Local Envy-Freeness in House Allocation Problems

Aurélie Beynier*, Yann Chevaleyre†, Laurent Gourvès†, Julien Lesca†,
Nicolas Maudet*, Anaëlle Wilczynski†
* Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
{aurelie.beynier,nicolas.maudet}@lip6.fr
† Université Paris-Dauphine, PSL, CNRS, LAMSADE, Paris, France
{yann.chevaleyre,laurent.gourves,julien.lesca,anaelle.wilczynski}@dauphine.fr

ABSTRACT

We study the fair division problem consisting in allocating one item per agent so as to avoid (or minimize) envy, in a setting where only agents connected in a given social network may experience envy. In a variant of the problem, agents themselves can be located on the network by the central authority. These problems turn out to be difficult even on very simple graph structures, but we identify several tractable cases. We further provide practical algorithms and experimental insights.

KEYWORDS

Object allocation; Envy-Freeness; Complexity; Algorithms

ACM Reference Format:
Aurélie Beynier, Yann Chevaleyre, Laurent Gourvès, Julien Lesca, Nicolas Maudet, Anaëlle Wilczynski. 2018. Local Envy-Freeness in House Allocation Problems. 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

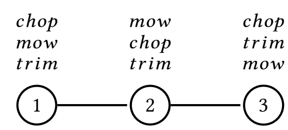
Fairly allocating resources to agents is a fundamental problem in economics and computer science, and has been the subject of intense investigations [10, 13]. Recently, several papers have explored the consequences of assuming in such settings an underlying network connecting agents [2, 4, 8, 13]. The most intuitive interpretation is that agents have limited information regarding the overall allocation. Two agents can perceive each other if they are directly connected in the graph.

A fairness measure, very sensitive to the information available to agents, is the notion of *envy* [20]. Indeed, envy occurs when an agent prefers the share of some other agents over her own. Accounting for a network topology boils down to replace "other agents" by "neighbors". The notion of envy can thus naturally be extended to account for the limited visibility of the agents. Intuitively, an allocation will be *locally envy-free* if none of the agents envies her neighbors. This notion has been referred as *graph*, *social*, or *local envy-freeness* [2, 4, 12, 13, 19].

In this paper, we are concerned with the allocation of indivisible goods within a group of agents. The setting we study in this paper is arguably one of the simplest in resource allocation, known in economics as house allocation [1, 23, 31]: agents have (strict) preferences over items, and each agent must receive exactly one item. In the case of a complete network, envy-freeness is not a very exciting notion in that setting. Indeed, for an allocation to be envy-free, each agent must get her top object (and this is obviously also a Pareto-optimal allocation in that case). When an agent is only connected to a subset of the other agents, she may not need to get her top-resource to be envy-free. The locations of the resources on the graph as well as the connections between the agents are then crucial issues in order to compute a locally envy-free allocation.

To see how the network can make a difference, consider the following scenario.

Example 1.1. Suppose for instance a team of workers taking their shifts in sequence, to which a central authority must assign different jobs. Workers have preferences regarding these jobs. As the shifts are contiguous and as the employees work at the same place, they have the opportunity to see the job allocated to some other workers, as one ends and the other one begins her shift. This would be modeled as a line topology in our setting as depicted on the graph below. To make things concrete, suppose there are three jobs, chop the tree, mow the lawn, and trim the hedge, and three gardeners (1, 2 and 3) with preferences $1 : chop \succ mow \succ trim$, $2 : mow \succ chop \succ trim$, $3 : chop \succ trim \succ mow$, taking shifts in order 1, 2 and finally 3. On the figure, rankings are mentioned over agents (with top jobs at the top, etc.)



By allocating the job *chop the tree* to agent 1, *mow the lawn* to agent 2, and *trim the hedge* to agent 3, we get an envy-free allocation if we disregard the fact that agent 1 and 3 may be envious of each other. Note that a locally envy-free allocation is not necessarily Pareto-optimal (take the same allocation, but the ranking of agent 1 to be $trim \succ chop \succ mow$), but that giving her top item to each agent if possible will always be an envy-free Pareto-optimal allocation in *any* network.

The reader may object that, in Example 1.1, agent 3 may still be envious of agent 1, because she knows that this agent must have received the task agent 2 didn't get, i.e. *chop the tree*. This is a valid point, to which we provide two counter-arguments. First, as a technical response, note that in general agents would not know exactly who gets the items they do not see. Thus, although agents may know that they must be envious of *some* agents, they cannot identify which one, which makes a significant difference in the case of envy. Our second point is more fundamental and concerns the model and the motivation of this work. Clearly, the

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. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

existence of a network may be due to an underlying notion of proximity (either geographical, or temporal as in our example) in the problem. However, another interpretation of the meaning of links must be emphasized: links may represent envy the central authority is concerned with. In other words, although there may theoretically be envy among all agents, the central authority may have reasons to only focus on some of these envy links. For instance, you may wish to avoid envy among members of the same team in your organization, because they actually work together on a daily basis (in that case links may capture team relationships). Under this interpretation, a network of degree $n-2$, that is the total number of agents minus 2, could for instance model a situation where agents team-up in pairs and conduct a task together, sharing their resources. In a similar vein, we may focus on avoiding envy among "similar" agents, because they may be legitimate to complain if they are not treated equally despite similar competences, for instance.

1.1 Related work

Our work is connected to a number of recent contributions–beyond the works directly addressing fair allocation on graphs already mentioned. Recently, house allocation settings have been discussed, notably in relation with swap dynamics [14, 22]. In particular, [22] show how graph structures can affect the complexity of some decision problems regarding such dynamics, for instance whether some allocation is reachable by a sequence of swaps among agents. The allocation of a graph has also recently been studied [9]. In this context, the nodes of the graph represent indivisible resources to allocate and edges formalize connectivity constraints between the resources. Some computational aspects of allocating agents on a line are also discussed in [5]: in that case the line concerns the items (eg. slots) to be allocated, and induces a domain restriction (stronger than single-peakedness). In these cases the model is really different, since the graph is capturing dependencies between the resources (like spatial dependencies for pieces of land). Several ways for a central authority to control fair division have been discussed in [6]: the structure of the allocation problem can be changed by adding or removing items to improve fairness. Interestingly our model introduces a new type of control action: locating agents on a graph. Finally, because envy-freeness is difficult to achieve in general (with indivisible items) [15], different notions of degree of envy have been studied, see *e.g.* [11, 17, 26, 27].

1.2 Contributions and organization

A formal definition of the model, together with the definition of the main problems that we address, are provided in Section 2. Section 3 is dedicated to the problem, called DEC-LEF, of deciding if a central planner, who has a complete knowledge of the social network and the agents' rankings of the objects, can allocate the objects such that no agent will envy a neighbor. We identify intractable and tractable cases of this decision problem, with respect to the number of neighbors of each agent, that is the degree of the nodes in the graph representing the social network. Another relevant parameter is the size of a *vertex cover*. A subset of agents N' forms a vertex cover of the social network if every agent is either in N', or at least one of her neighbors is in N'. We provide an algorithm which shows that DEC-LEF is in **XP** (parameterized by the size of a vertex cover) and a proof of **W[1]**-hardness.

Section 4 is dedicated to optimization problems with two different perspectives: maximizing the number of locally envy-free agents, and maximizing the degree of non-envy of the society. We provide approximation algorithms for both approaches.

A variant of DEC-LEF called DEC-LOCATION-LEF is studied in Section 5. This problem asks if one can decide both the placement of the agents and the object allocation so as to satisfy local envy-freeness. The problem is shown to be NP-complete, and a special case is resolved in polynomial time. To conclude, we report some experimental results (Section 6) and we provide open problems and future directions in Section 7.

2 OUR MODEL AND PROBLEMS

A set of objects O and a set of agents N are given. We assume that $|O| = |N| = n$. Each agent i has a preference relation $\succ_i$ over O (a linear order). The profile of all preference relations is denoted by $\succ$. We are also given a social network modeled as an undirected graph G with vertex set N and edge set E. Each edge in E represents a social relation between the corresponding agents. An instance of a resource allocation problem is thus described by a tuple $\langle N, O, \succ, G = (N, E)\rangle$. When the social network G is dense, it may be easier to describe it through its complement graph $\overline{G}$ which is the unique graph defined on the same vertex set and such that two vertices are connected iff they are not connected in G.

An *allocation* $\mathcal{A}$ is a bijection $N \rightarrow O$ where every agent i gets a single object $\mathcal{A}(i)$. We also generalize the notion of allocation to *partial allocations*, in which we allow objects to be unallocated.

Definition 2.1 (Envy-free). An allocation $\mathcal{A}$ is *envy-free* (EF) if no pair of agents i, j satisfies $\mathcal{A}(j) \succ_i \mathcal{A}(i)$.

Definition 2.2 (Locally envy-free). An allocation $\mathcal{A}$ is *locally envy-free* (LEF) if no pair of agents $\{i, j\} \in E$ satisfies $\mathcal{A}(j) \succ_i \mathcal{A}(i)$.

For a given allocation, an agent is LEF if she prefers her object to the object(s) of her neighbor(s).

Several notions of degrees of envy have been studied [11, 13, 26, 27]. In our context we shall study the number of envious agents, and a degree measure capturing some simple notion of intensity of envy, in terms of the difference of ranks between items (these two notions would correspond to $e^{sum,max,bool}$ and $e^{sum,sum,raw}$, up to normalization, under the classification of [13]).

Definition 2.3. (degrees of (non)-envy). Given an allocation $\mathcal{A}$, the degree of envy of agent i towards agent j is $e(\mathcal{A}, i, j) = \frac{1}{n-1}\max(0, r_i(\mathcal{A}(i)) - r_i(\mathcal{A}(j)))$, where $r_i(o)$ is the rank of object o in $i's$ preferences. The average degree of envy (respectively of non-envy) is $\mathcal{E}(\mathcal{A}) = \frac{1}{2|E|}\sum_{\{i,j\}\in E} e(\mathcal{A}, i, j) + e(\mathcal{A}, j, i)$ (respectively is $\mathcal{NE}(\mathcal{A}) = 1 - \mathcal{E}(\mathcal{A})$).

Note that for a given allocation $\mathcal{A}$, an agent i envies a neighboring agent j if and only if $e(\mathcal{A}, i, j) > 0$.

We mainly address four problems: DEC-LEF, MAX-LEF, MAX-NE and DEC-LOCATION-LEF. The first one is a decision problem regarding the existence of an LEF allocation over a given social network.

Definition 2.4 (DEC-LEF). Given an instance $\langle N, O, \succ, G = (N, E)\rangle$, is there an LEF allocation $\mathcal{A}$?

The second and the third ones are optimization problems in which an allocation that is as close as possible to local envy-freeness is sought, using the aforementioned criteria.

Definition 2.5 (MAX-LEF). Given an instance $\langle N, O, \succ, G = (N, E)\rangle$, find an allocation that maximizes the number of LEF agents.

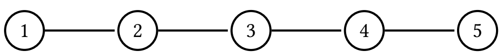
Definition 2.6 (MAX-NE). Given an instance $\langle N, O, \succ, G = (N, E)\rangle$, find an allocation that maximizes the average degree of non-envy $\mathcal{NE}(\mathcal{A})$.

In DEC-LOCATION-LEF, one has to place the agents on the network in addition to the allocation. This placement makes sense if we consider Example 1.1 where the agents take shifts.

Definition 2.7 (DEC-LOCATION-LEF). Given an undirected network (V, E), and $\langle N, O, \succ\rangle$, are there two bijections $\mathcal{A} : N \to O$ and $\mathcal{L} : N \to V$ ($\mathcal{A}$ and $\mathcal{L}$ determine the allocation of the objects and the location of the agents on the network, respectively) such that $\mathcal{A}(i) \succ_i \mathcal{A}(j)$ for every edge $\{\mathcal{L}(i), \mathcal{L}(j)\} \in E$?

Example 2.8. As a warm-up, consider 5 agents allocated on a line, as depicted below. Each agent has a strict ranking over objects.

a	c	a	b	c
b	a	b	a	e
c	b	d	d	b
d	d	c	c	a
e	e	e	e	d



Is there an LEF allocation of goods to agents? If not, what is the minimum number of envious agents? Finally, is it possible to find an LEF allocation by relocating agents on this line?

3 DECISION PROBLEM

This section is devoted to DEC-LEF. Our main findings settle the computational status of DEC-LEF with respect to the degree of the nodes in the social network, as well as the size of a vertex cover.

First of all, note that some objects cannot be assigned to certain agents for the allocation to be LEF. For example, the best object of an agent cannot be assigned to one of her neighbors. More generally, no better object than the one allocated to an agent can be assigned to one of her neighbors, leading to the following observations:

OBSERVATION 1. *In any LEF allocation, an agent with k neighbors must get an object ranked among her $n - k$ top objects.*

OBSERVATION 2. *In any LEF allocation, the best object for an agent is either assigned to herself or to one of her neighbors in $\overline{G}$.*

Observation 1 implies that an agent having $n - 1$ neighbors must receive her best object in any LEF allocation. Therefore, the computational complexity of DEC-LEF is not related to those agents, and in the following we assume that the social network does not contain a vertex of degree $n - 1$.

3.1 DEC-LEF and degree of nodes

Our first result shows that DEC-LEF is computationally difficult, even if the social network is very sparse, i.e. each agent has one neighbor in G. This is somewhat surprising as such a network offers very little possibility for an agent to be envious.

THEOREM 3.1. *DEC-LEF is* **NP**-*complete, even if G is a matching.*

PROOF. The reduction is from 3SAT [21]. We are given a set of clauses $C = \{c_1, \cdots, c_m\}$ defined over a set of variables $X = \{x_1, \cdots, x_p\}$. Each clause is disjunctive and consists of 3 literals. Is there a truth assignment which satisfies all the clauses?

Take an instance $\mathcal{I} = \langle C, X\rangle$ of 3SAT and create an instance $\mathcal{J}$ of DEC-LEF as follows.

The set of objects is $O = \{u_i^j : 1 \le i \le p,\ 1 \le j \le m\} \cup \{\overline{u}_i^j : 1 \le i \le p,\ 1 \le j \le m\} \cup \{q_j : 1 \le j \le m\} \cup \{t_{ij} : 1 \le i \le p,\ 1 \le j \le m\} \cup \{h_\ell : 1 \le \ell \le m(p-1)\}$. Here, u_i^j and $\overline{u}_i^j$ correspond to the unnegated and negated literals of x_i in clause c_j, respectively, q_j corresponds to the clause c_j, and the t_{ij}'s and h_ℓ's are gadgets. Thus, $|O| = 4mp$.

The set of agents N is built as follows. For each $(i, j) \in [p] \times [m]$, create a pair of variable-agents X_{ij}, X'_{ij} which are linked in the social network. For each $j \in [m]$, create a pair of clause-agents K_j, K'_j which are linked in the network. For each $\ell \in [m(p-1)]$, create a pair of garbage-agents L_ℓ, L'_ℓ which are linked in the network. Thus, the network consists of a perfect matching with $4mp$ agents.

Each clause c_j is associated with the pair of clause-agents (K_j, K'_j), q_j and 3 objects corresponding to its literals. For example, $c_2 = x_1 \vee x_4 \vee \overline{x}_5$ is associated with objects q_2, u_1^2, u_4^2, and $\overline{u}_5^2$. The preferences of the clause-agents are:

- $K_j : q_j \succ$ (the 3 objects related to the literals of c_j) $\succ$ rest
- $K'_j :$ (the 3 objects related to the literals of c_j) $\succ q_j \succ$ rest

where "rest" means the remaining objects. Both "rest" and the 3 objects corresponding to the literals of c_j are arbitrarily ordered, but in the same way for K_j and K'_j. Each variable x_i is associated with the m pairs of variable-agents (X_{ij}, X'_{ij}), $1 \le j \le m$. The preferences of these variable-agents are:

- $X_{i1} : u_i^1 \succ t_{i1} \succ \overline{u}_i^1 \succ t_{i2} \succ \text{rest}_i^1$
- $X'_{i1} : t_{i1} \succ u_i^1 \succ t_{i2} \succ \overline{u}_i^1 \succ \text{rest}_i^1$

- $X_{i2} : u_i^2 \succ t_{i2} \succ \overline{u}_i^2 \succ t_{i3} \succ \text{rest}_i^2$
- $X'_{i2} : t_{i2} \succ u_i^2 \succ t_{i3} \succ \overline{u}_i^2 \succ \text{rest}_i^2$
- $\vdots$
- $X_{im} : u_i^m \succ t_{im} \succ \overline{u}_i^m \succ t_{i1} \succ \text{rest}_i^m$
- $X'_{im} : t_{im} \succ u_i^m \succ t_{i1} \succ \overline{u}_i^m \succ \text{rest}_i^m$

where "rest_i^j" means the remaining objects arbitrarily ordered, but in the same way for X_{ij} and X'_{ij}. The preferences of the garbage-agents (L_ℓ, L'_ℓ), $1 \le \ell \le m(p-1)$ are:

- $L_\ell : h_\ell \succ U \succ$ rest
- $L'_\ell : U \succ h_\ell \succ$ rest

where $U = \{u_i^j, \overline{u}_i^j : i \in [p],\ j \in [m]\}$, "rest" is the set of remaining objects, and both U and "rest" are arbitrarily ordered in the same way for L_ℓ and L'_ℓ.

We claim that there is an LEF allocation in $\mathcal{J}$ if, and only if, there is a truth assignment satisfying $\mathcal{I}$.

Take a truth assignment which satisfies $\mathcal{I}$. One can allocate objects to each variable-agent pair (X_{ij}, X'_{ij}) in such a way that it is LEF: If $x_i = true$, then X_{ij} gets $\overline{u}_i^j$ and X'_{ij} gets $t_{i\,j+1}$ (where

$t_{i\,m+1} := t_{i1}$), otherwise $x_i = false$, X_{ij} gets u_i^j and X'_{ij} gets t_{ij}. One can allocate objects to each clause-agent pair (K_j, K'_j) in such a way that it is LEF: c_j is satisfied thanks to one of its literals; K_j gets q_j and K'_j gets an unallocated object corresponding to a literal of c_j. Finally, allocate objects to each garbage-agent pair (L_ℓ, L'_ℓ) in such a way that it is LEF: L_ℓ gets h_ℓ and L'_ℓ gets any unallocated objects of U.

Suppose an LEF allocation exists for $\mathcal{J}$. Consider a variable x_i. By construction of the preferences of the variable-agent pair (X_{i1}, X'_{i1}), we observe that there is absence of envy in only two cases: X_{i1} gets u_i^1 and X'_{i1} gets t_{i1}, otherwise X_{i1} gets $\overline{u}_i^1$ and X'_{i1} gets t_{i2}. If we are in the first case, then there is absence of envy between X_{im} and X'_{im} only if X_{im} gets u_i^m and X'_{im} gets t_{im} because t_{i1} is already allocated, and so on; the X_{ij}'s get all the u_i^j's (i is fixed but $1 \le j \le m$). If we are in the second case, then there is absence of envy between X_{i2} and X'_{i2} only if X_{i2} gets $\overline{u}_i^2$ and X'_{i2} gets t_{i3} because t_{i2} is already allocated, and so on; the X_{ij}'s get all the $\overline{u}_i^j$'s (i is fixed but $1 \le j \le m$). Thus, set x_i to $false$ (resp. x_i to $true$) if every X_{ij} gets u_i^j (resp. X_{ij} gets $\overline{u}_i^j$).

Consider any clause c_j. By construction of the preferences of the clause-agent pair (K_j, K'_j), we observe that there is absence of envy in only three cases: K_j gets q_j and K'_j gets one of the 3 objects associated with the literals of c_j. Since the allocation is LEF, there is some i^* such that K'_j gets either $u_{i^*}^j$ or $\overline{u}_{i^*}^j$, and this object is not allocated to a variable-agent. Thus, c_j is satisfied by the above truth assignment. To conclude, all the clauses are satisfied. □

The strength of this result lies on the fact that the network structure is extremely simple. As a consequence, it can easily be used as a building block to show hardness of a large variety of graphs. In short, it suffices to introduce additional dummy agents connecting agents from the original matching instance, and to make sure that each dummy agent x'_i ranks first "her" dummy resource d_i, followed by a copy of the ranking (minus d_i) of an (arbitrary) neighbor (so as to ensure that they must indeed receive their dummy resource for the allocation to be LEF).

Corollary 3.2. *DEC-LEF is* **NP***-complete on a line, or on a circle, and generally on graphs of maximum degree k for $k \ge 1$ constant.*

Given this result, one may suspect the problem to be hard on any graph structure beyond a clique. Our next result shows that if the social network is dense enough, then DEC-LEF is polynomial.

Theorem 3.3. *DEC-LEF in graphs of minimum degree $n-2$ is solvable in polynomial time.*

Proof. Note that $\overline{G}$ is a matching in that case. In order to simplify notations, we denote by $\phi(i)$ the neighbor of agent i in $\overline{G}$. Hence, $\phi(i)$ is the unique non-neighbor of agent i in the social network.

We reduce the problem to 2-SAT which is solvable in linear time [3]. Let us consider boolean variables x_{ij} for $1 \le i, j \le n$, such that x_{ij} is true iff object j is assigned to i. Denote by o_i^j the object at position j in the preference relation of agent i.

Consider the following formula φ:

$$\bigwedge_{i \in N} (x_{io_i^1} \vee x_{io_i^2}) \wedge \bigwedge_{\substack{1 \le i < \ell \le n \\ 1 \le j \le n}} (\neg x_{ij} \vee \neg x_{\ell j}) \wedge \bigwedge_{i \in N} (x_{io_i^1} \vee x_{\phi(i) o_i^1})$$

The first part of formula φ expresses that each agent must obtain an object within her top 2, as noted in Observation 1. By combination with the second part of φ, we get that the solution must be an assignment: each agent must obtain her first or second choice but not both since every object is owned by at most one agent and $|N| = |O|$. Observation 2 implies that the best object for agent i must be assigned either to agent i or $\phi(i)$. This condition is given by the last part of the formula. Hence, formula φ exactly translates the constraints of an LEF allocation. □

Interestingly, the status of DEC-LEF changes between social networks of degree at least $n-2$ and those of degree $n-3$. A *regular* graph is such that all of its nodes have the same degree.

Theorem 3.4. *DEC-LEF is* **NP***-complete in regular graphs of degree $n-3$.*

Proof. The reduction is from (3,B2)-SAT [25] which is a restriction of 3SAT where each literal appears exactly twice in the clauses, and therefore, each variable appears four times. Notations are the same as in the proof of Theorem 3.1. Take an instance $\mathcal{I} = \langle C, X \rangle$ of 3SAT and create an instance $\mathcal{J}$ of DEC-LEF as follows.

Instead of describing the social network in $\mathcal{J}$, we describe its complementary $\overline{G}$. Note that $\overline{G}$ is a regular graph of degree 2. Hence, $\overline{G}$ contains a collection of cycles. For each variable x_i, we introduce dummy variable-objects q_i^1 and q_i^2, literal-objects $u_i^1, u_i^2, \overline{u}_i^1$ and $\overline{u}_i^2$ corresponding to its first and second occurrence as an unnegated and negated literal, respectively, as well as a cycle in $\overline{G}$ containing literal-agents $X_i^1, \overline{X}_i^1, X_i^2$ and $\overline{X}_i^2$, connected in this order. Preferences are as follows:

- $X_i^j : q_i^j \succ q_i^{3-j} \succ u_i^j \succ \ldots$
- $\overline{X}_i^j : q_i^j \succ q_i^{3-j} \succ \overline{u}_i^j \succ \ldots$

Note that only the 3 top objects are represented since no object ranked below can lead to an LEF allocation (see Observation 1). Note also that in any LEF allocation, either q_i^1 and q_i^2 are allocated to agents X_i^1 and X_i^2 , either q_i^1 and q_i^2 are allocated to agents $\overline{X}_i^1$ and $\overline{X}_i^2$. The first case can be interpreted in $\mathcal{I}$ as assigning false to x_i, and the later case as assigning true to x_i.

For each clause c_i we introduce dummy clause-objects d_i^1 and d_i^2, as well as a cycle in $\overline{G}$ containing clause-agents K_i^1, K_i^2, K_i^3. The preferences of clause-agent K_i^j are:

- $K_i^j : d_i^1 \succ d_i^2 \succ \ell(i,j) \succ \ldots$

where $\ell(i,j)$ is the literal-object corresponding to the i^{th} literal of c_j. Note that an allocation is LEF if d_i^1, d_i^2 and one literal-object corresponding to a literal of c_i are assigned to K_i^1, K_i^2, K_i^3. This can be interpreted in $\mathcal{I}$ as the requirement for at least one literal of c_i to be true.

The reduction is almost complete but it remains to describe gadgets collecting all unassigned objects. Indeed, so far we have introduced $4m + 3p$ agents and $6m + 2p$ objects. It remains to construct garbage collectors for the $2m - p$ remaining objects. Note that no dummy object (neither variable nor clause) may be part of the remaining objects since they must be assigned to literal-agents or clause-agents in any LEF allocation. Let $\mathcal{L} = \{u_i^j, \overline{u}_i^j : 1 \leq i \leq m, 1 \leq j \leq 2\}$ denotes the set of literal-objects and let $\mathcal{L}(i)$ denote the i^{th} literal-object, where literal-objects are ordered arbitrarily. We describe a gadget collecting a single object of $\mathcal{L}$. Exactly $2m - p$ copies of this gadget will be used in the reduction to collect all the remaining literal-objects. For each $i \leq 4m$, we introduce objects t_i^1 and t_i^2 and a cycle in $\overline{G}$ containing gadget-agents L_i^1, L_i^2 and L_i^3. Furthermore, for each $i \leq 4m - 1$ we introduce gadget-object h_i. Preferences are as follows.

- $L_i^1 : t_i^1 \succ t_i^2 \succ h_{i-1} \succ \ldots$
- $L_i^2 : t_i^1 \succ t_i^2 \succ \ell(i) \succ \ldots$
- $L_i^3 : t_i^1 \succ t_i^2 \succ h_i \succ \ldots$

where h_0 and h_{4m} stand for h_1 and h_{4m-1}, respectively. Note that in any LEF allocation, objects t_i^1 and t_i^2 are allocated to agents belonging to $\{L_i^1, L_i^2, L_i^3\}$, and the remaining unassigned agent receives either h_{i-1}, h_i or $\ell(i)$. Since no more than $4m - 1$ agents can receive a gadget-object, at least one literal-object is assigned to agent L_i^2 for some $i \leq 4m$. Moreover, all gadget-objects must be assigned to gadget-agents since no other agent has a gadget-object in her top 3 objects. Therefore, in every LEF allocation, exactly one literal-object is allocated to an agent belonging to the gadget.

Now let us show that one can allocate objects without envy in the gadget. Let $\ell(i)$ be the literal-object assigned in the gadget. This object must be assigned to L_i^2. Assign t_i^1 and t_i^2 to agents L_i^1 and L_i^3, respectively. For any $j \neq i$, assign t_j^1 to agent L_j^2. Finally, for any $j > i$, h_{j-1} is assigned to L_j^1 and t_j^2 is assigned to L_j^3, and for any $j < i$, h_j is assigned to L_j^3 and t_j^2 is assigned to L_j^1.

It is easy to check that $\mathcal{I}$ is satisfiable iff $\mathcal{J}$ has an LEF allocation. Due to space limitation, we omit the proof of this statement. □

In the same idea as for Theorem 3.1, we can extend the previous hardness result to more general classes of graphs. It suffices to add, in the graph of the previous proof, dummy agents who are connected to three other agents. They have a dummy resource on top of their ranking, followed by the whole ranking of one of her neighbors. Each initial agent ranks last the dummy resources.

Corollary 3.5. *dec-LEF is* **NP***-complete on graphs of minimum degree* $n - k$ *for* $k \geq 3$ *constant.*

3.2 dec-LEF and vertex cover

So far the complexity of dec-LEF has been investigated through the degree of its nodes, but other parameters can be taken into account. Let us show how the size of a (smallest) *vertex cover* can help. A *vertex cover* C of $G = (N, E)$ is a subset of nodes such that $\{u, v\} \cap C \neq \emptyset$ for every edge $(u, v) \in E$. It follows that $I := N \setminus C$ must be an *independent set*, that is a set of pairwise non-adjacent vertices.

Theorem 3.6. *If the social network* G *admits a vertex cover of size* k*, then dec-LEF can be answered in* $O(n^{2k+3})$.

Proof. Find a vertex cover C of the social network and let $I := N \setminus C$. See [24] for a $O(2^k n)$ algorithm which decides and builds a vertex cover of size k in a graph with n vertices. Use brute force to assign k objects of O to C (the time complexity is $O(n^{2k})$). For each partial allocation $\mathcal{A}$ without envy within C, let $O_{-\mathcal{A}}$ be the set of unassigned objects (if no such partial allocation exists, then we can immediately conclude that no LEF allocation exists). Build a bipartite graph $(I, O_{-\mathcal{A}}; E')$ with an edge from agent $i \in I$ to object $o \in O_{-\mathcal{A}}$ if assigning o to i does not create envy. There is an LEF allocation for the entire network if and only if the bipartite graph admits a perfect matching ($O(n^3)$). □

Therefore the method is efficient when k is small. For instance, dec-LEF is polynomial if the social network is a star because the central node of a star is a vertex cover. More generally, Theorem 3.6 implies that dec-LEF is polynomial when $k = O(1)$.

Theorem 3.6 implies that dec-LEF belongs to **XP** when the fixed parameter under consideration is the size of a vertex cover. Recall that a problem belongs to **FPT** if there is an algorithm to solve it with time complexity in $O(f(k)poly(n))$, where f is an arbitrary function depending only on k. One could expect that dec-LEF also belongs to **FPT** for the same parameter since the problem of finding a vertex cover of size k is **FPT**. However, the following theorem shows that there is no hope that dec-LEF belongs to **FPT**.

Theorem 3.7. *dec-LEF parameterized by the size of a vertex cover is* **W[1]***-hard.*

Proof. We present a parameterized reduction from Multicolored Independent Set [18]. An instance of Multicolored Independent Set consists of a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, an integer k, and a partition $(\mathcal{V}_1, \ldots, \mathcal{V}_k)$ of $\mathcal{V}$. The task is to decide if there is an independent set of size k in $\mathcal{G}$ containing exactly one vertex from each set $\mathcal{V}_i$. We construct an instance of dec-LEF as follows. For each vertex v in $\mathcal{V}$, we introduce object o_v. Let O_i denote the set of objects $\{o_v : v \in \mathcal{V}_i\}$, and let $O_i^{\uparrow}$ denote an arbitrary order over the objects of O_i. For each edge $e = \{v, v'\}$ in $\mathcal{E}$, we introduce two agents X_e^v and $X_e^{v'}$, and two edge-objects o_e and o'_e. Let $O_{\mathcal{E}}$ denote the set of edge-objects, and let $O_{\mathcal{E}}^{\uparrow}$ denote an arbitrary ranking over the objects of $O_{\mathcal{E}}$. For each integer $i \leq k$, we introduce agent K_i. The agents of $\{K_i\}_{i \leq k}$ form a clique in the social network G. Furthermore, for each vertex $v \in V_i$ and for each edge $e = \{v, v'\}$ in $\mathcal{E}$, agent X_e^v is connected to agent K_i in G. Finally, for each integer $j \leq |\mathcal{V}| - k$, we introduce agent D_j. Preferences are the following:

- $K_i : O_i^{\uparrow} \succ O_1^{\uparrow} \succ \ldots \succ O_{i-1}^{\uparrow} \succ O_{i-1}^{\uparrow} \succ \ldots O_k^{\uparrow} \succ O_{\mathcal{E}}^{\uparrow}$
- $X_e^v : o_e \succ o_v \succ o'_e \succ \ldots$

Since agent D_j is isolated in G, her preferences may be arbitrary. It is easy to check that $\{K_i\}_{i \leq k}$ forms a vertex cover in the network.

We show that $\mathcal{G}$ has an independent set of size k containing one vertex in each set $\mathcal{V}_i$ iff an LEF allocation exists. Assume first that $\{v_1, \ldots, v_k\}$ is an independent set in $\mathcal{G}$, where $v_i \in \mathcal{V}_i$ for each $i \leq k$. We construct an LEF allocation as follows. For each $i \leq k$, assign o_{v_i} to K_i, and for each edge $e = (v_i, v') \in \mathcal{E}$, assign o_e to $X_e^{v_i}$. For each agent X_e^v such that v is not selected in the independent set, assign o_e to X_e^v if it is still available, and otherwise assign o'_e to X_e^v. Finally, assign the remaining objects arbitrarily. We claim that this allocation is envy-free. Indeed, each agent K_i receives an object of O_i. Furthermore, for each vertex v in $\mathcal{V}_i$ and for each edge e in $\mathcal{E}$, agent X_e^v has a single neighbor who is K_i. If K_i receives o_{v_i} and $v = v_i$ then $X_e^{v_i}$ receives o_e, and otherwise X_e^v receives o'_e.

Assume now that an LEF allocation exists. We claim that each agent K_i should receive an object of O_i. By contradiction, assume that agent K_i receives object $o \notin O_i$. Note that for any $j \neq i$, K_i and K_j are neighbors. Hence, for any object o', if $o \notin O_j$ and $o' \notin O_i \cup O_j$ then $o \succ_{V_i} o'$ iff $o' \succ_{V_j} o$ holds. This implies that if $o \notin O_j$ then an object of O_j must be assigned to K_j to avoid envy between agents K_i and K_j. Therefore, if $o \in O_{\mathcal{E}}$ then agent K_i envies agent K_j, a contradiction. On the other hand, if $o \in O_j$ for some $j \neq i$ then either K_i envies K_j or K_j envies K_i, since $o \succ_{V_i} o'$ iff $o \succ_{V_j} o'$ holds because $o' \in O_j$, a contradiction. Let o_{v_i} denote the object assigned to K_i. We claim that $\{v_1, \ldots, v_k\}$ forms an independent set in $\mathcal{G}$. By contradiction assume that edge e connects v_i and v_j in $\mathcal{G}$. This implies by construction that $X_e^{v_i}$ and $X_e^{v_j}$ are neighbors of K_i and K_j in G, respectively. On one hand, if $X_e^{v_i}$ does not receive o_e then she envies K_i. On the other hand, if $X_e^{v_j}$ does not receive o_e then she envies K_j. Therefore, o_e must be assigned to both $X_e^{v_i}$ and $X_e^{v_j}$, leading to a contradiction since o_e cannot be assigned twice. □

Our findings for DEC-LEF are summarized in Table 1.

degree d	$d \leq k$ ($k \geq 1$ fixed)	NP-c	Th. 3.1
	$d \geq n - k$ ($k \geq 3$ fixed)	NP-c	Th. 3.4
	$d \geq n - 2$	P	Th. 3.3
parameter k on the vertex cover size		XP	Th. 3.6
		W[1]-hard	Th. 3.7

Table 1: Complexity results of DEC-LEF

4 OPTIMIZATION

In light of Section 3, we know that both MAX-LEF and MAX-NE are **NP**-hard even on very simple graph structures. We present in this section approximation algorithms for MAX-LEF and MAX-NE.

4.1 Maximizing the number of LEF agents

This subsection is dedicated to MAX-LEF. A general method is proposed in Algorithm 1. For a maximization problem, an algorithm is ρ-approximate, with $\rho \in [0, 1]$, if it outputs a solution whose value is at least ρ-times the optimal value, for any instance.

Algorithm 1:

1 Find an independent set I of the network (in any opportune way)
2 **for all** $i \in I$ **do**
3 | Let $\mathcal{A}(i)$ be i's most preferred object within O
4 | Remove $\mathcal{A}(i)$ from O
5 Complete $\mathcal{A}$ (in any opportune way) and return $\mathcal{A}$

OBSERVATION 3. *Algorithm 1 is $\frac{|I|}{n}$-approximate for MAX-LEF.*

PROOF. By construction, every member of I is LEF, and the largest number of LEF agents is $|N| = n$. □

OBSERVATION 4. *The construction of I in Algorithm 1 (Step 1) can be done so that a polynomial time $(\Delta + 1)^{-1}$-approximation is produced, where Δ denotes the maximum degree in the social network.*

PROOF. The independent set is built as follows. I is initially empty and while $N \neq \emptyset$, do: choose $i \in N$, add i to I, and remove i and its neighbors from N. Since a node has at most Δ neighbors, I is an independent set of size at least $n/(\Delta + 1)$. Use Observation 3 to get the expected ratio of $(\Delta + 1)^{-1}$. □

The $(\Delta + 1)^{-1}$-approximation algorithm is long known for the MAXIMUM INDEPENDENT SET problem (that is, find an independent set of maximum cardinality), see for example in [28]. The following lemma shows that MAX-LEF shares exactly the same inapproximability results as MAXIMUM INDEPENDENT SET.

LEMMA 1. *Any r-approximate algorithm for MAX-LEF is also a r-approximate algorithm for MAXIMUM INDEPENDENT SET.*

PROOF. Suppose we have an r-approximation algorithm for MAX-LEF. Consider a set of n agents with identical preferences ($\succ_1 = \succ_2 = \ldots \succ_n$) on a graph. Our r-approximation algorithm computes for this instance an allocation $\mathcal{A}$ and a set I of non-envious agents. Because preferences are identical, a pair of connected agents cannot be locally envy-free, whatever the allocation is. In this setting, the set I is thus necessarily an independent set. So our algorithm is also a r-approximation algorithm for MAXIMUM INDEPENDENT SET. □

MAXIMUM INDEPENDENT SET in general is **Poly-APX**-hard, meaning it is as hard as any problem that can be approximated to a polynomial factor. Lemma 1 implies that MAX-LEF is also **Poly-APX**-hard. Thus, Algorithm 1 is asymptotically optimal.

Interestingly, there are graph classes where the size of an independent set can be expressed as a fraction of n. Therefore, this fraction corresponds to the approximation ratio of Algorithm 1.

PROPOSITION 4.1. *A polynomial time 0.5-approximate algorithm for MAX-LEF exists if the social network is bipartite.*

PROOF. Suppose the social network is a bipartite graph $(N_1, N_2; E)$. By definition both N_1 and N_2 are independent sets. If $|N_1| \geq |N_2|$, then run Algorithm 1 with $I := N_1$, otherwise run Algorithm 1 with $I := N_2$. Since $2|I| \geq |N_1| + |N_2| = |N| = n$, a polynomial time 0.5-approximation is reached. □

Proposition 4.1 can be easily extended to *k-partite graphs* (whose vertex set can be partitioned into k different independent sets), leading to a polynomial time k^{-1}-approximation algorithm.

Note that if the social network admits a vertex cover C of size k, then Algorithm 1 with $I := N \setminus C$ provides a $(1 - k/n)$-approximate solution to MAX-LEF.

4.2 Optimizing degree of (non)-envy

Instead of simply counting the number of non-envious agents, we will now focus on a more subtle criterion, measuring the degree of envy among agents. This leads to the MAX-NE optimization problem (defined in section 2) which consists in minimizing the *average degree of envy* $\mathcal{E}(\mathcal{A})$ (or equivalently maximizing the *average degree of non-envy* $\mathcal{NE}(\mathcal{A}) = 1 - \mathcal{E}(\mathcal{A})$). Before describing the algorithm, we first state the following lemma (due to lack of space, the proof of this lemma is omitted).

LEMMA 2. *Let $\mathcal{U}_n$ be the uniform distribution over all matchings from n agents to n objects. Then we have* $\mathbb{E}_{\mathcal{A} \sim \mathcal{U}_n} [\mathcal{NE}(\mathcal{A})] = \frac{5}{6} - o(1)$.

This tells us that with high probability, random matchings yield high degrees of non-envy. To get a deterministic algorithm based on this idea, we apply a standard derandomization technique. In our algorithm (Algorithm 2), at each step i, agent i receives one of the remaining unallocated objects. This object is chosen so as to minimize the conditional expectation of $\mathcal{E}$ (line 4). We will show below that this conditional expectation can be computed efficiently.

Algorithm 2:

1 $\mathcal{A}_0 \leftarrow \{\}$ is an empty allocation, $S \leftarrow \{\}$ is empty set of agents
2 **for** *each* $i \in 1 \dots n$ **do**
3   **for** *each object* x *unassigned in* $\mathcal{A}_{i-1}$ **do**
4         $\mathcal{A}_i^x \leftarrow \mathcal{A}_{i-1} \cup \{agent_i \leftarrow x\}$
5         $v_x \leftarrow \mathbb{E}_{\mathcal{A} \sim \mathcal{U}_n} \left[\mathcal{E}(\mathcal{A}) : \mathcal{A}(S \cup \{i\}) = \mathcal{A}_i^x\right]$
6     $x^* \leftarrow \arg\min_x v_x$
7     $\mathcal{A}_i \leftarrow \mathcal{A}_{i-1} \cup \{agent_i \leftarrow x^*\}$
8     $S \leftarrow S \cup \{i\}$

Proposition 1. *Algorithm 2 is a polynomial-time $\frac{5}{6} - o(1)$ approximation algorithm for* max*-NE.*

Proof. First, by standard arguments of the derandomization method (similar to e.g. page 132 of [30]) together with lemma 2, this algorithm outputs an allocation $\mathcal{A}$ such that $\mathcal{NE}(\mathcal{A}) \geq \frac{5}{6} - o(1)$. By design we have $\mathcal{NE}(\mathcal{A}) \leq 1$, so the approximation ratio holds. To show that the algorithm runs in polynomial time, we need to bound the computation time of v_x. If $\mathcal{A}$ is a partial allocation, define $P(\mathcal{A}, l)$ as the set of goods that agent l can own without violating $\mathcal{A}$. For example, if $\mathcal{A}$ is a complete allocation, $P(\mathcal{A}, l) = \mathcal{A}(l)$ and if $\mathcal{A} = \{\}$, then $P(\mathcal{A}, l) = O$. First note that v_x can be calculated as a sum of conditional expectations $\frac{1}{2|E|} \sum_{\{l,h\} \in E} \mathbb{E}_{\mathcal{A} \sim \mathcal{U}_n} \left[e(\mathcal{A}, l, h) + e(\mathcal{A}, h, l) : \mathcal{A}(S \cup \{i\}) = \mathcal{A}_i^x\right]$. Next, note that for any $l, h \in N$ the expectation $\mathbb{E}_{\mathcal{A} \sim \mathcal{U}_n}[e(\mathcal{A}, l, h) : \mathcal{A}(S \cup \{i\}) = \mathcal{A}_i^x]$ is equal to $\frac{1}{|Z_{l,h}|.(n-1)} \sum_{(a,b) \in Z_{l,h}} \max(0, r_l(a) - r_l(b))$ where $Z_{l,h} = \left\{(a, b) \in P(l, \mathcal{A}_i^x) \times P(h, \mathcal{A}_i^x) : a \neq b\right\}$. The computation of v_x can thus be done in $O(n^4)$. □

5 LOCATION AND ALLOCATION

This section is dedicated to dec-location-LEF. The following theorem shows that this problem is computationally challenging.

Theorem 5.1. *dec-location-LEF is* **NP***-complete.*

Proof. The reduction is from problem independent set which is **NP**-complete [21] and can be defined as follows. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a positive integer $k \leq |\mathcal{V}|$, is there an independent set $I \subseteq \mathcal{V}$ of size k?

Let s denote the size of $\mathcal{V}$, and $\mathcal{V} = \{v_1, \cdots, v_s\}$. We construct an instance of dec-location-LEF as follows. The set of objects is $O = Q \cup T$, where $Q = \{q_1, \cdots, q_{s-k}\}$ and $T = \{t_1, \cdots, t_k\}$. The set of agents is $N = \{X_1, \dots, X_{s-k}\} \cup \{L_1, \dots, L_k\}$. Let Q_{-i} denote the set $Q \setminus \{q_i\}$, and let $Q_{-i}^{\uparrow}$, $Q^{\uparrow}$ and $T^{\uparrow}$ denote partial orders over Q_{-i}, Q and T, respectively, where objects are ranked by increasing order of indices. Preferences are as follows:

- $X_i : q_i \succ Q_{-i}^{\uparrow} \succ T^{\uparrow}$
- $L_j : T^{\uparrow} \succ Q^{\uparrow}$

Finally, the social network is $G = \mathcal{G} = (\mathcal{V}, \mathcal{E})$.

We claim that the instance of dec-location-LEF is a yes-instance if, and only if, $\mathcal{G}$ contains an independent set of size k.

Assume that I is an independent set of size k in $\mathcal{G}$. We can assume without loss of generality that $I = \{v_1, \dots, v_k\}$. We construct $\mathcal{A}$ and $\mathcal{L}$ as follows. If $v_i \in I$ then $\mathcal{L}(L_i) = v_i$ and $\mathcal{A}(L_i) = t_i$. Otherwise, agents are placed arbitrarily on G and receive their best item ($\mathcal{A}(X_i) = q_i$). It is easy to check that $\mathcal{A}$ is LEF with respect to $\mathcal{L}$ since no two vertices $\mathcal{L}(L_i)$ and $\mathcal{L}(L_j)$ are neighbors in G.

Assume now that there exists $\mathcal{L}$ such that $\mathcal{A}$ is LEF w.r.t. $\mathcal{L}$. If $\mathcal{L}(L_i)$ and $\mathcal{L}(L_j)$ are neighbors in G then either $\mathcal{A}(L_i) \succ_{L_j} \mathcal{A}(L_j)$ or $\mathcal{A}(L_j) \succ_{L_i} \mathcal{A}(L_i)$ holds since L_i and L_j have the same preferences, leading to a contradiction with $\mathcal{A}$ LEF. Hence, $\{\mathcal{L}(L_1), \dots, \mathcal{L}(L_k)\}$ forms an independent set of size k in $G = \mathcal{G}$. □

Interestingly, the above reduction also holds when $\mathcal{A}$ is fixed, i.e. the allocation of objects to agents is imposed by the problem.

We shall extend the polynomial time result obtained for dec-LEF on social networks of degree at least $n-2$. Note that for dec-location-LEF, vertices of degree $n-1$ cannot be ignored, since we need to decide which agents they accommodate.

Observation 2 implies that two agents having the same top object must be neighbors in $\overline{G}$, otherwise one of them must be envious. Therefore, one can focus on $\mathbb{L}_{\succ}$, defined as the set of location functions such that each pair of agents having the same top object are neighbors in $\overline{G}$ (or equivalently, not neighbors in G).

If an instance contains three (or more) agents with the same top object then it must be a no-instance since each vertex in $\overline{G}$ has degree at most 1. The following lemma shows that the location functions of $\mathbb{L}_{\succ}$ are all equivalent for the search of an LEF allocation.

Lemma 5.2. *If $\mathcal{A}$ is an LEF allocation for some $\mathcal{L}$, and $\mathcal{A}$ cannot be improved by a swap between two agents without violating the LEF condition, then $\mathcal{A}$ is also LEF for any location function of$\mathbb{L}_{\succ}$.*

Proof. First of all, $\mathcal{L}$ must belong to $\mathbb{L}_{\succ}$ for $\mathcal{A}$ to be LEF. Let $\mathcal{L}'$ be a function of $\mathbb{L}_{\succ}$. It is easy to check that any pair of agents having the same top object have the same set of neighbors in G for both $\mathcal{L}$ and $\mathcal{L}'$. Therefore, if $\mathcal{A}$ is LEF for these agents under $\mathcal{L}$, then $\mathcal{A}$ is also LEF for these agents under $\mathcal{L}'$.

Let i be an agent who solely ranks some object o at the first position in her preferences. On one hand, if $\mathcal{L}(i)$ is a vertex of degree $n-1$ then Observation 2 implies that she must receive o. On the other hand, if $\mathcal{L}(i)$ is a vertex of degree $n-2$ and j is the unique neighbor of i in $\overline{G}$ then Observation 2 implies that o is assigned either to i or to j. But j must also be the unique agent to have some object o' ranked first in her preferences, where $o \neq o'$. Therefore, either agent i or j must receive o'. Since by hypothesis $\mathcal{A}$ cannot be improved by swapping objects o and o', o must be assigned to i and o' must be assigned to j. In all, agent i must receive her top object in $\mathcal{A}$. Therefore, $\mathcal{A}$ is also LEF for agent i. □

In order to solve dec-location-LEF, one can compute a function $\mathcal{L}$ of $\mathbb{L}_{\succ}$ by assigning the agents having the same top object to vertices connected in $\overline{G}$, and by assigning the other agents arbitrarily. Once $\mathcal{L}$ is fixed, one can use the algorithm presented in Theorem 3.3 to compute an LEF allocation if such an allocation exists. If an

LEF allocation is returned then the algorithm returns $\mathcal{L}$ and $\mathcal{A}$. Otherwise, we know by Lemma 5.2 that no function in $\mathbb{L}_{>}$ can lead to an LEF allocation, and the algorithm returns *false*. This algorithm clearly runs in polynomial time, leading to the following theorem:

THEOREM 5.3. *DEC-LOCATION-LEF in graphs of minimum degree $n-2$ is solvable in polynomial time.*

6 EXPERIMENTS

In this section we generate random instances of our decision and optimization problems and we solve these instances exactly using mixed integer linear program formulations. We build on the ones proposed by [16] (which address envy-freeness and the minimization of maximum pairwise envy among any two agents [26], in a context of additive utilities with several goods per agent). To fit our setting, we adapt it so as to account for graph constraints. We further design three variants, two where the objective functions correspond to MAX-LEF and MAX-NE, and another one where the locations of agents on the graph are treated as decision variables, to address the more challenging DEC-LOCATION-LEF.

For these experiments, we generate random k-regular graphs with 8 vertices for k ranging from 1 to 7. Agents' preferences are randomly drawn from impartial culture. Table 2 shows the results (averaged over 1000 runs). Note that MAX-LEF results are given in terms of the number of envious agents.

A natural question is how the likelihood to find an LEF evolves as the degree of the graph augments. It must clearly decrease (in the extreme case of a complete graph, recall that all agents must have a different preferred item, which occurs with a probability as low as $n!/n^n$). The question is how this drop will occur. Our experiments show that this decrease is sharp, and from a degree equal to half of the agents, it actually becomes highly unlikely to find an LEF. On the other hand, for graphs of small degrees, it is often the case that an LEF can be found, and, as expected, it becomes even more so as the number of agents and items augments. Further experiments on a higher number of agents confirm this. As a rule of thumb, this means for instance that from 20 agents, it is almost always possible to find an LEF on a grid-like network.

Degree	1	2	3	4	5	6	7
LEF	1	0.72	0.22	0.05	0.02	<0.01	<0.01
max-LEF	0	0.28	0.93	1.52	1.95	2.44	2.78
max-NE	1	0.99	0.99	0.99	0.98	0.98	0.98
MMPE	0	0.28	0.83	1.19	1.42	1.69	1.91
Loc-LEF	1	1	1	0.92	0.49	0.07	<0.01

Table 2: 8 agents, graphs of regular degree

The ability to allocate agents on the network gives the central authority some extra-power when it comes to find an LEF. However, note that this power heavily depends on the structure of the graph (for instance, it is useless when the graph is complete, as all the different ways to label the graph with agents are isomorphic). Table 2 shows that this power can be significant: the likelihood to find an LEF remains above 90% until degree 4, while it was as low as 5.5% in the basic problem.

We also report in Table 2 results regarding the measures we optimize (as well as the "classical" minimization of maximum pairwise envy (MMPE) of [26], which in our context can be interpreted as minimizing the maximum number of agents envied by any agent). Note in particular that even with a complete graph, it is on average possible to allocate items so as to make envious only about a third of the agents, and that no agent envies more than two other agents in our instance with 8 agents.

7 FUTURE WORK

We have studied different aspects of local envy-freeness in house allocation settings. There are several interesting future directions to explore. We give below some preliminary thoughts on the ones we find the most stimulating.

Constraints on the allocation. Two other relevant challenges related to DEC-LEF are: Given a *partial* allocation of the objects, can a full LEF allocation be found? Given some forbidden object-agent pairs, can an LEF allocation be found?

Oriented graphs. A natural extension of DEC-LEF is to consider a social network modeled with a directed graph. An arc (u, v) indicates that u possibly envies v, but it does not indicate that v possibly envies u, unless the arc (v, u) is also present. In this directed case, an allocation $\mathcal{A}$ is said to be LEF if $\mathcal{A}(j) \not\succ_i \mathcal{A}(i)$ for every arc (i, j). It is not difficult to see that DEC-LEF is NP-complete in this directed case (use the proof of Theorem 3.1 where each edge $\{u, v\}$ is replaced by the arcs (u, v) and (v, u)). Interestingly, the directed variant of DEC-LEF can be solved efficiently in *directed acyclic graphs* (DAGs). Indeed, if the social network is a DAG, then an LEF allocation must exist, and it can be computed in polynomial time. In fact, a DAG has at least one *source*, i.e. a vertex with indegree 0. If a source of a DAG is deleted, then we get a (possibly empty) DAG. The algorithm computing an LEF allocation works as follows: while the social network is non empty, find a source s, allocate s her most preferred object $o_s \in O$, remove o_s from O, and delete s. The algorithm also guarantees a Pareto-optimal allocation and mimics a serial dictatorship [29]. Note that DAGs actually characterize exactly those graphs guaranteeing LEF to exist (if a cycle exists, simply set the preferences of all agents to be exactly the same within the cycle to get a no-instance). But this leaves other interesting questions open: for instance, are there other natural classes of graphs admitting polynomial time algorithms for DEC-LEF in oriented graphs?

Domain restrictions. There is a long tradition in social choice to consider domain restrictions on agents' preferences to obtain positive results. This would be natural to study our setting under such assumptions. For example, we can fix the number of different rankings. To take a concrete question, how difficult DEC-LEF and DEC-LOCATION-LEF are when there are only two categories of agents: those with ranking $\succ_1$ on the objects and those with ranking $\succ_2$? More generally, can well-known domain restrictions, such as single-peakedness, be useful? Since the relevance of this domain restriction in the context of house allocation has recently been emphasized [7, 14], this might be an interesting road to pursue.

ACKNOWLEDGMENTS

This work is partially supported by the ANR project 14-CE24-0007-01 - CoCoRICo-CoDec.

REFERENCES

- [1] Atila Abdulkadiroğlu and Tayfun Sönmez. 1998. Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems. *Econometrica* 66, 3 (1998), 689–701.
- [2] Rediet Abebe, Jon Kleinberg, and David C. Parkes. 2017. Fair Division via Social Comparison. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-17)*. ACM, São Paulo, Brazil, 281–289.
- [3] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. 1979. A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas. *Inform. Process. Lett.* 8, 3 (1979), 121–123.
- [4] Haris Aziz, Sylvain Bouveret, Ioannis Caragiannis, Ira Giagkousi, and Jérôme Lang. 2018. Knowledge, Fairness, and Social Constraints. In *Proceedings of the 32nd AAAI conference on Artificial Intelligence (AAAI'18)*. AAAI Press, New Orleans, Louisiana, USA, –.
- [5] Haris Aziz, Jens L. Hougaard, Juan D. Moreno-Ternero, and Lars P. Østerdal. 2017. Computational Aspects of Assigning Agents to a Line. *Mathematical Social Sciences* 90 (2017), 93–99.
- [6] Haris Aziz, Ildikó Schlotter, and Toby Walsh. 2016. Control of Fair Division. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. IJCAI/AAAI Press, 67–73.
- [7] Sophie Bade. 2017. Matching with Single-Peaked Preferences. Technical report. (2017).
- [8] 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)*. ijcai.org, Melbourne, Australia, 3632–3638. https://doi.org/10.24963/ijcai.2017/508
- [9] 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)*. ijcai.org, Melbourne, Australia, 135–141. https://doi.org/10.24963/ijcai.2017/20
- [10] 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 12, 284–310. http://recherche.noiraudes.net/en/handbook.php
- [11] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. 2009. On Low-Envy Truthful Allocations. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT-2009)*. Springer, Venice, Italy, 111–119.
- [12] Susumu Cato. 2010. Local strict envy-freeness in large economies. *Mathematical Social Sciences* 59, 3 (2010), 319 – 322. https://doi.org/10.1016/j.mathsocsci.2010.01.002
- [13] Yann Chevaleyre, Ulle Endriss, and Nicolas Maudet. 2017. Distributed Fair allocation of Indivisible Goods. *Artificial Intelligence* 242 (2017), 1–22.
- [14] Anastasia Damamme, Aurélie Beynier, Yann Chevaleyre, and Nicolas Maudet. 2015. The Power of Swap Deals in Distributed Resource Allocation. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-15)*. ACM, Istanbul, Turkey, 625–633.
- [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 *Algorithmic Decision Theory, First International Conference, ADT 2009, Venice, Italy, October 20-23, 2009. Proceedings*. Springer, 98–110.
- [16] John P. Dickerson, Jonathan R. Goldman, Jeremy Karp, Ariel D. Procaccia, and Tuomas Sandholm. 2014. The Computational Rise and Fall of Fairness. In *Proceedings of the 28th Conference on Artificial Intelligence (AAAI-14)*. AAAI Press, Québec City, Québec, Canada, 1405–1411.
- [17] M. Feldman, K. Lai, and L. Zhang. 2009. The Proportional-Share Allocation Market for Computational Resources. *IEEE Transactions on Parallel and Distributed Systems* 20, 8 (Aug 2009), 1075–1088. https://doi.org/10.1109/TPDS.2008.168
- [18] Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. 2009. On the Parameterized Complexity of Multiple-Interval Graph Problems. *Theoretical Computer Science* 410, 1 (2009), 53–61.
- [19] Michele Flammini, Manuel Mauro, and Matteo Tonell. 2018. On Social Envy-Freeness in Multi-Unit Market. In *Proceedings of the 32nd AAAI conference on Artificial Intelligence (AAAI'18)*. AAAI Press, New Orleans, Louisiana, USA, –.
- [20] Duncan K. Foley. 1967. Resource Allocation and the Public Sector. *Yale Economic Essays* 7, 1 (1967), 45–98.
- [21] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- [22] Laurent Gourvès, 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)*. ijcai.org, Melbourne, Australia, 213–219. https://doi.org/10.24963/ijcai.2017/31
- [23] Aanund Hylland and Richard Zeckhauser. 1979. The Efficient Allocation of Individuals to Positions. *Journal of Political Economy* 87, 2 (1979), 293–314.
- [24] Jon M. Kleinberg and Éva Tardos. 2006. *Algorithm Design*. Addison-Wesley.
- [25] Jan Kratochvíl, Petr Savický, and Zsolt Tuza. 1993. One More Occurrence of Variables Makes Satisfiability Jump From Trivial to NP-Complete. *SIAM J. Comput.* 22 (1993), 203–210.
- [26] 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)*. ACM, New York, NY, USA, 125–131.
- [27] Trung Thanh Nguyen and Jörg Rothe. 2014. Minimizing Envy and Maximizing Average Nash Social Welfare in the Allocation of Indivisible Goods. *Discrete Applied Mathematics* 179 (2014), 54–68.
- [28] Vangelis Th. Paschos. 1992. A (Delta/2)-Approximation Algorithm for the Maximum Independent Set Problem. *Inform. Process. Lett.* 44, 1 (1992), 11–13.
- [29] Lars-Gunnar Svensson. 1999. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare* 16, 4 (1999), 557–567.
- [30] Vijay V. Vazirani. 2001. *Approximation Algorithms*. Springer.
- [31] Lin Zhou. 1990. On a Conjecture by Gale about One-Sided Matching Problems. *Journal of Economic Theory* 52, 1 (1990), 123 – 135.