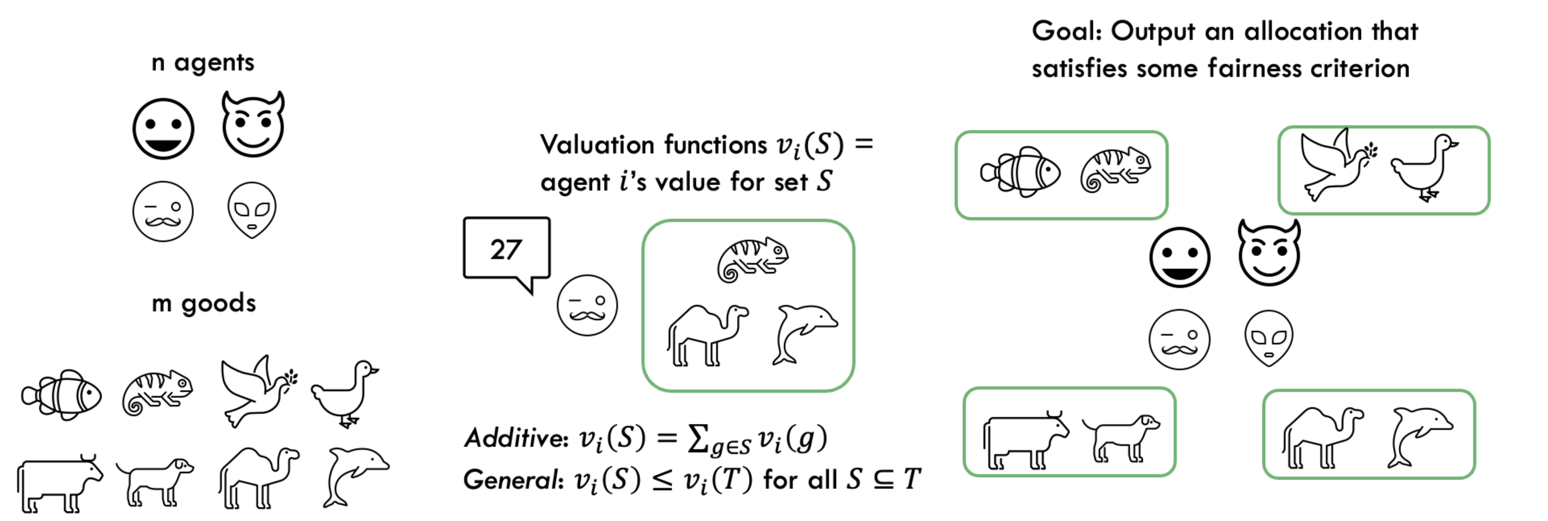
Relaxations of Envy-Freeness Over Graphs

Justin Payan Rik Sengupta Vignesh Viswanathan

University of Massachusetts Amherst

Fair Allocations

Given agents N, goods M, and valuations $v_i : 2^M \to \mathbb{R}_{\geq 0}$ for all $i \in N$, output a *fair allocation*.



Envy-free allocations would be great, but are not always possible. Can we ensure that our allocation is *envy-free up to any item* (EFX) or *envy-free up to k uniformly hidden items* (uHEF-k)?

EFX and Hidden Envy-Free Allocations on Graphs

An allocation is *EFX* when, for all pairs of agents i and j with bundles X_i and X_j, for all $g \in X_j$, we have $v_i(X_i) \geq v_i(X_j \setminus g)$.

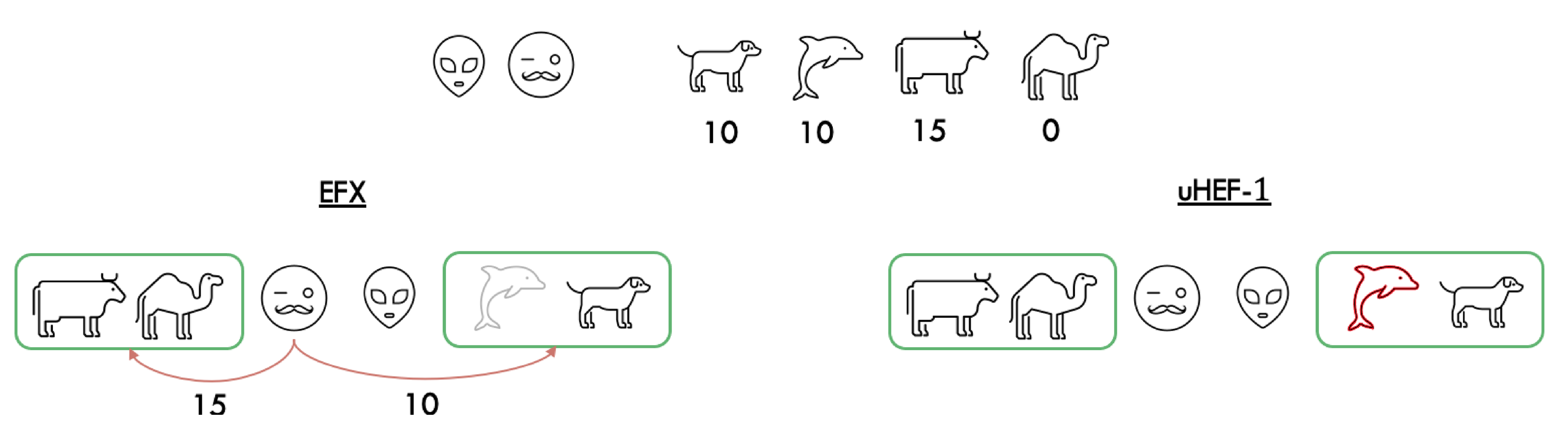
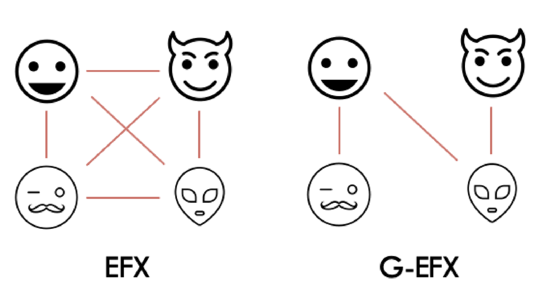


Figure 1. Example of an EFX allocation (left) and uHEF-1 allocation (right). Red goods are hidden.

An allocation is *uHEF-k* when, there exists a subset $S \subseteq M$, $|S| \leq k$, $|S \cap X_i| \leq 1$ for all i, such that for all pairs of agents i and j, we have $v_i(X_i) \geq v_i(X_j \setminus S)$.

We study graph theoretic relaxations of these problems, when the agents N are placed on the vertices of an undirected graph $G = (N, E)$, and we only wish to maintain the EFX or uHEF-k constraint among agent pairs i and j such that $(i, j) \in E$. We call these *G-EFX* or *G-uHEF-k* allocations.



Motivation

- A graph is a **natural** constraint, as stakeholders only care about those they interact with.
- The graph constraint **relaxes** the general problems ($G = K_n$, the complete graph).
- Solving G-EFX for simple graphs may lead to **critical insights** into the general problem.

Results for Hidden Envy on Graphs

Theorem (Upper Bound). When agents with general valuation functions are placed on a graph G with a vertex cover C of size k, we can find a G-uHEF-k allocation efficiently.

Pf. A round robin protocol where every agent in C picks each round before any agent not in C achieves this.

Theorem (Tightness). For any connected graph G with a vertex cover C of size k, there is an instance with additive valuation functions for which there is no G-uHEF-k' allocation for $k' < k$.

Pf. Technical, so omitted!

Despite the tightness of vertex cover, there are instances where the number of hidden items is much smaller than the vertex cover.

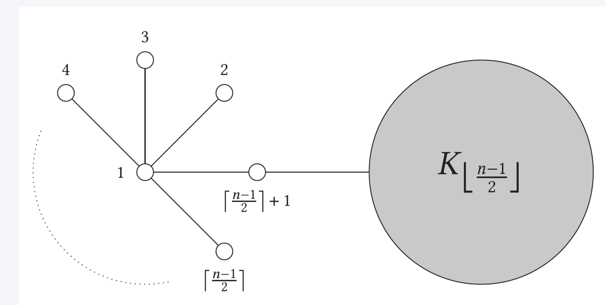


Figure 2. An instance admitting a G-uHEF-2 allocation, but any vertex cover has $\Theta(n)$ vertices.

Results for EFX on Graphs

Theorem (EFX on a Star). When agents with general valuation functions are placed on the vertices of the star $K_{1,n-1}$, a G-EFX allocation exists.

Pf. EFX allocations exist for agents with identical valuation functions [Plaut and Roughgarden, 2020]. We "imagine" all n agents to have the valuation function of the central agent, and let the outer agents choose first.

Generalizations:

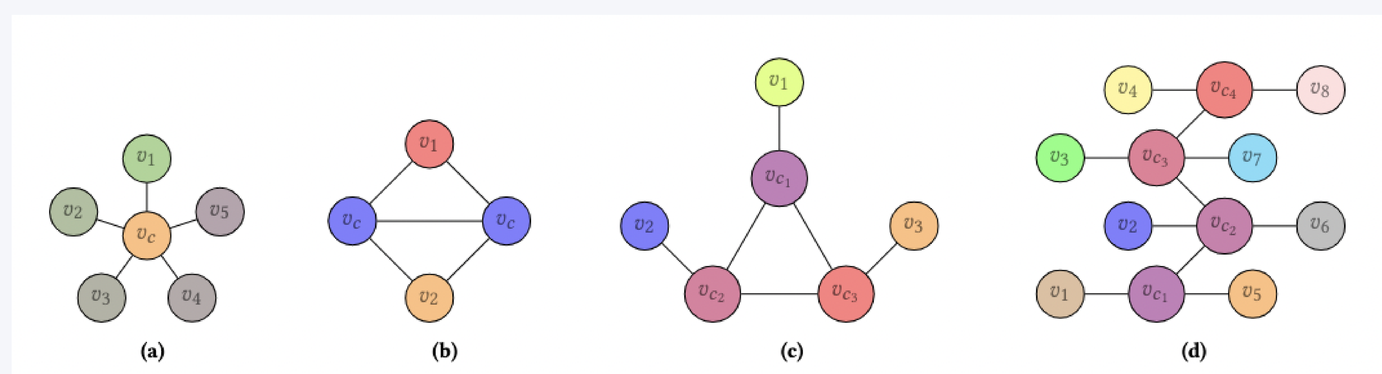


Figure 3. Generalizations of the star graph.

Theorem (Extension to Chores). When agents with additive valuation functions are placed on the vertices of the star $K_{1,n-1}$ and all items are chores (negative-valued goods), a G-EFX allocation exists.

Theorem (EFX on a 3-Edge Path). When agents with general valuation functions are placed on the vertices of the 3-edge path P_3, a G-EFX allocation exists.

Pf. EFX allocations exist for agents with two distinct "types" of valuation functions [Mahara, 2021]. We "imagine" the endpoints of the path to have the same valuation functions as their neighbors on that path, and let the endpoints choose first.

Generalizations:

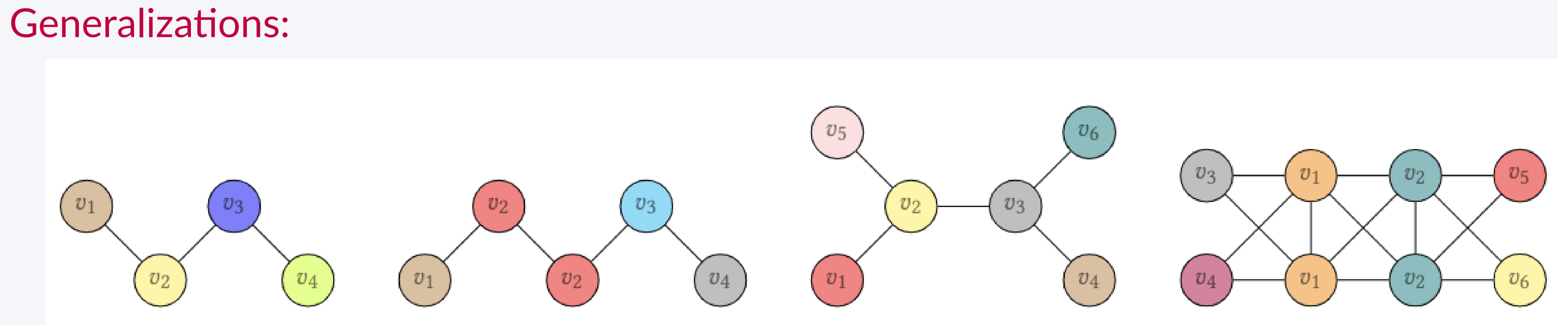


Figure 4. Generalizations of the path graph.

Results for Goods and Chores

Theorem (Hosseini et al. [2022]). EFX allocations do not necessarily exist for goods and chores, even for lexicographic valuations (exponential relative importance between items).

Theorem (Goods and Chores) G-EFX allocations exist (and can be found efficiently) for goods and chores for lexicographic valuations for all G with diameter 4 or more.

Pf. Assign goods to one side of the length 4 path, and chores to the other side. The chore endpoint receives a highly disliked chore, and the good endpoint receives the least important goods.

The Sweeping Algorithm

We study a "sweeping algorithm" on path graphs, that might generalize further. The algorithm iterates over edges, re-allocating goods to make edges EFX. We must prove termination.

Algorithm 1 Sweeping Algorithm

Require: n agents N, m goods M, valuations $v_i : 2^M \to \mathbb{R}$ for $i \in N$, path P_n
  Initialize allocation $X = (X_1, \dots X_n)$ with $X_1 = M$ and $X_i = \varnothing$ for all $i \neq 1$
  while $\exists (i, i+1) \in P_n$ s.t. i strongly envies $i+1$ or $i+1$ strongly envies i **do**
    for $i \in \{1, \dots, n-1\}$ and $i \in \{n-2, \dots, 1\}$ **do**
      $(X_i, X_{i+1}) \leftarrow \text{LocalEFX}(X_i, X_{i+1}, v_i, v_{i+1})$
  return X

Spliddit Experiments

To prove termination, we investigate *potential functions* – monotonic and bounded loop variants.

We create path graphs for 3392 Spliddit instance. The sweeping algorithm terminates in all instances with a G-EFX allocation. We examine several potential functions and plot where they fail.

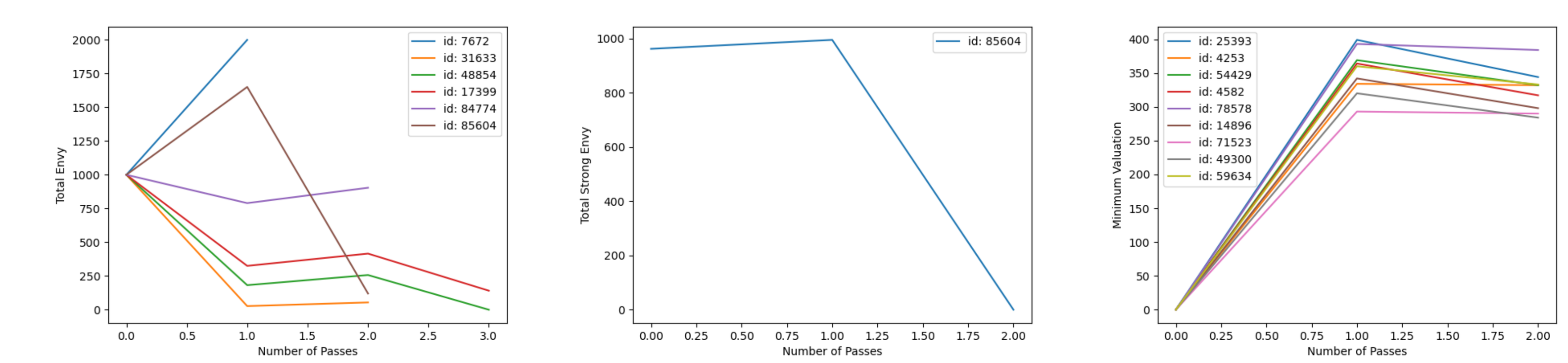


Figure 5. Potential functions on selected instances, plotted after each full pass of the algorithm.

The total strong envy is non-monotonic on exactly one instance. We can force strong envy to decrease monotonically by allowing the LocalEFX procedure to vary across edges, suggesting a proof technique.

References

Hadi Hosseini, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. Fairly dividing mixtures of goods and chores under lexicographic preferences, 2022.

Ryoga Mahara. Extension of additive valuations to general valuations on the existence of EFX. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA)*, volume 204, pages 1–15, 2021.

Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal of Discrete Mathematics*, 34:1039–1068, 2020.

Acknowledgments

We would like to thank Raghav Addanki for early conversations about this work. We also would like to thank Yair Zick, Rohit Vaish, and Hadi Hosseini for very helpful discussions and feedback.