Autonomous Agents and Multi-Agent Systems (2024) 38:42
https://doi.org/10.1007/s10458-024-09672-7

Graphical house allocation with identical valuations

Hadi Hosseini[1] · Andrew McGregor[2] · Justin Payan[2] · Rik Sengupta[2,3] · Rohit Vaish[4] · Vignesh Viswanathan[2]

Accepted: 8 August 2024 / Published online: 28 August 2024

Abstract
The classical house allocation problem involves assigning *n* houses (or items) to *n* agents according to their preferences. A key criterion in such problems is satisfying some fairness constraints such as envy-freeness. We consider a generalization of this problem, called Graphical House Allocation, wherein the agents are placed along the vertices of a graph (corresponding to a social network), and each agent can only experience envy towards its neighbors. Our goal is to minimize the *aggregate* envy among the agents as a natural fairness objective, i.e., the sum of the envy value over all edges in a social graph. We focus on graphical house allocation with identical valuations. When agents have identical and *evenly-spaced* valuations, our problem reduces to the well-studied Minimum Linear Arrangement. For identical valuations with possibly uneven spacing, we show a number of deep and surprising ways in which our setting is a departure from this classical problem. More broadly, we contribute several structural and computational results for various classes of graphs, including NP-hardness results for disjoint unions of paths, cycles, stars, cliques, and complete bipartite graphs; we also obtain fixed-parameter tractable (and, in some cases, polynomial-time) algorithms for paths, cycles, stars, cliques, complete bipartite graphs, and their disjoint unions. Additionally, a conceptual contribution of our work is the formulation of a structural property for disconnected graphs that we call *splittability*, which results in efficient parameterized algorithms for finding optimal allocations.

Keywords Fair allocation · House allocation · Envy minimization · Local envy

1 Introduction

The house allocation problem has attracted interest from the computer science and multiagent systems communities for a long time. The classical problem deals with assigning a set of *n* houses to *n* agents with (possibly) different valuations over the houses. It is often desirable to find assignments that satisfy some economic property of interest. In this work, we focus on the well-motivated economic notion of *fairness*, and in particular, study the objective of minimizing the *aggregate envy* among the agents.

A preliminary version of this paper appeared in the proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) [1].

Extended author information available on the last page of the article

Despite the historical interest in this problem, to the best of our knowledge, the house allocation problem has not been studied thoroughly over *graphs*, a setting in which the agents are placed on the vertices of an undirected graph G and each agent's potential envy is only towards its neighbors in G, with only a few exceptions [2, 3].

Incorporating the social structure over a graph enables us to capture the underlying restrictions of dealing with partial information, which is representative of constraints in many real-world applications. Thus, the classical house allocation problem is the special case of our problem when the underlying graph is complete.

Our work is in line with recent literature on examining various problems in computational social choice on social networks, including voting [4–6], fair division [7, 8], and hedonic games [9, 10]. By focusing on graphs, we aim to gain insights into how the structure of the social network impacts fairness in house allocation. We focus on *identical* valuation functions and show that even under this seemingly strong restriction, the problem is computationally hard, yet structurally rich. We provide a series of observations and insights about graph structures that help identify, and in some cases overcome, these computational bottlenecks.

1.1 Overview and our contributions

We assume that the agents are placed at the vertices of a graph representing a social network, and that they have identical valuation functions over the houses. Our objective is to find an allocation of the houses among the agents to minimize the total envy in the graph. We call this the problem of Graphical House Allocation with Identical Valuations, or Graphical House Allocation for short, where an instance of the problem consists of the underlying graph together with the set of values for any of the (identical) valuation functions.

This is a beautiful combinatorial problem in its own right, as it can be restated as the problem where, given an undirected graph and a multiset of nonnegative numbers, the numbers need to be placed on the vertices in a way that minimizes the sum of the edgewise absolute differences.

In Sect. 2, we present the formal model and set up some preliminaries.

In Sect. 3, we present computational lower bounds and inapproximability results for the problem, even for very simple graphs. We start by establishing the connection between the graphical house allocation problem and the Minimum Linear Arrangement problem, which has several notable similarities and differences. We show NP-hardness for the Graphical House Allocation problem, even when the graph is a disjoint union of paths, cycles, stars, or cliques, which all have known polynomial-time algorithms for linear arrangements.

In Sect. 4, we focus on connected graphs and completely characterize optimal allocations when the graph is a path, cycle, star, or a complete bipartite graph. We also prove a technically involved structural result for binary trees.

In Sect. 5, we focus on disconnected graphs, starting with a fundamental difference between graphical house allocation and linear arrangements, motivating our definitions of *splittable*, *strongly splittable*, and *unsplittable*[1] disconnected graphs. We employ these

[1] These terms were initially *separable*, *strongly separable*, and *inseparable* respectively in the conference version [1]. They have been changed subsequently to avoid ambiguity with other standard definitions of separability, as (different) graph theoretic properties; see Hammer and Maffray [11], for example.

characterizations to prove algorithmic results for a variety of graphs. In particular, we show that disjoint unions of paths, cycles, stars, and equal-sized cliques are strongly splittable and develop natural fixed parameter tractable (FPT) algorithms for these graphs.[2] Moreover, we show that disjoint unions of arbitrary cliques, as well as "balanced" complete bipartite graphs, satisfy splittability (but not strong splittability) and admit XP algorithms.[3]

Finally, in Sect. 6, we wrap up with a concluding discussion.

In the interest of readability, we defer the details of the more involved proofs to the appendix.

1.2 Related work

House allocation has been traditionally studied in the economics literature under the *housing market* model, where agents enter the market with a house (or an endowment) each and are allowed to engage in cyclic exchanges [12]. This model has found important practical applications, most notably in kidney exchange [13, 14].

While the initial work on house allocation focused on the economic notions of core and strategyproofness [15], subsequent work has explored *fairness* issues. [16] study the house allocation problem under ordinal preferences (specifically, weak rankings) and provide a polynomial-time algorithm for determining the existence of an envy-free allocation. By contrast, the problem becomes NP-hard when agents' preferences are specified as a set of pairwise comparisons [17]. Kamiyama et al. [18] study house allocation under cardinal preferences (similar to our work) and examine the complexity of finding a "fair" assignment for various notions of fairness such as proportionality, equitability, and maximizing the number of envy-free agents (they do not consider aggregate envy). They show that the latter problem is hard to approximate under general valuations, and remains NP-hard even for the restricted case of binary valuations. Madathil et al. [19] similarly study various notions of envy minimization and show that these problems are intractable for most classes of binary, cardinal and ordinal valuations. For binary valuations, the problem of finding the largest envy-free partial matching has also been studied [20]. Gross-Humbert et al. [21] introduce a notion of group envy-freeness for house allocation and present an algorithm to approximate this notion. Aziz et al. [22] study the computation of envy free allocations when agent preferences are uncertain. Choo et al. [23] study the minimum subsidy required to ensure envy-freeness in house allocation and its computational aspects. It is worth noting that many of these works [16–23] assume there are more houses than agents. On the other hand, in our work, we assume the number of houses is equal to the number of agents.

Recent studies have considered *graphical* aspects of house allocation (similar to our work), though with different objectives. For a comprehensive review of work on fairness objectives on graphs and other structured sets, we refer the reader to the survey by Biswas et al. [24]. For instance, Massand and Simon [2] consider house allocation under externalities and study various kinds of stability-based objectives. Beynier et al. [3] study *local envy-freeness* in house allocation, which entails checking the existence of an allocation with no envy along any edge of the graph. Their work is close to ours, but their model

[2] FPT is the class of problems solvable in time $O(f(k) \cdot \text{poly}(n))$, where n is the input length, k is a given parameter, and $f(\cdot)$ is a computable function.

[3] XP is the class of problems solvable in time $O(n^{f(k)})$, where n is the input length, k is a given parameter, and $f(\cdot)$ is a computable function.

involves agents with *distinct* ordinal preferences (as opposed to identical preferences). The problem under distinct preferences turns out to be computationally intractable even for simple graph structures like paths and matchings [3].

A recent follow-up work studies approximation algorithms for the Graphical House Allocation problem with identical valuations, and provides tight bounds on the approximability of the aggregate envy objective for many classes of graphs [25].

There is also a growing literature on fair allocation of indivisible objects among agents who are part of a social network. Bredereck et al. [8] study the computational complexity of finding locally envy-free allocations when agents correspond to nodes in a directed graph. This is very similar to our work with the key difference being that they study the setting where each agent can receive multiple items. They present fixed-parameter tractability results, mainly parameterized by the number of agents, though they leave results using graph structure to future work. Eiben et al. [26] extend these results, showing a number of parameterized complexity results relating the treewidth, cliquewidth, number of agent types, and number of item types to the complexity of determining if an envy-free allocation exists on a graph. It is worth noting that this line of work focuses on deciding if envy-free allocations exist, rather than minimizing envy.

Other works seek to obtain envy-free allocations, maximum welfare allocations, or other objectives by swapping objects along a graphical structure [27–30]. In particular, Gourvès et al. [29] studies the house allocation problem as well; the key difference being that their work considers global objectives (like Pareto efficiency) which need to be reached via swaps between neighbors in a graph, whereas our work studies local envy-freeness which is defined by a graphical structure.

House allocation has also been studied in the setting where the houses are nodes on a graph (representing the neighborhood the houses are in) [31]. In that work, agents have preferences not only over the houses, but also over their neighboring agents in the graph according to the computed allocation.

2 Preliminaries

For any natural number $t \geq 1$, we use $[t]$ to denote the set $\{1, 2, \dots, t\}$. There is a set of n *agents* $N = [n]$ and n *houses* $H = \{h_1, h_2, \dots, h_n\}$ (often called *items*). Each agent i has a *valuation function* $v_i : H \to \mathbb{R}_{\geq 0}$, where $v_i(h)$ indicates agent i's value for house $h \in H$. An *allocation* π is a bijective mapping from agents to houses. For each $i \in N$, $\pi(i)$ is the house allocated to agent i under the allocation π, and $v_i(\pi(i))$ is its utility.

Given N, H, and $\{v_i\}_{i \in N}$, our goal is to output an allocation π that is "fair" to all the agents, for some reasonable definition of fairness. A natural way to define fairness is using *envy*. An agent i is said to envy agent j under allocation π if $v_i(\pi(i)) < v_i(\pi(j))$. While we would ideally like to find *envy-free* allocations, this may not always be possible — consider a simple example with two agents and two houses, but (exactly) one of the houses is valued at 0 by both agents. Therefore, we instead focus on the magnitude of envy that agent i has towards agent j, for a fixed allocation π. This is defined as $envy_\pi(i,j) := \max\{v_i(\pi(j)) - v_i(\pi(i)), 0\}$.

We define an undirected graph $G = (N, E)$ over the set of agents, which represents the underlying social network. Our goal is to compute an allocation that minimizes the *total envy* along the edges of the graph, defined as $\mathsf{Envy}(\pi, G) := \sum_{(i,j) \in E} (envy_\pi(i,j) + envy_\pi(j,i))$;

note that edges are undirected. An allocation that minimizes the total envy is referred to as a *minimum envy allocation*. The minimum envy allocation may not be unique.

When the graph G is a complete graph K_n, a minimum envy allocation can be computed in polynomial time by means of a reduction to a bipartite minimum-weight matching problem. We prove this formally below.

Proposition 2.1 *When G is the complete graph K_n, and agents have arbitrary (non-identical) valuation functions, a minimum envy allocation can be computed in polynomial time.*

Proof Given N, H, and $\{v_i\}_{i\in N}$, we construct a weighted bipartite graph $\widehat{G}$. The constructed instance $\widehat{G}$ is a complete bipartite graph with bipartition (N, H), with edge weights as follows: for $i \in N$ and $h \in H$, the edge (i, h) has weight $\sum_{h'\in H\setminus h} \max\{v_i(h') - v_i(h), 0\}$.

Perfect matchings in $\widehat{G}$ correspond bijectively to allocations π. In fact, if a matching in $\widehat{G}$ corresponds to an allocation π, then the weight of the matching is $\sum_{i\in N} \sum_{h'\in H\setminus\pi(i)} \max\{v_i(h') - v_i(\pi(i)), 0\}$, which equals $\mathsf{Envy}(\pi, G)$. Therefore, computing a minimum envy allocation is equivalent to computing a minimum weight matching in $\widehat{G}$. It is well-known that this can be done in polynomial time [32]. □

Unfortunately, Proposition 2.1 cannot generalize much beyond complete graphs. It is known that for several other simple graphs like paths and matchings, computing a minimum envy allocation is NP-complete [3]. Given this computational intractability, we therefore explore a natural restriction of the problem, when all agents have identical valuations, to gain insights into the computational and structural aspects of fairness in social networks. We call this the Graphical House Allocation with Identical Valuations problem, or Graphical House Allocation for short. Formally, an instance of Graphical House Allocation consists of a set N of agents, a set H of houses, an undirected graph $G = (N, E)$, and a fixed valuation function $v : H \to \mathbb{R}_{\geq 0}$, that represents the common valuation function for all agents in N. Identical valuations capture a natural aspect of real-world housing markets, where the house prices are independent of agents.

When all agents have the same valuation function v, the total envy of an allocation π along the edges of a graph $G = (N, E)$ can be written as

$$\mathsf{Envy}(\pi, G) := \sum_{(i,j)\in E} |v(\pi(i)) - v(\pi(j))| \ .$$

This formulation also yields a new expression for envy along an edge $e = (i,j) \in E$ as $envy_\pi(e) = |v(\pi(i)) - v(\pi(j))|$. This value equals $envy_\pi(i,j) + envy_\pi(j,i)$, as one of those terms is zero under identical valuations.

When G is K_n, under identical valuations, an optimal allocation is trivially computable, as all allocations are equivalent.

For the rest of this paper, we will assume without loss of generality that the house values are all distinct unless stated otherwise (refer to Lemma A.1 in Appendix A for a formal justification). In particular, every agent's valuation function (denoted by v) gives each house a unique nonnegative value, with $v(h_1) < v(h_2) < \ldots < v(h_n)$. We will say $h_1 \prec h_2$ to mean $v(h_1) < v(h_2)$.

For an allocation π and a subset $N' \subseteq N$, we will refer to the set of houses received by N' as $\pi(N')$. If G' is a subgraph of G, we will use $\pi(G')$ in the same way. For graphs G_1 and G_2, we will use $G_1 + G_2$ to mean the disjoint union of G_1 and G_2.

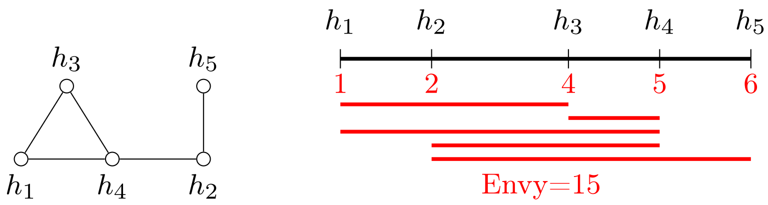


Fig. 1 (Left) A graph G on five agents along with a particular allocation π. The valuations are identical and are given by $\mathbf{v} = (1, 2, 4, 5, 6)$. (Right) The valuation interval is shown via the thick horizontal line in black. The five line segments in red denote the envy along the five edges of the graph G. The total length of these line segments is $\mathsf{Envy}(\pi, G) = 15$ (Color figure online)

Definition 2.2 For an instance of Graphical House Allocation, the *valuation interval* is defined as the closed interval $[v(h_1), v(h_n)] \subset \mathbb{R}_{\geq 0}$ with each $v(h_k)$ marked.

The motivation for Definition 2.2 is as follows. For an arbitrary allocation π, for each edge $e = (i,j) \in E$, we can draw a line segment from $v(\pi(i))$ to $v(\pi(j))$. This line segment has length $|v(\pi(i)) - v(\pi(j))| = envy_\pi(e)$. It follows that $\mathsf{Envy}(\pi, G)$ is the sum of the lengths of all such line segments. An optimal allocation π^* is any allocation that attains this minimum sum. See Fig. 1 for an example of a valuation interval, together with a graph G, and a particular allocation on G depicted under the valuation interval.

A subset of houses $H' = \{h_{i_1}, \dots, h_{i_k}\} \subseteq H$ with $h_{i_1} \prec \dots \prec h_{i_k}$ is called *contiguous* if there is no house $h' \in H \setminus H'$ with $h_{i_1} \prec h' \prec h_{i_k}$. Pictorially, the values in H' form an uninterrupted sub-interval of the valuation interval, with no value outside of H' appearing inside that sub-interval. In Fig. 1, the subsets $\{h_1, h_2, h_3\}$ and $\{h_5\}$ are contiguous, whereas the subsets $\{h_1, h_2, h_5\}$ and $\{h_3, h_5\}$ are not. Note that a subset of houses is contiguous if and only if it contains houses with only consecutive indices, i.e., $\{h_1, h_2\}$ is contiguous but $\{h_1, h_3\}$ is not.

We will often interchangeably talk about allocating h_i and allocating $v(h_i)$ to an agent, and we will also sometimes refer to houses as being marked points on the valuation interval. We close this section with the following useful definition.

Definition 2.3 Let (N, H, G, v) be an instance of the Graphical House Allocation problem and let π be any allocation for this instance. For any $S \subseteq N$ and $x \in \mathbb{R}$, we define $n^<_{S,\pi}(x)$ as the number of agents in S who are allocated a house with a value less than x in the allocation π. We define $n^>_{S,\pi}(x)$ similarly.

3 Hardness and lower bounds

In this section, we prove hardness results for the Graphical House Allocation problem. This section is divided into three subsections; in each of these subsections, we show hardness for a restricted version of the problem by either reducing to problems in graph theory (Sects. 3.1 and 3.2) or by reducing to the classic bin packing problem (Sect. 3.3).

3.1 Connection to the linear arrangement problem

The Minimum Linear Arrangement problem is the problem where, given an undirected n-vertex graph $G = (V, E)$, we want to find a bijective function $\pi : V \to [n]$ that minimizes $\sum_{(i,j)\in E} |\pi(i) - \pi(j)|$. Minimum Linear Arrangement is a special case of Graphical House Allocation where the valuation interval has evenly spaced values (or in other words, $\mathsf{image}(v) = [n]$ and v is one-to-one).

For specific graphs like paths, stars, and trees, Minimum Linear Arrangement can be solved in polynomial time [33]. However, finding a minimum linear arrangement is NP-hard for general graphs [34], with a best known run-time of $O(2^n m)$, where $|V| = n$ and $|E| = m$, using a dynamic programming algorithm [35]. The problem remains NP-hard even for bipartite graphs [36]. Both of these hardness results immediately extend to the Graphical House Allocation problem.

We also find that optimal arrangements satisfy properties that optimal house allocations are not guaranteed to satisfy. For example, we may assume without loss of generality that the underlying graph G in any instance of Minimum Linear Arrangement is connected. This is a consequence of the following observation.

Proposition 3.1 (Seidvasser [37]) *If G is any Minimum Linear Arrangement instance, then some optimal solution assigns a contiguous subset of $[n]$ to each connected component of G.*

Proof Consider an optimal solution π that does not assign contiguous subsets of $[n]$ to the connected components of G. Order the components of G in any arbitrary order, say $G_1 \cup \ldots \cup G_k$, where $G_i = (V_i, E_i)$ has n_i vertices, for each $i \in [k]$. Consider the allocation π', obtained by assigning the first n_1 values in $[n]$ to G_1, the next n_2 values to G_2, and so on, satisfying the constraint that for all intra-component pairs $j, j' \in G_i$, we have $\pi'(v_j) < \pi'(v_{j'})$ if and only if $\pi(v_j) < \pi(v_{j'})$. Note that π' is well-defined. Now, consider any edge $e = (u, v)$ of G, and say without loss of generality that $\pi(v) > \pi(u)$ (and therefore, $\pi'(v) > \pi'(u)$). The envy along this edge e in π' is the length $\pi'(v) - \pi'(u) = 1 + |\{w : \pi'(u) < \pi'(w) < \pi'(v)\}|$. But note that any vertex $w \in V$ satisfying $\pi'(u) < \pi'(w) < \pi'(v)$ must also satisfy $\pi(u) < \pi(w) < \pi(v)$ (the converse need not be true for all w). This means that the number of values falling between $\pi'(u)$ and $\pi'(v)$ cannot increase from π to π', i.e.,

$$\begin{aligned}\pi'(v) - \pi'(u) &= 1 + |\{w : \pi'(u) < \pi'(w) < \pi'(v)\}| \\ &\le 1 + |\{w : \pi(u) < \pi(w) < \pi(v)\}| \\ &= \pi(v) - \pi(u)\ .\end{aligned}$$

It follows that the edge e incurs at most as much envy under π' as it does under π. Since this is true for all edges e, it follows that π' incurs at most as much envy as π does. Therefore, π' is also optimal, and it is an allocation that assigns contiguous subsets of $[n]$ to the connected components of G. □

We will see in Sect. 5 that in the Graphical House Allocation problem, this property no longer holds; that is, it no longer suffices to only consider connected graphs. We use this property (or lack thereof) to establish a separation between the two problems. Specifically, we show that, when the graph is a disjoint union of paths (or cycles or stars), the optimal linear arrangement can be trivially found in linear time, but finding the optimal house allocation is NP-hard.

3.2 Connection to the minimum bisection problem

Our problem is NP-complete (for arbitrary graphs and valuations), because the special case of minimum linear arrangements is already NP-complete, as stated in Sect. 3.1. We next provide a different NP-completeness proof that uses a reduction from the Minimum Bisection problem. This hardness proof immediately implies inapproximability of the problem on general graphs.

Definition 3.2 The Minimum Bisection *problem* asks, for an n-vertex graph G and a natural number k, if there is a partition of $V(G)$ into two parts of size $n/2$, with at most k edges crossing the cut.

The Minimum Bisection problem is a known NP-complete problem [34, 38, 39]. Furthermore, it is also known to be hard to approximate efficiently, a fact that is useful in light of the following observation.

Theorem 3.3 *There is a polynomial-time reduction from the Minimum Bisection problem to the Graphical House Allocation problem with identical valuations.*

Proof In the decision version of the Minimum Bisection problem, we are given an instance $\langle G, k\rangle$, and we ask if there is a bisection of G with k or fewer edges crossing the cut. Given such an instance, we construct an instance of the Graphical House Allocation problem as follows. We use the same graph G, and our valuation interval has a cluster of $n/2$ values around 0 (within a subinterval of length ϵ) and a cluster of $n/2$ values around 1 (within a subinterval of length ϵ), where n is the number of vertices of G. We will choose ϵ later.

We claim that there is a bisection of G with k or fewer edges crossing the cut if and only if there is an allocation in our instance with total envy at most $k + n^2\epsilon$. The forward direction is trivial, just by allocating houses to G in accordance with the bisection. To see the converse, note that if the total envy is extremely close to k, then at most k edges can cross the length of the valuation interval between the two clusters.

To make this condition true, we set $\epsilon \approx n^{-3}$. Note that this is a polynomial-time reduction. □

It follows immediately that the inapproximability results for the Minimum Bisection problem carry over to the Graphical House Allocation problem. In particular, for any fixed constant $\epsilon > 0$, unless P = NP, there is no polynomial-time algorithm that can approximate the optimal bisection within an additive term of $n^{2-\epsilon}$ [39]. This implies that we cannot approximate the optimal total envy under the Graphical House Allocation problem within an additive term of $n^{2-\epsilon}(v(h_n) - v(h_1))$. Since on connected graphs, any allocation must incur an envy of at least $v(h_n) - v(h_1)$, this means that the problem cannot be approximated to an $n^{2-\epsilon}$ multiplicative factor unless P = NP.[4] Additionally, the minimum bisection

[4] It is worth remarking that *any* allocation is an n^2-approximation for connected graphs. The result above shows that we cannot improve this in general.

problem has no PTAS unless NP has randomized algorithms in subexponential time [38]; this result applies to Graphical House Allocation as well. Thus our problem is hard to approximate *even* with identical valuations.

3.3 Hardness of Graphical House Allocation with disconnected graphs

Finally, in this section, we show that Graphical House Allocation is NP-complete even on simple instances of graphs which can be solved in near-linear time in the case of linear arrangements using Proposition 3.1, such as disjoint unions of paths, cycles, cliques, or stars (and any combinations of them).

Theorem 3.4 (Hardness of Disjoint Unions) *Let $\mathcal{A}$ be any collection of connected graphs, such that there is a polynomial-time, one-to-one mapping from each nonnegative integer t (given in unary) to a graph in $\mathcal{A}$ of size t. Let $\mathcal{G}$ be the class of graphs whose members are the finite sub-multisets of $\mathcal{A}$ (as connected components). Then, finding a minimum envy allocation is NP-hard on the class $\mathcal{G}$.*

Proof We will reduce from the Unary Bin Packing problem.[5] In this problem, we are given a set I of items, item sizes $s(i) \in \mathbb{Z}^+$ for all $i \in I$, a bin size B, and a target integer k, all in unary. The problem asks, does there exist a *packing* of the items into at most k bins? A packing is a partition of the set of items into the bins, such that for any bin, the sum of the sizes of its constituent items does not exceed the bin size B.

Given an arbitrary instance $\langle I, s(\cdot), B, k\rangle$ of Unary Bin Packing, we create an instance of the Graphical House Allocation problem as follows. Fix some very large C and some very small $\epsilon > 0$, and let $n = kB$. For each item $i \in I$, take the graph in $\mathcal{A}$ that is the image of $s(i)$, and let G be the disjoint union of all of these graphs. To ensure G has exactly n nodes, we add isolated vertices $s(1)$ to the graph to make up for the gap between total item size and total bin capacity. Note that $G \in \mathcal{G}$, and it is also constructible in polynomial time using the one-to-one mapping. Define $H = \{h_1, \dots, h_n\}$, and for the valuation interval, define (identical) valuations $v(h_j) = \left\lfloor \frac{j}{B} \right\rfloor \cdot C + \frac{\epsilon j}{B}$. Note that this consists of k clusters of B values, each spanning length ϵ, with the distance between any two consecutive clusters at least C.

We wish to show that the given instance is in Unary Bin Packing if and only if the Graphical House Allocation instance (possibly padded with isolated vertices to add up to kB) has an allocation with envy less than C.

The forward direction is trivial; for the packing that attains the capacity constraints, put the graphs in the corresponding clusters on the valuation interval, putting the isolated vertices on the remaining values. No edge is between two different clusters, and so this allocation attains envy much smaller than C, as long as ϵ is small enough.

Conversely, if the envy is smaller than C, then no edge can span two distinct clusters. Therefore, each connected component can be mapped to a particular cluster on the valuation interval. Simply put the corresponding item in the corresponding bin to obtain a packing.

[5] The hardness of Unary Bin Packing can be shown using a straightforward reduction from the NP-complete problem Unary 3-Partition [40].

Note that this is a polynomial-time reduction, as the bin packing instance was given in unary. We can take C to be large enough and ϵ to be small enough, while still being polynomial in the input size. □

Corollary 3.5 *The* Graphical House Allocation *problem under identical valuations is NP-complete on: (a) disjoint unions of arbitrary paths, (b) disjoint unions of arbitrary cycles, (c) disjoint unions of arbitrary stars, and (d) disjoint unions of arbitrary cliques.*

In Sect. 5 we show that despite the hardness suggested by Corollary 3.5, it is possible to exploit a structural property to develop FPT algorithms for the first three problems (we also show a tractable approach to the fourth one).

4 Connected graphs

In this section, we characterize optimal house allocations when the underlying graph G is a star, path, cycle, complete bipartite graph, or binary tree.

These network structures are both mathematically convenient and ubiquitous in real-world social layouts. Any of these layouts can occur spatially; workers may be arranged in any of the above layouts in an office or factory, and allocating resources to these workers gives us an instance of Graphical House Allocation. Stars, paths, and cycles in particular naturally arise in scenarios where social networks reflect physical constraints. Properties and plots within the same neighborhood are often arranged in paths and cycles, while cities in a metro area are often spread out but connected through the central district (creating a star). The Graphical House Allocation problem appears when distributing items (e.g., patio sets or rain-water catchment systems) to these neighbors, or when distributing funded projects to cities in a metro area. In both cases, agents can model their preferences using the monetary values of the items or projects, leading to identical valuation functions. Complete bipartite graphs reflect us-versus-them social structures. In these scenarios, members of one group are not jealous of other members of the same group, but may be deeply offended if the other group receives strongly preferable items. Trees appear in any hierarchical social structure. For example, we might wish to avoid envy between managers and subordinates, while being less concerned with envy amongst peers [8]. A few illustrative graphs covered by our results are shown in Fig. 2.

4.1 Stars

Consider the *star graph* $K_{1,n-1}$, which has a single central node and $n-1$ other nodes of degree 1, all of them connected to the central node but not to each other.

Theorem 4.1 *If G is the star $K_{1,n-1}$, then the minimum envy allocation π^* under identical valuations corresponds to:*

- *for odd n, putting the unique median value in the center of the star, and all the houses on the degree-1 nodes in any order; the value of the envy is $\sum_{i>(n-1)/2+1} v(h_i) - \sum_{i\leq(n-1)/2} v(h_i)$.*

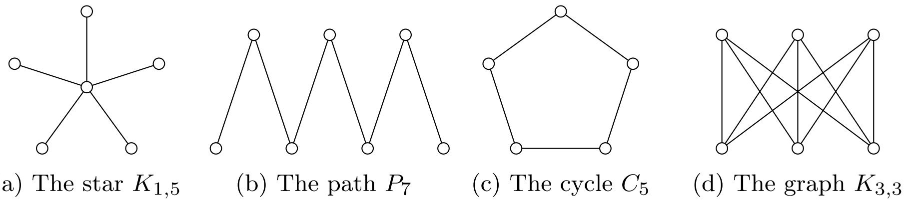


Fig. 2 Examples of characterized connected graphs

- *for even n, putting either of the medians in the center of the star, and all other houses on the degree-1 nodes in any order; the value of the envy for either median is* $\sum_{i>(n+1)/2} v(h_i) - \sum_{i<(n+1)/2} v(h_i)$.

Proof The proof is a restatement of the well-known fact that in any multiset of real numbers, the sum of the L_1-distances is minimized by the median of the multiset. It is easy to verify that for even n, both medians yield the same value. □

4.2 Paths and cycles

Consider the *path graph* P_n. We can characterize optimal allocations on these paths as follows.

Theorem 4.2 *If G is the path graph* P_n, *then the minimum envy allocation* π^* *under identical valuations attains a total envy of* $v(h_n) - v(h_1)$, *is unique (up to reversing the values along the path), and corresponds to placing the houses in sorted order along* P_n.

Proof The result is trivial when $n \leq 2$, so suppose $n > 2$. Fix an arbitrary allocation π, and observe that h_1 and h_n (the minimum and maximum-valued houses) have to be placed on some two vertices of P_n. Suppose the sub-path between them is $(i_1, \dots, i_k)$, with $\pi(i_1) = h_1$ and $\pi(i_k) = h_n$. Then, the envy along that sub-path is, using the triangle inequality repeatedly,

$$\sum_{r=1}^{k-1} |v(\pi(i_{r+1})) - v(\pi(i_r))| \geq |v(\pi(i_k)) - v(\pi(i_1))| = v(h_n) - v(h_1) .$$

It follows that $\mathsf{Envy}(\pi, P_n) \geq v(h_n) - v(h_1)$ for all allocations π. It is straightforward to see that this minimum is attained by sorting the houses in order along the path, and furthermore, this is unique. □

Now, consider the *cycle graph* C_n. We characterize optimal allocations on these cycles as follows.

Theorem 4.3 *If G is the cycle graph* C_n, *then any minimum envy allocation* π^* *under identical valuations attains a total envy of* $2(v(h_n) - v(h_1))$, *and corresponds to the following: place* h_1 *and* h_n *arbitrarily on any two vertices of the cycle, and then place the remaining houses so that each of the two paths from* h_1 *to* h_n *along the cycle consists of houses in sorted order.*

Proof The result is trivial when $n \leq 3$, so suppose $n > 3$. Fix an arbitrary allocation π, and observe that h_1 and h_n have to placed on some two vertices on the cycle C_n. As in the proof of Theorem 4.2, we know each of the two paths along the cycle from h_1 to h_n must have envy at least $v(h_n) - v(h_1)$, and so $\mathsf{Envy}(\pi, C_n) \geq 2(v(h_n) - v(h_1))$ for all allocations π. Once again, it is straightforward to see that this minimum is attained by sorting the houses in order along each of the two paths. □

Corollary 4.4 *For $n \geq 3$, the number of optimal allocations along the cycle C_n is 2^{n-3}, up to rotations and reversals.*

Proof We fix an arbitrary agent in C_n who receives h_1. Subsequently, we can choose an arbitrary subset of $H \setminus \{h_1, h_n\}$ to appear along one of the paths to h_n. Note that this choice completely determines an optimal allocation, as the other path contains the complement of the selected subset, and each subset appears in sorted order along the paths. The number of such subsets is 2^{n-2}. Since choosing the complement of our selected subset would have given us the same allocation up to a reversal and rotation, we have over-counted by a factor of two, and the result follows. □

Perhaps slightly non-obviously, the proofs of Theorems 4.2 and 4.3 can be seen as purely geometric arguments using the valuation interval. To see this, consider the path P_n, and take any allocation π that does not satisfy the form stated in Theorem 4.2, and consider how the allocation looks with respect to the valuation interval. First, observe that every sub-interval of the valuation interval between consecutive houses needs to be covered by some line segment from the allocation. Otherwise, there would be no edge with a house from the left to a house from the right of the sub-interval, which is impossible, as P_n is connected. But the only way to meet this lower bound of one line segment for each sub-interval of the valuation interval is to sort the houses along the path. The visualization of this argument for paths is shown in Fig. 3a. The geometric argument for cycles is similar, with the allocation shown in Fig. 3b.

4.3 Complete bipartite graphs

Let us start with the complete bipartite graph $K_{r,r}$ ($r \geq 1$) where both parts have equal size. Note that $n = 2r$ in this case.

Theorem 4.5 *When G is the graph $K_{r,r}$, the minimum envy allocation π^* has the following property: for every $i \in [r]$ the houses $\{h_{2i-1}, h_{2i}\}$ cannot be allocated to agents in the same side of the bipartite graph. Moreover, all allocations which satisfy this property have the same (optimal) envy.*

Proof For notational ease, let the graph have bipartition (L, R), with $|L| = |R| = r$. We refer to the property in the theorem statement as the *optimal* property. This proof will use the notation $n^{<}_{L,\pi}(x), n^{>}_{L,\pi}(x), n^{<}_{R,\pi}(x)$ and $n^{>}_{R,\pi}(x)$ from Definition 2.3.

Assume for contradiction that some optimal allocation π^* does not satisfy the optimal property. We will improve on this allocation, thereby reaching a contradiction.

(a) Optimal allocation on a path. (b) Optimal allocation on a cycle.

Fig. 3 Visualizations of the path and cycle optimal allocations

Because π^* does not satisfy the optimal property, there must exist an $i \in [r]$ such that both h_{2i-1} and h_{2i} are allocated to the same part. Let j be the least such i where this is true. Assume without loss of generality that h_{2j-1} and h_{2j} are allocated to agents in L.

Let $\{h_{2j-1}, h_{2j}, \dots, h_{2j+k}\}$ be the set of houses allocated to agents in L such that h_{2j+k+1} is allocated to some agent in R. Note that by our assumption, we have $k \geq 0$. Note that $2j + k + 1$ must be at most $2r$ because otherwise, the allocation allocates more items to L than R, which contradicts the definition of an allocation itself.

Construct an allocation π' from π^* by swapping the houses h_{2j+k} and h_{2j+k+1}. Note that we swap a house allocated to some agent in L with a house allocated to some agent in R. This has been pictorially described in Fig. 4.

Let us compute the difference in envy between allocations π^* and π'. In this analysis we slightly abuse notation and refer to the total envy between an agent and their neighbors as their envy *towards* their neighbors. For any agent with value less than $v(h_{2j+k})$ under π^* in L, their envy towards their neighbors in π' is less than their envy in π^* by exactly $v(h_{2j+k+1}) - v(h_{2j+k})$. Similarly, for any agent with value greater than $v(h_{2j+k})$ under π^* in L, their envy towards their neighbors in π' is greater than their envy in π^* by exactly $v(h_{2j+k+1}) - v(h_{2j+k})$. Extending this reasoning, we get the following expression for difference in envy

$$\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi^*, G) \\
&= \left[n^{>}_{L,\pi^*}(v(h_{2j+k})) - n^{<}_{L,\pi^*}(v(h_{2j+k}))\right]\left(v(h_{2j+k+1}) - v(h_{2j+k})\right) \\
&\quad + \left[n^{<}_{R,\pi^*}(v(h_{2j+k+1})) - n^{>}_{R,\pi^*}(v(h_{2j+k+1}))\right]\left(v(h_{2j+k+1}) - v(h_{2j+k})\right) \\
&= \left(v(h_{2j+k+1}) - v(h_{2j+k})\right)[n^{>}_{L,\pi^*}(v(h_{2j+k})) - n^{<}_{L,\pi^*}(v(h_{2j+k})) \\
&\quad + n^{<}_{R,\pi^*}(v(h_{2j+k+1})) - n^{>}_{R,\pi^*}(v(h_{2j+k+1}))] \\
&= \left(v(h_{2j+k+1}) - v(h_{2j+k})\right)[(r - (k + 2 + j - 1)) - (k + 1 + j - 1) \\
&\quad + (j - 1) - (r - j)] \\
&= \left(v(h_{2j+k+1}) - v(h_{2j+k})\right)\left[2j - 2(k + j) - 2\right] \\
&= \left(v(h_{2j+k+1}) - v(h_{2j+k})\right)[-2k - 2] \\
&< 0\,.
\end{aligned}$$

The third equality follows from our choice of j; for any $i < j$, exactly one of h_{2i-1} and h_{2i} is allocated to L under π^*. The inequality follows since $k \geq 0$ and $v(h_{2j+k+1}) - v(h_{2j+k}) > 0$. This implies that π' has a lower envy than π^*, which contradicts the optimality of π^*. It follows that all minimum envy allocations have the optimal property.

We now complete the proof by showing that in any allocation that satisfies the optimal property, for any $i \in [r]$, swapping h_{2i-1} and h_{2i} results in an allocation with equal envy.

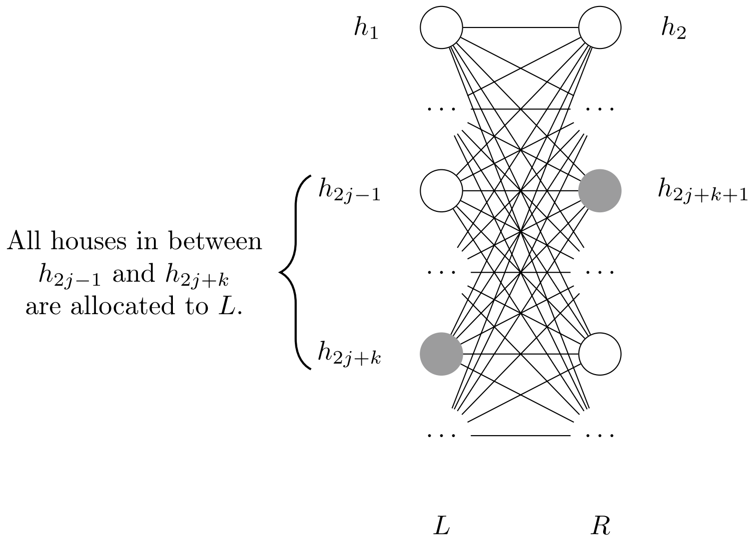


Fig. 4 A pictorial description of the allocation π in the proof of Theorem 4.5. To create the allocation π' we swap the houses allocated to the shaded nodes, i.e., we swap h_{2j+k} and h_{2j+k+1}

This observation can be repeatedly applied to show that any two allocations that satisfy the optimal property have the same envy. Note that permuting the allocation within a specific part (L or R) does not affect the total envy.

Formally, let π be any allocation that satisfies the optimal property. Pick an arbitrary $i \in [r]$ and swap h_{2i-1} and h_{2i} to create the allocation π'; Without loss of generality, assume h_{2i-1} is allocated to some agent in L in π. The difference in envy of the two allocations is given by:

$$\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G) \\
&\quad = \left[n^{>}_{L,\pi}(v(h_{2i-1})) - n^{<}_{L,\pi}(v(h_{2i-1}))\right]\left(v(h_{2i}) - v(h_{2i-1})\right) \\
&\qquad + \left[n^{<}_{R,\pi}(v(h_{2i})) - n^{>}_{R,\pi}(v(h_{2i}))\right]\left(v(h_{2i}) - v(h_{2i-1})\right) \\
&\quad = \left(v(h_{2i}) - v(h_{2i-1})\right)[n^{>}_{L,\pi}(v(h_{2i-1})) - n^{<}_{L,\pi}(v(h_{2i-1})) \\
&\qquad + n^{<}_{R,\pi}(v(h_{2i})) - n^{>}_{R,\pi}(v(h_{2i}))] \\
&\quad = \left(v(h_{2i}) - v(h_{2i-1})\right)[(r-i) - (i-1) + (i-1) - (r-i)] \\
&\quad = 0.
\end{aligned}$$

□

This also implies a straightforward polynomial time algorithm to compute a minimum envy allocation for instances on $K_{r,r}$.

We can now generalize this result to complete bipartite graphs where the two parts have unequal size. We relegate the proof to Appendix B, due to its similarity with the previous proof.

Theorem 4.6 *When G is the graph $K_{r,s}$ $(r > s)$, the minimum envy allocation π^* has the following property:*

(1) *If $r - s =: 2m$ is even, then the first and last m houses are allocated to the larger part, and for all $i \in [s]$, the houses h_{m+2i-1} and h_{m+2i} are allocated to different parts.*
(2) *If $r - s =: 2m + 1$ is odd, then the first m and last $m + 1$ houses are allocated to the larger part. For all $i \in [s]$, the houses h_{m+2i-1} and h_{m+2i} are allocated to the larger and smaller parts respectively.*

Moreover, all allocations which satisfy this property have the same (optimal) envy.

The following corollary is now due to a simple counting argument.

Corollary 4.7 *For any complete bipartite graph $K_{r,s}$ $(r \geq s)$,*

- *If $r - s$ is even, there are 2^s optimal allocations;*
- *If $r - s$ is odd, there is exactly one optimal allocation,*

up to permutations over allocations to the same side of the graph.

Proof For simplicity, let L and R denote the larger and smaller parts of the bipartition respectively. Therefore, $|L| = r \geq s = |R|$.

We wish to count the number of allocations which allocate a different set of houses to L (and therefore, R as well). There are of course, $r!$ allocations given a set of houses to allocate to agents in L but we ignore this factor.

When $r - s$ is even, there are s different choices we can make. That is, for each $i \in [s]$, we can choose which of h_{m+2i-1} and h_{m+2i} goes to L and which one goes to R (Theorem 4.6). This gives us 2^s different allocations.

When $r - s$ is odd, there is no choice since Theorem 4.6 shows that only one specific set of houses allocated to L achieves the optimal envy. Therefore, not counting permutations over allocations to the same part, there is only one unique allocation. □

It is easy to see that the simple structural characterization of optimal solutions from Theorem 4.6 implies a straightforward polynomial time algorithm for computing exact optimal allocations on general complete bipartite graphs. We remark here that, in fact, Theorem 4.6 generalizes Theorem 4.1 as well. When the number of outer (non-center) nodes in the star is odd, there are two possible houses that can be allocated to the center in an optimal allocation. But when the number of outer nodes is even, any optimal allocation allocates a unique house to the center.

4.4 Binary trees

In this subsection, we consider binary trees. A *binary tree T* is defined as a rooted tree where each node has either 0 or 2 children.

Our main result is a structural property characterizing at least one of the optimal allocations for any instance where the underlying graph is a binary tree. We call this the *local median* property.

Definition 4.8 (*Local Median Property*) An allocation on a binary tree satisfies the *local median property* if, for any internal node, exactly one of its children is allocated a house with value less than that of the node.

The proofs in this section will use the following lemma. We define the *inverse* of a valuation function v as a valuation function v^{inv} such that $v^{\text{inv}}(h) = -v(h)$ for all $h \in H$ (appropriately shifted so that all values of v are nonnegative). We note that any allocation has the same envy along any edge with respect to the inverted valuation and the original valuation.

Lemma 4.9 *The envy along any edge of the graph G under an allocation π with respect to the valuation v is equal to the envy along the same edge of the graph G under the allocation π with respect to the valuation v^{inv}.*

Proof For any edge (i, j) in the graph G and any allocation π, we have

$$|v(\pi(i)) - v(\pi(j))| = |(-v(\pi(i))) - (-v(\pi(j)))| = |v^{\text{inv}}(\pi(i)) - v^{\text{inv}}(\pi(j))|.$$

□

We will now show that at least one minimum envy allocation satisfies the local median property. More formally, we show the following: given a binary tree T and any allocation π, there exists an allocation that satisfies the local median property and has equal or lower total envy. The proof relies on the following lemma.

Lemma 4.10 *Let π be an allocation on a binary tree T, not satisfying the local median property. Let i be an internal node furthest from the root which is not allocated the median among the values given to it and its children. Then, there exists an allocation π' such that*

(a) *For the subtree T' rooted at i, we have that* $\mathsf{Envy}(\pi(T'), T') > \mathsf{Envy}(\pi'(T'), T')$;
(b) *For any other subtree T'' not contained by T', we have that* $\mathsf{Envy}(\pi(T''), T'') \geq \mathsf{Envy}(\pi'(T''), T'')$.

Proof Let the node i have value y under π and its children have values x_m and z_m respectively under π. By assumption, either $y < \min\{x_m, z_m\}$ or $y > \max\{x_m, z_m\}$. We show that in either case, the lemma holds. Since allocations are bijective and the values can be assumed to be distinct, we will refer to tree nodes using the value allocated to them in π.

Case 1 ($y < \min\{x_m, z_m\}$). Assume without loss of generality that $x_m < z_m$. We construct a path recursively as follows. Initialize the path as (y). If the final node on the path either has no children or has at least one child with allocated value lower than the value at the start of the path, i.e., y, then stop. Otherwise, pick the least valued child of the final node on the path and append it to the path. This gives us a path $(y, x_m, x_{m-1}, \dots, x_1)$ for some nodes with value $x_m, x_{m-1}, \dots, x_1$ in T. Note that by definition, this path has at least 2 vertices, i.e., $m \geq 1$. We construct a new allocation π'

from π by cyclically transferring houses as follows: we give the agent with value y the house with value x_m, we give the agent with value x_m the house with value x_{m-1} and so on till finally we give the agent with value x_1 the house with value y. This has been described in Fig. 5.

The solid edges and dashed edges in Fig. 5 cover all possible edges e in T where $envy_{\pi}(e) \neq envy_{\pi'}(e)$. From our path construction and our assumption that i is a node furthest from the root which does not satisfy the local median property, we have the following two properties: (a) $\max\{x_j, z_j\} > x_{j+1} > \min\{x_j, z_j\}$ for all $j \in [m-1]$, and (b) $x_j < z_j$ for all $j \in [m]$.

These two properties allow us to compare the envy along the solid edges:

$$\begin{aligned} envy_{\pi}(\text{solid}) &= (x_m - y) + (z_m - y) + \left[\sum_{j=2}^{m}((x_j - x_{j-1}) + (z_{j-1} - x_j))\right] \\ &= \left[\sum_{j=1}^{m-1}(z_j - x_j)\right] + (z_m - y) + (x_m - y)\,. \\ envy_{\pi'}(\text{solid}) &= (x_1 - y) + (z_1 - x_1) + \left[\sum_{j=2}^{m}((x_j - x_{j-1}) + (z_j - x_j))\right] \\ &= \left[\sum_{j=1}^{m}(z_j - x_j)\right] + (x_m - x_1) + (x_1 - y)\,. \end{aligned}$$

Combining the two values, we get

$$envy_{\pi'}(\text{solid}) - envy_{\pi}(\text{solid}) = y - x_m.$$

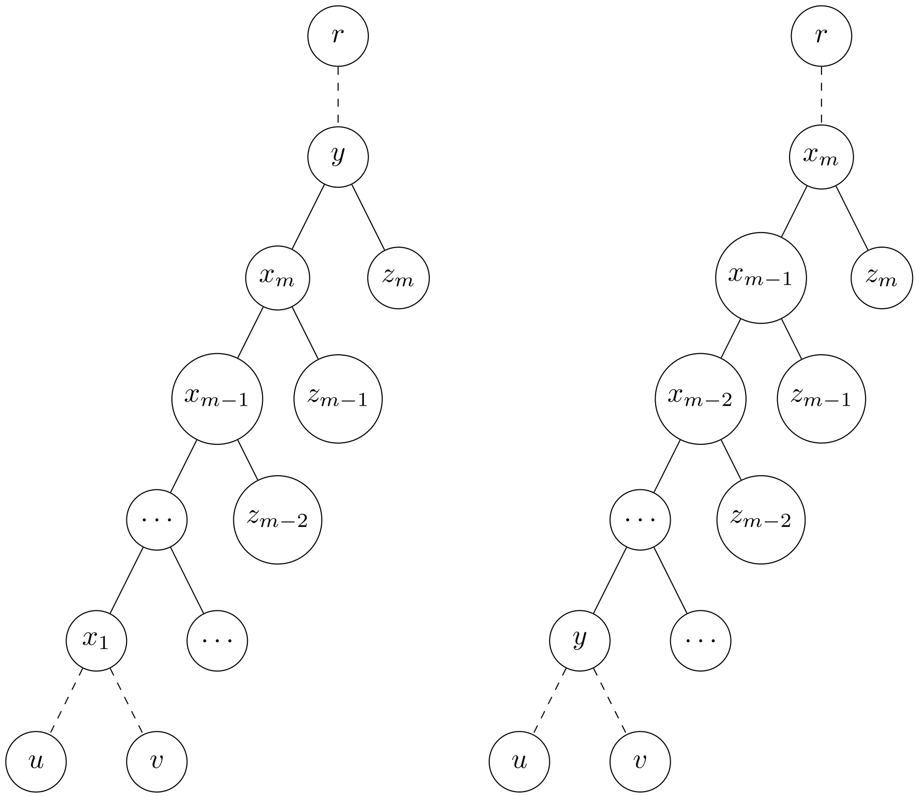
To compute the difference in envy along the dashed lines, some straightforward casework is required. There are many different possible relations between u, v, x_1, and y, and between r, y, and x_m. All possible cases and their corresponding results are summarized in Table 1. There are two assumptions made in Table 1. First, without loss of generality we assume $u < v$. Second, $u < y$, since this is the termination condition from our path construction.

If y is the root of the tree (i.e., r does not exist and $T' = T$), from Table 1, we get:

$$\begin{aligned} &\mathsf{Envy}(\pi', T') - \mathsf{Envy}(\pi, T') \\ &\quad = envy_{\pi'}(\text{solid}) - envy_{\pi}(\text{solid}) + envy_{\pi'}(\text{dashed}) - envy_{\pi}(\text{dashed}) \\ &\quad \leq (y - x_m) + 0 \\ &\quad < 0\,. \end{aligned}$$

Therefore, the total envy of π' is strictly less than that of π.

If r exists, the above analysis shows that the total envy along the subtree rooted at y (denoted by T') strictly reduces. Let us now study the envy of any tree T'' that is not contained by T'. Either T'' contains T', or T'' and T' are disjoint. If they are disjoint, then $\mathsf{Envy}(\pi, T'') = \mathsf{Envy}(\pi', T'')$, since the allocation on the subtree T'' is the same in π and π'. If T'' strictly contains T', T'' must contain the node r. From Table 1, we get:



(a) Before the swap (allocation π) (b) After the swap (allocation π').

Fig. 5 Cyclic swap to show the local median property holds (Lemma 4.10). Solid edges are guaranteed to exist. Dashed edges may or may not exist

Table 1 Cases for the possible values of $envy_{\pi'}(\text{dashed}) - envy_{\pi}(\text{dashed})$

Cases	r does not exist	$r < y < x_m$	$y < r < x_m$	$y < x_m < r$
u and v do not exist	0	$(x_m - y)$	$< (x_m - y)$	$(y - x_m)$
$u < y < v < x_1$	< 0	$< (x_m - y)$	$< (x_m - y)$	$< (y - x_m)$
$u < y < x_1 < v$	0	$(x_m - y)$	$< (x_m - y)$	$(y - x_m)$
$u < v < y < x_1$	< 0	$< (x_m - y)$	$< (x_m - y)$	$< (y - x_m)$

$$\begin{aligned}
&\mathsf{Envy}(\pi', T'') - \mathsf{Envy}(\pi, T'') \\
&\quad = envy_{\pi'}(\text{solid}) - envy_{\pi}(\text{solid}) + envy_{\pi'}(\text{dashed}) - envy_{\pi}(\text{dashed}) \\
&\quad \le (y - x_m) + (x_m - y) \\
&\quad = 0\,.
\end{aligned}$$

Therefore the total envy weakly decreases and we are done.

Case 2 ($y > \max\{x_m, z_m\}$). This implies $-y < \min\{-x_m, -z_m\}$. We can therefore apply Case 1 to the allocation π under the inverted valuations v^{inv}. It follows that, with respect to v^{inv}, there is an allocation π' which has a strictly lower total envy along the subtree T' rooted at i and a weakly lower total envy along any subtree T'' that is not contained by T'. Applying Lemma 4.9 with the allocations π' and π, we get the required result. □

Lemma 4.10 immediately gives rise to the following corollary.

Theorem 4.11 *For any binary tree T, at least one minimum envy allocation satisfies the local median property.*

Proof Given any tree T and a node i, we use T_i to denote the subtree of T rooted at node i. We also use $i.\mathsf{left}$ and $i.\mathsf{right}$ to refer to i's left and right child respectively.

Given any tree T rooted at some node i, consider the allocation π which lexicographically minimizes the vector:

$$\mathbf{u}(\pi, T) = (\mathsf{Envy}(\pi, T), \mathsf{Envy}(\pi, T_{i.\mathsf{left}}), \mathsf{Envy}(\pi, T_{i.\mathsf{right}}), \mathsf{Envy}(\pi, T_{i.\mathsf{left.left}}), \mathsf{Envy}(\pi, T_{i.\mathsf{left.right}}), \mathsf{Envy}(\pi, T_{i.\mathsf{right.left}}), \mathsf{Envy}(\pi, T_{i.\mathsf{right.right}}), \dots) .$$

It is easy to see that π is an optimal allocation. It is also easy to see that π satisfies the local median property as well. If π does not satisfy the local median property, applying Lemma 4.10, we get that there is an allocation π' such that $\mathbf{u}(\pi, T)$ is lexicographically greater than $\mathbf{u}(\pi', T)$, which is a contradiction. □

Unfortunately, the local median property is too weak to exploit for a polynomial time algorithm. Ideally, we would like to use the property to show that some minimum envy allocation satisfies an even stronger property called the *global median* property.

Definition 4.12 (*Global Median Property*) An allocation on a binary tree satisfies the *global median property* if, for every internal node, all the houses in one subtree of the node have value less than the house allocated to the node, and all the houses in the other subtree have value greater than the house allocated to the node.

If T is a binary tree of maximum depth d where an optimal allocation satisfies the global median property, there is a straightforward divide-and-conquer algorithm that computes an optimal allocation in time $O(4^d)$: the algorithm guesses which subtree of the root is allocated values more than that of the root, and which subtree is allocated values less than that of the root. The root has a unique allocation that satisfies the constraints placed by the guesses; the algorithm allocates the root this unique house and then applies the same procedure to each of the subtrees of the root. The time complexity comes from the recursive expression $T(d) \leq 4T(d-1) + O(1)$, where the 4 comes from the fact that we have to solve the problem on two subtrees for each of the two global median choices. Solving this gives us a runtime of $O(4^d)$. In particular, if T were close to being balanced, this algorithm would run in polynomial time in the size of T.

Conjecture 4.13 *There is an algorithm that computes an optimal allocation on a binary tree of maximum depth d in time $O(4^d)$. In particular, this algorithm runs in polynomial time on (nearly) balanced trees.*

It was recently shown in [25] that not all instances have an optimal allocation that satisfies the global median property; in fact, there is a counterexample even on a depth-3 complete binary tree. However, the counterexample does not rule out the possibility of the global median property being true on "most" trees, or of efficient algorithmic approaches not needing to exploit the local or global median properties, so Conjecture 4.13 remains open.

4.5 General trees

How do we take the approaches for binary trees and build towards arbitrary trees? Note that one consequence of Theorem 4.11 is that in at least one optimal allocation on a binary tree, the minimum and the maximum must both appear on leaves.

In the Minimum Linear Arrangement problem, it is known [37] that when the underlying graph is a tree, some optimal allocation assigns both the minimum and maximum values to leaves, and furthermore, the (unique) path from this minimum to the maximum consists of monotonically increasing values. This characterization is used crucially in designing the polynomial time algorithm on trees [33].

Empirically, this same property for trees seems to hold for non-uniformly spaced values as well. The proof technique used in Seidvasser [37] does not extend to our setting, but testing the problem on 1000 randomly generated trees and uniformly random values on the interval [0, 100] always gives us these properties on trees: the minimum and maximum values both end up on leaves.

It would be remarkable if this kind of structural characterizations held for our problem, but we remark here that the polynomial time algorithm exploiting these characterizations [33] does not generalize. Recently, in fact, Hosseini et al. [25] showed that the Graphical House Allocation problem, unlike Minimum Linear Arrangement, is NP-hard on trees.

5 Disconnected graphs

In this section, we consider disconnected graphs, starting with a structural characterization, and then using that to obtain upper bounds for several natural classes of disconnected graphs with hardness results (Sect. 3).

5.1 A structural characterization

Recall from Proposition 3.1 that optimal allocations in Minimum Linear Arrangement have the property that the connected components of the graph are assigned contiguous values. We might hope that this simple and elegant property is true for Graphical House Allocation as well. However, this turns out to be a crucial point of difference between these two problems: Proposition 3.1 is *false* in our setting, and so we can no longer assume our graph is connected without loss of generality. To see this, consider an instance when the underlying graph G is a disjoint union of an edge P_2 and a triangle C_3. The two valuation intervals in Fig. 6 yield very different optimal structures for this same instance.

We remark that this major departure from the Minimum Linear Arrangement problem implies that the spacing of the values along the valuation interval becomes a key

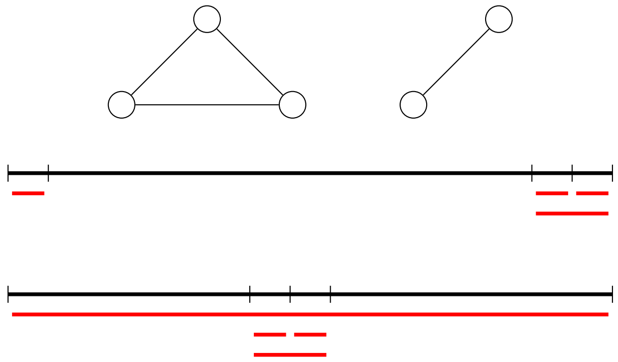


Fig. 6 For the valuation interval on top, the optimal allocation to $P_2 + C_3$ is to give the two low-valued houses to the edge, and to give the three high-valued houses to the triangle. This is the only allocation where the envy is negligible. For the valuation interval on the bottom, the optimal allocation to $P_2 + C_3$ is to give the two extreme-valued houses to the edge, and the cluster in the middle to the triangle. Any other allocation has to count one of the long halves of the interval multiple times, and is therefore strictly suboptimal. This is an instance where we see one of the connected components being "split" by another in the valuation interval. We prove in Theorem 5.15 that the graph $P_2 + C_3$ is *splittable*, because we can always assign a contiguous sequence of items to the C_3 component, and the P_2 component receives a contiguous sequence of items ignoring the other 3 items. It is not strongly splittable because P_2 does not always split C_3 (as in the second example above)

factor in the structure of optimal allocations in Graphical House Allocation. This serves as a motivation to classify disconnected graphs according to whether their connected components are always assigned contiguous values for all valuation interval instances. We call the relevant property *splittability*, defined as follows.

Definition 5.1 (*Splitting*) Let $G_1 = (N_1, E_1)$ and $G_2 = (N_2, E_2)$ be two of the connected components of $G = (N, E)$, and fix an arbitrary allocation π. We say G_1 *splits* G_2 *in* π if the values of $\pi(G_1)$ form a contiguous subset of the values in $\pi(G_1) \cup \pi(G_2)$.

Definition 5.2 (*Splittability and Strong Splittability*) Let G be a disconnected graph with connected components $G_1, \dots, G_k$. Then,

1. G is *splittable* if there exists an ordering $G_1, \dots, G_k$ of the components where, for all valuation intervals, there is an optimal allocation where for all $1 \le i < j \le k$, G_i splits G_j.
2. G is *strongly splittable* if, in addition to the above, G_j also splits G_i. Note that this is only possible if an optimal allocation assigns a contiguous subset of values to each connected component.
3. G is *unsplittable* if it is not splittable.

A class $\mathcal{A}$ of graphs is splittable (resp. strongly splittable) if every graph in it is splittable (resp. strongly splittable). Conversely, $\mathcal{A}$ is unsplittable if it contains an unsplittable graph.

Intuitively, splittability requires that the connected components of the graph G can be ordered such that each component receives a contiguous set of values, if we ignore items assigned to components appearing earlier in the ordering. This ordering of the components is fixed with respect to the graph structure, and does not depend on the valuation interval. For example, we see in Theorem 5.15 that for disjoint unions of cliques, for *any* valuation interval, the cliques can be ordered in decreasing order of size. We must assign a contiguous interval to the largest clique; upon removing those items, we must assign a contiguous interval to the second-largest clique; and so on. This is a descriptive statement rather than a computational statement, since it is non-trivial to determine *which* contiguous interval to allocate to each component of the graph.

For the graph to be strongly splittable, the set of values assigned to each component must be contiguous with respect to the entire valuation interval. In this case, any order suffices to show (strong) splittability, because any pair of components G_i and G_j would both split each other. Figure 6 shows a graph that is splittable but not strongly splittable, since C_3 always splits P_2, but P_2 may not split C_3, depending on the valuation interval.

We note that G is unsplittable precisely when there is a valuation interval where for each optimal allocation π, there are components G_1 and G_2 with $u, u' \in \pi(G_1)$ and $v, v' \in \pi(G_2)$ such that $u \prec v \prec u' \prec v'$. Furthermore, G is strongly splittable only if it is splittable.

In the Minimum Linear Arrangement problem, all disconnected graphs are strongly splittable, by Proposition 3.1. In contrast, for our problem, Fig. 6 already provides an example of a graph that is not strongly splittable. We discuss several examples of strongly splittable graphs in our problem in Sects. 5.2, 5.3, and 5.4; in particular, disjoint unions of paths, cycles, stars, identical cliques, or identical complete bipartite graphs satisfy strong splittability.

Our formulation of splittability and strong splittability has an immediate algorithmic consequence.

Proposition 5.3 *Suppose G has k connected components (where k is not necessarily a constant). If G is strongly splittable, and we can find a minimum envy allocation for each component in time* $O(\operatorname{poly}(n))$*, then we can find a minimum envy allocation on G in time* $O(\operatorname{poly}(n) \cdot k!)$*. If G is splittable, and we can find a minimum envy allocation for each component in time* $O(\operatorname{poly}(n))$*, then we can find a minimum envy allocation on G in time* $n^{O(k)}$*.*

Proof The proof follows straightforwardly from the definitions of (strong) splittability. If G is strongly splittable and has k connected components, we can try all $k!$ orderings of these components along the valuation interval. Each such ordering takes $O(\operatorname{poly}(n))$ to evaluate (since $k \leq n$), and one of the orderings is optimal by definition. If G is splittable, for any ordering of its components, we can place the first component on any of $O(n)$ contiguous subintervals along the interval, and then place the second component on any of the $O(n)$ contiguous subintervals among the remaining values, and so on. This ordering takes $n^{O(k)}$ time to output an optimal envy. We need to test this on all $k!$ orderings of the components, which costs $k! \cdot n^{O(k)}$, which is still $n^{O(k)}$, as $k \leq n$. □

It is not immediately obvious that there are splittable graphs that are not strongly splittable. Figure 6 shows an example of such a graph (Theorem 5.15 proves splittability). We will see more examples of this later, but we remark that there are even splittable forests that are not strongly splittable (Fig. 7). Even less obviously, *unsplittable* forests exist (Fig. 8). We formalize these below.

Proposition 5.4 *The following are both true.*

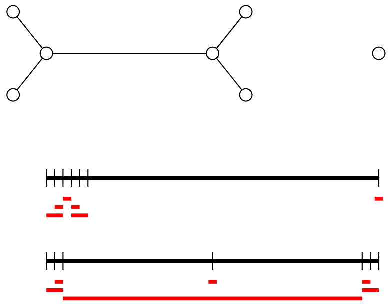


Fig. 7 Example of a splittable forest that is not strongly splittable. The forest is trivially splittable, as one component is just a single vertex. For the bottom valuation line, an optimal allocation must allocate the extreme clusters in the interval to the larger connected component

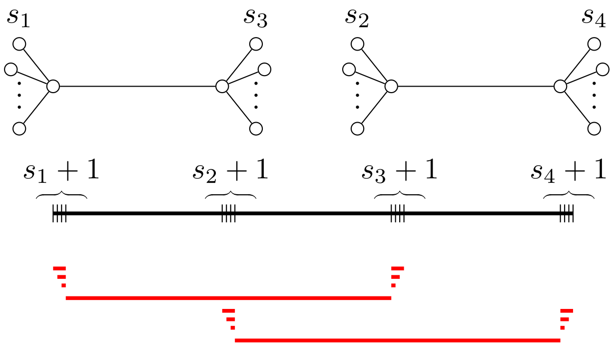


Fig. 8 Example of an unsplittable forest. Suppose $s_1 < s_2 < s_3 < s_4$, and they satisfy for all i, j, $|s_i - s_j| \geq 3$, and for all i, j, k, $s_i + s_j > s_k + 2$. Then, an optimal allocation on this instance must allocate the entire cluster of size $s_i + 1$ on the valuation interval to the corresponding star-like cluster of the given forest

1. *There exists a splittable forest that is not strongly splittable.*
2. *There exists an unsplittable forest.*

Proof We can prove these one part at a time.

1. The graph G given in Fig. 7 is a splittable forest that is not strongly splittable.

   It is trivial that G is splittable, as one component is a single vertex that always splits the other component on the valuation interval.

   Consider the lower valuation interval that is depicted in Fig. 7. Assume that the clusters along the valuation interval are sufficiently packed (each within a subinterval of length $\epsilon := 0.001/n^2$, where $n = 7$), and furthermore, the sole valuation in the middle is exactly at the center of the interval. Without loss of generality, assume the entire valuation interval has length 1. Note that the allocation that places the induced stars of G in the clusters attains a total envy of at most 1.001.

   We first claim that an optimal allocation cannot place both the degree-3 vertices in the same cluster. In such an allocation, one of the two large subintervals needs to be covered by at least two edges, and so the total envy is at least 3/2.

We next claim that an optimal allocation cannot place a degree-3 vertex in the center. If it does, then again by a similar casework as in the previous paragraph, one large subinterval has to be covered by at least two edges, and so the total envy is at least 3/2.

Therefore, every optimal allocation must place the degree-3 vertices in different clusters. The edge between those two vertices, therefore, incurs an envy of 1 by itself. Now, if the isolated vertex is anywhere but the center, there the center must be a leaf attached to a degree-3 vertex. The edge from this leaf to the degree-3 vertex incurs an additional envy of 1/2, pushing the total envy up to 3/2. It follows that the isolated vertex must be at the center.

2. The graph given in Fig. 8 is an unsplittable forest.

Assume that the clusters along the valuation interval are sufficiently packed (each within a subinterval of length ϵ), and furthermore, they are equispaced along the entire valuation interval, and without loss of generality assume the entire valuation interval has length 1.

Of course, note that each of the three "large" subintervals (of length 1/3 each) must be counted at least once in any allocation: the first must be counted since it is not possible to take a set of $s_1 + 1$ vertices of the forest with no edges going to its complement; the third must be counted for the same reason, using $s_i + s_j > s_k + 2$, making it impossible to pack in either of the components entirely within the last cluster; and the second must be counted because neither component can fit perfectly inside the first two clusters, again using $s_i + s_j > s_k + 2$ and $|s_i - s_j| \geq 3$. This immediately ensures an envy of at least 1, for any allocation.

Note that the allocation that places the induced stars of the given graph in the corresponding clusters along the valuation interval attains a total envy of at most $4/3 + 0.001$ (assuming ϵ is small enough). Let the four vertices of degree 2 or more be x_1, x_2, x_3, x_4, where x_i is incident to exactly s_i degree-1 vertices. Let us also number the clusters along the valuation interval 1, 2, 3, 4 from left to right.

We first claim that in any optimal allocation, x_i cannot be in cluster j for $j < i$. Otherwise, at least three of the s_i neighbors of x_i must lie in other clusters, so one of the three large subintervals must be counted three or more times. Together with the two other subintervals (which must be counted), it is then easy to see that the envy in this case would exceed 5/3. We next claim that x_i and x_j cannot be in the same cluster, for $i \neq j$. Otherwise, again, at least three edges pass over the same large subinterval, and so the envy exceeds 5/3 again.

It follows that x_i must belong to the ith cluster, for all i. The result follows immediately.

□

5.2 Disjoint unions of paths, cycles, and stars

We now move on to algorithmic approaches and characterizations of minimum envy allocations, and start with the setting where G is a disjoint union of paths. Suppose $G = P_{n_1} + \cdots + P_{n_r}$. What does an optimal allocation on G look like?

Theorem 5.5 *Let G be a disjoint union of paths, $P_{n_1} + \cdots + P_{n_r}$. Then, G is strongly splittable. Furthermore, in any optimal allocation, within each path, the houses appear in sorted order.*

Proof By Theorem 4.2, we know that each of the paths should have its allocated houses in sorted order. Now, suppose there are values $h_k \prec h_\ell \prec h_m$, with h_k and h_m being allocated to P_{n_i}, and h_ℓ to a different path P_{n_j}.

We can reallocate the houses only on these two paths and strictly improve the allocation. For instance, suppose $H_i := \pi(P_{n_i})$ and $H_j := \pi(P_{n_j})$. We can now allocate the n_i lowest-valued houses in $H_i \cup H_j$ to P_{n_i} and the n_j highest-valued houses in $H_i \cup H_j$ to P_{n_j}, keeping the rest of the allocation the same. Now, note that every subinterval among the values in $H_i \cup H_j$ counted by both paths in this new allocation was also counted by both paths in the old allocation. However, at least one subinterval (e.g., the subinterval between the n_i lowest values and the n_j highest values) is counted by strictly fewer paths in the new allocation. Therefore, this leads to an allocation with strictly lower envy than before, and this concludes the proof. □

The following corollary, which follows directly from Proposition 5.3, shows an FPT algorithm on the disjoint union of paths, parameterized by the number r of different paths. We simply check each of the $r!$ orderings of these paths, and return the one with the least envy.

Corollary 5.6 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of paths in time $\widetilde{O}(nr!)$, where r is the number of paths.*[6]

If G is a disjoint union of cycles, say $G = C_{n_1} + \cdots + C_{n_r}$, the same theorems characterizing optimal allocations go through, using Theorem 4.3. We omit the proofs, but state the results formally.

Theorem 5.7 *Let G be a disjoint union of cycles, $C_{n_1} + \cdots + C_{n_r}$. Then G is strongly splittable. Furthermore, in any optimal allocation, within each cycle, the houses appear in the form characterized in Theorem 4.3.*

Corollary 5.8 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of cycles in time $\widetilde{O}(nr!)$, where r is the number of cycles.*

If t is the number of different path (or cycle) *lengths*, then a straightforward dynamic programming algorithm computes the minimum envy allocation in time $O(tn^{t+1})$.

Proposition 5.9 *Let G be a disjoint union of paths. If t is the number of different path lengths in G, then we can find an optimal allocation on G for any instance in time $O(tn^{t+1})$.*

Proof The result for $t = 1$ is trivial. For $t > 1$, if the distinct path lengths are $n_1, \ldots, n_t$, then suppose $\varphi(r_1, \ldots, r_t, \ell)$ denotes the optimal envy using r_i paths of length n_i, for $i = 1, \ldots, t$, on the house set $\{h_1, \ldots, h_\ell\}$. Using Theorem 5.5 and Theorem 4.2, we have the recursion

[6] We suppress the logarithmic factors required for integer addition henceforth, in order to avoid the minor technical considerations of bit representation.

$$\varphi(r_1, \dots, r_t, \ell) = \min\{\varphi(r_1 - 1, r_2, \dots, r_t, \ell - n_1) + (v(h_\ell) - v(h_{\ell-n_1+1})), \dots, \varphi(r_1, \dots, r_{t_1}, r_t - 1, \ell - n_t) + (v(h_\ell) - v(h_{\ell-n_t+1}))\}.$$

Dynamically solving this yields an $O(tn^{t+1})$ algorithm to find the optimal allocation on the given instance. □

Corollary 5.10 *Let G be a disjoint union of cycles. If t is the number of different cycle lengths in G, then we can find an optimal allocation on G for any instance in time* $O(tn^{t+1})$.

Combining the two approaches from Corollary 5.6 and Proposition 5.9, we have a time complexity of $O(\min(nr!, tn^{t+1}))$. An immediate application of this dynamic programming algorithm is for graphs with degree at most one. These graphs are special cases of the disjoint union of paths where the path length can either be 0 or 1. By Proposition 5.9, we can find an optimal allocation for these instances in time $O(n^3)$.

Perhaps remarkably, there is no particularly elegant structural characterization when the underlying graph G is a disjoint union of paths *and* cycles, even when there is only one path and one cycle. This is a consequence of Fig. 6.

Finally, a similar result holds for disjoint unions of stars, though the proof is somewhat different. We omit the proof of Corollary 5.12, which follows from Theorem 5.11.

Theorem 5.11 *Let G be a disjoint union of stars,* $K_{1,n_1} + \dots + K_{1,n_r}$. *Then G is strongly splittable. Furthermore, in any optimal allocation, within each star, the houses appear in the form characterized in Theorem 4.1.*

Proof We "split" any two stars while improving on our objective. Consider two stars K_{1,n_1} and K_{1,n_2}. Let π be any optimal allocation that allocates the values $a_1, \dots, a_{n_1+1}$ to K_{1,n_1} and $b_1, \dots, b_{n_2+1}$ to K_{1,n_2}.

We provide a simple two-step procedure that creates a new allocation π' that allocates contiguous intervals to both stars and attains total envy at most that of π. In the first step, we simply re-arrange the values allocated to each star to ensure they satisfy the characterization for an optimal envy allocation from Theorem 4.1. In the second step, assuming without loss of generality the center of K_{1,n_1} has a lower value than that of K_{1,n_2}, we re-arrange the values allocated to the spokes of both stars by allocating the least n_1 values to K_{1,n_1} and the greatest n_2 values to K_{1,n_2}; crucially, we do not change the value allocated to the center of either star. It is easy to see that neither of these steps can increase the total envy: this is immediate by design in the first step, and follows from a similar argument to the proof of Theorem 5.5 in the second step.

It is also easy to see that, if the stars are not allocated contiguous intervals, the above two step procedure changes the allocation and strictly reduces the envy. This shows that not allocating contiguous intervals to each star is sub-optimal. □

Corollary 5.12 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of stars in time* $\widetilde{O}(nr!)$*, where r is the number of stars.*

5.3 Disjoint unions of cliques

We now turn our attention to disjoint unions of cliques. We first demonstrate that when all cliques have the same size, we maintain strong splittability.

Theorem 5.13 *Let G be a disjoint union of cliques with equal sizes,* $K^1_{n/r} + \cdots + K^r_{n/r}$. *Then, G is strongly splittable.*

Proof We prove the result for the case of two cliques $K_{n/2} + K_{n/2}$. The result for r cliques follows by showing that each pair of cliques must be split from each other.

Let (V, E) and (V', E') be the set of vertices and edges of each copy of $K_{n/2}$. Let $\tau : V \rightarrow V'$ be any bijective mapping from V to V'.

Let π be any allocation on $K_{n/2} + K_{n/2}$, we show that if π does not allocate contiguous intervals to each component, we can create a better allocation π'.

Let $a_1 < a_2 < \ldots < a_{n/2}$ be the values allocated to the nodes in V and $b_1 < b_2 < \ldots < b_{n/2}$ be the values allocated to the nodes in V' in some optimal allocation π. We rearrange the goods allocated to V' such that if node $v \in V$ receives a_i, then node $\tau(v)$ receives $b_{n/2-i}$. This does not change the total envy of the allocation.

If each component is not allocated a contiguous interval, the least-valued $n/2$ houses must have some a values and some b values. Let's call the least-valued $n/2$ houses H' and let's say there are k a_i's in H'. Therefore H' contains $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_{n/2-k}$.

We create a new allocation π' from π as follows. For all $i \in [k]$, we swap a_i with $b_{n/2-i}$. Note that for each house among the least-valued $n/2$ houses, if a_i is allocated to $v \in V$, we swap the houses given to v and $\tau(v)$. This has been pictorially described in Fig. 9.

Let us now compute the change in envy between π' and π. We do this by showing that, for every edge $(u, v) \in E$, the total sum of the envies along the edges (u, v) and $(\tau(u), \tau(v))$ decreases.

Case 1 u **and** v **are unaffected by the swap.** Then $\tau(u)$ and $\tau(v)$ are unaffected as well. Therefore the total envy along these two edges does not change.

Case 2 u **and** v **are both affected by the swap.** Then, $envy_{\pi'}(u, v) = envy_{\pi}(\tau(u), \tau(v))$ and $envy_{\pi}(u, v) = envy_{\pi'}(\tau(u), \tau(v))$. Therefore, the total envy along these two edges does not change.

Case 3 Only u **is affected by the swap.** This means $\tau(v)$ is not affected by the swap. The total envy along these two edges under π is

$$envy_{\pi}(u, v) + envy_{\pi}(\tau(u), \tau(v)) = (a_j - a_i) + (b_{n/2-i} - b_{n/2-j})$$

where $j > k > i$. This can be re-written as

$$\begin{aligned} envy_{\pi}(u, v) + envy_{\pi}(\tau(u), \tau(v)) = 2\min\{a_j, b_{n/2-i}\} + |a_j - b_{n/2-i}| \\ - 2\max\{a_i, b_{n/2-j}\} + |a_i - b_{n/2-j}| \ . \end{aligned}$$

The total envy along these two edges under π' is

$$envy_{\pi'}(u, v) + envy_{\pi'}(\tau(u), \tau(v)) = |a_j - b_{n/2-i}| + |a_i - b_{n/2-j}| \ .$$

The change in envy is

$$2\max\{a_i, b_{n/2-j}\} - 2\min\{a_j, b_{n/2-i}\} < 0 \ .$$

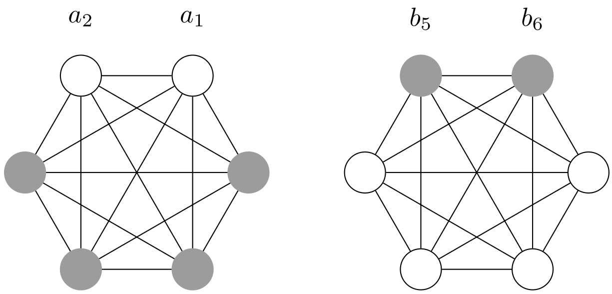


Fig. 9 A pictorial description of the allocation π in the proof of Theorem 5.13. Shaded nodes denote nodes that are allocated one of the highest $n/2$ valued houses. To construct π' from π, we swap the houses allocated to the unshaded nodes on the left clique with those allocated to the shaded nodes on the right clique

The inequality holds since $j > k > i$.

When $k \geq 1$, at least one edge belongs to Case 3 and so the total envy of π' is strictly less than the total envy of π. □

Because the cliques are all of equal sizes and agents have identical valuations, Theorem 5.13 implies that there is a trivial algorithm for assigning houses to agents. We can assign the first n/r houses to one clique, the next n/r houses to the next clique, and so on.

Corollary 5.14 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of equal-sized cliques in time $\widetilde{O}(n)$.*

We now turn our attention to the case when the cliques are not all of the same size.

As Fig. 6 demonstrates, strong splittability must be ruled out when cliques have different sizes. We will show that splittability still holds. We show further that the largest clique splits all other cliques, the second largest clique splits all cliques except (possibly) the largest one, and so on. The detailed proof is quite technical, and is relegated to Appendix C.

Theorem 5.15 *Let G be a disjoint union of cliques with arbitrary sizes, $K_{n_1} + \cdots + K_{n_r}$, where $n_1 \geq \ldots \geq n_r$. Then, G is splittable (but not necessarily strongly splittable if the n_i's are not all equal). In particular, for all $1 \leq i < j \leq r$, in every optimal allocation, K_{n_i} splits K_{n_j}.*

Theorem 5.15 implies an XP algorithm for finding a minimum envy allocation on unions of cliques. We state this formally as a corollary here.

Corollary 5.16 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of cliques in time $O(n^{r+2})$, where r is the number of cliques.*

Proof We sort the cliques in a non-increasing order of their size to get r cliques $K^1, K^2, \ldots, K^r$ such that $|K^1| \geq |K^2| \geq \ldots \geq |K^r|$. From Theorem 5.15, we know that K^1 receives a contiguous set of values in the optimal allocation, subject to which, K^2 must receive a contiguous set of values among the remaining houses, and so on.

This gives us a recursive procedure where we try out all possible contiguous sets of values of size $|K^1|$ to give to K^1 and subject to that, we try out all possible contiguous sets of values to give to K^2 and so on. From Theorem 5.15, we know that one of these allocations will be optimal, so we output the allocation we find with the lowest envy in this way.

The pseudocode is presented in Algorithm 1. The algorithm maintains a *partial* allocation π and updates it using recursive calls.

Algorithm 1 Minimum Envy House Allocation on Cliques

```
procedure FindMinEnvy(N, H, {K^i}_{i∈[r]}, v)
    sort {K_i} so that |K^1| ≥ ... ≥ |K^r|.
    if r = 1 then
        Let π be any allocation of the houses in H to agents in K^1
        envy = FindEnvy(π, v, K^1)
        return envy, π
    else
        π* ← ∅, envy* ← ∞
        for every |K^1|-sized contiguous subset of values S do
            Let π_{K^1} be any allocation of the houses in S to agents in K^1
            envy^S, π^S = FindMinEnvy(N − |K^1|, H \ S, {K^{i+1}}_{i∈[r−1]}, v)
            if envy* > envy^S + FindEnvy(π_{K^1}, v, K^1) then
                π* ← π^S
                envy* ← envy^S
        return envy*, π*
```

To analyze the time complexity, note that we compute at most $O(n^r)$ allocations. For each allocation, finding the envy of the allocation takes $O(n^2)$ time trivially. Note that the sorting step is just $O(r \log r)$, which is $o(n^2)$, and is therefore subsumed by the other term. This gives us a total time complexity of $O(n^{r+2})$. □

There seems to be a separation between unions of differently-sized cliques and unions of stars, cycles, paths, or equi-sized cliques. We suspect the problem may be W[1]-hard for unions of arbitrary cliques.

5.4 Disjoint unions of complete bipartite graphs

We can extend the techniques used in Sect. 5.3 to prove splittability guarantees for complete bipartite graphs as well. The proofs in this section are significantly more involved than the proofs in the previous section and are relegated to the appendix.

Combining techniques from Theorem 4.6 and Theorem 5.13, we can show that disjoint unions of identical complete bipartite graphs are strongly splittable.

Theorem 5.17 *If $G = K_{r,s}$ for any $r, s \in \mathbb{N}$, then $G + G$ is strongly splittable.*

Note that, as in Sect. 5.3, we can leverage Theorem 5.17 and Theorem 4.6 to give us an easy FPT algorithm on disjoint unions of *identical* complete bipartite graphs. We state this as a corollary without proof, as it is very similar to Corollary 5.14.

Corollary 5.18 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of identical complete bipartite graphs in time* $\widetilde{O}(n)$.

Next, we combine techniques from Theorem 4.6 and Theorem 5.15 to show that disjoint unions of (unequal-sized) symmetric bipartite graphs $\{K_{r,r}\}_{r\in\mathbb{N}}$ are splittable but not strongly splittable.

Theorem 5.19 *Let G be a disjoint union of symmetric complete bipartite graphs* $K_{n_1,n_1} + K_{n_2,n_2} + \cdots + K_{n_\ell,n_\ell}$, *where* $n_1 \geq n_2 \geq \ldots \geq n_\ell$. *Then G is splittable (but not necessarily strongly splittable if* $n_1 > n_\ell$*) and the order of splittability is* $K_{n_1,n_1}, \ldots, K_{n_\ell,n_\ell}$.

This is one of our most technically involved proofs, and it can be found in Appendix C.

The following proposition shows that for these graphs, strong splittability can be ruled out almost immediately, and so splittability is really the best property to hope for.

Proposition 5.20 *Disjoint unions of (unequal) symmetric complete bipartite graphs are not necessarily strongly splittable.*

Proof Consider $K_{1,1} + K_{2,2}$, which is the disjoint union of an edge and a 4-cycle. Consider an instance $\{h_1, \ldots, h_6\}$ where $v(h_1) = 0$, $v(h_6) = 1$, and the values $v(h_2), \ldots, v(h_5)$ are concentrated in an ϵ-interval around 0.5. Then, any optimal allocation assigns h_1 and h_6 to the $K_{1,1}$, showing that the graph is not strongly splittable. □

We end by noting that Theorem 5.19 immediately implies an XP algorithm to compute a minimum envy allocation over the disjoint union of symmetric complete bipartite graphs. We state this below but omit the proof, as it is similar to Corollary 5.16.

Corollary 5.21 *We can find an optimal allocation for an instance on an undirected n-agent graph G that is the disjoint union of symmetric complete bipartite graphs in time* $n^{O(r)}$, *where r is the number of symmetric complete bipartite graphs.*

We have shown strong splittability for disjoint copies of identical complete bipartite graphs and splittability for symmetric complete bipartite graphs. We conjecture that the disjoint unions of arbitrary complete bipartite graphs are splittable as well. This result would generalize Theorems 5.19 and 5.11.

Conjecture 5.22 *Let G be the disjoint union of arbitrary complete bipartite graphs. Then G is splittable.*

5.5 Splittability and graph properties

It is worth asking the question of whether there is a clear structural property of a graph that determines whether it is splittable or strongly splittable. This would allow us to generalize beyond specific classes of graphs, and state purely structural results that would generalize several results from Sects. 5.2, 5.3, and 5.4 under one compact umbrella.

From the graph classes considered in those sections, let us examine the ones that are regular. These would include disjoint unions of edges, cycles, equal-sized cliques, or identical symmetric complete bipartite graphs. By Theorems 5.5, 5.7, 5.13, and 5.17, we know that each of those graphs is strongly splittable. This might lead us to conjecture that disconnected regular graphs are strongly splittable as well. The following proposition shows that this is not the case. In fact, they need not even be splittable.

Proposition 5.23 *There exists a 3-regular unsplittable graph.*

Before we delve into the proof, we first need to define a *bicycle* graph.

Definition 5.24 For any odd number $2t + 1$, take the cycle C_{2t+1}, and suppose its vertices are $\{v_1, \dots, v_{2t+1}\}$ in order along the cycle. Now, add every edge (v_i, v_{t+i}) for $1 \leq i \leq t$. This defines a graph where every vertex except for v_{2t+1} has degree 3. Call this a $(2t + 1)$-*wheel* W_{2t+1}, and call v_{2t+1} its *rim*. Observe that m-wheels exist for every odd $m \geq 3$. Now, for any two odd numbers m_1, m_2, define the (m_1, m_2)-*bicycle* B_{m_1,m_2} as the graph obtained from a W_{m_1} and a W_{m_2} by joining the two rims by an edge. Note that every bicycle is 3-regular.

See Fig. 10 for examples of wheels and bicycles.

We are now ready to prove Proposition 5.23. Consider the graph G which is the disjoint union of two bicycles, $B_{401,201} + B_{301,101}$, and consider a valuation interval with four equispaced clusters with 401, 301, 201, and 101 values in those clusters in order, as shown in Fig. 11. It can be shown that any optimal allocation needs to place the entirety of W_{401} in the first cluster, the entirety of W_{301} in the second cluster, the entirety of W_{201} in the third cluster, and the entirety of W_{101} in the fourth cluster, contradicting splittability. The details of the proof are in Appendix C.

Finally, from the graph classes considered in Sects. 5.2, 5.3, and 5.4, let us examine the ones formed by taking the disjoint union of identical copies of the same graph. As stated before, every single one of those examples has corresponded to a strongly splittable graph, which again might lead to the very natural conjecture that disconnected graphs obtained by taking disjoint unions of the same connected graph are strongly splittable. The following proposition shows that this is not the case, and in fact, shows unsplittability.

Proposition 5.25 *There exists a connected graph G such that $G + G$ is unsplittable.*

Proof Consider the graph $G + G$ shown in Fig. 12, along with the valuation interval. The connected component G consists of a clique K_a, joined by an edge to a clique K_b, joined by an edge to a clique K_c, where $a \gg b \gg c \gg 1$. The valuation interval consists of six clusters of width ϵ each, consisting of a, a, b, b, c, and c values in order. Of course, in any optimal allocation, none of the K_a's can have any presence outside of the first two clusters, as then there will be many edges crossing over at least one of the intervals. By a similar argument, each of the K_a's needs be entirely within one of the first two clusters. By similar arguments, it can be shown that each of the K_b's needs to be inside one of the third and fourth clusters, and each of the K_c's needs to be inside one of the last two clusters. But now, no matter how we distribute the clusters among the two copies of G, this cannot be splittable, as neither copy can receive a contiguous subset of the values along the interval.

□

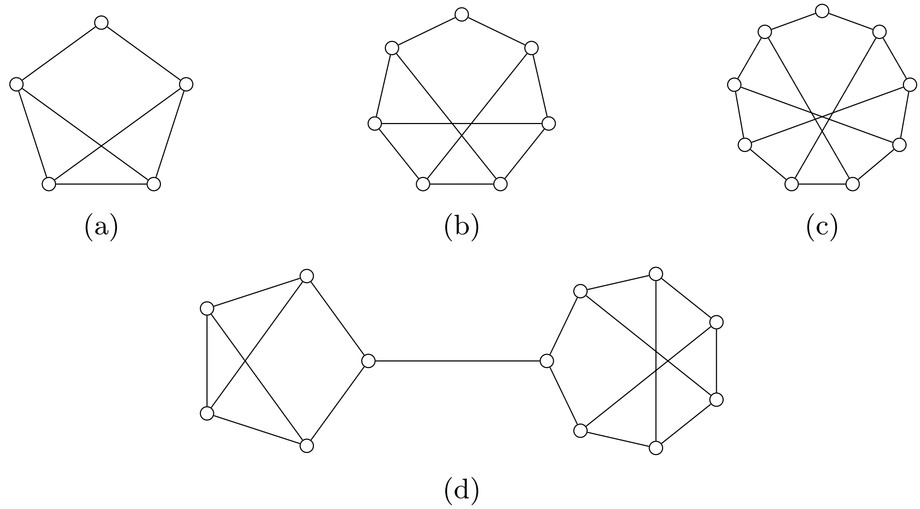


Fig. 10 **a–c** contain the wheels W_5, W_7, and W_9 respectively. In each case, the rim is the vertex of degree 2 at the top. **d** describes the bicycle $B_{5,7}$, which is 3-regular

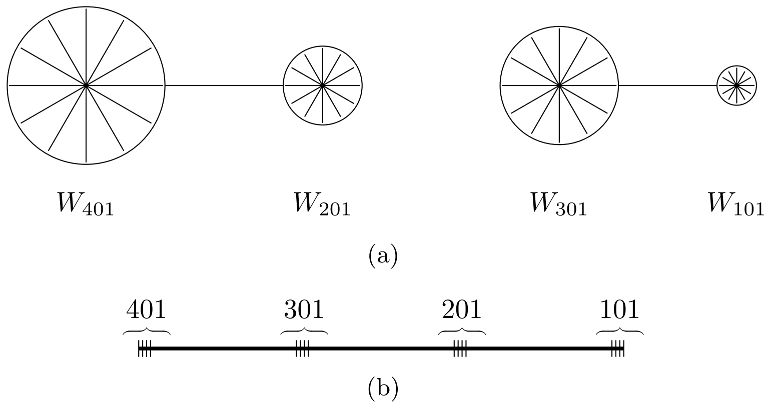


Fig. 11 The instance proving Proposition 5.23. Note that the vertices on the outer cycles are just connected to other vertices on these cycles, not to any central vertex

Propositions 5.23 and 5.25 above show counterexamples to seemingly quite reasonable conjectures, and pave the way for an in-depth investigation into the mysterious property of splittability. We relegate this to future work.

6 Conclusion and discussions

We investigated a generalization of the classical house allocation problem where the agents are on the vertices of a graph representing the underlying social network, under the condition that the agents have identical valuations. We wish to allocate the houses to the agents so as to

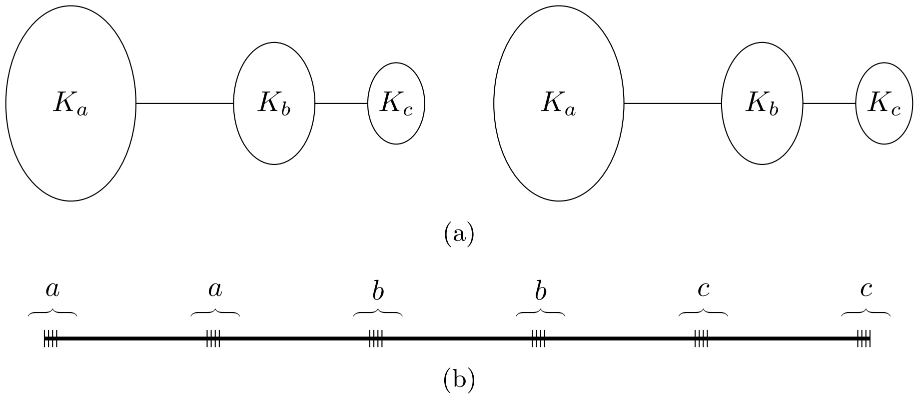


Fig. 12 The instance proving Proposition 5.25

minimize the aggregate envy among neighbors. Even for identical valuations, we showed that the problem is computationally hard and structurally rich. Furthermore, our structural insights facilitate algorithmic results for several natural and well-motivated graph classes.

There are a few natural questions for future research. We might consider other fairness objectives such as *minimizing the maximum envy* present on any edge of the graph. For evenly-spaced valuations, this corresponds to the classical graph theoretic property of *bandwidth*, which is also known to be NP-complete for general graphs, and hard to approximate as well [41, 42]. It would be interesting to know whether trees admit polynomial time characterizations of the minimum envy, or—more remarkably—whether they are NP-complete but admit the structural similarities to the Minimum Linear Arrangement problem discussed in Sect. 4.5. We might hope to completely characterize all strongly splittable graphs in terms of their graph theoretic structure. Another important future direction would be to extend some of these results for non-identical valuations.

7 Supplementary information

This article has an accompanying appendix which is 15 pages long. References appear after the appendix as required by the Springer Nature format. An accompanying information sheet has also been submitted as part of the supplementary material in accordance with the submission guidelines.

Appendix A: Distinct valuations

Lemma A.1 *Given any instance* (N, H, G, v) *of* Graphical House Allocation, *there exists a valuation function* v' *such that* v' *gives each house a distinct value, and any optimal allocation under* v' *is also optimal under* v.

Proof Let $\delta > 0$ be the smallest nonzero envy difference between two allocations of H to G under the valuation v, and let $\gamma > 0$ be the smallest nonzero difference between the values of two houses. If either δ or γ are not well-defined, then all allocations have the same optimal envy, and we can define any arbitrary one-to-one function v' to satisfy the lemma. So assume both δ and γ are well-defined and positive. Define $\epsilon = \min\{\delta/2, \gamma\}$. We will show that there is a one-to-one valuation function v', such that for any allocation π, the total envy under v' differs from the total envy under v by at most an additive term of ϵ. For $h_k \in H$, define

$$v'(h_k) := v(h_k) + \frac{\epsilon}{n^2 2^k}.$$

It is easy to see that this function is one-to-one by the definition of ϵ. For any allocation π on G, consider the envy between agents i and j. If $\pi(i) = h_k$ and $\pi(j) = h_\ell$, we have, using the triangle inequality,

$$\begin{aligned} \left|v'(\pi(i)) - v'(\pi(j))\right| &= \left|v(\pi(i)) - v(\pi(j)) + \frac{\epsilon}{n^2 2^k} - \frac{\epsilon}{n^2 2^\ell}\right| \\ &\leq \left|v(\pi(i)) - v(\pi(j))\right| + \frac{\epsilon}{n^2}\left|\frac{1}{2^k} - \frac{1}{2^\ell}\right| \\ &< \left|v(\pi(i)) - v(\pi(j))\right| + \frac{\epsilon}{n^2}. \end{aligned}$$

We also similarly have

$$\begin{aligned} \left|v'(\pi(i)) - v'(\pi(j))\right| &= \left|v(\pi(i)) - v(\pi(j)) + \frac{\epsilon}{n^2 2^k} - \frac{\epsilon}{n^2 2^\ell}\right| \\ &\geq \left|v(\pi(i)) - v(\pi(j))\right| - \frac{\epsilon}{n^2}\left|\frac{1}{2^k} - \frac{1}{2^\ell}\right| \\ &> \left|v(\pi(i)) - v(\pi(j))\right| - \frac{\epsilon}{n^2}. \end{aligned}$$

Summing over the at most n^2 edges of G, we have $\mathsf{Envy}_v(G,\pi) - \epsilon < \mathsf{Envy}_{v'}(G,\pi) < \mathsf{Envy}_v(G,\pi) + \epsilon$, as desired, where the subscripts v and v' denote the valuation functions being used in each case.

For any allocation π^* which minimizes envy under v', if we compare against another allocation π' such that π^* and π' have different total envies under v, we see that

$$\mathsf{Envy}_v(G,\pi^*) - \epsilon < \mathsf{Envy}_{v'}(G,\pi^*) \leq \mathsf{Envy}_{v'}(G,\pi') < \mathsf{Envy}_v(G,\pi') + \epsilon.$$

By the definition of $\epsilon = \min\{\delta/2, \gamma\}$, we can infer that if π^* is optimal under v', then it must be optimal under v as well. If π^* is not optimal under v, then there will be an allocation π' which is optimal under v that violates the inequality above; that is, we will have $\mathsf{Envy}_v(G,\pi') \leq \mathsf{Envy}_v(G,\pi^*) - 2\epsilon$ by the definition of ϵ. □

Appendix B: Technical Proofs from Section 4

Theorem 4.6 *When G is the graph $K_{r,s}$ ($r > s$), the minimum envy allocation π^* has the following property*:

(1) *If $r - s =: 2m$ is even, then the first and last m houses are allocated to the larger part, and for all $i \in [s]$, the houses h_{m+2i-1} and h_{m+2i} are allocated to different parts.*
(2) *If $r - s =: 2m + 1$ is odd, then the first m and last $m + 1$ houses are allocated to the larger part. For all $i \in [s]$, the houses h_{m+2i-1} and h_{m+2i} are allocated to the larger and smaller parts respectively.*

Moreover, all allocations which satisfy this property have the same (optimal) envy.

Proof This proof is very similar to that of Theorem 4.5. Again, for notational ease, let the graph have bipartition (L, R), with $|L| = r > s = |R|$. We refer to the properties in the theorem statement when $r - s$ is even and odd as the *optimal even* property and the *optimal odd* property respectively. This proof will also use the notation $n^{<}_{L,\pi}(x)$, $n^{>}_{L,\pi}(x)$, $n^{<}_{R,\pi}(x)$ and $n^{>}_{R,\pi}(x)$ defined in Definition 2.3.

Case 1 $r - s$ is even. We split the proof into two claims.

Claim B.1 *Any optimal allocation allocates the first m houses to agents in L.*

Proof of Claim B.1 Assume for contradiction that this is not true. That is, there is an optimal allocation π such that:

$$\pi(h_j) \in L \text{ for all } j \in [k] \text{ for some } 0 \le k < m,$$
$$\pi(h_{k+j}) \in R \text{ for all } j \in [l] \text{ for some } l > 0,$$
$$\pi(h_{k+l+1}) \in L.$$

Create an allocation π' from π by swapping h_{k+l} and h_{k+l+1}. We can now compare the aggregate envy of π and π' using arguments similar to those in Theorem 4.5.

$$\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G)\\
&= [n^{<}_{L,\pi}(v(h_{k+l+1})) - n^{>}_{L,\pi}(v(h_{k+l+1}))](v(h_{k+l+1}) - v(h_{k+l}))\\
&\quad + [n^{>}_{R,\pi}(v(h_{k+l})) - n^{<}_{R,\pi}(v(h_{k+l}))](v(h_{k+l+1}) - v(h_{k+l}))\\
&= (v(h_{k+l+1}) - v(h_{k+l}))\\
&\quad [n^{<}_{L,\pi}(v(h_{k+l+1})) - n^{>}_{L,\pi}(v(h_{k+l+1})) + n^{>}_{R,\pi}(v(h_{k+l})) - n^{<}_{R,\pi}(v(h_{k+l}))]\\
&= [k - (r - (k + 1)) + (s - l) - (l - 1)](v(h_{2i}) - v(h_{2i-1}))\\
&= [2k - (r - s) + 2 - 2l](v(h_{2i}) - v(h_{2i-1}))\\
&< 0.
\end{aligned}$$

The last inequality follows from the fact that $l \ge 1$ and $k < m = (r - s)/2$. This contradicts the optimality of π. □

Claim B.2 *In any optimal allocation, for any $i \in [s]$, h_{m+2i-1} and h_{m+2i} cannot be allocated to the same part.*

Proof of Claim B.2 Assume for contradiction that this is not true. Let π be an optimal allocation that satisfies Claim B.1 but not Claim B.2. Choose j as the least i such that h_{m+2i-1} and h_{m+2i} are allocated to the same part, say L. Let $\{h_{m+2j-1}, h_{m+2j}, \dots, h_{m+2j+k}\}$ be a set of houses allocated to agents in L such that $h_{m+2j+k+1}$ is allocated to some agent in R ($k \ge 0$). Create an allocation π' from π by swapping h_{m+2j+k} and $h_{m+2j+k+1}$. We can compare the envy between π' and π.

$$
\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G) \\
&\quad = [n^{>}_{L,\pi}(v(h_{m+2j+k})) - n^{<}_{L,\pi}(v(h_{m+2j+k}))](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\qquad + [n^{<}_{R,\pi}(v(h_{m+2j+k+1})) - n^{>}_{R,\pi}(v(h_{m+2j+k+1}))](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\quad = (v(h_{m+2j+k+1}) - v(h_{m+2j+k}))[n^{>}_{L,\pi}(v(h_{m+2j+k})) - n^{<}_{L,\pi}(v(h_{m+2j+k})) \\
&\qquad + n^{<}_{R,\pi}(v(h_{m+2j+k+1})) - n^{>}_{R,\pi}(v(h_{m+2j+k+1}))] \\
&\quad = (v(h_{m+2j+k+1}) - v(h_{m+2j+k}))[(r - (m + k + 2 + j - 1)) \\
&\qquad - (m + k + 1 + j - 1) + (j - 1) - (s - j)] \\
&\quad = [2j - 2(k + j) - 2](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\quad = [-2k - 2](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\quad < 0.
\end{aligned}
$$

The final inequality holds since $k \geq 0$. Again, we contradict the optimality of π. □

Claim B.2 also implies that none of the final $m = (r - s)/2$ houses are allocated to agents in R; this is because all agents in R have already been assigned houses by Claim B.2. We can therefore conclude that these houses must be allocated to agents in L in any optimal allocation.

To show that any allocation that satisfies the optimal even property has the same aggregate envy, we use a swapping based argument similar to Theorem 4.5. Let π be any allocation that satisfies the optimal even property. Pick an arbitrary $i \in [s]$ and let π' be the allocation that results from swapping h_{m+2i-1} and h_{m+2i} in π. Assume that h_{m+2i-1} is allocated to L in π. The proof for R flows similarly. Let us compare the envy of the two allocations.

$$
\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G) \\
&\quad = [n^{>}_{L,\pi}(v(h_{m+2i-1})) - n^{<}_{L,\pi}(v(h_{m+2i-1}))](v(h_{m+2i}) - v(h_{m+2i-1})) \\
&\qquad + [n^{<}_{R,\pi}(v(h_{m+2i})) - n^{>}_{R,\pi}(v(h_{m+2i}))](v(h_{m+2i}) - v(h_{m+2i-1})) \\
&\quad = (v(h_{m+2i}) - v(h_{m+2i-1})) \\
&\qquad [n^{>}_{L,\pi}(v(h_{m+2i-1})) - n^{<}_{L,\pi}(v(h_{m+2i-1})) + n^{<}_{R,\pi}(v(h_{m+2i})) - n^{>}_{R,\pi}(v(h_{m+2i}))] \\
&\quad = [(r - (i + m)) - (m + i - 1) + (i - 1) - (s - i)](v(h_{m+2i}) - v(h_{m+2i-1})) \\
&\quad = 0.
\end{aligned}
$$

Case 2 $r - s$ is odd. This is, unsurprisingly, very similar to the previous case. We similarly split the proof into two claims.

Claim B.3 *Any optimal allocation allocates the first m houses to agents in L.*

The proof of this claim is exactly the same as the proof to the Claim B.1. The key difference in this case is that $m = (r - s - 1)/2$ but this does not affect the proof as we can still use the inequality $k < (r - s)/2$ since $k < m$. So we move on to the second claim.

Claim B.4 *In any optimal allocation, for any $i \in [s]$, h_{m+2i-1} is allocated to some agent in L and h_{m+2i} is allocated to some agent in R.*

Proof This proof is again very similar to Claim B.2. However, there are some subtle differences.

Assume for contradiction that the claim is not true. Let π be an optimal allocation that satisfies Claim B.3 but not Claim B.4. Choose j as the least i where the claim is violated. That is, either h_{m+2j-1} is allocated to R or h_{m+2j} is allocated to L. In this proof, we assume the latter has occured. The proof for the former is very similar. In other words, both h_{m+2j-1} and h_{m+2j} are allocated to some agents in L. Let $h_{m+2j-1}, h_{m+2j}, \dots, h_{m+2j+k}$ be a set of houses allocated to agents in L such that $h_{m+2j+k+1}$ is allocated to some agent in R. Let π' be the allocation that results from swapping h_{m+2j+k} and $h_{m+2j+k+1}$. We can compare the envy between π' and π:

$$
\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G) \\
&\quad = [n^{>}_{L,\pi}(v(h_{m+2j+k})) - n^{<}_{L,\pi}(v(h_{m+2j+k}))](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\qquad + [n^{<}_{R,\pi}(v(h_{m+2j+k+1})) - n^{>}_{R,\pi}(v(h_{m+2j+k+1}))](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\quad = (v(h_{m+2j+k+1}) - v(h_{m+2j+k}))[n^{>}_{L,\pi}(v(h_{m+2j+k})) - n^{<}_{L,\pi}(v(h_{m+2j+k})) \\
&\qquad + n^{<}_{R,\pi}(v(h_{m+2j+k+1})) - n^{>}_{R,\pi}(v(h_{m+2j+k+1}))] \\
&\quad = (v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\qquad [(r - (m + k + 2 + j - 1)) - (m + k + 1 + j - 1) + (j - 1) - (s - j)] \\
&\quad = [2j - 2(k + j) - 1](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\quad = [-2k - 1](v(h_{m+2j+k+1}) - v(h_{m+2j+k})) \\
&\quad < 0.
\end{aligned}
$$

The final inequality holds since $k \geq 0$. The optimality of π has been contradicted. □

Claim B.4 also implies that none of the final $m + 1$ houses are allocated to agents in R. We can therefore conclude that these houses must be allocated to agents in L in any optimal allocation.

Note that the optimal odd property specifies exactly which houses must be allocated to L and R in any optimal allocation. Any two allocations which satisfy the optimal odd property can only differ over which agents in L and R houses are allocated to and not which houses are allocated to L and R. It is easy to see that this difference cannot lead to a difference in envy over the complete bipartite graph. □

Appendix C: Technical Proofs from Section 5

Theorem 5.15 *Let G be a disjoint union of cliques with arbitrary sizes, $K_{n_1} + \cdots + K_{n_r}$, where $n_1 \geq \ldots \geq n_r$. Then, G is splittable (but not necessarily strongly splittable if the n_i's are not all equal). In particular, for all $1 \leq i < j \leq r$, in every optimal allocation, K_{n_i} splits K_{n_j}.*

Proof Let π be any minimum envy allocation. Assume for contradiction that there exist two cliques (say K and K') such that $|K| > |K'|$ and K does not receive a contiguous set of valuations with respect to the houses in $K \cup K'$. The case where $|K| = |K'|$ has been shown in Theorem 5.13. Let the houses in $K \cup K'$ have values $\{a_1, a_2, \dots, a_{|K \cup K'|}\}$ such that $a_1 < a_2 < \dots < a_{|K \cup K'|}$. Since each house has a unique value, we refer to houses using their values for the rest of this proof.

By our assumptions, the houses allocated to K must be split. Therefore there must be some houses in K' that are better than the houses allocated to some nodes in K and worse than houses allocated to other nodes in K. This can be formalized as follows

$$\pi(a_j) \in K' \text{ for all } j \in [\ell] \text{ and some } \ell \geq 0$$
$$\pi(a_{l+j}) \in K \text{ for all } j \in [m] \text{ and some } m > 0$$
$$\pi(a_{l+m+j}) \in K' \text{ for all } j \in [k] \text{ and some } k > 0$$
$$\pi(a_{l+m+k+1}) \in K$$

We will frequently use the notation $n^{<}_{K,\pi}(x)$ and $n^{>}_{K,\pi}(x)$ (defined in Definition 2.3) for each clique K.

Construct the allocation π' starting at π and swapping the houses a_{l+m+k} and $a_{l+m+k+1}$. For any node in K whose value is less than $a_{l+m+k+1}$ under π, the total envy between them and their neighbors increases by $a_{l+m+k+1} - a_{l+m+k}$ in π'. For any node in K whose value is greater than $a_{l+m+k+1}$ under π, the total envy between them and their neighbors decreases by $a_{l+m+k+1} - a_{l+m+k}$ in π'. We can show something similar for K'. This gives us the total change in envy as

$$\begin{aligned}
&\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G) \\
&\quad = \mathsf{Envy}(\pi', K \cup K') - \mathsf{Envy}(\pi, K \cup K') \\
&\quad = \left[n^{<}_{K',\pi}(a_{l+m+k}) - n^{>}_{K',\pi}(a_{l+m+k})\right]\left(a_{l+m+k+1} - a_{l+m+k}\right) \\
&\qquad + \left[n^{>}_{K,\pi}(a_{l+m+k+1}) - n^{<}_{K,\pi}(a_{l+m+l+1})\right]\left(a_{l+m+k+1} - a_{l+m+k}\right) \\
&\quad = \left(a_{l+m+k+1} - a_{l+m+k}\right) \\
&\qquad \left[n^{<}_{K',\pi}(a_{l+m+k}) - n^{>}_{K',\pi}(a_{l+m+k}) + n^{>}_{K,\pi}(a_{l+m+k+1}) - n^{<}_{K,\pi}(a_{l+m+l+1})\right] \\
&\quad = \left[(l + k - 1) - (|K'| - l - k) + (|K| - (m + 1)) - m\right](a_{l+m+k+1} - a_{l+m+k}) \\
&\quad = \left[|K| - |K'| + 2(l + k) - 2m - 2\right](a_{l+m+k+1} - a_{l+m+k})
\end{aligned}$$

Note that due to the optimality of π, we must have $\mathsf{Envy}(\pi', G) - \mathsf{Envy}(\pi, G) \geq 0$. Since $a_{l+m+k+1} - a_{l+m+k} > 0$ by construction, this implies $|K| - |K'| + 2(l + k) - 2m - 2 \geq 0$. Removing the -2, we get $|K| - |K'| + 2(l + k) - 2m > 0$. This gives us the following observation.

Observation C.1 $|K'| - |K| - 2(l + k) + 2m < 0$

Construct another allocation π'' as follows: start at π and for every $j \in [\min\{m, k\}]$, swap $a_{l+m+1-j}$ with a_{l+m+j}. In each swap, we swap one house in K with one house in K'. Using a similar argument, we can compare the total envy of π'' and π.

$$
\begin{aligned}
&\mathsf{Envy}(\pi'', G) - \mathsf{Envy}(\pi, G)\\
&\quad = \mathsf{Envy}(\pi'', K \cup K') - \mathsf{Envy}(\pi, K \cup K')\\
&\quad = \left[n^{<}_{K,\pi}(a_{l+m+1-\min\{m,k\}}) - n^{>}_{K,\pi}(a_{l+m}) + n^{>}_{K',\pi}(a_{l+m+\min\{m,k\}}) - n^{<}_{K',\pi}(a_{l+m+1})\right]\\
&\qquad \left[\sum_{j\in[\min\{m,k\}]}(a_{l+m+j} - a_{l+m+1-j})\right]\\
&\quad = \left[(m - \min\{m,k\}) - (|K| - m) + (|K'| - (l + \min\{k,m\})) - l\right]\\
&\qquad \left[\sum_{j\in[\min\{m,k\}]}(a_{l+m+j} - a_{l+m+1-j})\right]\\
&\quad = \left[|K'| - |K| + 2m - 2(\min\{m,k\} + l)\right]\\
&\qquad \left[\sum_{j\in[\min\{m,k\}]}(a_{l+m+j} - a_{l+m+1-j})\right]
\end{aligned}
\tag{C1}
$$

Note that the second term is always strictly positive since $a_{l+m+j} > a_{l+m+1-j}$ for all $j \in \min\{m,k\}$. If we show that the first term $|K'| - |K| + 2m - 2(\min\{m,k\} + l)$ is negative, we contradict the optimality of π. We have two possible cases.

Case 1 $k \le m$. In this case, (C1) reduces to

$$
\begin{aligned}
&\mathsf{Envy}(\pi'', G) - \mathsf{Envy}(\pi, G)\\
&\quad = \left[|K'| - |K| + 2m - 2(k + l)\right]\left[\sum_{j\in[\min\{m,k\}]}(a_{l+m+j} - a_{l+m+1-j})\right]
\end{aligned}
$$

From Observation C.1, the first term is negative.

Case 2 $k > m$. In this case, (C1) reduces to

$$
\begin{aligned}
&\mathsf{Envy}(\pi'', G) - \mathsf{Envy}(\pi, G)\\
&\quad = \left[|K'| - |K| + 2m - 2(m + l)\right]\left[\sum_{j\in[\min\{m,k\}]}(a_{l+m+j} - a_{l+m+1-j})\right]\\
&\quad = \left[|K'| - |K| - 2l\right]\left[\sum_{j\in[\min\{m,k\}]}(a_{l+m+j} - a_{l+m+1-j})\right]
\end{aligned}
$$

Since $|K| > |K'|$ and $l \ge 0$, the first term is negative.

To conclude, it cannot be the case that the houses in K are split. □

Theorem 5.17 *If $G = K_{r,s}$ for any $r, s \in \mathbb{N}$, then $G + G$ is strongly splittable.*

Proof Let $(V = L \cup R, E)$ and $(V' = L' \cup R', E')$ be the set of vertices and edges of each copy of G. There exists a bijective mapping $\tau : V \mapsto V'$ such that for every node $v \in V$, $\tau(v) \in L'$ if and only if $v \in L$.

Let π be any allocation on $G + G$, we show that if π does not allocate contiguous intervals to each component, we can create a better allocation π'.

Let $a_1 < a_2 < \dots a_{r+s}$ be the values allocated to the nodes in V and $b_1 < b_2 < \dots b_{r+s}$ be the values allocated to the nodes in V' in some optimal allocation π. We rearrange the goods allocated to V' such that if node $v \in V$ receives a_i, then node $\tau(v)$ receives b_{r+s-i}. If the allocation of a values to V is optimal, then from our characterization of bipartite graphs (Theorem 4.6), we know that this allocation of b houses to V' is optimal as well.

If each component is not allocated a contiguous interval, the least valued $r + s$ houses must have some a values and some b values. Let's call the least valued $r + s$ houses H' and let's say there are k a_i's in H'. Therefore H' contains $a_1, a_2, \dots, a_k$ and $b_1, b_2, \dots, b_{r+s-k}$.

We create a new allocation π' starting at π and for all $i \in [k]$, we swap a_i with b_{r+s-i}. Note that for each house among the least-valued $r + s$ houses, if a_i is allocated to $v \in V$, we swap the houses given to v and $\tau(v)$, thereby creating π' from π.

Let us now compute the change in envy between π' and π. We do this by showing that, for every edge $(u, v) \in E$, the total sum of the envies along the edges (u, v) and $(\tau(u), \tau(v))$ decreases. Before we go into the math, note that if $(u, v) \in E$, then $(\tau(u), \tau(v)) \in E'$ by our definition of τ.

Case 1 u **and** v **are unaffected by the swap.** Then $\tau(u)$ and $\tau(v)$ are unaffected as well. Therefore the total envy along these two edges does not change.

Case 2 u **and** v **are both affected by the swap.** Then, $envy_{\pi'}(u, v) = envy_{\pi}(\tau(u), \tau(v))$ and $envy_{\pi}(u, v) = envy_{\pi'}(\tau(u), \tau(v))$. Therefore, the total envy along these two edges does not change.

Case 3 Only u **is affected by the swap.** This means $\tau(v)$ is not affected by the swap. The total envy along these two edges under π is

$$envy_{\pi}(u, v) + envy_{\pi}(\tau(u), \tau(v)) = (a_j - a_i) + (b_{r+s-i} - b_{r+s-j})$$

where $j > k > i$. This can be re-written as

$$\begin{aligned} envy_{\pi}(u, v) + envy_{\pi}(\tau(u), \tau(v)) = {} & 2\min\{a_j, b_{r+s-i}\} + |a_j - b_{r+s-i}| \\ & - 2\max\{a_i, b_{r+s-j}\} + |a_i - b_{r+s-j}| \end{aligned}$$

The total envy along these two edges under π' is

$$envy_{\pi'}(u, v) + envy_{\pi'}(\tau(u), \tau(v)) = |a_j - b_{r+s-i}| + |a_i - b_{r+s-j}|$$

The change in envy is

$$2\max\{a_i, b_{r+s-j}\} - 2\min\{a_j, b_{r+s-i}\} < 0$$

The inequality holds since $j > k > i$.

When $k \geq 1$, at least one edge belongs to Case 3 and so the total envy of π' is strictly less than the total envy of π. □

Theorem 5.19 *Let G be a disjoint union of symmetric complete bipartite graphs $K_{n_1,n_1} + K_{n_2,n_2} + \dots + K_{n_\ell,n_\ell}$, where $n_1 \geq n_2 \geq \dots \geq n_\ell$. Then G is splittable (but not necessarily strongly splittable if $n_1 > n_\ell$) and the order of splittability is $K_{n_1,n_1}, \dots, K_{n_\ell,n_\ell}$.*

Proof We prove complete symmetric bipartite graphs are not strongly splittable in Proposition 5.20, so we focus on proving splittability here. Consider two complete bipartite graphs $G_1 = K_{r,r}$ and $G_2 = K_{s,s}$ such that $r < s$. Assume houses with values $a_1, \dots a_{2r+2s}$ such that

$a_1 < \ldots < a_{2r+2s}$ are allocated to these two graphs. Since house values are unique, we will say the value a_i is allocated to a node j if the unique house with value a_i is allocated to the node j.

We need to show that $K_{s,s}$ is allocated a contiguous interval of values in at least one optimal allocation. Assume for contradiction that this is not true. Let π be an allocation where

$$a_1, \ldots, a_{\ell_1} \text{ is allocated to } G_1 \text{ for some } \ell_1 \geq 0$$
$$a_{\ell_1+1}, \ldots, a_{\ell_1+\ell_2} \text{ is allocated to } G_2 \text{ for some } \ell_2 > 0$$
$$a_{\ell_1+\ell_2+1}, \ldots, a_{\ell_1+\ell_2+\ell_3} \text{ is allocated to } G_1 \text{ for some } \ell_3 > 0$$
$$a_{\ell_1+\ell_2+\ell_3+1} \text{ is allocated to } G_2 \text{ for some } \ell_3 > 0$$

Since we assumed no optimal allocation gives a contiguous set of values to G_2, all optimal allocations must have the above structure for some ℓ_1, ℓ_2 and ℓ_3. If there are multiple optimal allocations, pick one such that ℓ_1 is maximized. Break any further ties by picking one such that ℓ_2 is maximized. Finally, break ties by ensuring $\ell_1 + \ell_2 + \ell_3$ is minimized. If there are still multiple envy minimizing allocations, pick one arbitrarily.

Since G_1 and G_2 are complete bipartite graphs, we refer to the nodes in the 'left' part of G_1 and G_2 using L_1 and L_2 respectively. Similarly, we refer to the 'right' part of nodes using R_1 and R_2. Since we assume π is optimal, the allocations to G_1 and G_2 must satisfy the structural properties from Theorem 4.5. Specifically, if the values $b_1, \ldots, b_{2y}$ are allocated to G_i for some $i \in [2]$, we assume $b_1, b_3, b_5, \ldots, b_{2y-1}$ are allocated to L_i.

Swap $a_{\ell_1+\ell_2+\ell_3}$ with $a_{\ell_1+\ell_2+\ell_3+1}$ in π to create a new allocation π'. Let us compare the envies of π and π'. Observe that

$$\begin{aligned}
&\mathsf{Envy}(\pi', G_1 + G_2) - \mathsf{Envy}(\pi, G_1 + G_2) \\
&= (a_{\ell_1+\ell_2+\ell_3+1} - a_{\ell_1+\ell_2+\ell_3}) \times \\
&\quad \Bigg[\left\lceil \frac{n^<_{G_1}(a_{\ell_1+\ell_2+\ell_3})}{2} \right\rceil - \left(r - \left\lceil \frac{n^<_{G_1}(a_{\ell_1+\ell_2+\ell_3})}{2} \right\rceil \right) \\
&\quad - \left\lceil \frac{n^<_{G_2}(a_{\ell_1+\ell_2+\ell_3+1})}{2} \right\rceil + \left(s - \left\lceil \frac{n^<_{G_2}(a_{\ell_1+\ell_2+\ell_3+1})}{2} \right\rceil \right) \Bigg] \\
&= (a_{\ell_1+\ell_2+\ell_3+1} - a_{\ell_1+\ell_2+\ell_3}) \times \left[2\left\lceil \frac{\ell_1 + \ell_3 - 1}{2} \right\rceil - 2\left\lceil \frac{\ell_2}{2} \right\rceil + s - r \right]
\end{aligned} \tag{C2}$$

where $n^<_{G_i,\pi}(x)$ and $n^>_{G_i,\pi}(x)$ are defined according to Definition 2.3.

An explanation for how this expression is computed is presented in Fig. 13. Note that (C2) must be strictly positive by our choice of optimal allocation — π' either has a bigger ℓ_2 or has a smaller $\ell_1 + \ell_2 + \ell_3$ than π. The first term in (C2) is always positive, the second term only contains integers, so it must be lower bounded by 1. This gives us the following observation:

Observation C.2 $2\lceil \frac{\ell_1+\ell_3-1}{2} \rceil - 2\lceil \frac{\ell_2}{2} \rceil + s - r \geq 1.$

Let us now construct a third allocation π'' from π by swapping $\{a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+1}, \ldots, a_{\ell_1+\ell_2}\}$ from G_2 with $\{a_{\ell_1+\ell_2+1}, \ldots, a_{\ell_1+\ell_2+\min\{\ell_2,\ell_3\}}\}$ from G_1. When we swap these two sets, we ensure we swap them in order. That is,

$$a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+1} \text{ is swapped with } a_{\ell_1+\ell_2+1},$$
$$a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+2} \text{ is swapped with } a_{\ell_1+\ell_2+2},$$

and so on. Note that we swap exactly $\min\{\ell_2, \ell_3\}$ values and with this careful swap, the edges between the values in each of these sets is preserved. That is, an edge between $a_{\ell_1+\ell_2+1}$ and $a_{\ell_1+\ell_2+2}$ exists in π'' if and only if it exists in π. Using an argument similar to Fig. 13, we can find the difference in envy between π'' and π as:

$$\begin{aligned}
&\mathsf{Envy}(\pi'', G_1 + G_2) - \mathsf{Envy}(\pi, G_1 + G_2)\\
&= c_1\Bigg[\Big\lceil\frac{n^{<}_{G_2}(a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+1})}{2}\Big\rceil - \left(s - \left\lfloor\frac{\min\{\ell_2,\ell_3\}}{2}\right\rfloor - \Big\lceil\frac{n^{<}_{G_2}(a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+1})}{2}\Big\rceil\right)\\
&\quad - \Big\lceil\frac{n^{<}_{G_1}(a_{\ell_1+\ell_2+1})}{2}\Big\rceil + \left(r - \left\lfloor\frac{\min\{\ell_2,\ell_3\}}{2}\right\rfloor - \Big\lceil\frac{n^{<}_{G_1}(a_{\ell_1+\ell_2+1})}{2}\Big\rceil\right)\Bigg]\\
&\quad + c_2\Bigg[\left\lfloor\frac{n^{<}_{G_2}(a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+1})}{2}\right\rfloor - \left(s - \Big\lceil\frac{\min\{\ell_2,\ell_3\}}{2}\Big\rceil - \left\lfloor\frac{n^{<}_{G_2}(a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+1})}{2}\right\rfloor\right)\\
&\quad - \left\lfloor\frac{n^{<}_{G_1}(a_{\ell_1+\ell_2+1})}{2}\right\rfloor + \left(r - \Big\lceil\frac{\min\{\ell_2,\ell_3\}}{2}\Big\rceil - \left\lfloor\frac{n^{<}_{G_1}(a_{\ell_1+\ell_2+1})}{2}\right\rfloor\right)\Bigg]\\
&\quad \text{where } c_1 = \sum_{j=0}^{\lceil\frac{\min\{\ell_2,\ell_3\}}{2}\rceil-1}\left(a_{\ell_1+\ell_2+2j+1} - a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+2j+1}\right)\\
&\quad \text{and } c_2 = \sum_{j=0}^{\lfloor\frac{\min\{\ell_2,\ell_3\}}{2}\rfloor-1}\left(a_{\ell_1+\ell_2+2j+2} - a_{\ell_1+\ell_2-\min\{\ell_2,\ell_3\}+2j+2}\right).
\end{aligned}$$

The only thing to keep in mind about c_1 and c_2 are that they are positive constants. The above expression can be simplified as

$$\begin{aligned}
&\mathsf{Envy}(\pi'', G_1 + G_2) - \mathsf{Envy}(\pi, G_1 + G_2)\\
&= c_1\left[2\Big\lceil\frac{\ell_2 - \min\{\ell_2,\ell_3\}}{2}\Big\rceil - 2\Big\lceil\frac{\ell_1}{2}\Big\rceil + r - s\right]\\
&\quad + c_2\left[2\left\lfloor\frac{\ell_2 - \min\{\ell_2,\ell_3\}}{2}\right\rfloor - 2\left\lfloor\frac{\ell_1}{2}\right\rfloor + r - s\right]\\
&\le (c_1 + c_2)\\
&\quad\left[\max\left\{2\Big\lceil\frac{\ell_2 - \min\{\ell_2,\ell_3\}}{2}\Big\rceil - 2\Big\lceil\frac{\ell_1}{2}\Big\rceil, 2\left\lfloor\frac{\ell_2 - \min\{\ell_2,\ell_3\}}{2}\right\rfloor - 2\left\lfloor\frac{\ell_1}{2}\right\rfloor\right\} + r - s\right]
\end{aligned} \tag{C3}$$

Again, (C3) must be strictly positive due to our choice of optimal allocation. c_1 and c_2 are positive constants, so this comes down to the second term. Note immediately that the second term cannot be positive if $\ell_2 \le \ell_3$. Therefore, we can assume $\ell_2 > \ell_3$, and using the fact that all the terms inside the second term are integers, we can make the following observation:

Observation C.3 $\max\left\{2\lceil\frac{\ell_2-\ell_3}{2}\rceil - 2\lceil\frac{\ell_1}{2}\rceil, 2\lfloor\frac{\ell_2-\ell_3}{2}\rfloor - 2\lfloor\frac{\ell_1}{2}\rfloor\right\} + r - s \ge 1.$

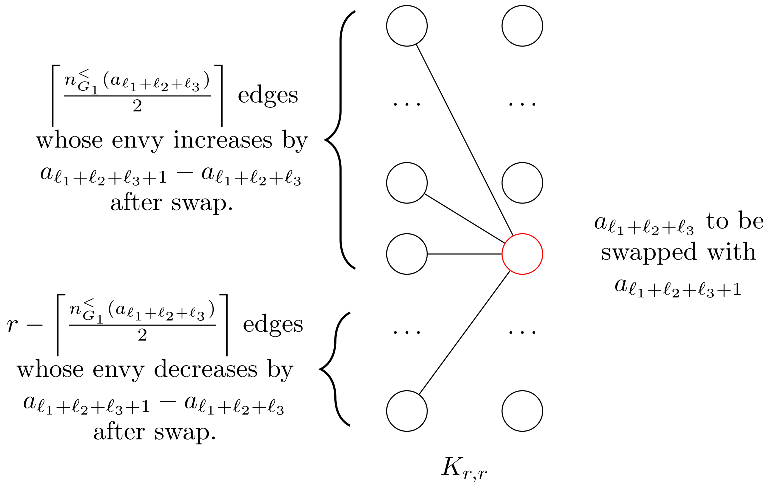


Fig. 13 Measuring the value $\mathsf{Envy}(\pi', G_1 + G_2) - \mathsf{Envy}(\pi, G_1 + G_2)$. We assume values are allocated in increasing order from the top to the bottom with least valued nodes at the top of the graph and the highest valued nodes at the bottom of the graph. Only the edges which see a change in envy are drawn. The exact change in envy for the edges in $K_{r,r}$ is described. A similar argument can be used to measure the exact change in envy in $K_{s,s}$

Adding up Observations C.2 and C.3, we get

$$\max\left\{2\left\lceil\frac{\ell_2-\ell_3}{2}\right\rceil - 2\left\lceil\frac{\ell_1}{2}\right\rceil + 2\left\lceil\frac{\ell_1+\ell_3-1}{2}\right\rceil - 2\left\lceil\frac{\ell_2}{2}\right\rceil,\right.$$
$$\left.2\left\lfloor\frac{\ell_2-\ell_3}{2}\right\rfloor - 2\left\lfloor\frac{\ell_1}{2}\right\rfloor + 2\left\lceil\frac{\ell_1+\ell_3-1}{2}\right\rceil - 2\left\lceil\frac{\ell_2}{2}\right\rceil\right\} \geq 2$$

It is easy to verify that the left hand side in the above inequality is upper bounded at 1; if there were no ceilings or floors, the left hand side would equal -1. The ceilings and floors, adversarially set, can only increase this value by 2. Therefore, the above expression can never be true and we have arrived at a glorious contradiction. □

Proposition 5.23 *There exists a* 3*-regular unsplittable graph.*

Proof The following lemma will prove to be useful.

Lemma C.4 *For any wheel* W_{2t+1}*, and any two non-rim vertices* $u_1, u_2 \in V(W_{2t+1})$*, there are three* u_1*-*u_2 *paths that are disjoint except at the endpoints.*

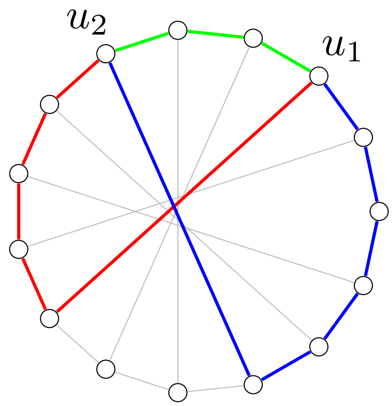


Fig. 14 Illustrative example of three disjoint paths between non-rim vertices u_1 and u_2, drawn here in three different colors: red, blue, and green

Proof The shortest path P_0 along the outer cycle is one path from u_1 to u_2. Call the remainder of the outer cycle the "longer u_1-u_2 path". Now, consider the path P_1 going from u_1 to its mate along its diagonal, and then to u_2 along the longer u_1-u_2 path. Also consider the path P_2 that takes u_1 to the mate of u_2 along the cycle on the longer u_1-u_2 path, and then across to u_2 on the diagonal. Note that P_0, P_1, and P_2 are all internally disjoint paths on this graph from u_1 to u_2. See Fig. 14 for an illustration. □

We continue with the proof. Consider the instance shown in Fig. 11. Call the inter-cluster gaps I_1, I_2, and I_3 respectively. By analyzing the size of any minimum cut in the given graph with exactly 401 vertices on one side, we can easily show that every allocation will need to have at least one edge go over I_1 (since there is no way to put 401 vertices of the graph without having at least one edge across the cut). Using a similar argument on minimum cuts with exactly 702 (resp. 903) vertices on one side, we can also show that at least two (resp. one) edges must go over I_2 (resp. I_3) in every allocation. So, the optimal envy must be at least $|I_1| + 2|I_2| + |I_3|$. Furthermore, this is realizable by the obvious allocation that maps the cluster sizes to the corresponding wheels. Therefore, any allocation that puts more than one edge on either I_1 or I_3, or more than two edges on I_2, must be strictly suboptimal.

Consider any optimal allocation. We first claim that W_{101} must be entirely inside the fourth cluster. Otherwise, some other wheel W' has its vertices appearing in the last cluster. If *only* the rim of W' appears in the last cluster, then its two neighbors in W' both appear in other clusters, so that I_3 has at least two edges passing over it, contradiction. So some non-rim vertex of W' appears in the fourth cluster. The fourth cluster is not enough to fit all of W', and so some non-rim vertex from W' appears in a different cluster as well. By Lemma C.4, this requires at least three edges over I_3, contradiction. Therefore, W_{101} fits snugly inside the fourth cluster.

We now claim that W_{201} must be entirely inside the third cluster. Otherwise, either W_{301} or W_{401} has some presence in the third cluster, say W_{301}. If this is a non-rim vertex, then again by Lemma C.4, we must have at least three edges over I_2, contradiction. So at best, the third cluster can have a rim vertex from W_{301}. This vertex's neighbors in W_{301} must be on either the first or second cluster, accounting for two edges above the interval I_2. But then, the third cluster must have some vertex from the bicycle $B_{401,201}$, but also does not have enough space to fit the entire bicycle. Hence, there must also be at least one edge over the interval I_2 from the bicycle $B_{401,201}$, accounting for a total of three or more edges over I_2, contradiction. A similar argument holds when W_{401} has some presence in the third cluster.

Finally, we claim that the copy of W_{401} must be entirely inside the first cluster. Otherwise, there is at least one vertex from W_{301} in the first cluster, and therefore at least one vertex from W_{401} in the second cluster. Of course, the second cluster cannot fit in at least 100 vertices from W_{401}, and so there is at least one non-rim W_{401}-vertex in the second cluster (otherwise its two neighbors correspond to two edges over I_1, contradiction), and at least one non-rim W_{401} vertex in the first cluster, which by Lemma C.4 is a contradiction. □

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10458-024-09672-7.

Acknowledgements The authors thank Cameron Musco and Yair Zick for extremely helpful discussions. Rohit Vaish acknowledges support from Science and Engineering Research Board (SERB) grant no. CRG/2022/002621 and Department of Science & Technology (DST) INSPIRE grant no. DST/INSPIRE/04/2020/000107. Andrew McGregor and Rik Sengupta acknowledge support from National Science Foundation (NSF) grants CCF-1934846 and CCF-1908849. This work was done in part while Andrew McGregor was visiting the Simons Institute for the Theory of Computing. Hadi Hosseini acknowledges support from National Science Foundation (NSF) grants IIS-2144413 and IIS-2107173. Justin Payan and Vignesh Viswanathan acknowledge support from National Science Foundation (NSF) grant IIS-2327057.

Author Contributions All authors contributed equally to this work.

Funding Rohit Vaish is funded by Science and Engineering Research Board (SERB) grant no. CRG/2022/002621 and Department of Science & Technology (DST) INSPIRE grant no. DST/INSPIRE/04/2020/000107. Andrew McGregor and Rik Sengupta are funded by National Science Foundation (NSF) grants CCF-1934846 and CCF-1908849. Hadi Hosseini is funded by National Science Foundation (NSF) grants IIS-2144413 and IIS-2107173. Justin Payan and Vignesh Viswanathan are funded by National Science Foundation (NSF) grant IIS-2327057.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent for publication All authors consent to publication.

References

1. Hosseini, H., Payan, J., Sengupta, R., Vaish, R., & Viswanathan, V. (2023). Graphical house allocation. In *Proceedings of the 22nd international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 161–169).
2. Massand, S., & Simon, S. (2019). Graphical one-sided markets. In *Proceedings of the 28th international joint conference on artificial intelligence (IJCAI)* (pp. 492–498).
3. Beynier, A., Chevaleyre, Y., Gourvès, L., Harutyunyan, A., Lesca, J., Maudet, N., & Wilczynski, A. (2019). Local envy-freeness in house allocation problems. *Autonomous Agents and Multi-Agent Systems, 33*, 591–627.
4. Doucette, J. A., Tsang, A., Hosseini, H., Larson, K., & Cohen, R. (2019). Inferring true voting outcomes in homophilic social networks. *Autonomous Agents and Multi-Agent Systems, 33*(3), 298–329.
5. Tsang, A., Doucette, J. A., & Hosseini, H. (2015). Voting with social influence: Using arguments to uncover ground truth. In *Proceedings of the 14th international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 1841–1842).
6. Grandi, U. (2017). Social choice and social networks. In U. Endriss (Ed.), *Trends in Computational Social Choice* (pp. 169–184). AI Access, open access.

7. Abebe, R., Kleinberg, J., & Parkes, D. C. (2017). Fair division via social comparison. In *Proceedings of the 16th international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 281–289).
8. Bredereck, R., Kaczmarczyk, A., & Niedermeier, R. (2022). Envy-free allocations respecting social networks. *Artificial Intelligence,305*.
9. Peters, D. (2016). Graphical hedonic games of bounded treewidth. In *Proceedings of the 30th AAAI conference on artificial intelligence (AAAI)* (pp. 586–593).
10. Igarashi, A., & Elkind, E. (2016). Hedonic games with graph-restricted communication. In *Proceedings of the 15th international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 242–250).
11. Hammer, P. L., & Maffray, F. (1990). Completely separable graphs. *Discrete Applied Mathematics, 27*(1), 85–99.
12. Shapley, L., & Scarf, H. (1974). On cores and indivisibility. *Journal of Mathematical Economics, 1*(1), 23–37.
13. Abdulkadiroğlu, A., & Sönmez, T. (1999). House allocation with existing tenants. *Journal of Economic Theory, 88*(2), 233–260.
14. Roth, A. E., Sönmez, T., & Ünver, M. U. (2004). Kidney exchange. *The Quarterly Journal of Economics, 119*(2), 457–488.
15. Svensson, L.-G. (1999). Strategy-proof allocation of indivisible goods. *Social Choice and Welfare, 16*(4), 557–567.
16. Gan, J., Suksompong, W., & Voudouris, A. A. (2019). Envy-freeness in house allocation problems. *Mathematical Social Sciences, 101*, 104–106.
17. Kamiyama, N. (2021). The envy-free matching problem with pairwise preferences. *Information Processing Letters, 172*, 106158.
18. Kamiyama, N., Manurangsi, P., & Suksompong, W. (2021). On the complexity of fair house allocation. *Operations Research Letters, 49*(4), 572–577.
19. Madathil, J., Misra, N., & Sethia, A. (2023). The complexity of minimizing envy in house allocation. In *Proceedings of the 22nd international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 2673–2675).
20. Aigner-Horev, E., & Segal-Halevi, E. (2022). Envy-free matchings in bipartite graphs and their applications to fair division. *Information Sciences, 587*, 164–187.
21. Gross-Humbert, N., Benabbou, N., Beynier, A., & Maudet, N. (2023). On the notion of envy among groups of agents in house allocation problems. In *Proceedings of the 26th European conference on artificial intelligence (ECAI)* (pp. 924–931).
22. Aziz, H., Iliffe, I., Li, B., Ritossa, A., Sun, A., & Suzuki, M. (2024). Envy-free house allocation under uncertain preferences. In *Proceedings of the 38th AAAI conference on artificial intelligence (AAAI)* (pp. 9477–9484).
23. Choo, D., Ling, Y. H., Suksompong, W., Teh, N., & Zhang, J. (2024). Envy-free house allocation with minimum subsidy. *Operations Research Letters,54*.
24. Biswas, A., Payan, J., Sengupta, R., & Viswanathan, V. (2023). The theory of fair allocation under structured set constraints. In A. Mukherjee, J. Kulshrestha, A. Chakraborty, & S. Kumar (Eds.), *Ethics in Artificial Intelligence: Bias, Fairness and Beyond* (pp. 115–129). Springer.
25. Hosseini, H., McGregor, A., Sengupta, R., Vaish, R., & Viswanathan, V. (2024). Tight approximations for graphical house allocation. In *Proceedings of the 23rd international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 825–833).
26. Eiben, E., Ganian, R., Hamm, T., & Ordyniak, S. (2020). Parameterized complexity of envy-free resource allocation in social networks. In *Proceedings of the 34th AAAI conference on artificial intelligence (AAAI)* (pp. 7135–7142).
27. Beynier, A., Maudet, N., & Damamme, A. (2018). Fairness in multiagent resource allocation with dynamic and partial observations. In *Proceedings of the 17th international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 1868–1870).
28. Lange, P., & Rothe, J. (2019). Optimizing social welfare in social networks. In *Proceedings of the 6th international conference on algorithmic decision theory (ADT)* (pp. 81–96).
29. Gourvès, L., Lesca, J., & Wilczynski, A. (2017). Object allocation via swaps along a social network. In *Proceedings of the 26th international joint conference on artificial intelligence (IJCAI)* (pp. 213–219).
30. Ito, T., Iwamasa, Y., Kakimura, N., Kamiyama, N., Kobayashi, Y., Nozaki, Y., Okamoto, Y., & Ozeki, K. (2022). Reforming an envy-free matching. In *Proceedings of the 36th AAAI conference on artificial intelligence (AAAI)* (pp. 5084–5091).
31. Elkind, E., Patel, N., Tsang, A., & Zick, Y. (2020). Keeping your friends close: Land allocation with friends. In *Proceedings of the 29th international joint conference on artificial intelligence (IJCAI)* (pp. 318–324).

32. Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Prentice Hall.
33. Chung, F. R. K. (1984). On optimal linear arrangements of trees. *Computers & Mathematics with Applications, 10*(1), 43–60.
34. Garey, M. R., Johnson, D. S., & Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science, 1*(3), 237–267.
35. Koren, Y., & Harel, D. (2002). A multi-scale algorithm for the linear arrangement problem. In *International workshop on graph-theoretic concepts in computer science* (pp. 296–309).
36. Even, S., & Shiloach, Y. (1978). NP-completeness of several arrangements problems. Technical Report, TR-43 The Technicon, 29.
37. Seidvasser, M. A. (1970). The optimal number of the vertices of a tree. *Diskref. Anal., 19*, 56–74.
38. Khot, S. (2006). Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing, 36*(4), 1025–1071.
39. Bui, T. N., & Jones, C. (1992). Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters, 42*(3), 153–159.
40. Garey, M. R., & Johnson, D. S. (1990). *Computers and intractability; A guide to the theory of NP-completeness*. W. H. Freeman & Co.
41. Papadimitriou, C. (1976). The NP-completeness of the bandwidth minimization problem. *Computing, 16*(3), 263–270.
42. Blache, G., Karpiński, M., & Wirtgen, J. (1997). *On approximation intractability of the bandwidth problem*. Inst. für Informatik.

Authors and Affiliations

Hadi Hosseini[1] · Andrew McGregor[2] · Justin Payan[2] · Rik Sengupta[2,3] · Rohit Vaish[4] · Vignesh Viswanathan[2]

✉ Vignesh Viswanathan
vviswanathan@umass.edu

Hadi Hosseini
hadi@psu.edu

Andrew McGregor
mcgregor@cs.umass.edu

Justin Payan
jpayan@umass.edu

Rik Sengupta
rik@ibm.com

Rohit Vaish
rvaish@iitd.ac.in

[1] Pennsylvania State University, University Park, PA, USA

[2] University of Massachusetts Amherst, Amherst, MA, USA

[3] MIT-IBM Watson AI Lab, Cambridge, MA, USA

[4] Indian Institute of Technology Delhi, New Delhi, India