p-NP-h Thm. 8

Table 1: Results overview for ERSA, where $\mathcal{G}_s = \mathcal{G}_t$ means that the sharing graph (\mathcal{G}_s) is the same as the underlying graph of the attention graph (\mathcal{G}_t), n is the number of agents, k is the number of envious agents after sharing, Δk is a drop in the number of envious agents, and m is the number of resources.

function values, structural parameters concerning the agent social networks, the number of agents, and the number of resources); Table 1 surveys our results in more detail. We show that the problem is polynomial-time solvable if the underlying undirected graph of the attention graph is the same as the sharing graph and has constant treewidth (close to a tree). We also identify an interesting contrast between the roles of the two graphs: When agents have the same utility function, the problem is solvable in polynomial time if the attention graph is a bidirectional clique, while the problem is NP-hard even if the sharing graph is a clique. Finally, we show that the problem is fixed-parameter tractable (FPT) for the parameter number of agents (giving hope for efficient solutions in case of a small number of agents) and polynomial-time solvable (in XP) for a constant number of resources. However, the problem is NP-hard even if the goal is to reduce the number of envious agents from one to zero.

Altogether, our main technical contributions are about exploring the potential to “overcome” the NP-hardness of decreasing the number of envious agents by exploiting several problem-specific parameters. Due to the lack of space, we refer the reader to the long version of the paper (Bredereck et al. 2021) for several proof details (marked with (\star)).

2 Preliminaries

For a set $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ of *agents* and a set $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ of *indivisible resources*, a (*simple*) *allocation* $\pi: \mathcal{A} \rightarrow 2^{\mathcal{R}}$ is a function assigning to each agent a collection of resources—a *bundle*—such that the assigned bundles are pairwise disjoint. An allocation is *complete* if every resource belongs to some bundle.

A *directed graph* G consists of a set V of *vertices* and a set $E \subseteq V \times V$ of *arcs* connecting the vertices; we do not allow self-loops (i.e., there are no arcs of form (v, v) for any vertex $v \in V$). A (simple) *undirected graph* $G = (V, E)$ consists of a set V of vertices and a set E of distinct size-2 subsets of vertices called *edges*. An *underlying undirected graph* of a directed graph G is the graph obtained by replacing all (directed) arcs with (undirected) edges. We say an undirected graph $G = (V, E)$ is a *clique* if $E = \binom{V}{2}$ and a directed graph is a *bidirectional clique* if $E = V \times V$. For some vertex $v \in V$, the set $I(v)$ of *incident arcs* (edges) is the set of all arcs (edges) with an endpoint in v .

Sharing Model. We fix an initial allocation π of resources in \mathcal{R} to agents in \mathcal{A} . A *sharing graph* is an undirected

graph $\mathcal{G}_s = (\mathcal{A}, \mathcal{E}_s)$ with vertices being the agents; it models possible sharings between the agents. The following definition of sharing says that two agents can only share resources held by one of them.

Definition 1. Function $\delta_\pi: \mathcal{E}_s \rightarrow 2^{\mathcal{R}}$ is a *sharing* for π if for every two agents a_i and a_j , with $\{a_i, a_j\} \in \mathcal{E}_s$, it holds that $\delta_\pi(\{a_i, a_j\}) \subseteq \pi(a_i) \cup \pi(a_j)$.

An initial allocation π and a corresponding sharing δ_π form a *sharing allocation*.

Definition 2. A sharing allocation induced by *allocation* π and *sharing* δ_π is a function $\Pi_\pi^{\delta_\pi}: \mathcal{A} \rightarrow 2^{\mathcal{R}}$ where $\Pi_\pi^{\delta_\pi}(a) := \pi(a) \cup \bigcup_{e \in I(a)} \delta_\pi(e)$.

Since the initial allocation π is fixed, for brevity, we use δ and Π^δ , omitting π whenever it is not ambiguous. For simplicity, for every agent $a \in \mathcal{A}$, we also refer to $\Pi^\delta(a)$ as a *bundle* of a .

Naturally, each allocation is also a sharing allocation with a trivial “empty sharing.” Observe a subtle difference in the intuitive meaning of a bundle of an agent between sharing allocations and (simple) allocations. For sharing allocations, a bundle of an agent represents the resources the agent has access to and can utilize, not only those that the agent possesses (as for simple allocations).

2-sharing. Definition 1 is very general and only requires that two agents share resources that one of them already has. In particular, it allows one agent to share the same resource with many other agents; and does not constrain the number of sharings an agent could participate in. In this paper, we assume that each resource can only be shared by *two agents* and each agent can participate in at most a bounded number of sharings. We express this requirement in Definition 3.

Definition 3. A 2-sharing δ is a sharing where, for any three agents a_i, a_j , and a_k , it holds that $\Pi^\delta(a_i) \cap \Pi^\delta(a_j) \cap \Pi^\delta(a_k) = \emptyset$. A b -bounded 2-sharing δ is a 2-sharing where, for each agent a , it holds that $|\bigcup_{e \in I(a)} \delta(e)| \leq b$. A simple 2-sharing δ is a 1-bounded 2-sharing, i.e., for each agent a , it holds that $|\bigcup_{e \in I(a)} \delta(e)| \leq 1$.

Herein, we count the number of sharings an agent participate in by the number of resources shared with other agents (either “shared to” other agents or received from other agents). Notably, in simple 2-sharing, each agent can either share or receive a single resource. Thus, every simple

2-sharing can be interpreted as matching in which each edge is labeled with a shared item.

Welfare and Fairness Measures. We assume agents having additive utility functions. For an agent a with *utility function* $u: \mathcal{R} \rightarrow \mathbb{N}_0$ and a bundle $R \subseteq \mathcal{R}$, let $u(R) := \sum_{r \in R} u(r)$ be the *value* of R as perceived by a . Let us fix a sharing allocation Π^δ of resources \mathcal{R} to agents $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ with corresponding utility functions u_1, u_2, \dots, u_n . We define the following social welfare measures:

$$\begin{aligned} \text{utilitarian: } \text{usw}(\Pi^\delta) &:= \sum_{i \in [n]} u_i(\Pi^\delta(a_i)) \text{ and} \\ \text{egalitarian: } \text{esw}(\pi) &:= \min_{i \in [n]} u_i(\Pi^\delta(a_i)). \end{aligned}$$

Notice that we assume each agent a_i gets the full utility for all resources in $\Pi^\delta(a_i)$. We will discuss a generalization of this assumption in Section 5.

A directed graph $\mathcal{G}_t = (\mathcal{A}, \mathcal{E}_t)$ with vertices being the agents is an *attention graph*; it models social relations between the agents. We say that an agent a_i *looks at* another agent a_j if $(a_i, a_j) \in \mathcal{E}_t$. An agent is *envious* on \mathcal{G}_t under Π^δ if it prefers a bundle of another agent it looks at over its own one; formally, a_i *envies* a_j if $u_i(\Pi^\delta(a_i)) < u_i(\Pi^\delta(a_j))$ and $(a_i, a_j) \in \mathcal{E}_t$. We denote the set of envious agents in Π^δ as $\text{Env}(\Pi^\delta)$. For a given (directed) attention graph \mathcal{G}_t over the agents, a sharing allocation is \mathcal{G}_t -*envy-free* if no agent envies its out-neighbors.

3 Improving Social Welfare by Sharing

In this section, we study the problem of improving utilitarian (and egalitarian) welfare through sharing, defined as follows.

Definition 4. Given an initial complete allocation π of resources \mathcal{R} to agents \mathcal{A} , a sharing graph \mathcal{G}_s , and a non-negative integer k , b-Bounded Utilitarian Welfare Sharing Allocation (*b-UWSA*) asks if there is a *b*-bounded 2-sharing δ such that $\text{usw}(\Pi^\delta) \geq k$; b-Bounded Egalitarian Welfare Sharing Allocation (*b-EWSA*) asks if there is a *b*-bounded 2-sharing δ such that $\text{esw}(\Pi^\delta) \geq k$.

We first consider *b-UWSA*. When $b = 1$, since every simple 2-sharing corresponds to a matching, we can easily reduce 1-UWSA to MAXIMUM WEIGHTED MATCHING. Thus, 1-UWSA is solvable in polynomial time. When $b > 1$, however, the problem is not just finding c matchings such that the total is maximized. Notice that in 2-sharing each resource can only be shared once. Nevertheless, we show that we can still reduce *b-UWSA* to MAXIMUM WEIGHTED MATCHING via a more involved reduction.

Theorem 1. *b-UWSA* is solvable in polynomial time for any $b \geq 1$.

Proof. Given an instance of *b-UWSA*, we construct an instance $(G = (V, E), w)$ of MAXIMUM WEIGHTED MATCHING as follows. For simplicity, assume each agent $a_i \in \mathcal{A}$ has at least b resources in the initial allocation π ; otherwise we can ensure this by adding enough resources that are valued as 0 by all agents. For each agent $a_i \in \mathcal{A}$ and each resource $r_j \in \mathcal{R}$, we add a vertex v_i^j into V . In addition, for each agent $a_i \in \mathcal{A}$, we add $n_i = |\pi(a_i)| - b$

dummy vertices $\{\bar{v}_i^k\}_{k=1,2,\dots,n_i}$ into V . For each pair of agents a_{i_1}, a_{i_2} , for each vertex $v_{i_1}^{j_1}$ corresponding to a_{i_1} and each vertex $v_{i_2}^{j_2}$ corresponding to a_{i_2} , we add an edge between $v_{i_1}^{j_1}$ and $v_{i_2}^{j_2}$ with weight $\max\{u_{i_1}(r_{j_2}), u_{i_2}(r_{j_1})\}$. Finally, for each v_i^j and each \bar{v}_i^k corresponding to the same agent a_i , we add a dummy edge between them with weight $W = \max_{a_i \in \mathcal{A}, r_j \in \mathcal{R}} u_i(r_j)$. This finishes the construction of the instance $(G = (V, E), w)$. Notice that there always exists a maximum weighted matching that contains n_i dummy edges for each agent a_i . Let $P = W \sum_{a_i \in \mathcal{A}} n_i$ be the weight of those edges. Next we show that there is a *b*-bounded 2-sharing δ such that $\text{usw}(\Pi^\delta) \geq k$ if and only if there is matching M in graph G with weight $\sum_{e \in M} w(e) \geq k - \text{usw}(\pi) + P$, where $\text{usw}(\pi) = \sum_{a_i \in \mathcal{A}} u_i(\pi(a_i))$.

\Rightarrow : Assume there is a *b*-bounded 2-sharing δ such that $\text{usw}(\Pi^\delta) \geq k$. Based on δ , we can find a matching M with the claimed weight by including the corresponding edges for each $e \in \mathcal{E}_s$ with $\delta(e) \neq \emptyset$ and edges between the remaining normal vertices and all dummy vertices. Formally, for each edge $(a_{i_1}, a_{i_2}) \in \mathcal{E}_s$ such that $\delta(a_{i_1}, a_{i_2}) = r_{j_1}$ and $r_{j_1} \in \pi(a_{i_1})$ we add the edge $(v_{i_1}^{j_1}, v_{i_2}^{j_2})$ into matching M , where $r_{j_2} \in \pi(a_{i_2})$ is an arbitrary resource except for the resource that is shared by a_{i_2} with some other agent under δ . Notice that $w(v_{i_1}^{j_1}, v_{i_2}^{j_2}) = \max\{u_{i_1}(r_{j_2}), u_{i_2}(r_{j_1})\} \geq u_{i_2}(r_{j_1})$. After this, for each agent a_i there are at least n_i normal vertices not matched and exactly n_i dummy vertices not matched, so we can add n_i dummy edges of the same weight W into M , each containing one normal vertex and one dummy vertex. Since $\text{usw}(\Pi^\delta) \geq k$, we have that $\sum_{e \in M} w(e) = \text{usw}(\Pi^\delta) - \text{usw}(\pi) + P \geq k - \text{usw}(\pi) + P$.

\Leftarrow : Assume there is matching M in graph G with the claimed weight. Without loss of generality, we can assume M contains n_i dummy edges for every agent a_i since the weight of dummy edges is no smaller than that of non-dummy edges. Then for each agent a_i , M contains at most c non-dummy edges. Based on these non-dummy edges in M we can find the corresponding *b*-bounded 2-sharing δ such that $\text{usw}(\Pi^\delta) \geq k$ as follows. For each non-dummy edge $(v_{i_1}^{j_1}, v_{i_2}^{j_2}) \in M$, we set $\delta(a_{i_1}, a_{i_2}) = r_{j_1}$ if $u_{i_2}(r_{j_1}) \geq u_{i_1}(r_{j_2})$ and $\delta(a_{i_1}, a_{i_2}) = r_{j_2}$ otherwise. We have that

$$\begin{aligned} \text{usw}(\Pi^\delta) &= \text{usw}(\pi) + \sum_{e=(v_{i_1}^{j_1}, v_{i_2}^{j_2}): \delta(a_{i_1}, a_{i_2}) \neq \emptyset} w(e) \\ &= \text{usw}(\pi) + \sum_{e \in M} w(e) - P \geq k. \end{aligned}$$

Since M contains at most b non-dummy edges for each a_i , a_i participates in at most b sharings in δ . \square

Next we consider *b-EWSA*. When $b = 1$, we show in Lemma 1 that we can reduce the problem to MAXIMUM MATCHING. The idea is to partition all agents into two subgroups according to the target k and build a bipartite graph characterizing whether one agent from one group can improve the utility of one agent from the other group to k by sharing, then the problem is just to find a maximum matching on the bipartite graph.

Lemma 1 (*). 1-EWSA is solvable in polynomial time.

On the other hand, we show that b -EWSA is NP-hard when $b \geq 2$. Notice that there is a trivial reduction from 3-PARTITION to b -EWSA if $b \geq 3$. To show the result for any $b \geq 2$, we reduce from the strongly NP-hard NUMERICAL THREE-DIMENSIONAL MATCHING (N3DM) (Garey and Johnson 1979).

Theorem 2. b -EWSA is NP-hard for any constant $b \geq 2$.

Proof. We present a polynomial-time many-one reduction from the NP-hard N3DM. Therein, given 3 multisets of positive integers X, Y, Z , each containing m elements, and a bound T , the task is to decide whether there is a partition S_1, S_2, \dots, S_m of $X \cup Y \cup Z$ such that each S_i contains exactly one element from each of X, Y, Z and the sum of numbers in each S_i is equal to T . Given an instance (X, Y, Z) of N3DM, we construct an instance of b -EWSA as follows. Without loss of generality, assume all elements from $X \cup Y \cup Z$ are smaller than T and the sum of them is equal to $B := mT$. We set the goal $k = (B^2 + B + 1)T$. We create 3 groups of agents corresponding to the 3 multisets X, Y, Z . For each $x_i \in X$, we create an agent a_i^1 in group 1 who holds a *large* resource that is valued as k by itself and $B^2T + x_i$ by all other agents. For each $y_i \in Y$, we create an agent a_i^2 in group 2 who holds a *middle* resource that is valued as k by itself and $BT + y_i$ by all other agents. For each $z_i \in Z$, we create an agent a_i^3 in group 3 who holds a *small* resource that is valued as z_i by itself and 0 by all other agents. In the initial allocation, all $2m$ agents in group 1 and group 2 have utility exactly k and all m agents from group 3 have utility less than k . The proof of correctness is presented in the long version of the paper (Bredereck et al. 2021). \square

4 Reducing Envy by Sharing

In this section, we study the problem of reducing envy through sharing, defined as follows.

Definition 5. Given an initial complete allocation π of resources \mathcal{R} to agents \mathcal{A} , a sharing graph \mathcal{G}_s , an attention graph \mathcal{G}_t , and a non-negative integer k , Envy Reducing Sharing Allocation (ERSA) asks if there is a simple 2-sharing δ such that the number of envious agents $|\text{Env}(\Pi^\delta)| \leq k$.

Notice that in the above definition we restrict that the sharing is a *simple 2-sharing*, i.e., 1-bounded 2-sharing, because the problem for reducing envy in this setting is already NP-hard, even in a special case when the attention graph and the sharing graph are (bidirectional) cliques and the goal is to decrease the number of envious agents by one, as shown in Theorem 3.

Theorem 3 (*). ERSA is NP-hard even if the attention graph and the sharing graph are (bidirectional) cliques, and the goal is to reduce the number of envious agents by at least one.

Theorem 3 in fact constitutes a strong intractability result and it calls for further studies on other specific features of the input. We counteract the intractability result of ERSA (Theorem 3) by considering cases with few agents, tree-like graphs, identical utility functions, or few resources.

Reducing Envy for Few Agents

The simple fact that for n agents and m resources there are at most m^n possible 2-sharings leads to a straightforward brute-force algorithm that runs in polynomial time if the number of agents is constant (see Observation ?? in the supplementary material). However, due to the factor m in the base, the running time of such an algorithm skyrockets with large number of resources (even for small, fixed values of n). We improve this by showing that ERSA is fixed-parameter tractable with respect to the number of agents.

Theorem 4. ERSA with n agents and m resources is solvable in $O((2n)^n m^2)$ time.

The high-level idea behind Theorem 4 is as follows. In order to find a desired sharing, our algorithm guesses *target agents*—a set of at least $n - k$ agents that do not envy in the desired sharing—and a *sharing configuration*—a set of ordered pairs of agents that share some resource in the desired sharing. Then, for such a guessed pair, the algorithm tests whether the desired sharing indeed exists. If it is true for at least one guessed pair, then the algorithm returns “yes”; otherwise, it returns “no.” The main difficulty in checking the existence of the desired sharing is that we need to maintain the envy-freeness within target agents while increasing their utilities.

Before stating the algorithm more formally, we give some notation and definitions. Let C be a fixed set of target agents.

Definition 6. A sharing configuration M for a set C of target agents is a set of arcs such that

1. M is a set of vertex-disjoint arcs and
2. if $(i, j) \in M$, then $\{i, j\} \in \mathcal{E}_s$ and $j \in C$.

A simple 2-sharing δ is called a realization of M if δ only specifies the shared resource for each arc in M ; formally, for each $(i, j) \in M$, we have that $(\delta(\{i, j\}) \neq \emptyset) \wedge (\delta(\{i, j\}) \in \pi(i))$, and for each $\{i, j\}$ with $\delta(\{i, j\}) \neq \emptyset$, we have that $((i, j) \in M \wedge \delta(\{i, j\}) \in \pi(i)) \vee ((j, i) \in M \wedge \delta(\{i, j\}) \in \pi(j))$. A realization δ is feasible if $C \cap \text{Env}(\Pi^\delta) = \emptyset$. i.e., no agent in C will be envious under δ .

Note that a sharing configuration does not only describe shares of a proper simple 2-sharing but also ensures that the shared resources are indeed shared “to” the target agents; we justify this restriction later in Lemma 3.

Let us fix a sharing configuration M for C . For each target agent a_i , we define a set P_i^0 of *initially possible resources* that a_i might get in some realization of M . For convenience, we augment each P_i^0 with a dummy resource d_i that has utility zero for every agent. Formally, we have

$$P_i^0 := \begin{cases} \pi(j) \cup \{d_i\} & \text{if } \exists j \text{ such that } (j, i) \in M, \\ \{d_i\} & \text{otherwise.} \end{cases} \quad (1)$$

For each target agent $a_i \in C$, we define a *utility threshold* t_i —the smallest utility agent a_i must have such that a_i will not envy agents *outside* C . Formally, if there is at least one agent $a_j \notin C$ such that $(a_i, a_j) \in \mathcal{G}_t$, then

$$t_i := \max_{a_j \notin C, (a_i, a_j) \in \mathcal{G}_t} u_i(\pi(j)), \quad (2)$$

```

DoesFeasibleRealizationExist( $\mathcal{G}_t, \pi, \{u_i\}_{i \in C}, C, M$ )
for each agent  $a_i \in C$  do
     $P_i \leftarrow P_i^0 \setminus \{r \in P_i \mid u_i(\pi(i) \cup \{r\}) < t_i\}$ ;
repeat
     $B \leftarrow \bigcup_{a_i \in C} F_i(\{P_1, P_2, \dots, P_{|C|}\})$ ;
     $P_i \leftarrow P_i \setminus B$ ;
until  $B = \emptyset$ ;
if  $\exists i$  with  $P_i = \emptyset$  then return "no" else return "yes";

```

Algorithm 1: Testing existence of a feasible realization of sharing configuration M for set C of target agents.

otherwise, $t_i := 0$. If some target agent cannot achieve its utility threshold by obtaining at most one of its initially possible resources, then there is no realization of M such that the agent does not envy. We express it more formally in Observation 1.

Observation 1. *There is no feasible realization of M in which some agent $a_i \in C$ gets a resource $r \in P_i^0$ such that $u_i(\pi(i) \cup \{r\}) < t_i$.*

For each target agent $a_i \in C$, we define a set of *forbidden resources*.

Definition 7. *Let $C = \{a_1, a_2, \dots, a_q\}$ and let $\mathcal{P} = \{P_1, P_2, \dots, P_q\}$ be a family of sets of possible resources for the target agents. Then resource $r \in P_i$ is a forbidden resource for some target agent a_i if there is some target agent a_j with $(a_j, a_i) \in \mathcal{G}_t$ such that*

$$\max\{u_j(\pi(j) \cup \{r'\}) \mid r' \in P_j\} < u_j(\pi(i) \cup \{r\}),$$

that is, if agent a_i gets resource r , then agent a_j will envy a_i even if a_j gets its most valuable resource from P_j . We denote the set of all forbidden resources for a_i as $F_i(\mathcal{P})$.

Observe that in every feasible realization no target agent gets one of its forbidden resources since otherwise there is another target agent that envies.

Observation 2. *Let \mathcal{P} be a family of possible resources for the target agents. In every feasible realization no target agent a_i gets a resource from $F_i(\mathcal{P})$.*

Based on the above observations, Algorithm 1 tests whether for a pair of a set C of target agents and a sharing configuration M there is a feasible realization. The algorithm keeps track of the possible resources P_i for each target agent a_i . Starting with each P_i equal to the corresponding set of initially possible resources, it utilizes Observation 1 and removes the “low-utility” resources. Then, utilizing Observation 2, the algorithm finds all forbidden resources for a particular collection of the possible resources for the target agents and eliminates the forbidden resources. This procedure is repeated exhaustively. Finally, if at least one of the possible resource sets is empty, the algorithm outputs “no.” Otherwise, the algorithm returns “yes” since at least one resource remained in the set of possible resources for every target agent.

After applying Observation 1 and 2 to Algorithm 1 and proving its correctness (Lemma 2 and Lemma 3), we finish the proof of Theorem 4.

Lemma 2 (★). *In Algorithm 1, if some P_i is empty after the repeat-loop, then there is no feasible realization for M for C .*

Lemma 3 (★). *If there is a simple 2-sharing σ such that $C \cap \text{Env}(\Pi^\sigma) = \emptyset$, then there is a sharing configuration M for C that has a feasible realization and Algorithm 1 outputs “yes”; otherwise, the algorithm outputs “no”.*

Proof of Theorem 4. According to Lemma 3, to solve an instance of ERSA, it is enough to test whether there is a pair of a target subset and a sharing configuration that has a feasible realization. Since, checking a feasible realization, due to Lemma 3, can be done by Algorithm 1, we check all such possible pairs and return “yes” if there is (at least) one that has a feasible realization.

There are $O(2^n)$ possible target sets and at most n^n possible sharing configurations per target set, which gives $O((2n)^n)$ cases. For each case, we apply Algorithm 1. Therein, the for-loop takes $O(nm)$ time. Concerning the repeat-loop that runs at most m times, computing the set B takes $O(nm)$ time; thus, the repeat-loop takes $O(nm^2)$ time. Finally, Algorithm 1 runs in $O(nm^2)$ time and ERSA can be solved in $O((2n)^n m^2)$ time. \square

Next, we show that restricting the parameter k “number of envious agents” does not help to make ERSA solvable in polynomial time.

Theorem 5 (★). *ERSA is NP-hard even if the goal is to reduce the number of envious agents from one to zero.*

Reducing Envy for Tree-like Graphs

We study how the tree-like structure of the sharing graph and the attention graph influences the computational complexity of ERSA. Studying tree-likeness, we hope for tractability for quasi-hierarchical social networks, where agents at the same level of the hierarchy influence each other but they rather do not do so in a cross-hierarchical manner.

Theorem 3 shows that when both graphs are (bidirectional) cliques ERSA is NP-hard. We continue to focus on the case when the underlying graph of the attention graph is the same as the sharing graph. Note that this restriction appears naturally when assuming that one may envy everybody one knows and one may share only with known people. Theorem 6 shows that in this case, if the sharing graph is a path, a tree or being very close to a tree (corresponding to a “hierarchical network”), then we can solve ERSA in polynomial time, while, intuitively, for sharing graphs being “far from a path,” presumably there is no algorithm whose exponential growth in the running time depends only on the “distance from path”.

Theorem 6 (★). *When the underlying graph of the attention graph is the same as the sharing graph, ERSA can be solved in polynomial time if the sharing graph has a constant treewidth (assuming the tree decomposition is given), and ERSA is W[1]-hard with respect to the pathwidth of the sharing graph.*

Reducing Envy for Identical Utility Functions

We proceed by studying the natural special case where all agents have the same utility function. Already in this constrained setting of homogeneous agents allocation problems frequently become hard (Bouveret and Lang 2008). This is why this scenario also attracts quite some attention in the fair allocation literature (Nguyen, Roos, and Rothe 2013; Biswas and Barman 2018; Barman, Krishnamurthy, and Vaish 2018; Bouveret and Lang 2008). Here, we focus on cliques that naturally model small, dense communities and allow for convenient comparison with the classical setting of indivisible, non-shareable resources (where the attention graph is implicitly assumed to be a bidirectional clique).

Theorem 3 already shows that restricting the attention graph and the sharing graph to be cliques is not enough to make ERSA solvable in polynomial time. However, if all agents have the same utility function, Theorem 7 shows that restricting the attention graph to be a clique alone can achieve polynomial-time solvability. The idea behind the proof of Theorem 7 is that unenvious agents are exactly those with the highest utility, so we can reduce the problem to a bounded number of MAXIMUM MATCHING.

Theorem 7 (★). *ERSA is solvable in polynomial time if the attention graph is a bidirectional clique and all agents have the same utility function.*

We complement Theorem 7 by showing in Theorem 8 that identical utility functions together with the sharing graph being a clique are not sufficient to make ERSA solvable in polynomial time. Note that Theorem 7 and Theorem 8 show an interesting contrast between the influence of the completeness of the attention graph and the sharing graph on the problem’s computational complexity.

Theorem 8 (★). *ERSA is NP-hard even if the sharing graph is a clique, all agents have the same utility function, and the maximum initial bundle size is one. For the same constraints, ERSA is W[1]-hard with respect to the parameter “number of resources.”*

Finally, we observe that for the scenario with a constant number of shared resources, there is a naive brute-force algorithm running in polynomial time.

Observation 3 (★). *ERSA is solvable in polynomial time if the number of (shared) resources is a constant.*

5 Extensions

So far, we have assumed that agents get the full utility of the shared resources and there is no cost of sharing, which may not hold for some situations. Nonetheless, many of our algorithms can be easily adapted to deal with (computational) issues that arise. Consider the case when agents get only a fraction of the full utility from shared resources. Then our algorithms for improving utilitarian welfare (Theorem 1) and egalitarian welfare (Lemma 1) still work with minor changes. For reducing envy it might be that after sharing agents lose some utility and thus become envious. Again, our algorithms for few agents (Theorem 4) or identical utility functions (Theorem 7) still work with minor changes. It is also natural to assume that the central authority has to pay

some cost for each sharing to incentivize agents to share resources and there is a budget on the total cost, then the goal of the central authority is to improve the allocation through sharing with the total budget constraint. For this generalized setting, our algorithms for improving egalitarian welfare (Lemma 1), reducing envy for few agents (Theorem 4) or identical utility functions (Theorem 7) still work with minor changes. We refer to the long version (Bredereck et al. 2021) for details and proofs of the above claims.

6 Conclusion

We brought together two important topics—fair allocation of resources and resource sharing. Already our very basic and simple model where each resource can be shared by neighbors in a social network and each agent can participate in a bounded number of sharings led to challenging computational problems. We shed light at their fundamental computational complexity limitations (in the form of computational hardness) and provided generalizable algorithmic techniques (as mentioned in Section 5). Our results are of broader interest in at least two respects. First, we gained insight into a recent line of research aiming at achieving fairness without relaxing its requirements too much. Second, there is rich potential for future research exploring our general model of sharing allocations (in its full power described by Definition 1).

Beyond 2-sharing. We focused on sharing resources between neighbors in a social network. Yet, there are many scenarios where sharing resources among a large group of agents can be very natural and wanted. If every resource can be shared with everyone, then there is a trivial envy-free allocation. Hence, it is interesting to further study the limits of existence of envy-free allocations under various sharing relaxations (concerning parameters such as number of shared resources, number of agents sharing a resource, etc.).

No initial allocation. In our model, the sharing builds up on an initial allocation. It is interesting to study the case without initial allocation, i.e., allocating indivisible but shareable resources to achieve welfare and/or fairness goal.

Combinations of graph classes. We showed that ERSA is NP-hard even if both input graphs are (bidirectional) cliques, but it is polynomial-time solvable if two graphs are the same and have constant treewidth. Based on this, analyzing various combinations of graph classes of the two social networks might be valuable.

Strategic concerns and robustness. We have assumed that all utility values are truthfully reported and correct, and that the agents need not to be incentivized to share resources. Neither of these assumptions might be justified in some cases—the agents might misreport their utility, the utility values might be slightly incorrect, or a sharing can come at a cost for agents (splitting utility from shared resources, as described in Section 5, is an example of the latter). Tackling this kind of issues opens a variety of directions, which includes studying strategic misreporting of utilities, robustness of computed solutions against small utility values perturbations, and finding allocations that incentivize sharing.

Acknowledgements

We are grateful to the AAAI'22 reviewers for their insightful comments. This work was started when all authors were with TU Berlin. Andrzej Kaczmarczyk was supported by the DFG project “AFFA” (BR 5207/1 and NI 369/15) and by the European Research Council (ERC). Junjie Luo was supported by the DFG project “AFFA” (BR 5207/1 and NI 369/15) and the Singapore Ministry of Education Tier 2 grant (MOE2019-T2-1-045).

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 101002854).



European Research Council
Established by the European Union

References

- Abebe, R.; Kleinberg, J.; and Parkes, D. C. 2017. Fair Division via Social Comparison. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS’17)*, 281–289.
- Aziz, H.; Bouveret, S.; Caragiannis, I.; Giagkousi, I.; and Lang, J. 2018. Knowledge, Fairness, and Social Constraints. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI’18)*, 4638–4645.
- Barman, S.; Krishnamurthy, S. K.; and Vaish, R. 2018. Finding Fair and Efficient Allocations. In *Proceedings of the 19th ACM Conference on Economics and Computation (EC’19)*, 557–574.
- Bei, X.; Qiao, Y.; and Zhang, S. 2017. Networked Fairness in Cake Cutting. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI’17)*, 3632–3638.
- Belk, R.; Eckhardt, G.; and Bardhi, F. 2019. *Handbook of the Sharing Economy*. Edward Elgar Pub.
- Beynier, A.; Chevaleyre, Y.; Gourvés, L.; Harutyunyan, A.; Lesca, J.; Maudet, N.; and Wilczynski, A. 2019. Local Envy-Freeness in House Allocation Problems. *Autonomous Agents and Multi-Agent Systems*, 33: 591–627.
- Biswas, A.; and Barman, S. 2018. Fair Division Under Cardinality Constraints. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI’18)*, 91–97.
- Bouveret, S.; Cechlárová, K.; Elkind, E.; Igarashi, A.; and Peters, D. 2017. Fair Division of a Graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI’17)*, 135–141.
- Bouveret, S.; Chevaleyre, Y.; and Maudet, N. 2016. Fair Allocation of Indivisible Goods. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*, chapter 13, 311–329. Cambridge University Press.
- Bouveret, S.; and Lang, J. 2008. Efficiency and Envy-freeness in Fair Division of Indivisible Goods: Logical Representation and Complexity. *Journal of Artificial Intelligence Research*, 32(1): 525–564.
- Bredereck, R.; Kaczmarczyk, A.; Luo, J.; Niedermeier, R.; and Sachse, F. 2021. On Improving Resource Allocations by Sharing. *CoRR*, 2112.07525.
- Bredereck, R.; Kaczmarczyk, A.; and Niedermeier, R. 2018. Envy-Free Allocations Respecting Social Networks. In *Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems (AAMAS’18)*, 283–291.
- Brustle, J.; Dippel, J.; Narayan, V. V.; Suzuki, M.; and Vetta, A. 2020. One Dollar Each Eliminates Envy. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC’20)*, 23–39.
- Budish, E. 2011. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, 119(6): 1061–1103.
- Caragiannis, I.; Gravin, N.; and Huang, X. 2019. Envy-Freeness Up to Any Item with High Nash Welfare: The Virtue of Donating Items. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC’19)*, 527–545.
- Caragiannis, I.; and Ioannidis, S. 2020. Computing envy-freeable allocations with limited subsidies. *CoRR*, abs/2002.02789.
- Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2019. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Transactions on Economics and Computation*, 7(3): 12:1–12:32.
- Chaudhury, B. R.; Kavitha, T.; Mehlhorn, K.; and Sgouritsa, A. 2021. A Little Charity Guarantees Almost Envy-Freeness. *SIAM Journal on Computing*, 50(4): 1336–1358.
- Chevaleyre, Y.; Endriss, U.; and Maudet, N. 2007. Allocating Goods on a Graph to Eliminate Envy. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI’07)*, 700–705.
- Chevaleyre, Y.; Endriss, U.; and Maudet, N. 2017. Distributed fair allocation of indivisible goods. *Artificial Intelligence*, 242: 1–22.
- Eiben, E.; Ganian, R.; Hamm, T.; and Ordyniak, S. 2020. Parameterized Complexity of Envy-Free Resource Allocation in Social Networks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI’20)*, 7135–7142.
- Friedman, E.; Psomas, C.-A.; and Vardi, S. 2015. Dynamic Fair Division with Minimal Disruptions. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC’15)*, 697–713.
- Friedman, E.; Psomas, C.-A.; and Vardi, S. 2017. Controlled Dynamic Fair Division. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC’17)*, 461–478.
- Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Gourvès, L.; Lesca, J.; and Wilczynski, A. 2017. Object Allocation via Swaps along a Social Network. In Sierra, C.,

ed., *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, 213–219.

Halpern, D.; and Shah, N. 2019. Fair Division with Subsidy. In *Proceedings of the 12th International Symposium on Algorithmic Game Theory (SAGT'19)*, 374–389.

He, J.; Procaccia, A. D.; Psomas, A.; and Zeng, D. 2019. Achieving a Fairer Future by Changing the Past. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, 343–349.

Huang, S.; and Xiao, M. 2019. Object Reachability via Swaps along a Line. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'19)*, 2037–2044.

Lange, P.; and Rothe, J. 2019. Optimizing Social Welfare in Social Networks. In *Proceedings of the 6th International Conference on Algorithmic Decision Theory (ADT'19)*, 81–96.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On Approximately Fair Allocations of Indivisible Goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC'04)*, 125–131.

Nguyen, T. T.; Roos, M.; and Rothe, J. 2013. A survey of approximability and inapproximability results for social welfare optimization in multiagent resource allocation. *Annals of Mathematics and Artificial Intelligence*, 68(1-3): 65–90.

Sandomirskiy, F.; and Segal-Halevi, E. 2019. Fair Division with Minimal Sharing. *CoRR*, abs/1908.01669.

Schor, J. B.; and Cansoy, M. 2019. The Sharing Economy. In Wherry, F. F.; and Woodward, I., eds., *The Oxford Handbook of Consumption*, 51–74. Oxford University Press.

Segal-Halevi, E. 2019. Fair Division with Bounded Sharing. *CoRR*, abs/1912.00459.

Williams, L.; Kessler, R. R.; Cunningham, W.; and Jeffries, R. 2000. Strengthening the case for pair programming. *IEEE Software*, 17(4): 19–25.